



UNIVERSITÄT ZU LÜBECK  
INSTITUT FÜR ROBOTIK  
UND KOGNITIVE SYSTEME

# Smart Sensor, Navigation and Learning Strategies for low-cost Lawn Care Systems

Nils Rottmann

Dissertation

University of Lübeck  
Institute for Robotics and Cognitive Systems





UNIVERSITÄT ZU LÜBECK

From the Institute for Robotics and Cognitive Systems  
of the University of Lübeck  
Director: Prof. Dr.–Ing. Achim Schweikard

**Smart Sensor, Navigation and Learning Strategies for  
low-cost Lawn Care Systems**

Dissertation  
for Fulfillment of Requirements  
for the Doctoral Degree  
of the University of Lübeck

from the Department of Computer Sciences and Technical Engineering

Submitted by  
Nils Rottmann  
from Nienburg (Weser)

Lübeck, 2021



1. First referee: Univ.-Prof. Dr. Elmar Rückert

2. Second referee: Prof. Dr. Georg Schildbach

Date of oral examination: 04. August 2021

Approved for printing. Lübeck, 10. August 2021



Nature is the source of all true  
knowledge. She has her own logic, her  
own laws, she has no effect without  
cause nor invention without necessity.

---

Leonardo da Vinci



## Abstract

Autonomous systems, including robots, are increasingly in demand to progressively take over tasks of everyday life, such as mowing the lawn or vacuuming. To make such systems affordable for the end user and at the same time intelligent, cost-efficient sensors and smart navigation and planning strategies are required. While developing such systems, the special characteristics of the working environment as well as the intended task have to be considered.

Currently, household robots, e.g., lawn mowers, often apply simple, random planning strategies, as they can only rely on in-/outside area measurements for navigation. Inspired by probabilistic approaches for mapping, navigation and planning, in the first part of this thesis a mapping and a complete coverage path planning approach for domestic robots under high pose uncertainty is developed. In particular, it is shown that accurate map estimates of enclosed environments can be generated with only in-/outside area detectors. Further, given a map of the enclosed environment, intelligent path planning strategies are developed that far exceed the performance of standard random walk behavior. Here, probabilistic models are key to enable an estimation of the coverage of the workspace.

The second part of this thesis discusses the development of a low-cost sensor for autonomous lawn mowers. This addresses the rising expectations in solving complex everyday tasks with new smart sensor technologies. In particular, the focus there is on developing a low-cost sensor that can replace the currently used perimeter wire to avoid high setup and maintenance costs. The sensor development follows the basic idea of optimally exploiting the characteristic features of the working environment. In this case, by exploiting chlorophyll fluorescence which is one of the main characteristics of plants in general. Therefore, chlorophyll fluorescence is actively stimulated by the proposed sensor concept and the chlorophyll fluorescence response is measured. This allows for efficiently detecting the working area for autonomous lawn mowers.

The last part of this thesis deals with the control and monitoring of autonomous robots, especially in outdoor scenarios. For this purpose, an application for the Robot Operating System (ROS) for mobile devices is developed. The leading idea for the development is to facilitate robot control and monitoring for the end-user while offering various modules. Especially the use of mobile devices as intuitive control panels allows for simple operational control.

Together, the methodological, sensor and software developments of this thesis provide the basis for building novel, cost-efficient autonomous systems which support humans in everyday tasks such as lawn care in particular or gardening in general.

**Keywords.** Mobile Robots, Sensors, Mapping, Navigation, Planning, Chlorophyll Fluorescence, Plant Classification, Reinforcement Learning



## Kurzfassung

Autonome Systeme, darunter auch Roboter, sind zunehmend gefragt um, nach und nach Aufgaben des täglichen Lebens zu übernehmen, wie zum Beispiel das Rasenmähen oder Staubsaugen. Um solche Systeme für den Endanwender erschwinglich und gleichzeitig intelligent zu machen, sind kosteneffiziente Sensoren und intelligente Navigations- und Planungsstrategien erforderlich. Bei der Entwicklung solcher Systeme müssen die besonderen Eigenschaften der Arbeitsumgebung sowie die vorgesehene Aufgabe berücksichtigt werden.

Derzeit verwenden Haushaltsroboter, z. B. Rasenmäher oft einfache, zufällige Planungsstrategien, da sie sich nur auf Innen-/Außenbereichsmessungen zur Navigation verlassen können. Inspiriert von probabilistischen Ansätzen für Kartierung, Navigation und Planung wird im ersten Teil dieser Arbeit ein Kartierungs- und ein flächendeckender Pfadplanungsansatz für Haushaltsroboter unter hoher Posenunsicherheit entwickelt. Insbesondere wird gezeigt, dass genaue Kartenschätzungen von umschlossenen Umgebungen nur mit Innen-/Außenbereichsdetektoren erzeugt werden können. Darüber hinaus werden intelligente Bahnplanungsstrategien entwickelt, die die Leistung von Standard-Random-Walk-Verhalten weit übertreffen. Hier sind probabilistische Modelle der Schlüssel, um eine Schätzung der Abdeckung des Arbeitsraums zu ermöglichen.

Der zweite Teil dieser Arbeit befasst sich mit der Entwicklung eines Low-Cost-Sensors für autonome Rasenmäher. Damit wird den steigenden Erwartungen an die Lösung komplexer Alltagsaufgaben mit neuen intelligenten Sensortechnologien Rechnung getragen. Insbesondere liegt der Fokus hier auf der Entwicklung eines Low-Cost-Sensors, der den derzeit verwendeten Begrenzungsdraht ersetzen kann, um hohe Einrichtungs- und Wartungskosten zu vermeiden. Die Sensorentwicklung folgt dem Grundgedanken, die charakteristischen Merkmale der Arbeitsumgebung optimal auszunutzen. In diesem Fall durch Ausnutzung der Chlorophyll-Fluoreszenz, die eine der Haupteigenschaften von Pflanzen im Allgemeinen ist. Dabei wird die Chlorophyll-Fluoreszenz durch das vorgeschlagene Sensorkonzept aktiv angeregt und die Reaktion der Chlorophyll-Fluoreszenz gemessen. Dies ermöglicht eine effiziente Erkennung des Arbeitsbereichs für autonome Rasenmäher.

Der letzte Teil dieser Arbeit beschäftigt sich mit der Steuerung und Überwachung von autonomen Robotern, insbesondere in Outdoor-Szenarien. Zu diesem Zweck wird eine Anwendung für das Robot Operating System (ROS) für mobile Geräte entwickelt. Der Leitgedanke bei der Entwicklung ist es, die Steuerung und Überwachung von Robotern für den Endanwender zu erleichtern und dabei verschiedene Module anzubieten. Insbesondere die Nutzung von mobilen Geräten als intuitives Bedienpanel ermöglicht eine einfache Betriebssteuerung.

Zusammengenommen bilden die methodischen, sensorischen und softwaretechnischen Entwicklungen dieser Arbeit die Grundlage für den Aufbau neuartiger, kostengünstiger

autonomer Systeme, die den Menschen bei alltäglichen Aufgaben wie der Rasenpflege im Speziellen oder der Gartenarbeit im Allgemeinen unterstützen.

**Schlüsselwörter.** Mobile Roboter, Sensoren, Kartierung, Navigation, Planung, Chlorophyll-Fluoreszenz, Pflanzenklassifikation, Reinforcement Learning

## Acknowledgments

I like to thank my supervisor Elmar Rückert for his guidance and support throughout my doctoral studies. In particular for his inspiring visions, conceptual ideas and persevering patience. Special thanks go also to Achim Schweikard and Floris Ernst for supporting my doctoral studies within the institute. Also, I like to thank Georg Schildbach for taking time to review my thesis and to attend my defense.

I especially like to thank my co-authors, with whom the work has been a lot of fun and very successful. To mention here in particular would be Ralf Bruder, Jannis Hagenah and Nico Studt. Special thanks go also to my colleagues from Stihl, namely Stephanie Jung and Andreas Strohmaier and their supervisor Stephan Meyer.

I am also very grateful for the support provided by current and former colleagues at the Institute of Robotics and Cognitive Systems. In particular, I would like to thank Cornelia Rieckhoff, who was always available for organizational support.

Last but not least, I like to thank my family and friends for their tremendous support during my doctoral studies, especially during the corona crisis. In particular to mention are my parents, Ute and Heinrich, and my brother and sister, Eric and Kea.



## Mathematical Notation

In this thesis, standard mathematical notation is used such that scalars are represented as single letters, i.e.,  $a \in \mathcal{R}^1$ , vectors as bold letters, i.e.,  $\mathbf{w} \in \mathcal{R}^{10}$ , and matrices as bold capital letters, i.e.,  $\mathbf{M} \in \mathcal{R}^{\{10 \times 10\}}$ . Subscripts are used to denote elements in vectors, i.e.,  $w_i$  is part of  $\mathbf{w} = [w_1, w_2, \dots, w_i]$ . Functions are written as single letters, i.e.,  $x = f(y)$  which implements the transformation  $f(y) : \mathcal{R}^1 \rightarrow \mathcal{R}^1$ . Sets are written as capital letters, i.e.,  $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ . For better understanding, subscripts for variables or functions might be used, i.e.,  $f_{\text{Chl}}(\lambda)$  which implements the emission spectrum for chlorophyll fluorescence.

Standard notation for commonly used variables is used throughout this thesis. This includes for example  $\mathbf{p}$  for the robot's pose,  $\mathbf{x}$  for the robot's position or  $\mathbf{R}$  for the rotation matrix. Other variables, especially which represent quantity information must be read according to the respective context. For example,  $N$  can be number of odometric constraints for pose graph optimization or the number of particles for a particle filter. Important non-commonly used variables and parameters appearing within this thesis are summarized in the nomenclature, partitioned into the three main chapters. Thereby, variables and parameters are designated for each chapter individually.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	A short History of industrial and domestic Robots . . . . .	1
1.2	Domestic Robots - Guiding Research Questions . . . . .	3
1.3	Organization of the Thesis . . . . .	5
<b>2</b>	<b>Navigation &amp; Planning</b>	<b>7</b>
2.1	Introduction . . . . .	8
2.2	Related Work . . . . .	8
2.2.1	Mapping & Navigation . . . . .	9
2.2.2	Complete Coverage Path Planing . . . . .	10
2.2.3	Hyper-Parameter Learning . . . . .	11
2.3	Scientific Contributions . . . . .	11
2.4	Robot Motion Models . . . . .	12
2.4.1	Robot Setup . . . . .	12
2.4.2	Velocity Motion Model . . . . .	13
2.4.3	Odometry Motion Model . . . . .	14
2.4.4	Model Parameter Determination . . . . .	15
2.5	Map Generation for closed Environments . . . . .	18
2.5.1	Pose Graph Representation . . . . .	19
2.5.2	Loop Closure Detection . . . . .	22
2.5.3	Hyper-Parameter Learning . . . . .	25
2.5.4	Evaluation . . . . .	28
2.6	Path Planning under high Pose Uncertainty . . . . .	32
2.6.1	Probabilistic Coverage Map and Dynamical System . . . . .	33
2.6.2	Particle Filter . . . . .	35
2.6.3	Complete Coverage Path Planning . . . . .	35
2.6.4	Evaluation . . . . .	38
2.7	Discussion . . . . .	42
2.7.1	Map Generation . . . . .	42
2.7.2	Hyper-Parameter Learning . . . . .	43
2.7.3	Complete Coverage Path Planning . . . . .	43
2.8	Conclusion . . . . .	44
2.8.1	Detailed Future Work . . . . .	44
2.9	Acknowledgments . . . . .	44

<b>3</b>	<b>Sensor Development for Working Area Detection and Classification</b>	<b>47</b>
3.1	Introduction . . . . .	48
3.2	Related Work . . . . .	49
3.2.1	Working Area Detection . . . . .	49
3.2.2	Plant Classification . . . . .	51
3.3	Scientific Contributions . . . . .	52
3.4	Chlorophyll Fluorescence . . . . .	52
3.4.1	General Principle . . . . .	53
3.4.2	Kautsky Effect & Lifetime . . . . .	54
3.5	Chlorophyll Sensor . . . . .	55
3.5.1	Core Sensor Design . . . . .	55
3.5.2	Signal Path Analysis . . . . .	57
3.5.3	Hardware Component Analysis . . . . .	61
3.5.4	Evaluation . . . . .	62
3.6	Plant Classification . . . . .	64
3.6.1	Data Acquisition & Preprocessing . . . . .	65
3.6.2	Deep Learning Classification Approach - Benchmark . . . . .	66
3.6.3	Low-cost system . . . . .	68
3.6.4	Evaluation . . . . .	69
3.7	Discussion . . . . .	73
3.7.1	Working Area Detection . . . . .	73
3.7.2	Plant Classification . . . . .	74
3.8	Conclusion . . . . .	74
3.8.1	Detailed Future Work . . . . .	75
3.9	Acknowledgments . . . . .	75
<b>4</b>	<b>Robot Control and Monitoring – ROS-Mobile</b>	<b>77</b>
4.1	Introduction . . . . .	77
4.2	Related Work . . . . .	79
4.3	Scientific Contributions . . . . .	80
4.4	ROS-Basics . . . . .	80
4.4.1	ROS Components . . . . .	81
4.4.2	ROS Code Hierarchy . . . . .	82
4.5	Mobile Application Design . . . . .	82
4.5.1	Software Architecture . . . . .	82
4.5.2	Functionality and Extensibility . . . . .	83
4.6	Use-Case Scenarios . . . . .	84
4.6.1	Demo Example . . . . .	84
4.6.2	Application Scenarios . . . . .	85
4.7	User Acceptance . . . . .	86
4.8	Conclusion . . . . .	87
4.8.1	Detailed Future Work . . . . .	87
4.9	Acknowledgements . . . . .	88

<b>5 Conclusion</b>	<b>89</b>
5.1 Summary . . . . .	89
5.2 Conclusion on the Guiding Research Questions . . . . .	90
5.3 Future Work . . . . .	92
<b>Appendix</b>	<b>93</b>
A Wall Following Algorithm/Controller . . . . .	93
B Bayesian Optimization . . . . .	95
C Optimal traveled Distance . . . . .	96
D Sensor Circuit Diagrams . . . . .	97
E PT & LED Spectra . . . . .	100
<b>List of own Publications</b>	<b>103</b>
<b>List of Figures</b>	<b>105</b>
<b>List of Tables</b>	<b>107</b>
<b>Acronyms</b>	<b>109</b>
<b>Nomenclature</b>	<b>111</b>
<b>Bibliography</b>	<b>115</b>
<b>Curriculum Vitae</b>	<b>129</b>



# 1 Introduction

In this chapter, a brief introduction to robotics in industrial and domestic contexts is given. On this basis, the research questions related to household robots that lead through this thesis are presented. Finally, the overall structure of this thesis is introduced.

## Contents

---

<b>1.1</b>	<b>A short History of industrial and domestic Robots . . . . .</b>	<b>1</b>
<b>1.2</b>	<b>Domestic Robots - Guiding Research Questions . . . . .</b>	<b>3</b>
<b>1.3</b>	<b>Organization of the Thesis . . . . .</b>	<b>5</b>

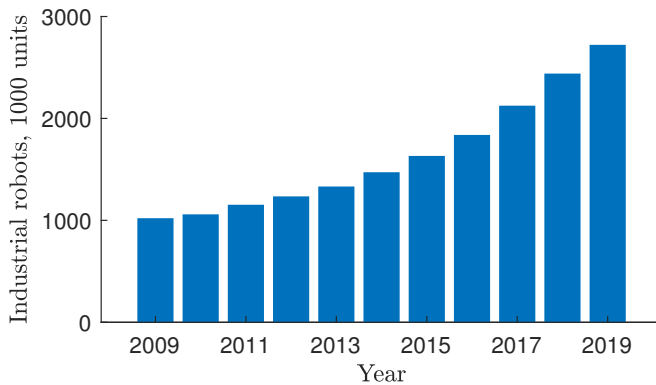
---

## 1.1 A short History of industrial and domestic Robots

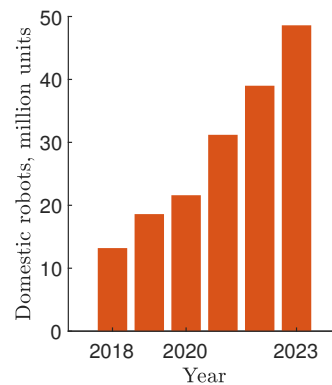
Robots, or autonomous systems in general, were invented during the 20th century and became increasingly popular. Nowadays, robots, e.g., industrial robots, are indispensable in many areas, such as car manufacturing. The history of robots began with the 1920 published drama “R.U.R. – Rossum’s Universal Robots” by the czech writer Karel Čapek [1], where for the first time the word “robot” was introduced. Then, in 1942, another science-fiction author, Isaac Asimov, postulated the three laws of “robotics” in his short story “Runaround” [2]:

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.
2. A robot must obey orders given it by human beings except where such orders would conflict with the First Law.
3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.

Asimov’s work, especially the laws of “robotics”, became the basis for several modern science-fiction stories, e.g., the in 2004 released movie “I, Robot”.



(a) The growth of the operational stock of industrial robots worldwide during the past years.



(b) The growth of the unit sales of service robots for domestic tasks and the potential development.

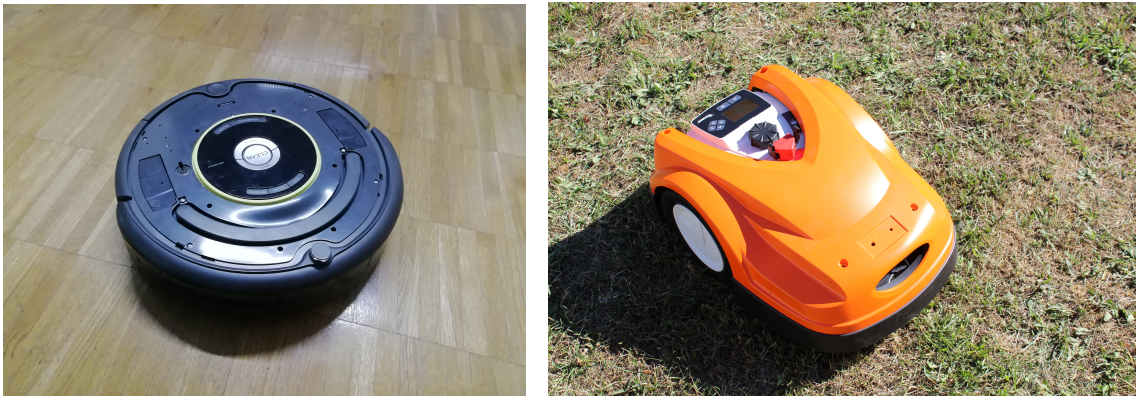
Figure 1.1: Growth of robot stocks and sales in the industrial and domestic area. Data from the IFR Press Conference, 2020, [https://ifr.org/downloads/press2018/Presentation\\_WR\\_2020.pdf](https://ifr.org/downloads/press2018/Presentation_WR_2020.pdf).

**Industrial Robots** Industrial robots first made their appearance in 1954 with the invention of the “Unimate”, the first industrial robot of the world. In the following decades, industrial robots experienced a rapid development, where four “generations” of robots can be identified [3]:

First Manipulators (1950-1967), Sensorized Robots (1968-1977), Industrial Robots (1978-1999) and Intelligent Robots (2000-now).

A detailed survey of the development of industrial robots in the early decades is given by Gasparetto and Scalera in [4]. Nowadays, the use of industrial robots continues to grow exponentially, Figure 1.1a. This is based on the ever-increasing automation of industrial production. Moreover, through recent advances in artificial intelligence, robots become more and more flexible in their fields of application. This allows for a wide range of application scenarios, e.g., co-working, which might increase the use of industrial robots even further.

**Domestic Robots** A more recent history than industrial robots have domestic robots, e.g., vacuum cleaners or lawn mowers. A detailed overview for this field can be found in [5]. The development of such domestic robots, beginning mainly with cleaning robots, started in the early 1990’s. However, in the early days such robots were way too expensive for private households, such that it took until 2002 before a small and cheap vacuum cleaner, the “Roomba” from iRobot, was successful on the market. This was the start of the rapidly growing spread of such autonomous domestic robots throughout society which lasts until today and will probably go on in the future, Figure 1.1b.



(a) Autonomous vacuum cleaner.

(b) Autonomous lawn mower.

Figure 1.2: Examples for domestic robot types.

## 1.2 Domestic Robots - Guiding Research Questions

Vacuum cleaners and lawn mowers are certainly the most prominent representatives in the field of domestic robots, Figure 1.2. However, due to their low-cost design those domestic robots often only have limited sensing capabilities, e.g., lawn mowers where in most cases a perimeter wire sensor is used to signal the robot if it is on the lawn or not. Because of these limiting sensing capabilities, domestic robots often behave “stupid”, e.g., by performing a random walk, instead of acting “intelligent”, e.g., following a complete coverage path plan. These incapacities of domestic robots lead to different research questions concerning efficient navigation and planning algorithms as well as smart sensor development. Finding efficient solutions to these research questions is of great interest to the general public, as it allows household robots to be improved so that they can take on more and different tasks. Moreover, such systems can lay the foundation for the development of more complex systems which further increase the automation of everyday tasks. In the following, the research questions guiding through this thesis are introduced.

### **Q1.a: Is it possible to develop planning and navigation strategies for autonomous systems with only limited sensing capabilities?**

In order to optimally plan and navigate, a mobile robot first requires an intuition of the environments in which it acts, thus a map. Hence, the first part of the answer to this question must deal with efficient and accurate map estimation for those types of robots. This can be based on already existing methods, such as Pose Graph Optimization, where sufficient adjustments can be made. The second part of the answer must address the planning and navigation based on a known map of the environment. The main difficulty here is to deal with the inaccuracy of the position estimate and still perform an efficient complete coverage path planning procedure. Therefore, probabilistic coverage maps can

be generated to allow the robot, in combination with localization techniques, e.g., particle filter, to optimally cover its working area. Most of the mentioned techniques require users to tune hyper-parameters to adjust the algorithms to different environments, e.g., small versus large lawns. However, it would be preferable if robots would be able to act truly autonomously. This leads to the next, directly related research question.

### **Q1.b: Can reinforcement learning be used for hyper-parameter learning to enable true autonomy for lawn mowers?**

In order to allow the robot to adjust to any appropriate environment, it has to learn required hyper-parameter by itself, simply by exploring the environment. Therefore, reinforcement learning can be used where a cost function is required which changes dependent on the choice of hyper-parameters. Efficient optimization techniques can then be used, e.g., Bayesian Optimization, to find the optimal hyper-parameter setting for the given environment. Such a workflow allows then for truly autonomous systems. Being able now to cope with the limited sensing capabilities of domestic robots, the next step can be to adjust the sensing itself, which raises the next question.

### **Q2.a: How to efficiently exploit characteristics of the working environment for navigation and planning for autonomous lawn mowers?**

The most of the current autonomous lawn mowers use a perimeter wire for detecting the boundary lines of the lawn environment. This method neither exploits characteristics of the environments nor is it cost efficient, since a perimeter wire has to be buried alongside the boundary and constantly energized. A possible solution to replace this perimeter wire technique is to detect the chlorophyll fluorescence of the lawn. Therefore, an active, cost-sensitive chlorophyll sensor can be developed. Such sensors can then be used to efficiently determine the working area for autonomous lawn mowers. However, chlorophyll fluorescence is not a unique feature for lawn but for plants in general. This raises the question if we can also use the chlorophyll fluorescence detection to distinguish between different plant species.

### **Q2.b: Can machine learning approaches be used to learn to distinguish different plant species based on their chlorophyll fluorescence response?**

Distinguishing between different plant species can be crucial for real smart gardening, e.g., the detection of moss or weed in the lawn. Recent advances in deep learning methods allow to evaluate the complete chlorophyll emission spectrum for different plant species and learn an efficient classification approach. This shows if plant classification based on the plant emission spectra can work and how accurate such a classification can be. However, recording the complete spectra requires an expensive spectrometer. Thus, in a next step cost efficient approaches can be evaluated which are applicable to autonomous lawn mowers. Given all the above introduced concepts, e.g., map estimation, complete coverage path

planning, plant detection and classification, the question arises how autonomous robots applying these concepts can efficiently be monitored and controlled.

### **Q3: What are intuitive controlling and monitoring techniques for autonomous robots in outdoor environments?**

In order to efficiently control and monitor robots in outdoor environments it is favorable to use mobile devices. Nowadays, many robots are running with the Robot Operating System (ROS) as middleware, which provides hardware abstraction, extensive libraries, package management and many more. However, for ROS there are no sufficient good mobile applications available which fulfill all requirements. Those requirements are, beneath other, a general applicability to all ROS operated systems, a high customizability and extensibility, a user-friendly interface and a consistent integration into the mobile device workflow. Thus, in this thesis a new mobile application based on the Model View View-Model (MVVM) architecture is created which enables the user to intuitively interact with any robotic system. Here, the MVVM architecture allows a separation between the UI and the model logic and thus enables for a consistent mobile device workflow.

## **1.3 Organization of the Thesis**

The structure of this thesis is orientated on the above presented guiding research questions and divided into three chapters. Thereby, each chapter can be read on its own but is part of an overall narrative. Each chapter starts with discussing relevant related work followed by pointing out the scientific contributions and ends with a conclusion.

In Chapter 2, the research questions Q1.a and Q1.b are addressed. Therefore, a map generation procedure as well as a complete coverage path planning method for robots with only in-/outside area detectors are developed. Furthermore, methods for learning the hyper-parameters of the proposed algorithms are introduced.

Chapter 3 then deals with research questions Q2.a and Q2.b, where a low-cost sensor is developed which efficiently exploits the characteristics of the working environment, namely the chlorophyll fluorescence. This sensor allows then for efficiently in-/outside area detection of the lawn environment. Moreover, plant classification based on chlorophyll fluorescence is investigated to provide enhanced information, e.g., for better navigation and planning.

The last research question Q3 is addressed in Chapter 4. There, a mobile application is introduced to easily and efficiently control robots in outdoor environments with mobile devices, e.g., smartphones or tablets. In Chapter 5, the thesis is concluded.



## 2 Navigation & Planning

In this chapter, map generation and path planning methods for low-cost robots with only limited sensing capabilities are discussed. These methods use statistical evaluation and optimization techniques to generate reliable map estimates and motion plans. Application domains here include autonomous lawn mower or vacuum cleaner.

### Contents

---

<b>2.1</b>	<b>Introduction</b>	<b>8</b>
<b>2.2</b>	<b>Related Work</b>	<b>8</b>
2.2.1	Mapping & Navigation	9
2.2.2	Complete Coverage Path Planing	10
2.2.3	Hyper-Parameter Learning	11
<b>2.3</b>	<b>Scientific Contributions</b>	<b>11</b>
<b>2.4</b>	<b>Robot Motion Models</b>	<b>12</b>
2.4.1	Robot Setup	12
2.4.2	Velocity Motion Model	13
2.4.3	Odometry Motion Model	14
2.4.4	Model Parameter Determination	15
<b>2.5</b>	<b>Map Generation for closed Environments</b>	<b>18</b>
2.5.1	Pose Graph Representation	19
2.5.2	Loop Closure Detection	22
2.5.3	Hyper-Parameter Learning	25
2.5.4	Evaluation	28
<b>2.6</b>	<b>Path Planning under high Pose Uncertainty</b>	<b>32</b>
2.6.1	Probabilistic Coverage Map and Dynamical System	33
2.6.2	Particle Filter	35
2.6.3	Complete Coverage Path Planning	35
2.6.4	Evaluation	38
<b>2.7</b>	<b>Discussion</b>	<b>42</b>
2.7.1	Map Generation	42
2.7.2	Hyper-Parameter Learning	43

2.7.3 Complete Coverage Path Planning . . . . .	43
<b>2.8 Conclusion . . . . .</b>	<b>44</b>
2.8.1 Detailed Future Work . . . . .	44
<b>2.9 Acknowledgments . . . . .</b>	<b>44</b>

---

## 2.1 Introduction

As discussed in Chapter 1, domestic robots, such as lawn mowers, are becoming increasingly popular. However, most of these robots employ thereby simplistic navigation strategies, e.g., random walk, due to the lack of suitable maps and accurate sensors required for successful path planning. While most existing work for mapping and localization utilizes long-range sensors, such as Light Detection And Ranging (LiDAR) sensors, RGB-D cameras or time of flight sensors, domestic robots lack such sensor-richness. This is due to the cost-effective design of domestic robots, aiming for low acquisition and maintenance costs, or because certain sensors are not suitable for outdoor environments, e.g., due to reflections or direct sunlight. However, intelligent navigation and planning with low cost hardware is essential for domestic robots to realize their full potential. For example, autonomous lawn mowers employed with random walk policies are limited to simple environments, e.g., they cannot enter small corridors.

In this chapter, it is discussed to what extent optimal planning and navigation strategies can be found for domestic robots with limited sensing capabilities, especially with only in-/outside area sensing (Q1.a in Subsection 1.2). Furthermore, methods are highlighted to enable true autonomy for domestic robots by learning required hyper-parameters (Q1.b in Subsection 1.2).

The chapter begins with an overview of related work and a list of contributions, Section 2.2 and Section 2.3. In Section 2.4, the fundamentals required for the in this chapter proposed methods are introduced. This includes the general setup of the investigated robot as well as the required motion models. The actual method part starts with introducing a map generation method for closed environments, Section 2.5. Given such a map of the environment, a smart navigation and path planning strategy is proposed in Section 2.6. There, a probabilistic approach for efficient complete coverage path planning under high uncertainties is introduced. In Section 2.7, application limitations and implications for the proposed methods are discussed and in Section 2.8 the chapter concludes.

## 2.2 Related Work

In this section, different approaches for navigation, e.g., Simultaneous Localization and Mapping (SLAM), and different complete coverage path planning strategies, e.g., cellular decomposition methods such as the "Seed Spreader" algorithm, are discussed. Additionally, hyper-parameter learning techniques are reviewed.

### 2.2.1 Mapping & Navigation

Most of the existing work in the field of localization and mapping deals with the SLAM problem, for example FastSLAM [6], which is a SLAM approach based on particle filtering, or graph-based SLAM [7] which utilizes pose graphs. In general, these methods rely on long-range sensors, such as LiDAR sensors or cameras [8, 9, 10, 11], by using sensor fusion and probabilistic reasoning, e.g., particle filter [6] or extended kalman filter [12]. Nevertheless, there are some approaches which try to handle the SLAM problem using only sparse sensor data to avoid expensive sensing, e.g., [13] where the SLAM approach is modified by grouping multiple sparse scans for identifying features, which results in higher data density but comes with larger measurement uncertainty. For this approach, however, long-range sensor were used, even if only sparse data were provided. In comparison, existing work for low-range sensors, such as sonars or infrared sensors, require linear features, e.g., straight walls. Thus, they are mostly designed only for indoor robots and lay their focus on map generation. For example, in [14] and [15] mapping methods for low-cost robots equipped with low-range sensors utilizing those linear features are introduced. Both approaches rely on the Extended Kalman Filter (EKF), where the former requires more strict conditions, such that the environment must be indoor and rectangular. In [16], the sensing capability of the mobile robot is even further restricted utilizing only bumpers or wall sensors for applying wall following behavior for indoor mapping. Again, a rectangular environment is assumed for rectification of the recorded path data. Those conditions, linear features and/or a rectangular environment, are not necessarily present, especially not in outdoor environments. Hence, a mapping approach with more relaxed conditions is required. In [17], a map generation approach based the loop closure detected by returning to the home station is introduced. Thereby, the lawn mower was driving along the boundary wire while measuring movements with the wheel odometry. However, using only a single loop closure requires to distribute the error along all estimated positions equally.

For navigation, existing work can also be categorized into the above implied two extreme cases. Either computational and monetary expensive sensors like cameras or LiDAR sensors are installed [18, 19, 20] or simplistic navigation strategies such as a random walk are used, as in current household robots available for purchase [21]. When using limited sensing, only few navigation strategies have been proposed. In [22], a minimalist approach with contact sensors, a compass, angular and linear odometers in three different sensor configurations is used in order to determine how complex a sensor system of a robot really has to be in order to localize itself. An entropy based approach in order to determine uncertainty-reducing motions for active localization is proposed in [23]. Here, the localization problem is addressed for a blind robot with only a clock and a contact sensor. Probabilistic techniques are used to discretize the boundary of the environment into small cells. A probability  $P_{k,i}$  of the robot being in the cell  $i$  at the time step  $k$  is allocated to each cell and updated in every iteration. In [24], a localization method based on Monte Carlo Localization (MCL) [25] using only a single short-range sensor in a fixed position is introduced. The open challenge however that remains is how to efficiently localize a robot equipped only with binary sensors, for example for lawn mowers equipped with perimeter wire systems. Those

sensors only return the information whether the sensor is in the field or outside. The aforementioned approaches can either not be applied to such localization tasks because they require low-range sensors [24] or they are trying to solve the problem with even less sensor information [23].

### 2.2.2 Complete Coverage Path Planning

In the past, many Complete Coverage Path Planning (CCPP) strategies have been proposed. Those strategies can be partitioned into cellular decomposition, landmark-based or grid-based methods. For cellular decomposition methods, the free-space of the working area is divided into individual, non-overlapping cells such that the free-space is completely filled. Those cells are then covered by generating an efficient path through all cells, e.g., a “mowing the lawn” pattern. Two prominent examples are given with [26] and [27]. In the former, the problem of complete terrain acquisition with arbitrary shaped obstacles is addressed which produced the “Seed Spreader” algorithm. In the latter, the Boustrophedon Cell Decomposition (BCD) was developed which allows for non-polygonal obstacles within the operation space.

Landmark-based approaches use topological maps detecting natural landmarks for navigation and planning. Those methods are still utilizing the BCD. Such an algorithm is introduced in [28], where cellular decomposition is used for coverage path planning by generating a planar graph  $G$  with a set of nodes  $N$  and edges  $E$ . The overall algorithm is then designed as a finite state machine given the three states “boundary”, “normal” and “travel”. The method shows coverage accuracies of 99% in simulation and 85% for real robots.

The most famous class of CCPP algorithms utilizes grid maps which were firstly introduced in [29]. Those grid maps are simple to create and to maintain but suffer from exponential growth. One of the first methods applying CCPP on grid maps has been given in [30] where a complete coverage path is planned offline. For a detailed overview of CCPP algorithms, it is referred to [31].

Other approaches for CCPP focus on certain areas of application. For example, in [32] the authors proposed a CCPP method for agricultural machines, where trajectories are selected which guarantee complete coverage while minimizing overlapping. The best studied robots for CCPP are vacuum cleaners since they are widely used nowadays and they operate in a simplistic indoor setting. A complete setup for this type of robots, including a simple CCPP method, is presented in [33]. More studies covering efficient probabilistic robot cleaning strategies are presented in [34] and [35]. In the former, dirt grid maps are introduced which are modeled by Poisson Processes. Based on the modeled dirt distribution, a Traveling Salesman Problem (TSP) is solved for optimally cleaning the working space. In the latter, high-confidence cleaning guarantees under uncertainties are studied. Therefore, a particle filter is used to estimate the dirt distribution assigning a particle to each grid cell of the map. These particles are then updated based on random samples from the robots motion model. The robots path is then updated solving again a TSP.

For autonomous lawn mowers, little research considering efficient path planning methods has been done. In [36], this problem is addressed designing different planning methods with respect to minimal time or energy consumption. However, the proposed method requires an exact pose estimate for the robot utilizing real-time positioning system and self-navigation. Such precise positioning is not available for consumer lawn mowers, as these are only equipped with in-/outside area detection and odometry sensors. Also, most of the previously mentioned methods are not applicable to lawn mowers, since they mostly require remote sensing such as sonar. An approach for CCPP considering only contact sensors is presented in [37]. However, it requires a rectilinear structure of the working environment. Those limitations, either that remote sensing techniques are needed or assumptions of the structure of the environment have to be made, require new probabilistic approaches for CCPP to be applicable on autonomous lawn mowers.

### 2.2.3 Hyper-Parameter Learning

Mapping or SLAM methods, such as GMapping [38, 39], RTabMap [40, 41] or the in Subsection 2.2.1 mentioned approaches, e.g., FastSLAM [6], require the tuning of a large number of hyper-parameters. A correct setting of these parameters is crucial for the performance of these algorithms [42]. Usually, most of the mapping and navigation methods require the accurate setting of hyper-parameters, which might account for, e.g., the inaccuracies of the individual sensors. In general, finding convenient hyper-parameters for a certain mapping or navigation task requires a-prior knowledge on the structure of the environment and the robot itself. However, truly autonomous systems are expected to be able to adapt themselves to any environment and thus, being able to learn the required parameters autonomously. A well-known method for such meta-parameter learning problems is classical Reinforcement Learning (RL) [43], more specifically Bayesian Optimization (BO) [44, 45]. BO is a black box optimizer that only requires a definition of a cost function for efficiently learning required hyper-parameters. However, finding a proper definition of such a cost function is critical for the success of the parameter learning procedure. For mapping algorithms, a natural choice would be to define the cost as the difference between the estimated map and the respective ground truth. However, since ground truth is not known a priori other cost measures have to be developed for hyper-parameter learning.

## 2.3 Scientific Contributions

In the following, the novel scientific contributions presented within this chapter are summarized.

**Mapping & Navigation** A simple but efficient method for loop closure detection using only odometry sensors and in-/outside area detectors is proposed. Based on the detected

loop closures a map estimate can be generated using pose graph optimization techniques. The approach is tested in various realistic simulation environments as well as in different real garden settings. The proposed approach enables robots with limiting sensing capabilities, e.g., in-/outside area detection sensors, to operate intelligently by considering a generated map estimate for both path planning and localization.

**Complete Coverage Path Planning** A probabilistic CCP approach for closed environments applicable to autonomous lawn mowers and other low-cost systems is proposed which can cope with high uncertainties regarding the pose estimates. For this purpose, the neural network based CCP approach from [46, 47] is adapted and combined with ideas for high-confidence cleaning guarantees for vacuum cleaners from [35]. The contributions thereby are three-fold: (1) Adaptation of a neural net planning method for highly uncertain pose estimates, (2) a definition of a probabilistic coverage map and (3) a demonstration of the proposed method in different realistic simulation scenarios.

**Hyper-Parameter Learning** A RL scheme for learning the hyper-parameters for the introduced mapping approach is proposed. Therefore, a cost function is defined which utilizes the fact that the robot is operating in a closed environment.

## 2.4 Robot Motion Models

As the basis for the development and evaluation of the different methods for mapping, navigation and planning, a differential drive robot is used. In this section, first the general robot setup is defined, e.g., the sensing capabilities and the actuators, followed by the velocity and odometry motion models [48]. The velocity motion model requires thereby that the robot is controllable through a translational and an angular velocity whereby the odometry motion model relies on wheel encoder information. Thus, the velocity motion model, while in general less accurate than the odometry motion model, can be used for path planning whereas information from the odometry motion model is only available after the robots movement and can be used for filtering algorithms. Here, the velocity motion model is used for simulating the differential drive robot and the odometry motion model for pose estimation methods, e.g., the particle filter. At the end of this section, the model parameters are determined based on recorded real data using Maximum Likelihood Estimation (MLE) [49].

### 2.4.1 Robot Setup

The differential drive robot used throughout this thesis, an adjusted Viking MI 422P, has two actuators which drive the wheels. Position encoders (Hall-sensors) provide the number of ticks  $n_r$  and  $n_l$  counted during one iteration. Given the number of ticks per complete rotation  $n_{\text{Hall}}$  and the diameters of the wheels  $d_r$  and  $d_l$ , the unrolled distance by each

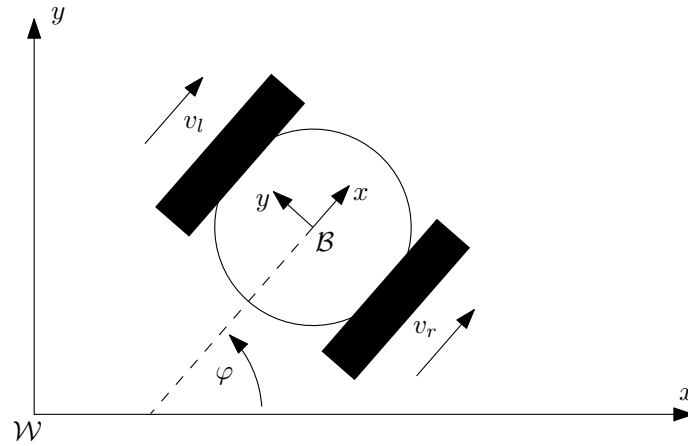


Figure 2.1: A differential drive robot presented in the world frame  $\mathcal{W}$  with the body frame  $\mathcal{B}$ .

wheel is

$$l_r = \frac{d_r \pi n_r}{n_{\text{Hall}}}, \quad l_l = \frac{d_l \pi n_l}{n_{\text{Hall}}}. \quad (2.1)$$

The actuators are velocity controlled via Serial Peripheral Interface (SPI) by the main processor, where a motor driver provides the conversion into the required motor signals. Sensors, e.g., chlorophyll fluorescence sensors or boundary wire sensors, are placed right and left at the front of the robot and are directly connected via Universal Serial Bus (USB) with the main processor, which is a Raspberry Pi 3B+. The operating system running on the Raspberry Pi is Raspberry Pi OS (previously called Raspbian). The robot control is realized by means of the Robot Operating System (ROS). A right-handed coordinate system with the  $x$ -axis pointing forward, the  $y$ -axis left and the  $z$ -axis up is defined for the differential drive robot, Figure 2.1.

### 2.4.2 Velocity Motion Model

For the velocity controlled differential drive robot the input signals to the system are the translational velocity in body frame  $x$ -direction,  $v$ , as well as the angular velocity around the  $z$ -axis,  $\omega$ . Given the velocities for the right and left wheel,  $v_r$  and  $v_l$ ,

$$v = \frac{v_r + v_l}{2}, \quad \omega = \frac{v_r - v_l}{2L}. \quad (2.2)$$

Performing a backward calculation leads to

$$v_r = L\omega + v, \quad v_l = L\omega - v. \quad (2.3)$$

Here,  $2L$  is the wheelbase of the differential drive robot. Given the velocities  $v$  and  $\omega$ , a first order system of differential equations can be drawn to represent the kinematic of the

differential drive robot

$$\begin{aligned}\dot{x} &= v \cos(\varphi), \\ \dot{y} &= v \sin(\varphi), \\ \dot{\varphi} &= \omega.\end{aligned}\tag{2.4}$$

Discretizing this first order system with time step  $\Delta t = t_i - t_{i-1}$  and under the assumption of constant  $v$  and  $\omega$  during a time period  $\Delta t$ , yields

$$\begin{aligned}x_i &= x_{i-1} + \frac{v}{\omega} [\sin(\varphi_i) - \sin(\varphi_{i-1})], \\ y_i &= y_{i-1} + \frac{v}{\omega} [\cos(\varphi_{i-1}) - \cos(\varphi_i)], \\ \varphi_i &= \varphi_{i-1} + \omega \Delta t.\end{aligned}\tag{2.5}$$

To ensure correct approximation for non-constant velocities,  $\Delta t$  should be chosen small. Also, notice that Equation (2.5) has a singularity at  $\omega \rightarrow 0$ . Here, the robot does not turn, thus moves in a straight line and the circle radius of the robot motion  $r = \|\frac{v}{\omega}\|$  goes to infinity. In order to avoid this singularity, Equation (2.4) can be solved for  $\omega = 0$  which leads to

$$\begin{aligned}x_i &= x_{i-1} + v \cos(\varphi_{i-1}) \Delta t, \\ y_i &= y_{i-1} + v \sin(\varphi_{i-1}) \Delta t, \\ \varphi_i &= \varphi_{i-1}.\end{aligned}\tag{2.6}$$

Equation (2.5) and Equation (2.6) are derived for an ideal robot. However, robot motion is in general subject to noise. Here, the noise for the translational and angular velocities are modeled as zero-mean Gaussian variables as proposed by [48]

$$\begin{aligned}\hat{v} &= v + \epsilon_v & \epsilon_v &\sim \mathcal{N}(0, \alpha_1|v| + \alpha_2|\omega|) \\ \hat{\omega} &= \omega + \epsilon_\omega & \epsilon_\omega &\sim \mathcal{N}(0, \alpha_3|v| + \alpha_4|\omega|).\end{aligned}\tag{2.7}$$

Inserting these noisy velocities to the motion model from Equation (2.5) leads to

$$\begin{aligned}x_i &= x_{i-1} + \frac{\hat{v}}{\hat{\omega}} [\sin(\varphi_{i-1} + \hat{\omega} \Delta t) - \sin(\varphi_{i-1})], \\ y_i &= y_{i-1} + \frac{\hat{v}}{\hat{\omega}} [\cos(\varphi_{i-1}) - \cos(\varphi_{i-1} + \hat{\omega} \Delta t)], \\ \varphi_i &= \varphi_{i-1} + \hat{\omega} \Delta t.\end{aligned}\tag{2.8}$$

The noise parameters  $\alpha_1, \dots, \alpha_4$  are robot specific and have to be determined, for example using maximum likelihood estimation.

### 2.4.3 Odometry Motion Model

A different approach to estimate the pose of a differential drive model is to use an odometry model. Here, the wheel movement data,  $l_r$  and  $l_l$ , provided by the odometers of the wheels

are used to estimate the current pose

$$\mathbf{p}_i = \begin{bmatrix} x_i \\ y_i \\ \varphi_i \end{bmatrix} = \begin{bmatrix} x_{i-1} \\ y_{i-1} \\ \varphi_{i-1} \end{bmatrix} + \begin{bmatrix} \cos(\varphi) & 0 \\ \sin(\varphi) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta \varphi \end{bmatrix} \quad (2.9)$$

given the previous pose  $\mathbf{p}_{i-1} = [x_{i-1}, y_{i-1}, \varphi_{i-1}]^\top$ . The translational and angular movements are calculated as  $\Delta s = \frac{l_r + l_l}{2}$  and  $\Delta \varphi = \frac{l_r - l_l}{2L}$ . A convenient noise model, as proposed by [48], requires the division of the robot's movement into an initial rotation  $\delta_{R1}$ , a straight translatorial movement  $\delta_T$  and a concluding rotation  $\delta_{R2}$  as depicted in Figure 2.2. With  $\Delta \mathbf{p}_i = \mathbf{p}_i - \mathbf{p}_{i-1}$ , the individual movement components can be derived as

$$\begin{aligned} \delta_{R1} &= \text{atan2}(\Delta p_{i,y}, \Delta p_{i,x}) - p_{i-1,\varphi} \\ \delta_T &= \sqrt{\Delta p_{i,x}^2 + \Delta p_{i,y}^2} \\ \delta_{R2} &= \Delta p_{i,\varphi} - \delta_{R1}. \end{aligned} \quad (2.10)$$

Note, that if  $\delta_T$  is close to zero, it may come to instabilities calculating  $\delta_{R1}$  which may lead to unrealistic high rotations. A practical solution is to catch any case where  $\delta_T \rightarrow 0$  and set  $\delta_{R1} = \delta_{R2} = \frac{\Delta p_{i,\varphi}}{2}$ . Another case which has to be addressed is the backward motion of the robot where  $\Delta s < 0$ . Equation (2.10) turns then into

$$\begin{aligned} \delta_{R1} &= \text{atan2}(-\Delta p_{i,y}, -\Delta p_{i,x}) - p_{i-1,\varphi} \\ \delta_T &= -\sqrt{\Delta p_{i,x}^2 + \Delta p_{i,y}^2} \\ \delta_{R2} &= \Delta p_{i,\varphi} - \delta_{R1}. \end{aligned} \quad (2.11)$$

In order to account for noise in the odometry data, random Gaussian samples with zero mean are added to the motion components  $\delta_{R1}$ ,  $\delta_T$  and  $\delta_{R2}$  as proposed in [48]:

$$\begin{aligned} \hat{\delta}_{R1} &= \delta_{R1} + \epsilon_{R1} & \epsilon_{R1} &\sim \mathcal{N}(0, a_1|\delta_{R1}| + a_2|\delta_T|) \\ \hat{\delta}_T &= \delta_T + \epsilon_T & \epsilon_T &\sim \mathcal{N}(0, a_3|\delta_T| + a_4(|\delta_{R1}| + |\delta_{R2}|)) \\ \hat{\delta}_{R2} &= \delta_{R2} + \epsilon_{R2} & \epsilon_{R2} &\sim \mathcal{N}(0, a_1|\delta_{R2}| + a_2|\delta_T|). \end{aligned} \quad (2.12)$$

The noise parameters  $a_1, \dots, a_4$  are robot specific and have to be determined, for example using maximum likelihood estimation.

#### 2.4.4 Model Parameter Determination

The velocity as well as the odometry motion model require accurate estimates for the noise parameters  $\alpha_1, \dots, \alpha_4$  and  $a_1, \dots, a_4$  respectively in order to allow for realistic motion estimation. Therefore, MLE can be used for determining those parameters and the results can be validated utilizing the Kolmogorow-Smirnow-Test [50].

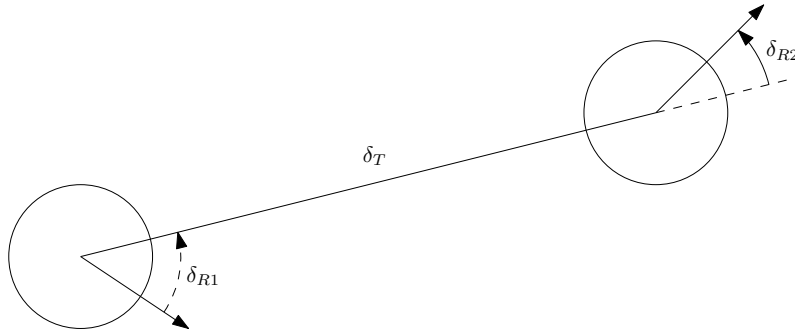


Figure 2.2: Odometry model where the movement of the robot is divided into two rotations  $\delta_{R1}$ ,  $\delta_{R2}$  and a straight translatorial movement  $\delta_T$ . This sketch has been redrawn from [48].

**Maximum Likelihood Estimation** Let there be  $N$  measurements with  $\mathbf{y}_{1:N}$  being the true values and  $\hat{\mathbf{y}}_{1:N}$  being the estimated values given by a certain motion model, e.g.,  $[\delta_{R1}, \delta_T, \delta_{R2}]_{1:N}$  and  $[\hat{\delta}_{R1}, \hat{\delta}_T, \hat{\delta}_{R2}]_{1:N}$ . Let further the error between the estimate and the true value be normally distributed with zero mean and variance  $\Sigma = \text{diag}[\sigma_1^2, \dots, \sigma_M^2]$

$$\boldsymbol{\epsilon} = \hat{\mathbf{y}} - \mathbf{y} \sim \mathcal{N}(\mathbf{0}, \Sigma). \quad (2.13)$$

The variances  $\sigma_j^2 = \sigma_j^2(\mathbf{y}, \boldsymbol{\theta})$  are functions of the measurements  $\mathbf{y}$  and the model parameters  $\boldsymbol{\theta}$ , e.g.,  $\boldsymbol{\theta} = [a_1, \dots, a_4]$ ,  $\sigma_{R1}^2 = a_1|\delta_{R1}| + a_2|\delta_T|$ . The probability density distribution for the error  $\boldsymbol{\epsilon}$  is the product

$$f = \prod_{j=1}^M f_j \quad \text{with} \quad f_j(\epsilon_j | \sigma_j^2) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{1}{2} \left[\frac{\epsilon_j}{\sigma_j}\right]^2\right). \quad (2.14)$$

The likelihood given the  $N$  measurements is the product of their marginal densities

$$\mathcal{L} = \prod_{i=1}^N \prod_{j=1}^M f_{j,i}(\epsilon_{j,i} | \sigma_j^2(\mathbf{y}_i, \boldsymbol{\theta})) = \prod_{j=1}^M \prod_{i=1}^N f_{j,i}(\epsilon_{j,i} | \sigma_j^2(\mathbf{y}_i, \boldsymbol{\theta})). \quad (2.15)$$

Taking the logarithm of the likelihood results in

$$\ln \mathcal{L} = \sum_{j=1}^M \sum_{i=1}^N l_{j,i} + M \frac{N}{2} \ln(2\pi) \quad \text{with} \quad l_{j,i} = -\ln(\sigma_j^2(\mathbf{y}_i, \boldsymbol{\theta})) - \frac{1}{2} \left[\frac{\epsilon_{j,i}}{\sigma_j(\mathbf{y}_i, \boldsymbol{\theta})}\right]^2. \quad (2.16)$$

In order to generate estimates for the model parameters  $\boldsymbol{\theta}$ , Equation (2.16) has to be maximized with respect to those parameters. Therefore, global optimization methods such as pattern search [51] can be used.

**Kolmogorow-Smirnow-Test** In order to validate the parameter estimation results, a Kolmogorow-Smirnow-Test can be used to check if an unseen data set  $[\mathbf{y}_{1:K}^*, \hat{\mathbf{y}}_{1:K}^*]$  leads to the

Table 2.1: Estimated motion model parameters and measured robot parameters.

	Kinematic Model	Odometry Model		Robot
$\alpha_1/a_1$	0.011230	0.002361	$d_r$	0.215
$\alpha_2/a_2$	0.003417	0.000346	$d_l$	0.215
$\alpha_3/a_3$	0.193604	0.000223	$L$	0.1825
$\alpha_4/a_4$	0.180664	0.000069	$n_{\text{Hall}}$	703

desired normal distribution. Therefore, the empirical data first has to be adjusted based on the assumed variance  $\sigma_j^2(\mathbf{y}_k^*, \boldsymbol{\theta})$ ,

$$\bar{\epsilon}_j = \frac{\hat{y}_j^* - y_j^*}{\sigma_j^2(\mathbf{y}_k^*, \boldsymbol{\theta})}. \quad (2.17)$$

Let  $\bar{\epsilon}_1 \leq \bar{\epsilon}_2 \leq \dots \leq \bar{\epsilon}_K$  without loss of generality,  $F_{\text{emp}}(\bar{\epsilon}_k)$  be an empirical distribution function

$$F_{\text{emp}}(\bar{\epsilon}) = \frac{1}{K} \sum_{k=1}^K I_{-\infty, \bar{\epsilon}}(\bar{\epsilon}_k) \quad \text{with} \quad I_{-\infty, \bar{\epsilon}}(\bar{\epsilon}_k) = \begin{cases} 1 & \bar{\epsilon}_k \leq \bar{\epsilon} \\ 0 & \bar{\epsilon}_k > \bar{\epsilon} \end{cases} \quad (2.18)$$

and  $F_0(\bar{\epsilon}_k)$  the cumulative probability distribution of the null hypothesis,  $\mathcal{N}(0, 1)$ ,

$$F_0(\bar{\epsilon}_k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\bar{\epsilon}} \exp\left(-\frac{t^2}{2}\right) dt. \quad (2.19)$$

Then the upper and lower differences

$$d_{u,k} = |F_{\text{emp}}(\bar{\epsilon}_k) - F_0(\bar{\epsilon}_k)| \quad \text{and} \quad d_{l,k} = |F_{\text{emp}}(\bar{\epsilon}_{k-1}) - F_0(\bar{\epsilon}_k)| \quad (2.20)$$

are calculated for all data point  $k = 1, \dots, K$  and compared to the critical value  $d_\alpha$ . If  $d_{\max}$ , thus the maximum value for all  $d_{u,k}$  and  $d_{l,k}$  is larger than  $d_\alpha$ , the null hypothesis will be rejected. The critical value can be approximately calculated using the significance level  $\alpha$  and the size of the data set  $K$  as

$$d_\alpha = \frac{\sqrt{-0.5 \ln\left(\frac{\alpha}{2}\right)}}{\sqrt{K}}. \quad (2.21)$$

**Motion Model Parameters** In order to generate training and test data for the MLE and the Kolmogorow-Smirnow-Test, the lawn mower was driven in hand steering mode over a test lawn. Thereby, the exact pose of the lawn mower was tracked using a motion capture system<sup>1</sup>. The motion capture data were recorded and freed of measurement errors in [52]. Based on these data, the errors  $[\epsilon_v, \epsilon_\omega]$  for the velocity motion model and  $[\epsilon_{R1}, \epsilon_T, \epsilon_{R2}]$  for

<sup>1</sup>OptiTrack

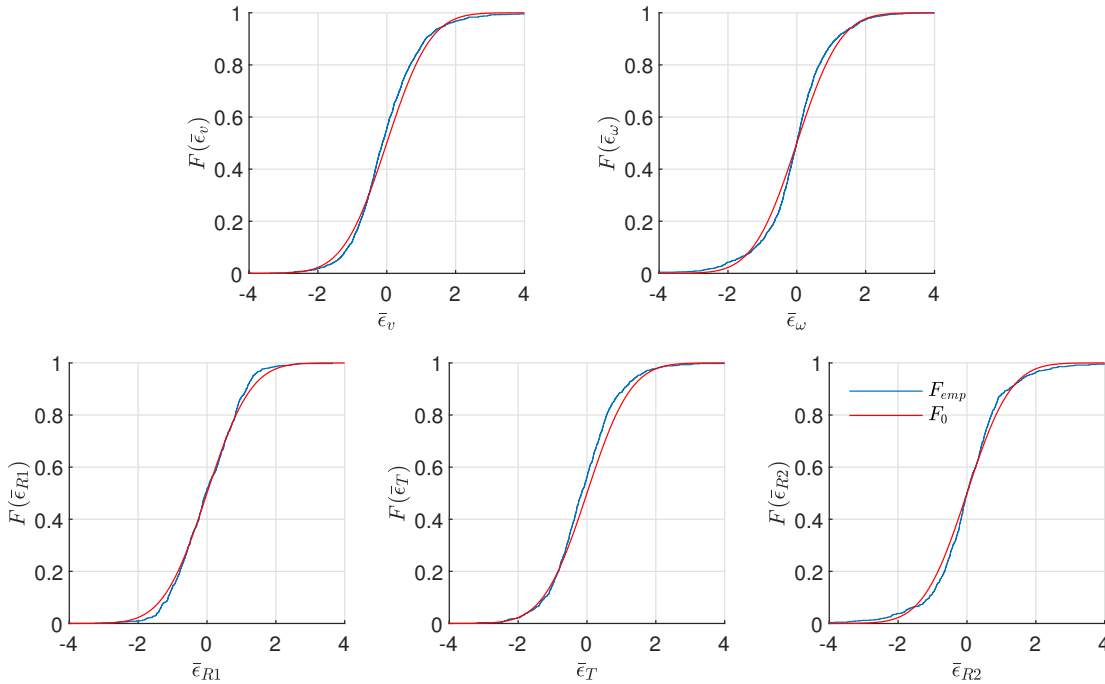


Figure 2.3: Cumulative distribution functions  $F_{emp}$  for the empirical error data  $[\bar{\epsilon}_v, \bar{\epsilon}_\omega]$  and  $[\bar{\epsilon}_{R1}, \bar{\epsilon}_T, \bar{\epsilon}_{R2}]$ , normalized with respect to the expected variances. All tests did not reject the null hypothesis of a standard normal distribution,  $\mathcal{N}(0, 1)$ , with the cumulative distribution function  $F_0$ .

the odometry motion model can be determined. A MLE leads then to the in Table 2.1 noted model parameter values. Thereby, the data set was divided into a training set for the MLE with 95 % of the data and a test set for the Kolmogorow-Smirnow-Test with the rest 5 % of the data. The Kolmogorow-Smirnow-Test confirmed for all parameter settings a match with the desired normal distribution. In Figure 2.3, comparisons between the empirical cumulative distribution functions for the five different error measures and the cumulative distribution function of the standard normal distribution are shown.

## 2.5 Map Generation for closed Environments

The first step towards development of optimal planning and navigation strategies requires an efficient and accurate map estimate generation. In this section, an intelligent mapping method for closed environments for low cost systems with limited sensing capabilities is introduced. The proposed method utilizes the well known Pose Graph Optimization (PGO) technique by providing an efficient loop closure detection scheme. Therefore, the robot cycles along the boundary line of the closed environment utilizing a wall following

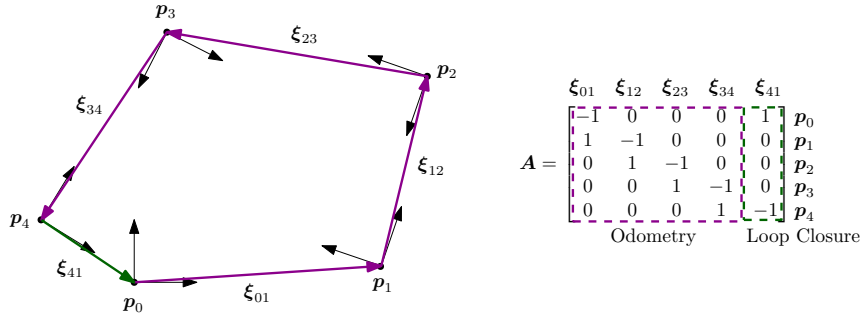


Figure 2.4: Pose graph with five vertices connected with five edges. Four of the edges are odometric constraints and one is a loop closing constraint. On the right, the incidence matrix is shown divided into the parts containing the odometric constraints and the loop closing constraints.

approach (Appendix, Section A) to generate odometry data which are then used for the PGO and for the loop closure detection. The section starts with introducing the pose graph representation and optimization in general, followed by the proposed approach for loop closure detection.

### 2.5.1 Pose Graph Representation

The pose graph representation was firstly introduced by Lu and Milios 1997, [53]. Let  $\mathbf{p} = \{\mathbf{p}_0, \dots, \mathbf{p}_N\}$  be a set of  $N + 1$  poses representing the position and orientation of a mobile robot in a two dimensional space, hence  $\mathbf{p}_i = [\mathbf{x}_i^\top, \varphi_i]^\top$ . Here,  $\mathbf{x}_i \in \mathbb{R}^2$  is the cartesian position of the robot and  $\varphi_i \in [-\pi, \pi]$  the corresponding orientation as an Euler angle with the integer  $i = 0:N$ . The relative measurement between two poses  $i$  and  $j$  is then given as

$$\boldsymbol{\xi}_{ij} = \begin{bmatrix} \mathbf{R}_i^\top(\mathbf{x}_j - \mathbf{x}_i) \\ \varphi_j - \varphi_i \end{bmatrix} = \mathbf{p}_j \ominus \mathbf{p}_i, \quad (2.22)$$

where  $\mathbf{R}_i = \mathbf{R}_i(\varphi_i)$  is a planar rotation matrix and  $\ominus$  the pose compounding operator which was introduced in [53]. These relative measurements are, in general, affected by noise. Thus, including a zero mean Gaussian noise  $\boldsymbol{\epsilon}_{ij} \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_{ij})$  leads to the noisy relative measurements

$$\hat{\boldsymbol{\xi}}_{ij} = \boldsymbol{\xi}_{ij} + \boldsymbol{\epsilon}_{ij}. \quad (2.23)$$

In general, there are two different types of relative pose measurements: Odometric constraints and loop closing constraints. Here, the first constraints are generated by the wheel odometry of the differential drive robot. The second type of constraints are provided by the robot recognizing a match between actual measurements and past measurements by revisiting places. Subsection 2.5.2 introduces how to efficiently identify and add those loop closing constraints to the pose graph for robots with limited sensing in closed environments.

The pose graph is thereby represented as a directed graph  $\mathcal{G}(\mathcal{V}, \mathcal{E})$  with  $N+1$  vertices and  $N+M$  edges, where  $N$  is the number of odometric constraints and  $M$  the number of loop closing constraints. The connection between the vertices by the edges can be compactly written using an incident matrix  $\mathbf{A}$ . There, every column represents an edge connecting two vertices with each other. The row number thereby represents the vertex from which the edge starts, denoted by  $-1$ , and the vertex where the edge leads to, denoted by  $1$ . In Figure 2.4 a pose graph is exemplarily shown in combination with the according incidence matrix  $\mathbf{A}$ .

**Path Segmentation/Data Pruning** During the mapping procedure, a vast amount of odometry data is collected which results into a path with many data points. For example, a robot sampling with 20Hz generates 1200 path points per minute. This makes PGO intractable such that path segmentation/data pruning strategies are required to reduce the number of data points. Therefore, path points are clustered into straight line segments similarly as in [16].

Assume the position estimates based on the odometry are given as  $X = \{\mathbf{x}_0, \dots, \mathbf{x}_n\}$ . A set of dominant points is initialized as  $DP = \{\mathbf{x}_0\}$  and a temporarily subset as  $S = \{\mathbf{x}_0\}$ . The set of dominant points  $DP$  represents the pruned data whereas the subset  $S$  contains the data points that are currently being pruned. Successively, all odometry data points are included to the temporary subset  $S \leftarrow \{S, \mathbf{x}\}$  whereby with each integration of a new data point  $\mathbf{x}$  the following conditions are checked:

$$L_{\min} > \|DP_{\text{end}} - S_{\text{end}}\| \quad (2.24)$$

and

$$e_{\max} > \frac{1}{|S| - 2} \sum_{i=2}^{|S|-1} e_i \quad (2.25)$$

with  $DP_{\text{end}}$  and  $S_{\text{end}}$  being the last items in the respective sets,  $e_i$  the shortest distance between the point  $S_i$  and the vector  $v = S_{\text{end}} - S_1$  and  $|S|$  the cardinality of the set  $S$ . The adjustable parameters  $L_{\min}$  and  $e_{\max}$  reflect the minimum pruning length and the maximum pruning error respectively. If both conditions are true, thus the minimum pruning length is reached and the maximum pruning error is exceeded,  $S_{\text{end}-1}$  is added to the set  $DP \leftarrow \{DP, S_{\text{end}-1}\}$  and the temporary subset  $S$  is set back to  $S = \{S_{\text{end}}, \mathbf{x}\}$ . An example data set with original odometry and pruned data is shown in Figure 2.5.

Based on the pruned data set  $DP$ , the poses for the pose graph are generated as

$$\mathbf{p} = \{[DP_1^\top, \varphi_1]^\top, \dots, [DP_{|DP|-1}^\top, \varphi_{|DP|-1}]^\top\}, \quad (2.26)$$

where  $|DP|$  is the cardinality of the set of dominant points and

$$\varphi_i = \text{atan2}(v_{i,y}, v_{i,x}) \quad \text{with} \quad \mathbf{v}_i = DP_{i+1} - DP_i. \quad (2.27)$$

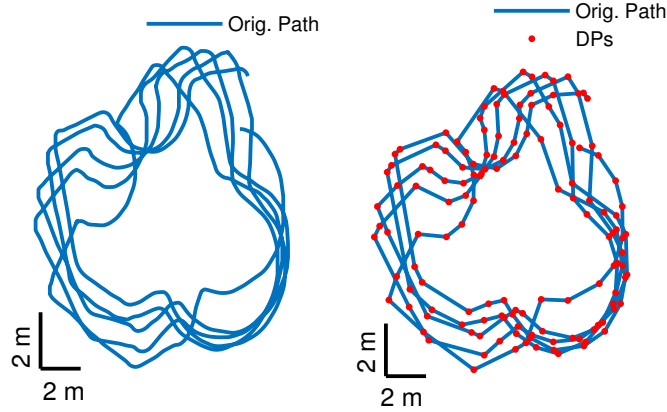


Figure 2.5: Path Segmentation: In the left panel the original odometry data with 37687 data points is shown. In the right panel the path segmentation with 137 dominant points, marked as red dots, is presented.

For the pose graph,  $N = |DP| - 2$  and the relative measurements  $\hat{\xi}$  can be calculated using Equation (2.22). An example for the generation of the pose graph based on the set of dominant points is shown in Figure 2.6.

**Pose Graph Optimization** Given the pose graph with  $N + 1$  poses,  $N$  odometric and  $M$  loop closing constraints, the overall optimization problem is to minimize the sum of weighted residual errors  $\mathbf{r}_{ij}(\mathbf{p})$  with respect to the pose estimates  $\mathbf{p}$

$$\min_{\mathbf{p}} \sum_{(i,j) \in \mathcal{E}} \|\mathbf{r}_{ij}(\mathbf{p})\|_{\mathbf{P}_{ij}}^2 \quad (2.28)$$

with

$$\|\mathbf{r}_{ij}(\mathbf{p})\|_{\mathbf{P}_{ij}}^2 = [(\mathbf{p}_j \ominus \mathbf{p}_i) - \hat{\xi}_{ij}]^\top \mathbf{P}_{ij}^{-1} [(\mathbf{p}_j \ominus \mathbf{p}_i) - \hat{\xi}_{ij}]. \quad (2.29)$$

The covariance matrix corresponding to the relative measurements  $\hat{\xi}_{ij}$  is thereby given as  $\mathbf{P}_{ij}$ . Rewriting Equation (2.28) with  $\mathbf{e}_{ij}(\mathbf{p}_i, \mathbf{p}_j) = (\mathbf{p}_j \ominus \mathbf{p}_i) - \hat{\xi}_{ij}$  and  $\mathbf{\Omega}_{ij} = \mathbf{P}_{ij}^{-1}$  as the information matrix leads to the more compact form

$$\min_{\mathbf{p}} \sum_{(i,j) \in \mathcal{E}} \mathbf{e}_{ij}^\top \mathbf{\Omega}_{ij} \mathbf{e}_{ij}. \quad (2.30)$$

Here, covariance matrices for the odometric constraints and covariance matrices for the loop closing constraints must be distinguished. Where the former can be modeled as demonstrated in Section 2.4.3, the determination of the latter covariance matrices is method specific and will be addressed in the following sections.

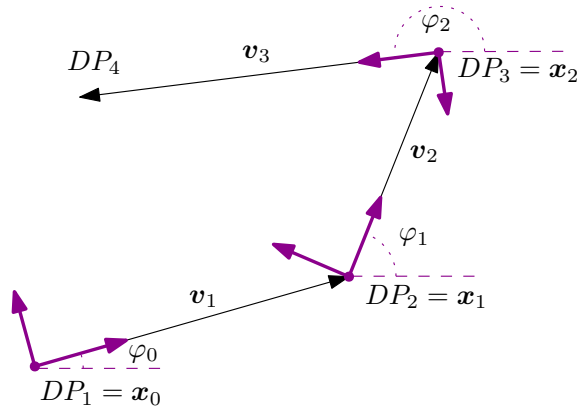


Figure 2.6: The figure shows how the dominant points are transformed to a set of poses. Thereby, the procedure starts with the first dominant point as the initial pose given by  $\mathbf{p}_0 = [\mathbf{x}_0^\top, \varphi_0]^\top$  with  $\mathbf{x}_0 = DP_1$  and  $\varphi_0 = \text{atan2}(\mathbf{v}_{1,y}, \mathbf{v}_{1,x})$ ,  $\mathbf{v}_1 = DP_2 - DP_1$ .

The pose graph optimization problem, Equation (2.30), has been addressed in several studies, e.g., TORO [54], g2o [55], iSAM2 [56] and LAGO [57]. Thereby, TORO is based on gradient descent and is an extension of Olson’s algorithm [58]. It applies a tree-based parameterization to distribute residual errors across the graph that improves the performance. The “general graph optimization” framework, g2o, has been designed to perform the optimization of different least squares problems, which can be represented as a graph. It thereby relies on the Gauss-Newton method. The iSAM2 method applies Bayes trees using incremental variable re-ordering and fluid relinearization to solve sparse nonlinear incremental optimization problems. LAGO addresses the pose graph optimization problem by decoupling the orientation and translation. Here, the simpler Levenberg-Marquardt algorithm is used, which works reasonably well for the addressed pose graph optimization problem. The Levenberg-Marquardt algorithm approximates the function by its first order Taylor expansion around the current initial guess  $\hat{\mathbf{p}}$  in order to then find the solution to the optimization problem iteratively. A detailed description how to apply the Levenberg-Marquardt algorithm for the given PGO problem is given in [7].

### 2.5.2 Loop Closure Detection

The idea for efficiently detecting loop closures with only limited sensing capabilities is to cycle along the boundary line of the closed environment. Based on the so generated pose graph, loop closing constraints are detected by comparing the shape of the neighborhood regions of each vertex with another. Therefore, a piecewise linear function

$$\theta(x) = \phi_i \quad \text{for } l_{i-1} \leq x < l_i, \quad i = 0, 1, \dots, N \quad (2.31)$$

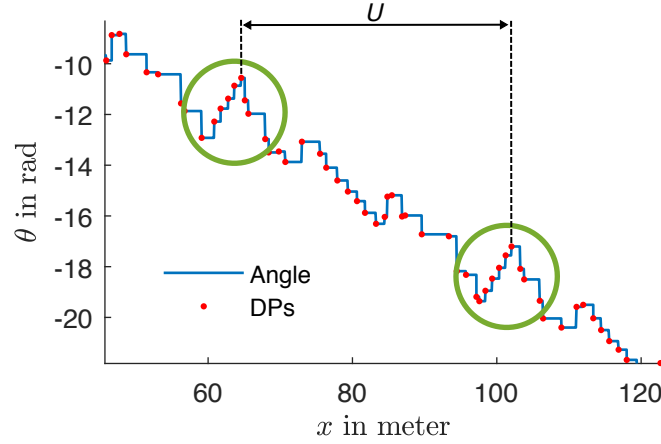


Figure 2.7: Example for the piecewise linear orientation function  $\theta(x)$ . The green circled regions show similar path segments. The vertices or dominant points (DPs) of the pose graph are pictured as red dots. The estimated circumference  $U$  for the closed environment is exemplarily depicted for a possible loop closing pair.

representing the shape of the pose graph is constructed by accumulating the orientation and distance differences between the poses

$$\begin{aligned}\phi_i &= \phi_{i-1} + \Delta\phi_i \\ l_i &= l_{i-1} + \|\mathbf{v}_i\|.\end{aligned}\tag{2.32}$$

Here,  $\mathbf{v}_i = \mathbf{x}_i - \mathbf{x}_{i-1}$ ,  $\Delta\phi_i = \phi_i - \phi_{i-1}$  with  $\mathbf{p}_i = [\mathbf{x}_i^\top, \phi_i]^\top$  and the procedure starts with  $\phi_0 = \varphi_0$  and  $l_0 = 0$ . An example for such a piecewise orientation function based on the data depicted in Figure 2.5 is shown in Figure 2.7.

By defining the neighborhood of a vertex  $i$  as  $[l_i - L_{\text{NH}}, l_i + L_{\text{NH}}]$ , a comparison error between two vertices  $i$  and  $j$  is given as

$$\mathbf{C}_{ij} = \int_{-L_{\text{NH}}}^{+L_{\text{NH}}} |[\theta(l_i + x) - \phi_i] - [\theta(l_j + x) - \phi_j]| dx.\tag{2.33}$$

Rewriting Equation (2.33) as a sum over  $m$  linearly distributed evaluation points  $x_k$  leads to

$$\mathbf{C}_{ij} = \frac{1}{m} \sum_{k=1}^m |[\theta(l_i + x_k) - \phi_i] - [\theta(l_j + x_k) - \phi_j]| \tag{2.34}$$

with  $x_1 = -L_{\text{NH}}$ ,  $x_m = +L_{\text{NH}}$ . An example error matrix representing the comparison errors for the in Figure 2.7 introduced orientation function is shown in Figure 2.8. Based on the Comparison error  $C_{ij}$ , a loop closing pair  $SP_k = \{\mathbf{p}_i, \mathbf{p}_j\}$  for  $i \neq j$  is defined as a local minimum of  $C_{ij}$  for which holds  $C_{ij,\text{min}} < c_{\text{max}}$ . A local minimum represents thereby

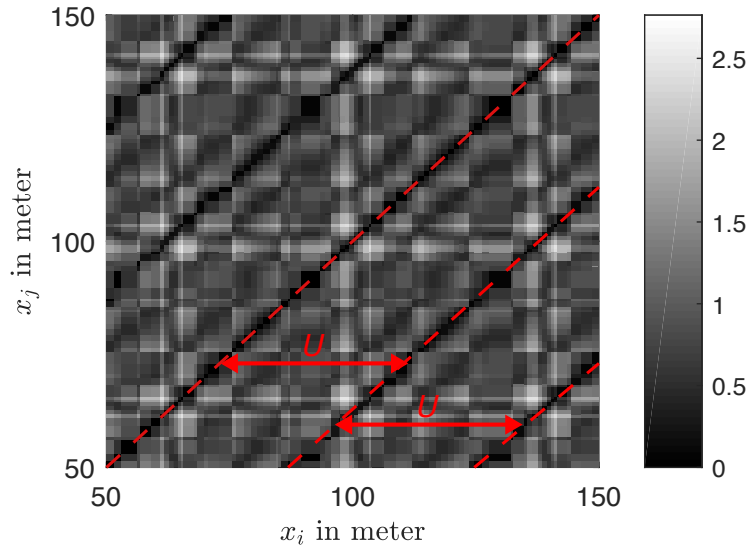


Figure 2.8: Comparison error of the shapes of the neighborhood between the vertices of the pose graph. For better reading the error is plotted in the form  $\log(\mathbf{C}_{ij} + 1)$  and only a section of the matrix is shown. The variables  $x_i$  and  $x_j$  are representing the position  $l$  of the vertices  $i$  and  $j$  in meter along the pose graph. The estimated circumference  $U$  for the closed environment can be read directly from the graphic.

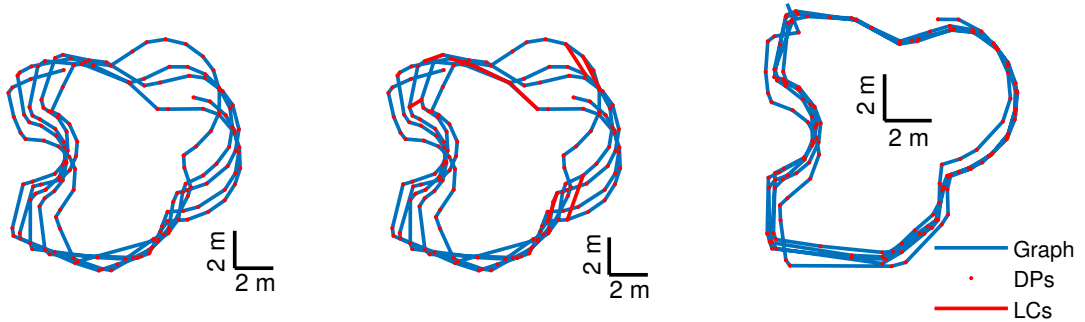
the best possible loop closure in a certain region of the error matrix and the threshold  $c_{\max}$  ensures that not every local minimum is selected as loop closing pair, but only sufficient accurate ones. The hyper-parameters  $L_{\text{NH}}$  and  $c_{\max}$  are problem specific and have to be thoroughly adapted.

After detecting a loop closure between the vertices  $i$  and  $j$  of the pose graph, the relative measurement  $\hat{\xi}_{ij}$  has to be added to the graph. Therefore, the neighborhood regions for both vertices  $i$  and  $j$  are discretized as distinct points, represented by the sets  $X_i = \{\mathbf{x}_{i,1}, \dots, \mathbf{x}_{i,K}\}$  and  $X_j = \{\mathbf{x}_{j,1}, \dots, \mathbf{x}_{j,K}\}$ , and transformed such that both poses  $i$  and  $j$  are equal with  $\hat{\mathbf{p}}_i = \hat{\mathbf{p}}_j = [0, 0, 0]^\top$ . By using an adapted ICP approach [59], which minimizes the distance error

$$\min_{\mathbf{R}_\phi, \mathbf{t}} E_{\text{dist}}(\mathbf{R}_\phi, \mathbf{t}) = \min_{\mathbf{R}_\phi, \mathbf{t}} \sum_{k=1}^K \|\mathbf{R}_\phi \mathbf{x}_{i,k} + \mathbf{t} - \mathbf{x}_{i,k}^*\| \quad (2.35)$$

a planar rotation  $\mathbf{R}_\phi$  with  $\phi$  being the rotation angle and a translation vector  $\mathbf{t} = [t_x, t_y]^\top$  can be calculated. Here,  $\mathbf{x}_{i,k}^*$  is the point of  $X_j$  closest to  $\mathbf{x}_{i,k}$ . The loop closing constraint can then be derived using Equation (2.22) by transforming  $\hat{\mathbf{p}}_j$  given the calculated rotation  $\phi$  and translation  $\mathbf{t}$  which leads to

$$\hat{\xi}_{ij} = [t_x \quad t_y \quad \phi]^\top. \quad (2.36)$$



(a) Pruned graph from the odometry data with marked dominant points (DPs). (b) The pose graph together with the loop closing constraints (LCs). (c) The optimized pose graph.

Figure 2.9: Exemplary optimization procedure for finding an optimal map representation for closed loop environments utilizing only odometry data.

The corresponding covariance matrix can be calculated using the correlation error  $C_{ij}$  and hyper-parameters  $\gamma_1$  and  $\gamma_2$

$$\mathbf{P}_{lc,ij} = \text{diag}([\gamma_1 \ \gamma_1 \ \gamma_2]) C_{ij}. \quad (2.37)$$

The parameters  $\gamma_1$  and  $\gamma_2$  are problem specific and have to be thoroughly adjusted. In Figure 2.9, the optimization procedure for an exemplary pose graph is shown.

### 2.5.3 Hyper-Parameter Learning

In order to learn unknown hyper-parameters (e.g.  $L_{NH}$ ) for the above in Subsection 2.5.2 introduced loop closure detection method, an optimization problem can be defined as

$$\min_{\boldsymbol{\theta}} c(\boldsymbol{\theta}). \quad (2.38)$$

Here,  $c(\boldsymbol{\theta})$  is a specific cost function and  $\boldsymbol{\theta}$  are the hyper-parameters which have to be optimized. A natural choice for  $c(\boldsymbol{\theta})$  would be a “mapping error” between a groundtruth and the map estimate. However, since the groundtruth is not known a priori (otherwise the mapping would be obsolete), different cost measures utilizing probabilistic hypotheses are used in this subsection. The most important assumption for the derivation of the cost function is that the odometry error has zero mean with some covariance matrix  $\mathbf{P}$ ,  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{P})$ . This holds for the here investigated differential drive robot as has been shown in Subsection 2.4.3.

To optimize the hyper-parameters, a two-stage optimization process is proposed. In the first stage, a cost function  $c_1(\boldsymbol{\theta})$  is defined for tuning the parameters for loop closure

detection,  $L_{\text{NH}}$  and  $c_{\text{max}}$ . A resulting by-product, the circumference of the closed environment, is then used for defining the second cost function  $c_2(\boldsymbol{\theta})$ , with which the pose graph parameters  $\gamma_1, \gamma_2$  are tuned.

**Stage 1 – Optimization of Loop Closing Parameters** Let the odometry error between two poses  $i$  and  $j$  be on average zero, thus  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{P})$  with  $\mathbf{P}$  being the covariance matrix. Let  $u$  then denote the traveled path distance along the pose graph between a loop closing pair  $i, j$

$$u = \sum_{k=i}^{j-1} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|. \quad (2.39)$$

Given the assumption of zero mean odometry error, the traveled path distances for all loop closing pairs  $\mathbf{u} = [u_1, u_2, \dots, u_M]$ , identified by cycling around a closed environment, are on average multiples of the circumference of the closed environment,  $nU$ . Here,  $n \in \mathbb{N}^+$  is a positive integer, representing the number of cycles before loop closure detection. Hence, if the loop closures are detected properly, a histogram of the path distances  $\mathbf{u}$  shows distinct peaks at positions  $nU$ . In Figure 2.10 such histograms are shown for ill-detected loop closures, Figure 2.10b, and well-detected loop closures, Figure 2.10d. To transform this idea into a cost function, a Gaussian Mixture Model (GMM) [60] can be learned from observed path distances  $\mathbf{u}$  given the probability distribution

$$p(\mathbf{u}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{u} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \quad (2.40)$$

Here,  $K$  is the number of mixture components and  $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$  the mixture weight, the mean and the variance of the  $k$ -th component, respectively. For fitting the GMM, the iterative Expectation-Maximization (EM) algorithm is used [61], [62]. The EM algorithm starts with a randomly selected model and then alternately optimizes the allocation of the data  $\mathbf{u}$ , i.e., the weighting  $\pi_k$ , to the individual parts of the model and the parameters of the model  $\boldsymbol{\mu}_k$  and  $\boldsymbol{\Sigma}_k$ . If there is no significant improvement, the procedure is terminated.

For the cost function, the negative log likelihood of the GMM,

$$-\mathcal{L} = -\ln p(\mathbf{u} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\sum_{i=1}^M \ln \left[ \sum_{k=1}^K \pi_k \mathcal{N}(u_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right] \quad (2.41)$$

over the data set  $\mathbf{u} = [u_1, u_2, \dots, u_M]$ , is used. The log likelihood decreases if the dataset  $\mathbf{u}$  meets the above assumption of evenly distributed peaks at positions  $nU$ . A common strategy for training GMMs is to iteratively increase  $K$  until the log likelihood does not improve further. In Figure 2.10, the evolution of the negative log likelihood with respect to the number of components  $K$  is shown for ill-detected loop closures, Figure 2.10a, and well-detected loop closures, Figure 2.10c.

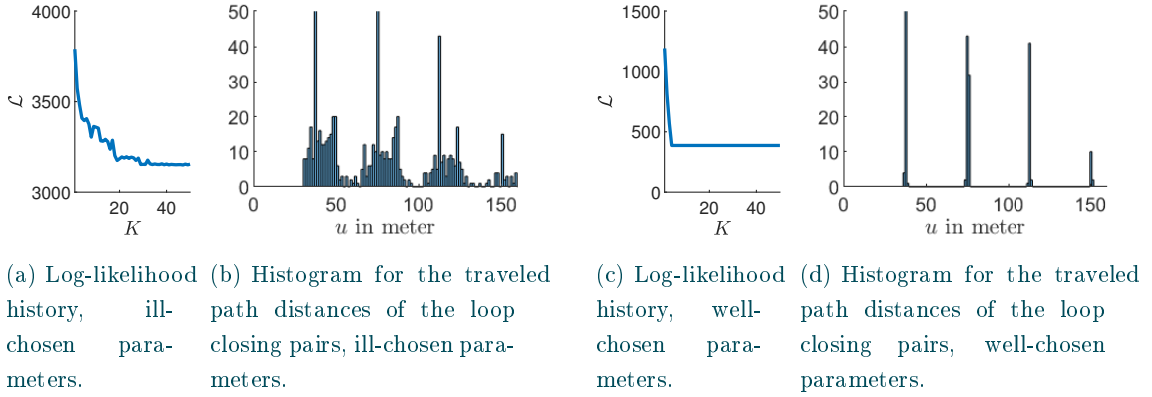


Figure 2.10: Examples for ill- and well-chosen hyper-parameters for loop closure detection.

The cost function for optimizing the hyper-parameters for loop closure detection is then

$$\min_{\theta} c_1(\theta) = \min_{\theta} \left( \frac{-\mathcal{L}}{M} - \ln(M) \right), \quad (2.42)$$

with the unknown hyper-parameters  $\theta = [L_{\text{NH}}, c_{\text{min}}]$ , the length of the neighborhood and the minimum comparison error, and  $M$  being the number of loop closures found. After finding accurate hyper-parameters, the circumference of the closed environment  $U$  can be estimated using the best GMM fit. The estimated closed environment circumference  $U$  is then used in stage 2.

**Stage 2 – Optimization of Pose Graph Parameters** Given the circumference estimate of the closed environment  $U$  from stage 1, the hyper-parameters for the loop closing covariance for pose graph optimization,  $\gamma = [\gamma_1, \gamma_2]$ , can be learned. Therefore, the cost function is defined as

$$\min_{\gamma} c_2(\gamma) = \min_{\gamma} |U - \hat{U}|, \quad (2.43)$$

where  $\hat{U}$  represents the circumference estimate after pose graph optimization. The idea is to punish deviations between the estimated circumference based on the original pose graph and the optimized one. The underlying assumption here is that the determined circumference  $U$  already represents closely the true circumference of the environment and should therefore be changed as little as possible. In order to determine the circumference after pose graph optimization, a fit onto GMMs is performed.

**Global Optimization - Bayesian Optimization** In order to find optimal parameters, Equation (2.42) and Equation (2.43) have to be minimized. Since there are no derivatives for both, Equation (2.42) and Equation (2.43), derivative free optimization techniques are required. Moreover, both optimization problems are highly cluttered, as exemplarily

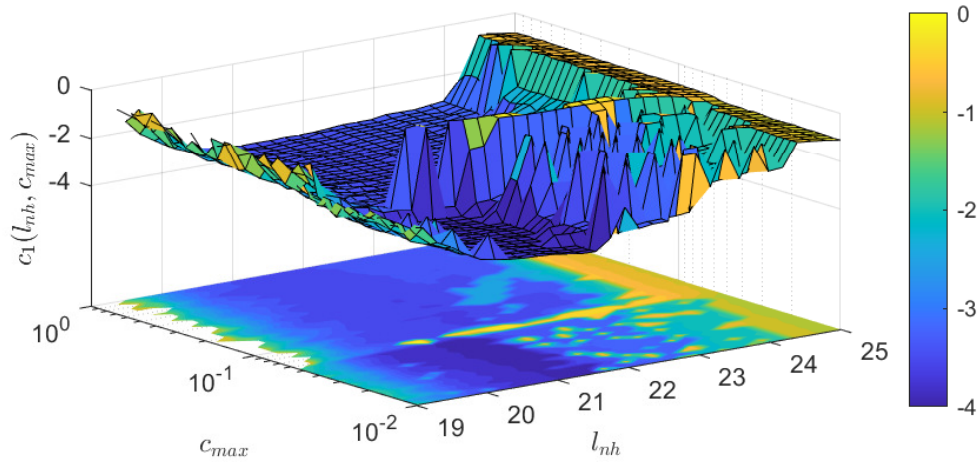


Figure 2.11: An example of the above introduced cost function, Equation (2.42). The structure is highly cluttered such that local optimization procedures could get stuck in local minima. Hence, global optimization techniques have to be used.

shown in Figure 2.11. Hence, global optimization techniques are required. Here, episodic BO [45] with Expected Improvement (EI) [63] is used to iteratively find a sufficiently good approximation of the global minimum. A brief summary concerning BO can be found in the appendix, Section B.

#### 2.5.4 Evaluation

The above proposed mapping approach is first tested on real data collected in different garden environments using the Viking MI 422P lawn mower. The approach generates accurate map estimates for the different closed garden environments. Additionally, simulations were performed to evaluate the approach in various challenging environments and for different odometry parameters. In order to compare the mapping results, an intuitive error measure, which calculates the deviation between a ground truth map and the map estimate, is used.

**Error Determination** In order to compare the optimized pose graph data with a ground-truth map, the pose graph has to be transformed into a closed map representation. Therefore, the prefix and the suffix of the pose graph are cut off, because these parts have not been optimized due to missing loop closing constraints. Thereby, the prefix is the part of the pose graph before the first loop closing constraint and the suffix is the part of the pose graph after the last loop closing constraint. In order to generate a closed map representation, the first loop closing pair is taken which represents a complete turn around the borderline. Setting the points of this loop closing pair onto each other leads to a closed

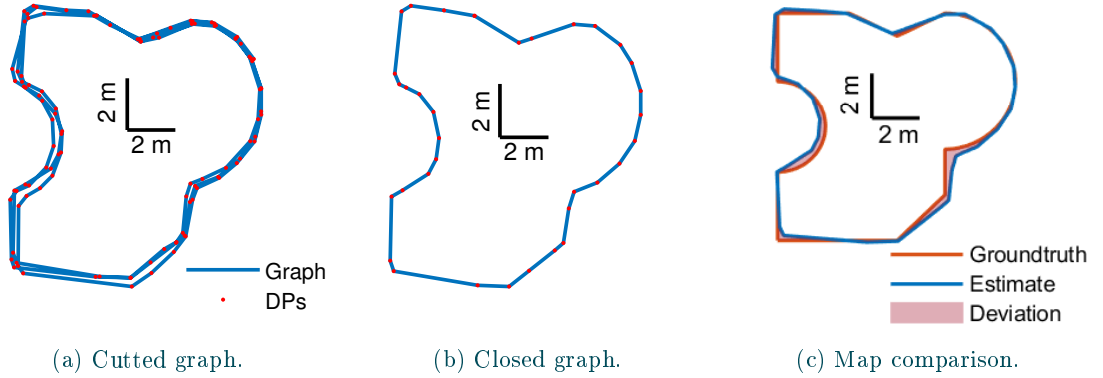


Figure 2.12: Exemplary map comparison procedure. The figure shows the different processing steps to compare an optimized pose graph with a groundtruth map.

trajectory as presented in Figure 2.12b.

To compare different map estimates with each other the deviation of area between the estimated closed map and the true shape of the environment is calculated. Therefore, the true shape of the environment is represented as a polygon defined by the points  $X_{\text{true}}$ . The idea is to determine a rotation matrix  $\mathbf{R}$  and a translation vector  $\mathbf{t}$  which transforms the set of points  $X$ , representing the closed map estimate, onto the set of points  $X_{\text{true}}$

$$X \leftarrow \mathbf{R} \cdot X + \mathbf{t}, \quad (2.44)$$

such that

$$\Delta A = 1 - \frac{A_{\text{true}} \cap A_{\text{estimate}}}{A_{\text{true}} \cup A_{\text{estimate}}} \quad (2.45)$$

is minimized. Here,  $\Delta A$  represents the difference between the areas of the map estimate and the true shape of the environment. To find a convenient rotation matrix and translation vector for Equation (2.44), optimization techniques such as gradient descent can be used. However, local optimization procedures require a sufficient accurate initial guess to not get stuck in local minima. In Figure 2.12c, a transformed map estimate together with the groundtruth and the deviation area between both is plotted.

**Real Garden Examples** For generating real data, the lawn mower was driven along the boundary line of two different lawn areas, the university courtyard and a private garden environment. The velocity of the lawn mower was set to  $0.3 \text{ ms}^{-1}$ . The odometry data was sampled with a frequency of 20 Hz.

In Figure 2.13, the university courtyard, the respective measured odometry data and a resulting map estimate are shown. The ground truth is available as CAD data, such that a comparison between the map estimate and the true closed environment can be made

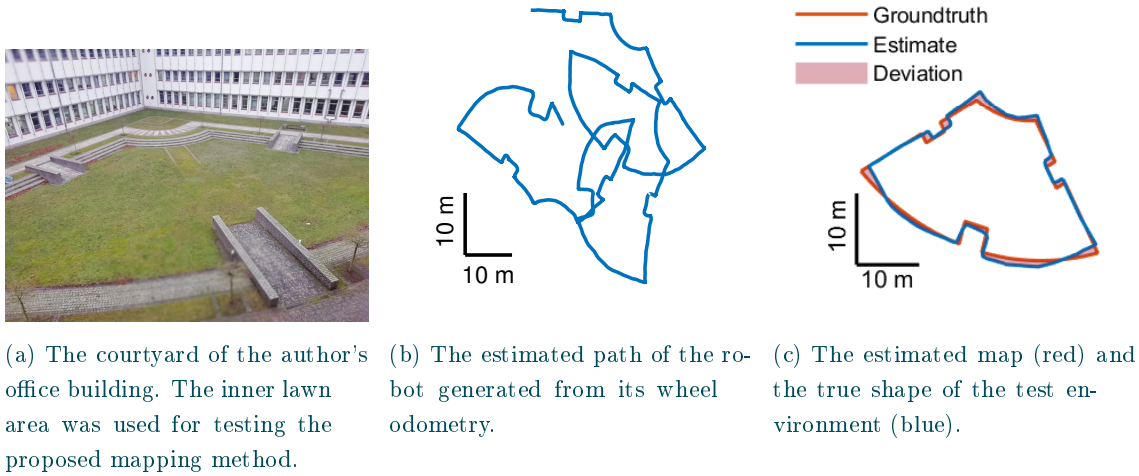


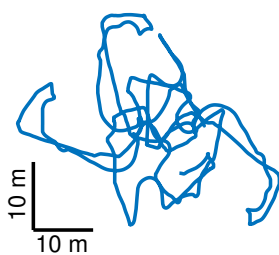
Figure 2.13: The real courtyard depicted (a), the collected odometry data (b) and the map estimate (c). The comparison error between the map estimate and the groundtruth map is  $\Delta A = 8.24\%$

according to Equation (2.45). In a first step, the mapping parameters were set manually based on the known circumference  $U = 106.8\text{ m}$  and the complexity of the environment with  $L_{\text{NH}} = 30$ ,  $c_{\text{max}} = 0.3$ ,  $\gamma_1 = 1$ ,  $\gamma_2 = 0.001$ . The resulting mapping error between the map estimate and the groundtruth map is then  $\Delta A = 8.24\%$ . In a second step, the hyper-parameter estimation procedure was evaluated. Therefore, 20 runs were performed from which all led to good loop closing detection with  $L_{\text{NH}} = 36.16 \pm 1.81$  and  $c_{\text{max}} = 0.56 \pm 0.03$ . However, the hyper-parameters for the pose graph optimization,  $\gamma_1, \gamma_2$  differ largely within the searching range of  $\gamma_1 = 0.01 \dots 10$  and  $\gamma_2 = 0.0001 \dots 1$ . The overall mapping difference over the 20 runs, according to Equation (2.45), is  $\Delta A = 0.1668 \pm 0.0797$ . There are outliers which reflect that in some cases the optimization procedure did not find suitable hyper-parameters. Here, 8 out of the 20 runs imply with mapping differences  $\Delta A > 20$  that non-optimal hyper-parameters are found. Neglecting these outliers and considering only the remaining 12 successful runs, the mapping difference can be calculated as  $\Delta A = 0.1045 \pm 0.0126$ .

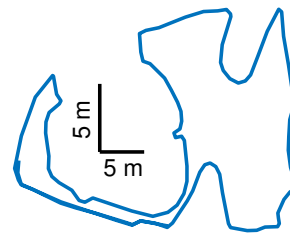
Additionally, the mapping approach was evaluated in a second real environment, a representative of a typical private lawn. In Figure 2.14 from left to right, a part of the private lawn, the measured odometry data and the map estimate are shown. Since no ground truth data is available for this lawn, the map results are compared qualitatively with the image of the real garden. As demonstrated, the approach is capable of mapping large closed environments with narrow corridors based on severely distorted odometry data.



(a) The top view onto a part of a lawn from a typical private household.



(b) The estimated path of the robot generated from its wheel odometry.



(c) The estimated map.

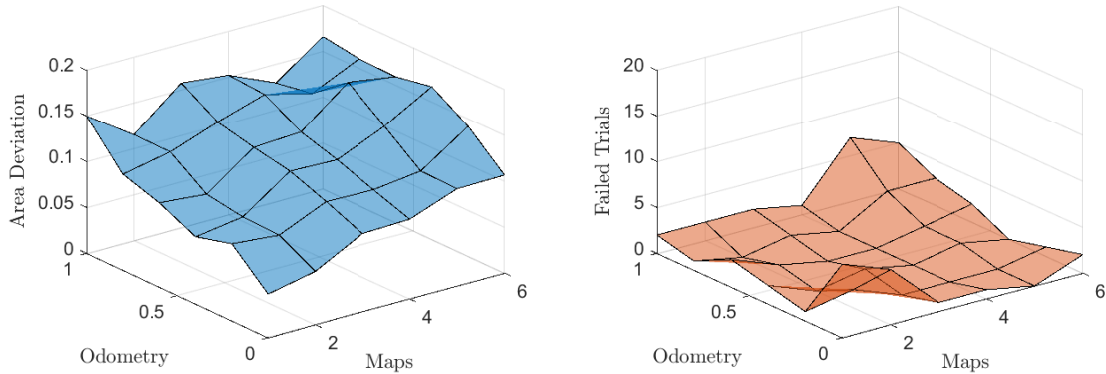
Figure 2.14: A typical lawn (a), the collected odometry data (b) and the map estimate with learned parameters (c).

**Simulated Data** The proposed mapping approach was tested on 6 different maps with increasing degree of complexity (Figure 2.22) and with different levels of odometry error. For every map-odometry error pair, 20 runs were performed where the hyper-parameters were learned according to the in Equation (2.42) and Equation (2.43) proposed cost functions. For path segmentation, the parameters were set to  $L_{\min} = 0.1$ ,  $e_{\max} = 0.001$  and for the BO procedure a number of 30 iterations was chosen.

Figure 2.15a shows the averaged mapping results whereas Figure 2.15b shows the number of failed trials, thus the number of runs for each map-odometry error where no convenient loop closing pairs were found by the algorithm. Additionally, the averaged results are also listed in Table 2.2. The results show that the mapping results get more accurate for more simplistic maps but the failed trials increase for those simplistic maps, even if only small numbers of cases occur. Also, the mapping results strongly increase with decreasing

Table 2.2: Mapping method performance based on Equation (2.45).

	Odometry Error Level					
	0.0	0.2	0.4	0.6	0.8	1.0
1	0.047	0.084	0.073	0.092	0.105	0.150
2	0.058	0.079	0.081	0.087	0.115	0.115
3	0.085	0.102	0.117	0.124	0.134	0.158
4	0.087	0.100	0.115	0.127	0.150	0.152
5	0.106	0.107	0.137	0.167	0.137	0.130
6	0.107	0.142	0.167	0.159	0.161	0.167



(a) Comparison results between estimated map and groundtruth data. (b) Failed trials where the algorithm did not find convenient loop closing pairs.

Figure 2.15: Comparison of the here proposed map estimation method. The algorithms have been tested in different maps shown in Figure 2.22, and with different odometry error levels.

odometry error and approach to the optimal solution ( $\Delta A = 0$ ) but never reach it. This is mainly caused by errors from the data segmentation.

Additionally, in Figure 2.16 map estimation examples for the 6 different map environments are shown. In Figure 2.16a, the paths measured by the odometry of the robot are shown and in Figure 2.16b the generated map estimates (red) and the groundtruth map data (blue). It illustrates that the proposed method in general is able to generate a fair map estimate even with only highly distorted odometry data available.

## 2.6 Path Planning under high Pose Uncertainty

Assume a map of the closed environment is given as a binary occupancy grid  $\mathcal{M}$ . Such a map can be generated utilizing only the given in-/outside area detectors by following the map generation procedure introduced in Section 2.5. Given the map of the environment  $\mathcal{M}$ , a probability occupancy grid map  $\mathcal{C}$  is used to model the coverage of the closed environment. Therefore, a similar dynamical system model as proposed in [34, 35] for high-confidence cleaning guarantees is used. To account for the high uncertainty of the pose estimate given only the odometry and the in-/outside area measurements, a particle filter is used to estimate the pose and the coverage, respectively. The CCPP approach then uses a neural network where each cell of the coverage map represents a neuron, as introduced in [46, 47]. Based on the neural activity of the neurons and the current pose, the planner chooses the neighboring cell to move to. This, in combination with the probabilistic coverage information, leads to efficient CCPP under high uncertainties.

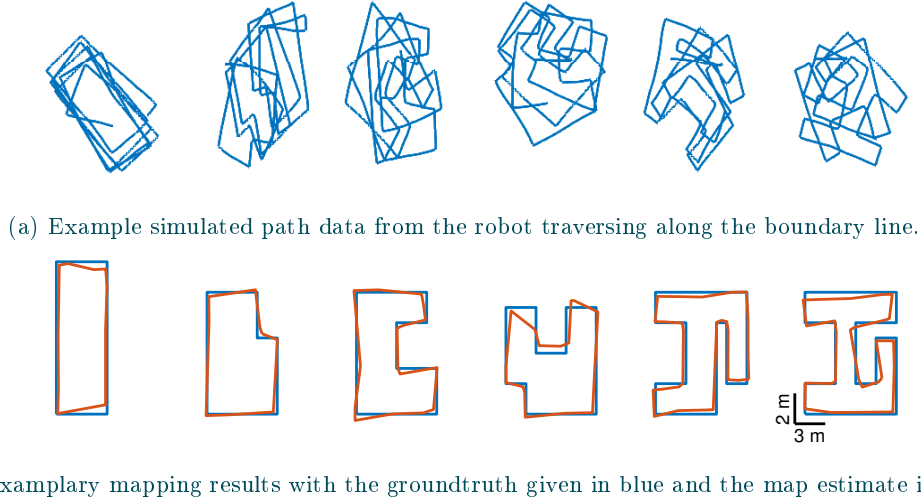


Figure 2.16: Exemplary map estimation results for different test maps as shown in Figure 2.22.

### 2.6.1 Probabilistic Coverage Map and Dynamical System

Given the map of the environment  $\mathcal{M}$ , a probability occupancy grid map  $\mathcal{C}$  is generated with  $c_i \in [0, 1]$  being the probability of the cell  $i$  to be covered by the robot initialized as  $c_i = 0, \forall i$ . Let  $\mathbf{p}_t$  be the pose of robot at time step  $t$ ,  $\mathbf{u}_t$  the input signals to the robot,  $\mathbf{z}_t$  the sensor measurements and  $\mathbf{c}_t$  the coverage states of all cells, then the path of the robot and the coverage can be defined as joint posterior distribution

$$\begin{aligned} \text{prob}(\mathbf{p}_{0:t}, \mathbf{c}_{0:t} | \mathbf{u}_{1:t}, \mathbf{z}_{1:t}) &= \eta \underbrace{\text{prob}(\mathbf{z}_t | \mathbf{p}_t)}_{\text{sensor model}} \underbrace{\text{prob}(\mathbf{c}_t | \mathbf{c}_{t-1}, \mathbf{p}_{t-1}, \mathbf{p}_t)}_{\text{coverage model}} \\ &\quad \underbrace{\text{prob}(\mathbf{p}_t | \mathbf{p}_{t-1}, \mathbf{u}_t)}_{\text{motion model}} \underbrace{\text{prob}(\mathbf{p}_{0:t-1}, \mathbf{c}_{0:t-1} | \mathbf{u}_{1:t-1}, \mathbf{z}_{1:t-1})}_{\text{prior distribution}}. \end{aligned} \quad (2.46)$$

In comparison to [35], there is no coverage sensor model which gives additional information for pose estimation. By considering the Markov property

$$\text{prob}(a_t | a_{t-1:0}) = \text{prob}(a_t | a_{t-1}), \quad (2.47)$$

Equation (2.46) turns into

$$\begin{aligned} \text{prob}(\mathbf{p}_t, \mathbf{c}_t | \mathbf{u}_t, \mathbf{z}_t) &= \eta \text{prob}(\mathbf{z}_t | \mathbf{p}_t) \text{prob}(\mathbf{c}_t | \mathbf{c}_{t-1}, \mathbf{p}_{t-1}, \mathbf{p}_t) \\ &\quad \text{prob}(\mathbf{p}_t | \mathbf{p}_{t-1}, \mathbf{u}_t) \text{prob}(\mathbf{p}_{t-1}, \mathbf{c}_{t-1} | \mathbf{u}_{t-1}, \mathbf{z}_{t-1}) \end{aligned} \quad (2.48)$$

which results into an iterative update rule for the pose estimate and the coverage based on the current inputs and sensor measurements. In the following, the models introduced in Equation (2.46) are shortly summarized.

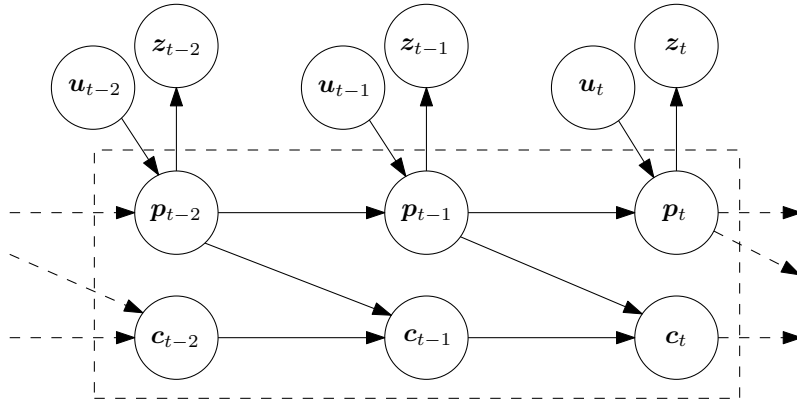


Figure 2.17: The dynamical system, Equation (2.46), illustrated as graph.

**Sensor Model** The robot, as introduced in Section 2.4.1, is equipped with two in-/outside area detectors, e.g., chlorophyll fluorescence sensors or boundary wire sensors, which give the information whether the sensors are over grass (inside) or not (outside). The sensors are placed at the left and right front of the robot. Based on the current pose  $\mathbf{p}$ , an estimate of the measurements can be made given the information from the map  $\mathcal{M}$ .

**Coverage Model** The coverage model reflects the change in coverage based on the movements of the robot. A general probabilistic model for the coverage of a certain grid cell  $i$  can be defined as

$$c_{i,t} = c_{i,t-1} + P_i(\mathbf{p}_{t-1}, \mathbf{p}_t) \cdot (1 - c_{i,t-1}), \quad (2.49)$$

where  $P_i(\mathbf{p}_{t-1}, \mathbf{p}_t)$  is the probability of the robot covering a certain grid cell  $i$  by moving from pose  $\mathbf{p}_{t-1}$  to  $\mathbf{p}_t$ . In comparison to [35], where unweighted area sampling is used to determine  $P_i(\mathbf{p}_{t-1}, \mathbf{p}_t)$  (Figure 2.18b), here a more conservative yet computationally faster approach is used in order to lower the computational burden onto the system (Figure 2.18a). The approach is based on simple line drawing algorithms where a cell is marked as covered if the robot passed it. Therefore, the resolution of the coverage grid has to be chosen as

$$\text{Resolution} \geq \frac{2\sqrt{2}}{d_{\text{effector}}}, \quad (2.50)$$

where  $d_{\text{effector}}$  is the diameter of the robot's effector. This ensures the coverage of the whole cell if visited. The coverage probability is then  $P(\mathbf{p}_{t-1}, \mathbf{p}_t) = 1$  if the robot traversed the certain grid cell and  $P(\mathbf{p}_{t-1}, \mathbf{p}_t) = 0$  otherwise.

**Motion Model** For the motion model, the odometry model proposed in [48] is used as presented in Section 2.4.3.

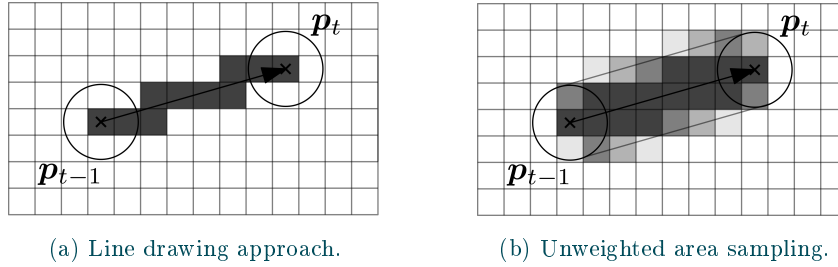


Figure 2.18: Different coverage models where the circles represent the robot’s effector, e.g., the cutter. The first approach is conservative and computationally fast but underestimates the true coverage whereas the latter is more accurate but comes with higher computational costs and might slightly overestimate the coverage.

### 2.6.2 Particle Filter

To efficiently generate a pose estimate for the robot by fusing the odometry and sensor data, a standard particle filter algorithm [48] is used to handle the binary in-/outside area measurements. The general idea of the particle filter is to represent the probability distribution of the posterior by a set of samples, called particles, instead of using a parametric form as the Kalman Filter does. Here, each particle

$$\mathcal{P}_t = \mathbf{p}_t^{[1]}, \mathbf{p}_t^{[2]}, \dots, \mathbf{p}_t^{[N]} \quad (2.51)$$

represents a concrete instantiation of the state at time  $t$ , where  $N$  denotes the number of particles used. The belief  $bel(\mathbf{p}_t)$  is then approximated by the set of particles  $\mathcal{P}_t$ . The pose estimate of the robot can be calculated by taking the mean over all  $N$  particles. The Bayes filter posterior is used to include the likelihood of a state hypothesis  $\mathbf{p}_t$

$$\mathbf{p}_t^{[i]} \sim \text{prob}(\mathbf{p}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}). \quad (2.52)$$

Here,  $\mathbf{z}_{1:t}$  and  $\mathbf{u}_{1:t}$  represent the measurement history and the input signal history respectively.

Given the positions of the  $N$  particles, the coverage map is updated when the robot traversed from one cell to another. According to the coverage model proposed in Equation (2.49) and Figure 2.18a the update rule can be defined as

$$c_{i,t} = c_{i,t-1} + \frac{n_{i,t}}{N}(1 - c_{i,t-1}), \quad (2.53)$$

where  $c_{i,t}$  is the coverage probability of cell  $i$  at time  $t$  and  $n_{i,t}$  the number of particles in cell  $i$  at time  $t$ .

### 2.6.3 Complete Coverage Path Planning

Given the estimation of the robots trajectory and the estimation of the coverage, an efficient path planning scheme is left to define. Due to the high uncertainty of the robot’s pose

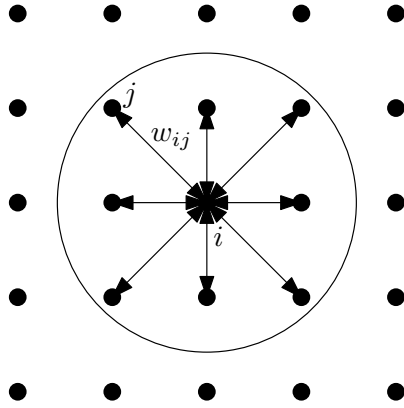


Figure 2.19: A schematic of the neural network with the neighborhood of a cell and the cell connections.

and the associated uncertainties in map coverage, such a path planning scheme requires rapid adaptability. Hence, solving a TSP on the fully connected graph over all cells of the map as in [35] is not feasible. Instead, the neural network approach introduced in [46, 47] is adapted to cope with uncertain pose and coverage estimates. The neural network approach is thereby derived from the shunting equation [64] which was inspired by a model for a patch of a membrane introduced in [65]. In this approach, neurons are generated which each have a neural activity. These neural activities then provide an attraction on the basis of which the robot plans its next steps.

Let each cell of the coverage map represent a neuron and  $q_i$  the neural activity of the neuron  $i$ , then the change of neuronal activity can be described according to [46] as

$$\frac{dq_i}{dt} = -Aq_i + (B - q_i) \left( [I_i]^+ + \sum_{j=1}^k w_{ij}[q_j]^+ \right) - (D + q_i)[I_i]^-. \quad (2.54)$$

Here,  $A$ ,  $B$  and  $D$  are non-negative parameters representing the passive decay rate, the upper and the lower bound of the neural activity. The function operators  $[a]^+$  and  $[a]^-$  are defined as  $[a]^+ = \max\{a, 0\}$  and  $[a]^- = \max\{-a, 0\}$ . The neuronal activities are initialized as  $q_i = 0, \forall i$  but receive an external input  $I_i$ . Following the suggestion from [46] and considering the coverage probabilities  $c_i$ , the external input is proposed to be

$$I_i = \begin{cases} (1 - c_i) E & \text{if cell } i \text{ is inside the working area} \\ -E & \text{if cell } i \text{ is outside the working area} \end{cases}. \quad (2.55)$$

Here, the external input accelerator  $E$  should be chosen such that  $E \gg B$ . The neurons are each connected to their neighboring cells, as illustrated in Figure 2.19. The connections weights can be defined as

$$w_{ij} = f(\|\mathbf{x}_i - \mathbf{x}_j\|), \quad (2.56)$$

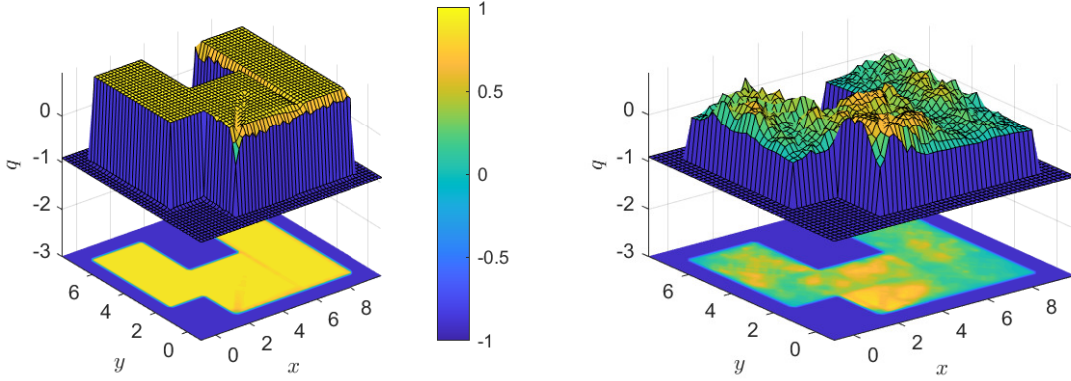


Figure 2.20: Neural activities  $q_i$  for different coverage values. The left panel shows the neural activities after 10% coverage is reached and the right panel the neural activities after 95% coverage is reached.

where  $\mathbf{x}_i$  represents the location of the  $i$ -th neuron. For example, [46] proposes the weighting function

$$f(a) = \begin{cases} \frac{\mu}{a} & \text{if } 0 < a < r \\ 0 & \text{if } a \geq r \end{cases} \quad (2.57)$$

with  $\mu$  and  $r$  being positive constants. Here, the neurons are ordered in a regular grid such that a constant symmetric weighting matrix can be proposed

$$\mathbf{W} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 1 & \frac{1}{\sqrt{2}} \\ 1 & 0 & 1 \\ \frac{1}{\sqrt{2}} & 1 & \frac{1}{\sqrt{2}} \end{bmatrix} \quad (2.58)$$

for efficiently using image filtering techniques to determine the input from the neighboring cells. Since Equation (2.54) only allows for positive neural activities to propagate between neighboring neurons, negative neural activities stay local. In other words, uncovered areas inside the working area attract the robot globally while areas outside the working space only locally push the robot away. Samples of neural activity distributions over the cells are shown in Figure 2.20 for two different coverage states.

Following [46], the robot's path is planned based on the neural activity landscape. More precisely, let  $\mathbf{x}_i$  be the robot's current position and  $q_i$  the corresponding neural activity, then the next position  $\mathbf{x}_{\text{next}}$  the robot is sent to can be defined as the position of the neighboring cell with the largest neural activity. Thus,

$$q_{\mathbf{x}_{\text{next}}} = \max(q_j, j = 1, \dots, k) \quad (2.59)$$

with  $k$  being the number of neighboring cells of  $q_i$ . Considering the robot's motion model, it is advantageous to avoid unnecessary turns with the robot since turning movements are

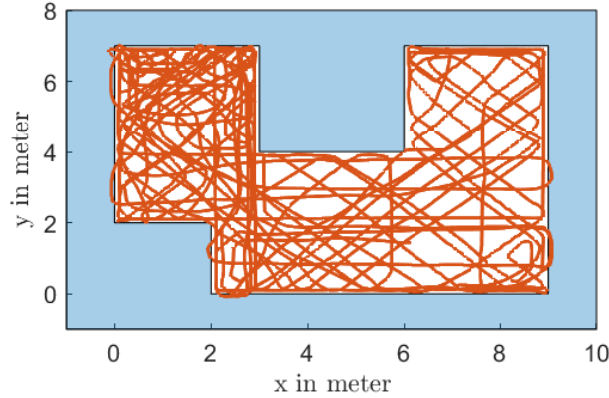


Figure 2.21: Example path after reaching 90% coverage.

much more prone to odometry errors. Thus, an additional term to Equation (2.59) is added which takes the current orientation  $\varphi_i$  of the robot into account

$$q_{\mathbf{x}_{\text{next}}} = \max(q_j - \gamma g_j, j = 1, \dots, k) \quad (2.60)$$

with

$$g_j = |\varphi_i - \text{atan2}(\mathbf{x}_j - \mathbf{x}_i)| \quad (2.61)$$

and  $\gamma$  being a parameter which has to be set appropriately. This modification ensures a preferred next cell allocation based on the robots current pose. In Figure 2.21, an example path after reaching 90% coverage is shown.

**Relocalization** Since the robot has a large odometry error but only in-/outside area detection sensors, the robot receives only few valuable measurement data during path execution. Thus, it is required that the robot stops path execution and starts to relocalize itself when the pose estimate becomes quite uncertain, thus exceeds a certain variance value  $\sigma_{\text{max}}$ . For the relocalization, the robot drives along the boundary to get different measurement signals for improving the accuracy of the pose estimate by the particle filter until the pose estimates variance is below another threshold  $\sigma_{\text{min}}$  after which the robot continues with path execution. The method for applying a wall following behavior to the robot is summarized in the appendix, Section A.

#### 2.6.4 Evaluation

The proposed CCPP approach is evaluated in different challenging simulation scenarios based on the odometry and velocity motion model from Section 2.4. This enables realistic simulation studies for analyzing the performance of the proposed method in detail.

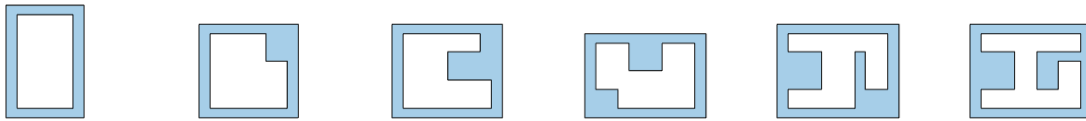


Figure 2.22: Different artificially created maps with increasing number of complexity but with the same enclosed area of  $A = 50 m^2$ . The maps are numbered from left to right, beginning with Map 1 and ending with Map 6.

**Evaluation Criteria** As measurement of the performance, the traveled distance required to reach a certain total coverage percentage is used which here is a coverage value of 95%. In total, the proposed method was evaluated on 6 different maps with different degrees of complexity but same size of coverage space  $A = 50 m^2$ , see Figure 2.22, and for different levels of odometry errors. Here, an odometry level error of one signals a full odometry error as represented by the odometry model parameters from Table 2.1 and an odometry level error of zero signals no odometry error at all.

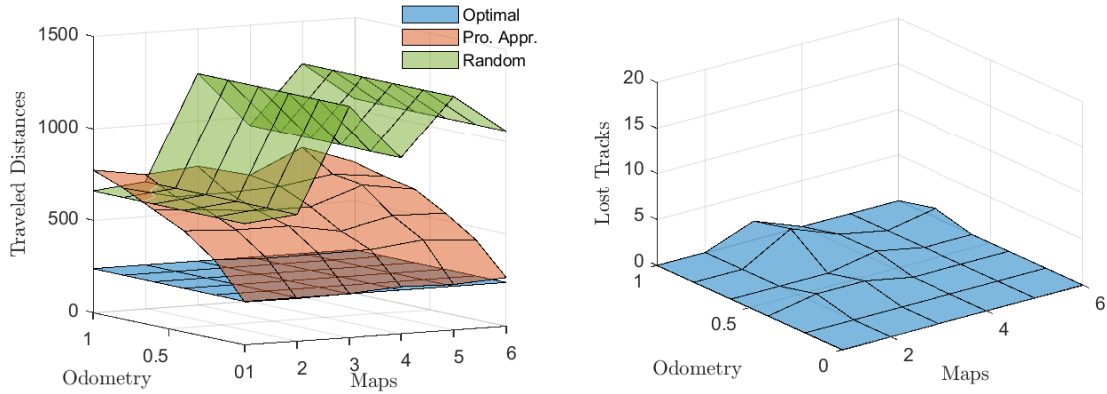
The optimal traveled distance traversing over the centers of the grid map cells to reach 95% coverage for a coverage space of  $A = 50 m^2$  can be determined as  $T_{\text{opt}} = 237.4 m$  (Appendix, Section C). Based on this optimal distance, an optimality criterion for the method can be defined as

$$\text{Opt} = 1 - \frac{T - T_{\text{opt}}}{T_{\text{rand}} - T_{\text{opt}}}, \quad (2.62)$$

where  $T$  is the average traveled distance the method requires and  $T_{\text{rand}}$  the average traveled distance a random walk pattern requires. Here, a value close to one signals a performance close to the optimal pattern where a value close to zero signals a performance close to a random walk pattern. A negative value therefore shows a worse performance than a random walk pattern.

**Statistical Evaluation** For the simulation study, 20 runs were performed for each pair of map–odometry error level. In total,  $6 \times 6$  different combination pairs were evaluated with respect to their traveled distance after reaching the 95% total coverage level. Since the approach is probabilistic, the 20 runs recorded for each map–odometry error level pair were averaged. The method parameters used for the evaluation are tabulated in Table 2.3.

In Figure 2.23a, the calculated average traveled distances are shown together with the optimal traveled distances and the average traveled distances reached with a random walk pattern. Additionally, in Table 2.4 the performance of the method according to Equation (2.62) is presented. The evaluation shows, that the method in general outperforms the commonly used random walk pattern except for the most simplistic maps under high odometry error. A strong correlation between the performance of the method and the odometry error level can be found such that with lower odometry error the approach performs



(a) Comparison between optimal, random walk and the here proposed path planning algorithms. (b) Trials where the particle filter has lost track of the robots pose estimate during path execution.

Figure 2.23: Comparison of the here proposed complete coverage method with the commonly used random walk approach and the optimal approach. For the optimal approach the exact pose of the robot is assumed to be known. For the evaluation, the algorithms have been tested in different maps shown in Figure 2.22, and with different odometry error levels.

better and approaches the optimal performance. This is to be expected, since less odometry error allows for better path performance and requires less relocalization. In addition, the complexity of the map favors the proposed approach, as the random walk pattern has trouble covering narrow areas of the map.

**Coverage Tracking** The robot should be able to estimate the current coverage of the workspace as efficiently as possible. Thus, the correlation between the true coverage and the estimated coverage by the proposed approach is of importance. In Figure 2.24, a comparison between both values is shown for map 4 and an odometry error level of 1.0. The estimated coverage always slightly underestimates the true coverage and thus can be used to conservatively estimate the true coverage.

Table 2.3: CCpp method parameters

Method	Particle Filter		Path Planner & Map			
	$N$	500	Resolution	$5 \text{ m}^{-1}$	$E$	100
$N_{\text{thr}}$	$0.7N$	$A$	10	$\gamma$	0.1	
		$B$	1	$\sigma_{\text{max}}$	0.09	
		$D$	1	$\sigma_{\text{min}}$	0.03	

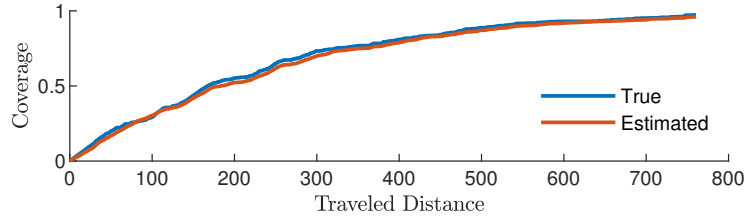


Figure 2.24: Comparison between the true and the estimated coverage with respect to the traveled distance of the robot for an odometry level of 1.0 and map 4.

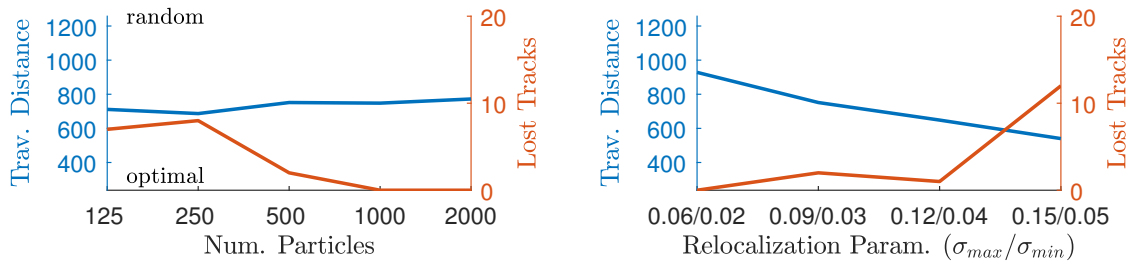
**Particle Filter Performance Analysis** One disadvantage of the proposed approach is the use of the particle filter, as it sometimes loses track of the pose estimate, Figure 2.23b, which leads to an abort of the path planning execution. This happens especially with increasing odometry error. Possible error sources, according to [48], are:

(1) The approximation error of the probability distribution due to the finite number of particles used, (2) the approximation error induced by the randomness of the resampling phase, (3) the divergence of the proposal and target distribution if only deterministic measurements are available and (4) the particle deprivation problem, where no particles might be in the vicinity of the correct state.

In general, a larger number of particles is beneficial for reducing most of the mentioned error sources but comes with a higher computational burden. Nevertheless, an increasing number of particles reduces the number of lost tracks significantly, as shown in Figure 2.25a. Other measures to be taken are the use of additional sensors to either improve the odometry error, e.g., IMU and odometry sensor fusion, or the external sensory information. Also, a boundary transition zone can be defined to reduce the deterministic character of the in-/outside measurements. Additionally, the parameters for relocalization,  $\sigma_{\max}$ ,  $\sigma_{\min}$ , can be adjusted, such that the robot more often relocalizes itself. This reduces the number of lost tracks but also reduces the performance of the method, as shown in Figure 2.25b.

Table 2.4: CCpp method performance based on Equation (2.62).

	Odometry Error Level					
	0.0	0.2	0.4	0.6	0.8	1.0
1	1.00	0.54	0.26	0.07	-0.12	-0.26
2	1.00	0.60	0.34	0.17	0.10	-0.08
3	0.99	0.82	0.70	0.66	0.60	0.49
4	0.97	0.74	0.65	0.51	0.41	0.38
5	0.99	0.79	0.69	0.57	0.54	0.44
6	0.96	0.76	0.62	0.51	0.47	0.42



(a) Effect of the number of particles on the method performance.

(b) Effect of the relocalization on the method performance.

Figure 2.25: Method performance (traveled distance, lost tracks) with respect to the number of particles and the relocalization parameter  $\sigma_{max}$ .

## 2.7 Discussion

In the previous sections, Section 2.5 and Section 2.6, methods for map estimation and complete coverage path planning for autonomous lawn mowers with only limited sensing capabilities were proposed. In the following, application limitations and implications for the proposed methods are discussed in detail.

### 2.7.1 Map Generation

The proposed loop closure detection and map generation method is able to reliably generate an accurate map estimate for closed environments. However, there are three main issues which should be discussed here: The effects of the odometry error and map complexity, the here proposed map evaluation strategy and the influence of the hyper-parameters.

**Odometry Error & Map Complexity** As demonstrated in Section 2.5.4, the accuracy and reliability of the proposed mapping approach depend highly on the underlying odometry data and the complexity of the environment which should be mapped. Whereas the former might be quite obvious, since with more accurate odometry measurements less optimization is required, the latter can be explained by the different amount of characteristic features between complex and less complex environments. Here, those features are basically structures of the boundary line of the enclosed environment, e.g., corners, which allows for efficient loop closure detection. Many and more complex features stabilize the loop closure detection process but reduce odometry accuracy due to additional required turns by the robot following the boundary line. Thus, complex environments may be more robust for the mapping approach but might result into slightly worse mapping results.

**Map Evaluation** The in Equation (2.45) introduced error measurement shall serve as a reference to test the quality of the here proposed mapping algorithm. It allows the reader to estimate the performance of the mapping algorithm in an intuitive manner. However,

the error measurement is highly tailored to the here presented mapping problem of an enclosed environment. A different error metric for example is proposed in [66], where the relative displacements between poses are measured. However, for this error metric the true poses are required which are not known for the robot in real outdoor environments.

**Hyper-Parameters** The correct choice of hyper-parameters is crucial for the performance of the here proposed mapping algorithm. On the basis of preliminary information about the environment, it is possible to adjust those hyper-parameters properly, e.g., the neighborhood length  $L_{\text{NH}}$  based on preliminary information of the circumference of the enclosed environment. In general, however, this prior information is not available in advance, so the hyper-parameters must be either guessed or learned by the robot.

### 2.7.2 Hyper-Parameter Learning

Learning the hyper-parameters for the proposed mapping algorithm requires convenient cost functions as presented in Equation (2.42) and Equation (2.43). However, these cost functions face two main issues: (1) They are highly tailored to the given problem and make strong assumptions about the odometry error and (2) they are highly non-linear.

In general, cost functions for learning hyper-parameters are tailored to the specific problem. However, the assumption of a zero mean odometry error might not hold due to irregular errors, e.g., slippage, even if for the robot model this assumption holds. Nevertheless, it is a good approximation and allows to specifically design a cost function for hyper-parameter learning without knowledge of the true environment.

As demonstrated in Figure 2.11, the cost functions are highly non-linear which makes it hard for any optimization algorithm to find the global optimum. This is especially true for gradient-based methods. Hence, a non-gradient based method (BO) is used to find the global optimum. However, the number of iterations are limited and thus the global is merely approximated.

### 2.7.3 Complete Coverage Path Planning

The proposed CCPP method works well, especially with decreasing odometry error and increasing map complexity. However, it requires continuous relocalization by applying a simple wall following method since within the working space no useful sensory information can be collected. If the odometry error gets too large and the map complexity drops down, the algorithm requires frequent relocalization whereas a random walk pattern increases its performance due to the lower map complexity. This leads to a point where the proposed method no longer outperforms a random walk pattern as shown in Figure 2.23a. However, in general the CCPP method outperforms the random walk pattern even under large odometry errors.

## 2.8 Conclusion

In this chapter, a mapping approach as well as complete coverage path planning approach for robots with limited sensing capabilities operating in closed environments are introduced. These approaches allow robots to efficiently plan and navigate even if only in-/outside area detectors are available.

**Mapping Generation** The proposed mapping method for enclosed environments does not require any additional assumptions like for example a rectilinear structured environment [16] or linear features [15], [14]. Thus, it enables robots equipped with only simplistic sensors, e.g., lawn mowers, to generate an accurate map estimate of any closed working environment. Therefore, required odometry data is collected using a wall following scheme for which low range sensors or binary sensors are sufficient. As demonstrated, the approach performed well in real as well as in simulated environments. Such an accurate map estimate can then be used for efficient navigation, key to complex path planning strategies.

**Path Planning** The proposed CCP method outperforms the commonly used random walk pattern while coping with high pose uncertainties due to only limited sensing capabilities, e.g., only in-/outside area detectors. The performance of the method was thereby analyzed in different challenging scenarios and with different odometry accuracy levels. Thereby, the method proved to be efficient, especially for complex environments.

### 2.8.1 Detailed Future Work

In future work, the proposed mapping and planning strategies can be tested on a variety of different mobile robots, e.g., different lawn mowers or vacuum cleaners, in order to evaluate in more detail the stability and performance of these methods. Also, the hyper-parameters for the planning method could be learned with the required traveled distance as cost factor. In addition, the hyper-parameter learning could be adopted for other mapping or SLAM algorithms.

## 2.9 Acknowledgments

The part of this chapter concerning the loop closure detection and map generation method is based on the conference paper [67] which was written by Nils Rottmann (NR), Ralf Bruder (RB), Achim Schweikard (AS) and Elmar Rueckert (ER). NR developed the basic ideas, set up the simulation environment and implemented the required algorithms. Together, NR and RB prepared the electronics, the software and conducted the real garden experiments. The paper was written by NR with highly appreciated assistance from RB, AS and ER. AS and ER both greatly improved the paper by contributing important ideas on how to visualize and highlight the key concept.

The part of this chapter concerning the hyper-parameter learning for the proposed loop closure detection and map generation method is based on the workshop paper [68] which was written by Nils Rottmann (NR), Ralf Bruder (RB), Achim Schweikard (AS) and Elmar Rueckert (ER). NR developed the basic ideas and implemented the algorithms. Together, all authors discussed the underlying assumptions for the proposed cost function. The paper was written by NR with contribution from ER to the optimization part.

The part of this chapter concerning the complete coverage path planning is based on the conference paper [69] which was written by Nils Rottmann (NR), Robin Denz (RD), Ralf Bruder (RB) and Elmar Rueckert (ER). NR developed the basic ideas and the general concept. Together, NR and RD set up the simulation environment and conducted the experiments. NR and RB defined the parameter settings for the simulation environment based on real robot data. The paper was written by NR with highly appreciated assistance from RD and ER. ER greatly improved the paper by contributing important ideas on how to statistically evaluate the methods performance.



## 3 Sensor Development for Working Area Detection and Classification

In this chapter, novel chlorophyll fluorescence based approaches for mowing area detection and plant classification are introduced. These approaches rely on active chlorophyll fluorescence stimulation and allow the robot to efficiently detect and classify its working space.

### Contents

---

<b>3.1</b>	<b>Introduction</b>	<b>48</b>
<b>3.2</b>	<b>Related Work</b>	<b>49</b>
3.2.1	Working Area Detection	49
3.2.2	Plant Classification	51
<b>3.3</b>	<b>Scientific Contributions</b>	<b>52</b>
<b>3.4</b>	<b>Chlorophyll Fluorescence</b>	<b>52</b>
3.4.1	General Principle	53
3.4.2	Kautsky Effect & Lifetime	54
<b>3.5</b>	<b>Chlorophyll Sensor</b>	<b>55</b>
3.5.1	Core Sensor Design	55
3.5.2	Signal Path Analysis	57
3.5.3	Hardware Component Analysis	61
3.5.4	Evaluation	62
<b>3.6</b>	<b>Plant Classification</b>	<b>64</b>
3.6.1	Data Acquisition & Preprocessing	65
3.6.2	Deep Learning Classification Approach - Benchmark	66
3.6.3	Low-cost system	68
3.6.4	Evaluation	69
<b>3.7</b>	<b>Discussion</b>	<b>73</b>
3.7.1	Working Area Detection	73
3.7.2	Plant Classification	74

<b>3.8 Conclusion</b> . . . . .	<b>74</b>
3.8.1 Detailed Future Work . . . . .	75
<b>3.9 Acknowledgments</b> . . . . .	<b>75</b>

---

### 3.1 Introduction

In the previous chapter, intelligent navigation and planning strategies for robots in closed environments with focus on autonomous lawn mowers have been discussed. These strategies require a cost-effective and accurate detection method for the working area. At present, this is realized by means of perimeter wire, which leads to high setup and maintenance costs. Therefore, in the first part of this chapter, an active low-cost sensor approach for detecting chlorophyll fluorescence is proposed. The novel and innovative sensing concept allows for a robust working area detection. The lawn detection is thereby based on the averaging of multiple measurements using LED pulses and sensed fluorescence responses. Selecting only low-cost consumer components for the sensor design allows for high-volume production under low-cost aspects. The sensor system thereby is evaluated by analyzing theoretically the signal path, including among other the sampling frequency, the sensed surface area and environmental influences. In real world experiments, the performance of the sensor in an exemplary garden and on collected grass samples was evaluated. The theoretical and practical evaluations show that the sensor's classification result is robust under different environmental conditions, such as changes in lawn quality.

The second part of this chapter discusses possibilities for efficient working area classification, more precisely plant classification by chlorophyll fluorescence analysis. This enables further automation for gardening, e.g., efficient weed removal or scarification, and can help to improve the navigation accuracy due to additional available features. Furthermore, efficient plant classification methods can be used in many other areas, such as monitoring of species diversity for environmental protection or crop harvesting in agriculture. Here, plant classification utilizing the chlorophyll fluorescence responses is analyzed. These responses differ between the individual plant species because of the different composition of chlorophyll a, chlorophyll b, carotenes or carotinoids. The classification approach is first tested using deep learning techniques, more precisely convolutional neural networks, for classifying five different plant species. Therefore, a large data set of spectral responses by stimulating the chlorophyll fluorescence with light of different wavelengths and recording the re-emission using a spectrometer was collected. The results show that using a convolutional neural network leads to promising classification results with a mean accuracy of more than 95%. In a second step, a low-cost approach substituting the spectrometer by commercial consumer phototransistors is evaluated. This reduces available input data and thus classification accuracy but allows for plant classification in the gardening area.



Figure 3.1: Plants species used for testing the proposed plant classification approach.

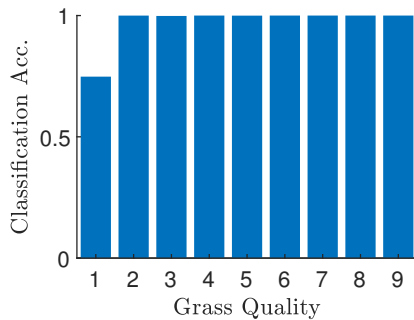
Initially, the related work is discussed (Section 3.2) and the contributions of this chapter are highlighted (Section 3.3). Afterwards, a summary of the most important aspects of chlorophyll fluorescence is given (Section 3.4). In the following, it is further discussed how to efficiently exploit these characteristics of the working environment for navigation and planning (Q2.a in Subsection 1.2) which results into the development of an active chlorophyll fluorescence sensor (Section 3.5). In the second half of this chapter, in Section 3.6, the possibility of accurate plant classification based on the chlorophyll fluorescence emissions of the plants is discussed (Q2.b in Subsection 1.2). Therefore, the data acquisition and preprocessing procedures, a deep learning classification approach as well as a low-cost classification approach are studied. The chapter ends with a discussion in Section 3.7 and a conclusion in Section 3.8.

## 3.2 Related Work

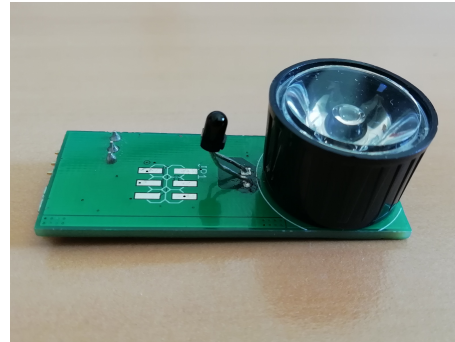
Related work for the working area detection and for the plant classification is discussed. This includes reviewing existing methods for lawn area detection in general and for chlorophyll fluorescence detection specifically. In the second part, state-of-the-art plant classification strategies are summarized and grouped according to their individual classification techniques.

### 3.2.1 Working Area Detection

Many strategies were proposed to detect the boundaries of the working area for autonomous lawn mowers, for example vision based localization and mapping strategies [70, 71] or capacity based sensor technology for detecting humidity [72]. However, since for autonomous mowers the safety impact on leaving the mowing area is high, the sensory systems have to be reliable. Vision based systems use color and texture identifiers to detect grass-containing regions using statistical methods, e.g., Bayes classifier [73] and reach accuracies of 90 %, shaded grass, and 95 %, illuminated grass [74]. Capacity based systems have to be calibrated and are sensitive with respect to change in air conditions, such as rain or fog. In addition, there are local positioning systems which rely on active beacons using trian-



(a) The classification accuracy for the chlorophyll fluorescence sensor for different grass qualities.



(b) The novel chlorophyll fluorescence based sensor. The main components are the LED and the phototransistor.

Figure 3.2: The remote chlorophyll fluorescence sensor and the classification accuracy with respect to different lawn types, Figure 3.11.

gulation [75]. However, such local positioning systems require a-priori an exact map of the environment. The only working electronics in consumer markets nowadays use bounding wire, electro-magnetic field measurement technology which safely detects wire crossing and in-/outside area estimation. Such technique was firstly introduced in lawn mowers in [76]. However, it requires the installation of a perimeter wire surrounding the lawn which results in additional time and maintenance costs. In order to overcome these problems, here a cost-efficient grass detection system based on remote chlorophyll fluorescence sensing, Figure 3.2, is introduced.

Current remote chlorophyll fluorescence sensing systems can be grouped into ground based measurement and long distance systems [77]. The ground based measurement systems can be further partitioned into active and passive ones. The most popular group of sensors for active chlorophyll fluorescence sensing are FLiDAR (Fluorescence Light Detection and Ranging) [78], where brief periodic excitation pulses ( $< 1 \mu s$ ) with defined wavelength (e.g.  $355 nm$ ) are used for excitation. Current FLiDARs are using multiple excitation wavelengths, e.g., for identifying plant species [79] or the stress level [80]. Passive remote sensing, in comparison, relies on the fluorescence induced by the natural sunlight. Since the fluorescence represents only a very small fraction of the recorded spectrum, Fraunhofer Lines are used in order to measure the fluorescence signal, for example using FLD (Fraunhofer Line Discrimination) [81], which was extended in [82],[83]. In general, passive remote sensing techniques can be partitioned into radiance-based (including FLD) and reflectance-based methods, e.g., using the physiological reflectance index (PRI) [84], where reflectance at  $531 nm$  and  $570 nm$  is used for indexing. For a more detailed description about passive remote sensing techniques the reader is referred to a more comprehensive review [85]. Lastly, there are long distance chlorophyll fluorescence sensing techniques, e.g.,

using satellite images to detect chlorophyll in cyanobacterial blooms [86] or globally identifying the functional status of vegetation [87].

Long distance sensing techniques are considered as not applicable for autonomous mowers. FLiDARs on the other hand can be used for accurately identifying the mowing area. However, since autonomous mowers are designed for low purchase and maintenance costs, FLiDARs in general are too expensive. Passive sensing might be a cost effective solution but requires sunlight which limits its applicability, for example when the mowing time should be over night. The same argument holds for vision based systems which are in addition unreliable due to their statistical nature. To overcome these problems, i.e., low maintenance and acquisition costs, reliable detection of the mowing area and constant operational readiness, here a novel active chlorophyll sensor is introduced

### 3.2.2 Plant Classification

Many plant classification approaches are using visual properties such as texture-based and morphological features, for example the leaf shape, the size or the aspect ratio [88, 89]. Such features are often predefined by botanists or chosen arbitrarily. A solution for avoiding predefined features is autonomous feature extraction using for example the principal component analysis (PCA), which leads to high classification accuracies as shown in [89] and [90]. However, in order to achieve such high classification accuracies certain prerequisites have to be fulfilled, for example leaves should not overlap and light conditions should not change drastically. With the rising popularity of deep neural networks, plant classification based on image analysis can nowadays even overcome the above prerequisites [91] and reach similar or better results as with hand-crafted features. A large data base for training such deep neural nets is given in [92]. However, overlapping or clutter may drastically reduce the classification accuracy.

Another procedure for plant classification relies on the recording of reflectance spectra, where overlapping leaves or clutter barely interfere with the classification procedure. However, ambient light, plant age, growth condition and plant health may affect the spectra and hence, complicate classification. Also, a large number of species proved to be difficult to distinguish [93]. However, recently there have been advances in machine learning techniques regarding classification of different weed species using hyperspectral sensing [94]. In [95], a classification accuracy of 97% was achieved exploiting attained reflectance spectra of corn and silver beet. Here, the authors used support vector machines for classification.

Other methods are using active or passive broad- and narrowband spectrometers. They have been effectively used detecting diseases, stress or metabolic deficiencies [96, 97]. In order to extract features from the spectral data, feature extraction techniques such as the principal component analysis are employed. An alternative is chlorophyll fluorescence fingerprinting [98, 99]. However, generating the chlorophyll fluorescence fingerprints requires a certain experimental setup and cannot be easily extended for outdoor applications.

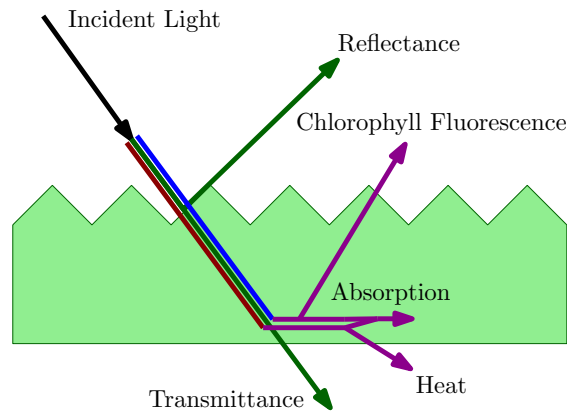


Figure 3.3: Example diagram for chlorophyll fluorescence inspired by [100]. About 78% of the incident radiation is absorbed, while the rest is either transmitted or reflected. About 20% is dissipated through heat and only 2% emitted as fluorescence.

### 3.3 Scientific Contributions

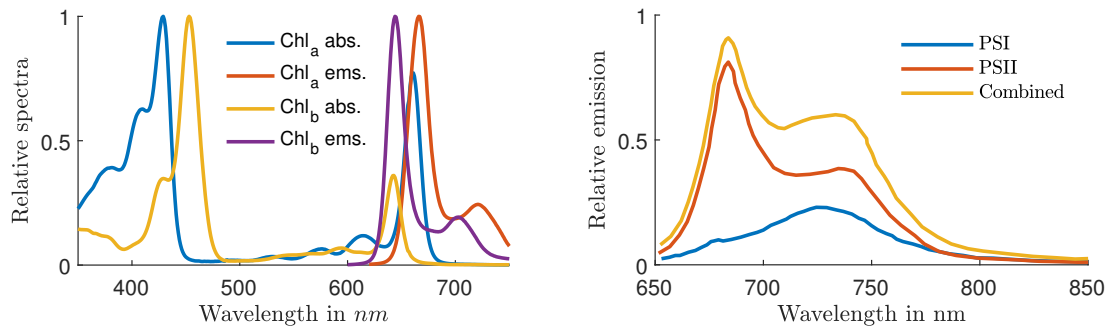
In the following, the novel scientific contributions presented within this chapter are summarized according to the content structure from above.

**Working Area Detection** A novel chlorophyll fluorescence sensor for working area detection is proposed which (1) stimulates the chlorophyll fluorescence by emitting blue light with a standard  $432\text{ nm}$  light emitting diode (LED), (2) detects the chlorophyll fluorescence response using a standard infrared phototransistor and (3) filters the sunlight response by using high stimulation frequencies.

**Plant Classification** The contribution for plant classification based on chlorophyll fluorescence emission spectra is threefold. First, a vast dataset is presented which serves as a comparison standard for chlorophyll fluorescence based plant classification. Second, the general feasibility of distinguishing plant species based on their individual fluorescence spectra is shown. Therefore, a deep learning approach is used for classifying five different plants including a profound analysis of the network hyper-parameters and the generalization performance. Third, a cost-efficient yet accurate approach for plant classification based on chlorophyll fluorescence is proposed which utilizes only standard LEDs and phototransistors along with computationally efficient machine learning methods.

### 3.4 Chlorophyll Fluorescence

In this section, chlorophyll fluorescence as a unique feature of plants and grass is introduced. Therefore, the main concepts of chlorophyll fluorescence such as the absorption



(a) Absorption and emission fluorescence spectra for chlorophyll a and b in diethyl-ether taken from the PhotochemCAD database [103],[104]. (b) Relative chlorophyll emission spectra for Photosystem I (PSI) and Photosystem II (PSII) under room temperature corresponding to the maximal fluorescence states using green barley leaves [105].

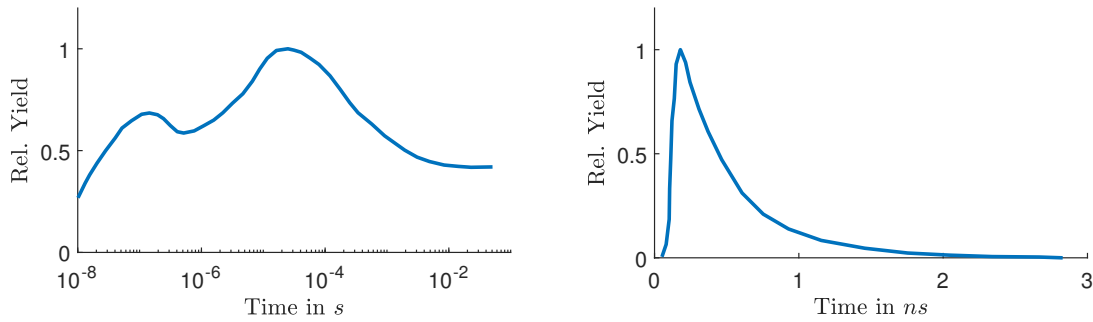
Figure 3.4: Chlorophyll a,b absorption and emission spectra, left panel, and Photosystem I (PSI), Photosystem II (PSII) emission spectra, right panel.

and emission fluorescence spectra, the “Kautsky-Effect” and the chlorophyll fluorescence life time are briefly summarized. A more detailed survey can be found in [101] and [102].

### 3.4.1 General Principle

Light energy absorbed by plants, more specifically by the chlorophyll molecules, can either drive photosynthesis reaction, it can be dissipated as heat or re-emitted as light which is called chlorophyll fluorescence, Figure 3.3. These three processes are in competition to each other, thus a decrease in efficiency at one process will increase the efficiency at another. In general, the light energy re-emitted by the chlorophyll fluorescence is of a magnitude much lower than the absorbed light, between 1 – 2%. During the chlorophyll fluorescence process, chlorophyll molecules turn into an excited state absorbing light energy and, by falling back into a non-excited state, re-emitting light energy. Thereby, chlorophyll either absorbs high-energy blue (350 – 500 nm) or low-energy red light (600 – 680 nm) and re-emits low-energy red/infrared light (650 nm - 800 nm), Figure 3.4a. Green light (500 – 600 nm) is reflected, which gives plants the typical leafy green appearance. This effect can be efficiently exploited for sensory systems to detect the presence of plants by stimulating chlorophyll fluorescence with high energy blue light and sensing the re-emitted low-energy red/infrared light.

For the chlorophyll fluorescence two protein complexes, Photosystem I (PSI) and Photosystem II (PSII), are relevant. PSI consists of chlorophyll a, chlorophyll b and carotene and PSII of chlorophyll a, chlorophyll b and xanthophyll. Both systems lead to characteristic chlorophyll fluorescence emission spectra as shown in Figure 3.4b. In combination,



(a) The figure shows the chlorophyll a fluorescence yield change in the dark-adapted cells of the green alga *Chlorella* after a saturating nanosecond laser flash (adjusted from [110],[111]).

(b) The figure shows the chlorophyll fluorescence decay for a spinach leaf measured at  $684\text{ nm}$  after an excitation impulse with a width of  $70 - 80\text{ ps}$  (adjusted from [112]).

Figure 3.5: The “Kautsky-Effect”, left panel, and the chlorophyll lifetime, right panel.

they lead to the characteristic spectral response with the two distinct peaks at around  $684\text{ nm}$  and around  $738\text{ nm}$ . The proportions of PSI and PSII in the spectral response vary depending on the molecule composition of chlorophyll a, chlorophyll b, carotene and xanthophyll in the investigated plant species. This allows utilizing the chlorophyll fluorescence effect for plant classification by comparing the spectral responses for different plant species.

### 3.4.2 Kautsky Effect & Lifetime

An important characteristic for chlorophyll fluorescence is the so called “Kautsky-Effect” [106], where a plant’s reaction to sudden light changes is investigated, e.g., the plant is unveiled in the sunlight. This change of setting results in an increase in the yield of chlorophyll fluorescence. This holds for the first second after which the fluorescence level drops down over a few minutes until it reaches a steady state. This drop down effect is known as fluorescence quenching [107]. In Figure 3.5a, the “Kautsky-Effect” is shown which is induced by the PSII [108] reaction centers being in a “closed” state. This decreases the photosynthesis process which on the other hand increases the chlorophyll fluorescence. Based on the “Kautsky-Effect”, certain effects such as plant stress can be measured by recording the relative fluorescence yield from plants even under full sunlight [109].

An important characteristic for exploiting chlorophyll fluorescence for sensory systems is the life time of the chlorophyll fluorescence. This induces a lower bound for the sampling period of any sensor. The life time is thereby the time after stimulation in which chlorophyll fluorescence can be measured and lies around one nanosecond. For example, Schmuck and Moya [112] showed for spinach leaves that at steady state conditions the mean lifetime is  $0.415\text{ ns}$  and when closing all reaction centers of the PSII, thus enhancing chlorophyll

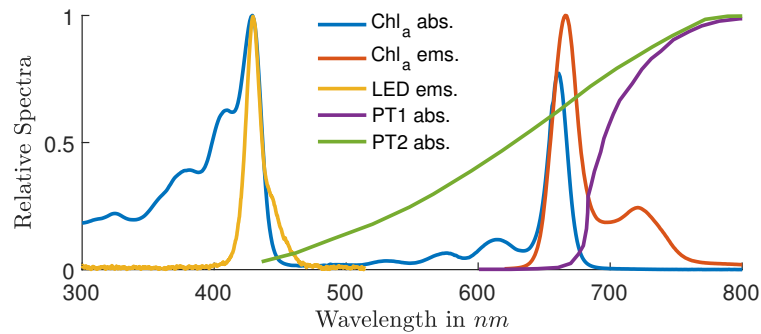


Figure 3.6: Examples of different normalized absorption and emission spectra. The blue and red lines show the absorption and emission fluorescence spectra for Chlorophyll a in diethyl-ether taken from the PhotochemCAD database [103],[104]. The yellow line represents the measured emission spectrum of a consumer LED with emission peak at  $432\text{ nm}$ . The purple and green lines show the spectral sensitivities for the RPT-37PB3F Phototransistor from Rohm Semiconductor and the PT480 from SHARP.

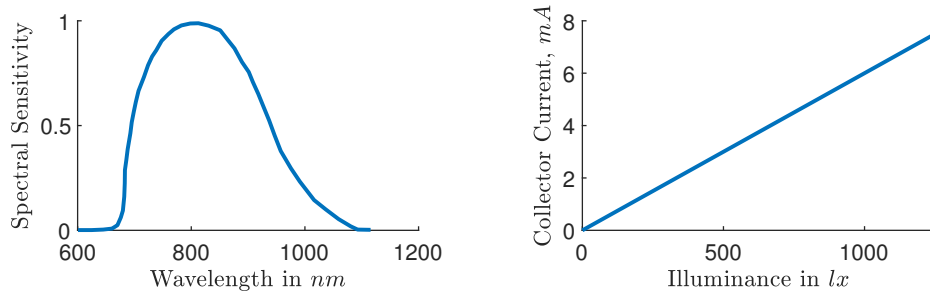
fluorescence, the mean lifetime is around  $2\text{ ns}$ , shown also in Figure 3.5b. Similar results have been achieved in [113] with maple and spinach leaves and in [114] with maize and spruce leaves.

## 3.5 Chlorophyll Sensor

The basic idea behind the sensor design presented here is that chlorophyll fluorescence is actively excited with blue high energy light and the red/infrared low energy light response measured. Therefore, a consumer Light-Emitting Diode (LED) with a peak wavelength of around  $430\text{ nm}$  for the excitation and a consumer Phototransistor (PT) with spectral sensitivity in the red/infrared area are used. Figure 3.6 illustrates the idea based on the spectral curves of the chlorophyll fluorescence and the LED and PT. In the following, the development and evaluation of the proposed chlorophyll sensor is described in detail. This includes the core sensor design, a fundamental analysis of the signal path as well as an examination of used hardware components.

### 3.5.1 Core Sensor Design

As illustrated in Figure 3.6 the absorption spectrum of chlorophyll is particularly strong in the range around  $430\text{ nm}$ , whereas the emission spectrum is located in the area of  $650 - 750\text{ nm}$ . Thus, a consumer LED with emission peak at around  $430\text{ nm}$  is required as a stimulation source and a standard PT with a sufficient good spectral sensitivity between  $650 - 750\text{ nm}$  is required as an absorption sink. Moreover, the emission spectrum of the chosen LED and the spectral sensitivity of the PT should not overlap. Otherwise it is not



(a) Spectral sensitivity of the RPT-37PB3F with respect to the wavelength of the received light.

(b) Resulting collector current with respect to the illuminance filtered by the spectral sensitivity.

Figure 3.7: Spectral sensitivity and collector current of the RPT-37PB3F.

possible to distinguish between fluorescence response and LED radiation. Here, a problem might be encountered since a preferred spectral response of a PT should have high yield in the relevant area ( $650 - 750 \text{ nm}$ ) and zero yield elsewhere. However, real PTs produced for the consumer market have a spectral response either in the visible or infrared range where the emission response of chlorophyll lies between these ranges. Hence, a trade-off between an optimal spectral response and cost-efficiency has to be made. The emission spectrum for the here chosen LED (yellow) together with the spectral sensitivities for two different consumer PTs (purple, green) are drawn in Figure 3.6. The RPT-37PB3F has just a low spectral sensitivity in the desired area whereas the PT480 shows there high yield but intersects with the emission spectrum of the LED. Here, we use the RPT-37PB3F<sup>1</sup> since its spectral sensitivity does not intersect with the LED emission spectrum. In Figure 3.7, a detailed view of the spectral sensitivity and the resulting collector current with respect to the received illuminance is given.

In order to distinguish between the excited chlorophyll fluorescence and the ambient light (e.g. sunlight), the LED signal is modulated with a certain frequency  $f_{\text{LED}}$ . The current signal captured by the PT is transformed using a current-to-voltage converter (transimpedance amplifier), the output voltage is further amplified and the resulting signal band pass filtered such that it is freed of ambient light influences. The control unit of the sensor is a small microprocessor, the *ATMEGA32U4*<sup>2</sup>, which generates the excitation signal for the LED and receives the amplified and filtered chlorophyll fluorescence signal. To achieve high excitation frequencies, the in- and output signals are generated and captured by directly using interrupt routines. In addition, plastic lenses were used for emitting

<sup>1</sup>RPT-37PB3F Datasheet, ROHM Semiconductor, [https://fscdn.rohm.com/en/products/databook/datasheet/opto/optical\\_sensor/photo\\_transistor/rpt-37pb3f-e.pdf](https://fscdn.rohm.com/en/products/databook/datasheet/opto/optical_sensor/photo_transistor/rpt-37pb3f-e.pdf)

<sup>2</sup>ATmega16U4/ATmega32U4 Datasheet, Atmel, [http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7766-8-bit-AVR-ATmega16U4-32U4\\_Datasheet.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7766-8-bit-AVR-ATmega16U4-32U4_Datasheet.pdf)

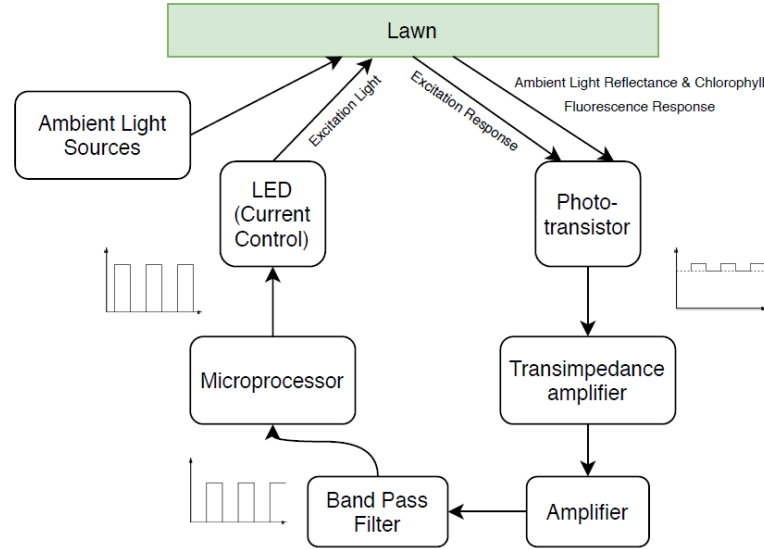


Figure 3.8: The signal path for the proposed low cost sensor. The microprocessor controls the LED which emits a pulsed light for stimulating the chlorophyll process. The light radiated back is then absorbed by the PT and the result further processed and sent back to the microprocessor for evaluation.

and receiving optics to focus the light onto and from the measurement area. The whole setting allows for high-volume lowest cost sensors. A detailed overview over the electronic components and circuit diagrams can be found in the appendix, Section D.

### 3.5.2 Signal Path Analysis

In the following, the signal path as presented in Figure 3.8 is analyzed. Therefore, various parameters have to be considered when designing the sensor in order to achieve robust lawn classification results, e.g., sensor apertures or electronic component characteristics. Parameters which play essential roles in the analysis are listed in Table 3.1. While analyzing the signal path, different spectral functions  $f(\lambda)$  similar to those as shown in Figure 3.6 are required, where  $\lambda$  is the wavelength. If referred to function  $f(\lambda)$  as to be normalized, then

$$\int f(\lambda)d\lambda = 1, \quad (3.1)$$

and if referred to function  $f(\lambda)$  as to be relative, then

$$f(\lambda) \in [0, 1] \quad \forall \lambda. \quad (3.2)$$

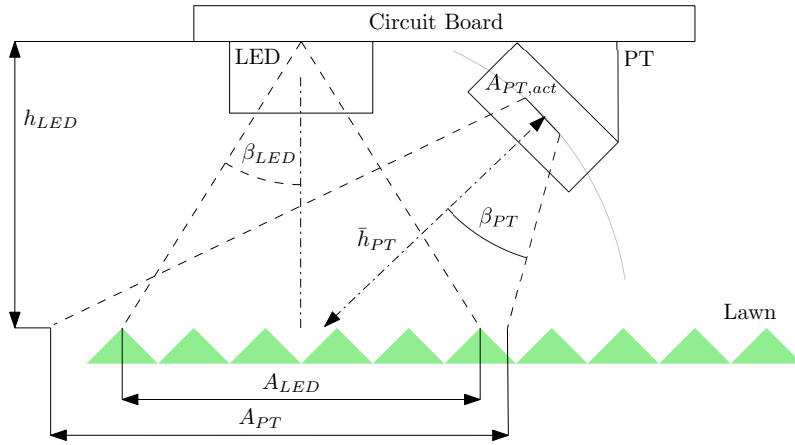


Figure 3.9: A sketch of the sensor emission (LED) and absorption (PT) apertures. Optimally, the apertures are aligned to the same focus point with  $A_{LED} = A_{PT}$ .

**Sensor Apertures & Chlorophyll Fluorescence** In Figure 3.9, a sketch of the apertures for the LED and PT of the designed sensor are shown. Let  $\Phi_{LED}$  be the luminous flux of the LED. First, the light of the LED is conically sent to the grass and illuminates a certain area

$$A_{LED} = \pi (\tan(\beta_{LED}) h_{LED})^2, \quad (3.3)$$

where  $\beta_{LED}$  is the lens angle and  $h_{LED}$  the distance of the sensor to the ground. In order to estimate the resulting luminous flux response of the chlorophyll to the PT the normalized emission spectrum  $f_{LED}(\lambda)$  of the LED and the relative absorption spectrum  $g_{Chl}(\lambda)$  of chlorophyll *a* are used. In combination with the re-emission magnitude for chlorophyll fluorescence  $\gamma$ , this leads to

$$\Phi_{Chl} = \gamma \Phi_{LED} \int f_{LED}(\lambda) g_{Chl}(\lambda) d\lambda. \quad (3.4)$$

The amount of luminous flux from the chlorophyll fluorescence response received by the PT varies depending on the lens aperture. Here, it is assumed that PT's lens aperture is such positioned, that the complete illuminated area  $A_{LED}$  can be detected, which leads to a luminous flux from the chlorophyll fluorescence response into the PT of

$$\Phi_{PT,AC} = \varphi \Phi_{Chl} \int f_{Chl}(\lambda) g_{PT}(\lambda) d\lambda. \quad (3.5)$$

Here,  $f_{Chl}(\lambda)$  is the normalized emission spectrum for the chlorophyll fluorescence,  $g_{PT}(\lambda)$  the relative spectral sensitivity of the PT and  $\varphi$  the part of the scattered light from the lawn surface which is received by the PT. For calculating  $\varphi$ , the reflected scattered light is assumed to be equally distributed in a half sphere from each point of the illuminated area. The distance between the active measurement surface  $A_{PT,act}$  and the illuminated area is  $h_{PT}$ , where  $h_{PT}$  differs with respect to the considered points of the active measurement

surface and the illuminated area. To simplify the calculations, it is assumed that a mean distance between the PT and the illuminated lawn area  $\bar{h}_{PT}$  can be defined. The part of the hemisphere of the emitted fluorescence which is received by the active measurement surface is then defined as

$$\varphi = A_{PT,act}/(2\pi\bar{h}_{PT}^2). \quad (3.6)$$

**Sun Radiation** In addition to the desired high frequency signal from the LED excitation, the PT also registers a low frequency signal from an ambient light source, the sun, which directly emits with  $E_{Sun}$ . To determine the amount of luminous flux reflected by grass and received by the PT, the surface Albedo  $\alpha \approx 0.25$  [115] and the normalized reflectance spectrum of grass  $f_{Grass}(\lambda)$  [116] are used. With the same assumptions and simplifications as before, this leads to

$$\Phi_{PT,DC} = \alpha \varphi E_{Sun} A_{PT} \int f_{Grass}(\lambda) g_{PT}(\lambda) d\lambda \quad (3.7)$$

for the luminous flux to the PT based on emitted sun radiation. Consider here that  $\varphi$  is not necessarily the same value as in Equation (3.5), since the mean distance between the PT and the scanned area and the mean distance between the PT and illuminated lawn area are not necessarily the same. However, for simplification it is assumed that the apertures are so aligned that the same  $\varphi$  holds for both settings.

**Light Emission Acquisition** The luminous flux  $\Phi_{PT} = \Phi_{PT,DC} + \Phi_{PT,AC}$  induces a response of the PT, resulting in a collector current  $I_C$  based on the collector emitter voltage  $V_{CE}$  applied to the PT. For example, let  $V_{CE} = 5V$  then

$$I_C \approx \frac{3mA}{500lx} \frac{\Phi_{PT}}{A_{PT,act}}, \quad (3.8)$$

where the active measurement surface of the PT is given with  $A_{PT,act} = 7.55mm^2$ . The Alternating Current (AC) part of the PT response generated by the pulsing LED is around  $I_{C,AC} \approx 27\mu A$  whereas the the Direct Current (DC) part increases linearly with increasing sun radiation, e.g., for  $E_{Sun} = 10^5lx$  a value of  $I_{C,DC} \approx 3mA$  can be calculated. However, the *VSOP98260*<sup>3</sup> can only resolve currents up to  $400\mu A$ . Thus, the maximum sun radiation with which the sensor works is around  $E_{Sun} = 10^4lx$ . To solve this problem, the sensor can be shaded or a resistance bridge can be included to discharge part of the current. These options are discussed in more detail in Section 3.7. Finally, the *VSOP98260* amplifies and filters the signal and forwards it to the microprocessor.

**Signal Generation & Classification** Autonomous lawn mowers in general move with a maximum velocity of  $v_{max} \approx 1m/s$ . Given an area detection radius for the sensor of

<sup>3</sup>VSOP98260 Datasheet, Vishay Semiconductor, <https://www.vishay.com/docs/82447/vsop98260.pdf>

Table 3.1: Important parameters for the analysis of the signal path.

Symbol	Value	Unit	Description
$A_{LED}$	$2.30 \times 10^{-3}$	$m^2$	illuminated area by the LED
$A_{PT}$	$3.40 \times 10^{-3}$	$m^2$	scanned area by the PT
$A_{PT,act}$	$7.55 \times 10^{-6}$	$m^2$	active measurement surface of the PT
$E_{Sun}$	$0 - 10^5$	$lx$	illuminance of the sun with $10^5$ in summer and by clear sky
$f_{LED}$	38000	$Hz$	pulsing frequency of the LED
$f_{sensor}$	179	$Hz$	classification frequency of the sensor
$h_{LED}$	0.1	$m$	distance from the sensor to the ground
$\bar{h}_{PT}$	0.12	$m$	mean distance between the PT and the illuminated area
$I_{C,AC}$	$2.7 \times 10^{-5}$	$A$	collector current of the PT, AC signal
$I_{C,DC}$	$0 - 3 \times 10^{-3}$	$A$	collector current of the PT, DC signal
$f_{LED}(\lambda)$	see Figure 3.6	1	normalized emission spectrum of the LED
$f_{chl}(\lambda)$	see Figure 3.6	1	normalized emission spectrum of the chlorophyll fluorescence
$f_{Grass}(\lambda)$	see [116]	1	normalized reflection spectrum of grass
$g_{chl}(\lambda)$	see Figure 3.6	1	relative absorption spectrum of chlorophyll a
$g_{PT}(\lambda)$	see Figure 3.6	1	relative spectral sensitivity of the PT
$\alpha$	0.25	1	surface Albedo
$\beta_{LED}$	15	$^\circ$	lens angle of the LED
$\gamma$	0.02	1	re-emission magnitude for the chlorophyll fluorescence
$\varphi$	$8.34 \times 10^{-5}$	1	part of scattered light from the lawn surface received by the PT
$\Phi_{LED}$	145	$lm$	luminous flux of the LED under full power
$\Phi_{chl}$	2.112	$lm$	luminous flux from chlorophyll fluorescence response of the grass
$\Phi_{PT,AC}$	$3.39 \times 10^{-5}$	$lm$	luminous flux received by the PT, modulated AC signal
$\Phi_{PT,DC}$	$0 - 3.8 \times 10^{-3}$	$lm$	luminous flux received by the PT, sunlight DC signal

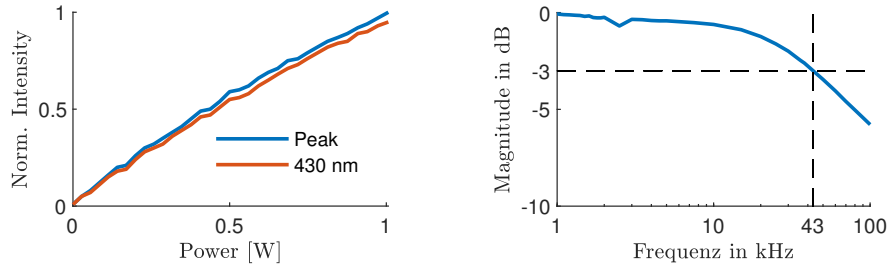
$r_{LED} \approx 0.025$  m and requiring an overlap between measurement areas of  $r_{LED}$ , the sensor's measurement frequency has to be at least

$$f_{sensor,min} = \frac{v_{max}}{r_{LED}} = 40 \text{ Hz.} \quad (3.9)$$

Given the Bode plot and cut off frequency of the used LED, Figure 3.10b, a preamplifier circuit (*VSOP98260*) with a carrier frequency of  $f_{VSOP} = 38$  KHz and a range for the pass band of 20 KHz – 60 KHz is chosen. Thus, pulsing the LED with the carrier frequency  $f_{VSOP}$  exploits optimal the trade-off between high sampling frequencies and sensor light intensity. The chlorophyll fluorescence response speed lies around 2 ns, Subsection 3.4.2, and thus is with approximately 500 MHz multiple orders of magnitude faster than the proposed sensor. However, in order to ensure stability in the presence of measurement noise which might result to false positives or false negatives,  $N = 200$  measurements are performed. The sensor frequency then becomes

$$f_{sensor} = \frac{1}{N} f_{LED} - \epsilon \approx 180 \text{ Hz} - \epsilon, \quad (3.10)$$

where  $\epsilon$  reflects the time required for sending the data over the serial connection during which no measurements are processed. For the developed sensor a frequency of  $f_{sensor} \approx 179$  Hz could be measured and thus  $\epsilon \approx 1$  Hz. The classification output from



(a) Relative light emission of the used high-power LED in proportion to the power consumption.

(b) Bode plot for the LED in use. The cutoff frequency is at  $f_c = 43 \text{ kHz}$ .

Figure 3.10: Relative light emission and Bode plot for the used high-power LED.

the sensor is given by

$$c = \frac{n}{N}, \quad (3.11)$$

where  $n$  is the number of measured chlorophyll fluorescence responses within  $N$  LED pulses.

### 3.5.3 Hardware Component Analysis

In this part, the individual hardware components of the sensor are evaluated in order to determine optimal settings with respect to range, ambient light tolerance and noise suppression.

**Light Emission** For the light emission, a high-power LED is used with  $1 \text{ W}$  maximum power consumption and a resulting luminous flux of approximately  $145 \text{ lm}$ . First, the LED was examined for its behavior with increasing power supply under constant current flow using the CCS200 spectrometer from *Thorlabs, Inc.* In Figure 3.10a, the measured data are depicted where the red curve shows the relative intensity at  $430 \text{ nm}$  and the blue curve the relative intensity at the peak of the measured spectrum. The emitted light intensity increases nearly linear with a small decrease in the slope. Second, the frequency response of the LED was evaluated measuring the light emission with a *BPX 65* photodiode connected to an oscilloscope (DSO6014A from *Agilent Technologies*). The photodiode has a rise and fall time of  $t_r, t_f = 0.012 \mu\text{s}$ . Thus, it can resolve a maximum frequency of  $41, \bar{6} \text{ MHz}$ . To drive the LED with different frequencies a programmable function generator (HM8131.2 from *HAMEG*) was used. In Figure 3.10b, a bode plot for the LED emission is shown. The cutoff frequency here is approx.  $f_c = 43 \text{ kHz}$ .

**Fluorescence Detection** For the chlorophyll fluorescence signal detection a infrared, high-sensitivity PT, the RPT-37PB3F, is used. First, the frequency response of the PT is evaluated utilizing an oscilloscope (DSO6014A from *Agilent Technologies*) and a *OVP214VC*

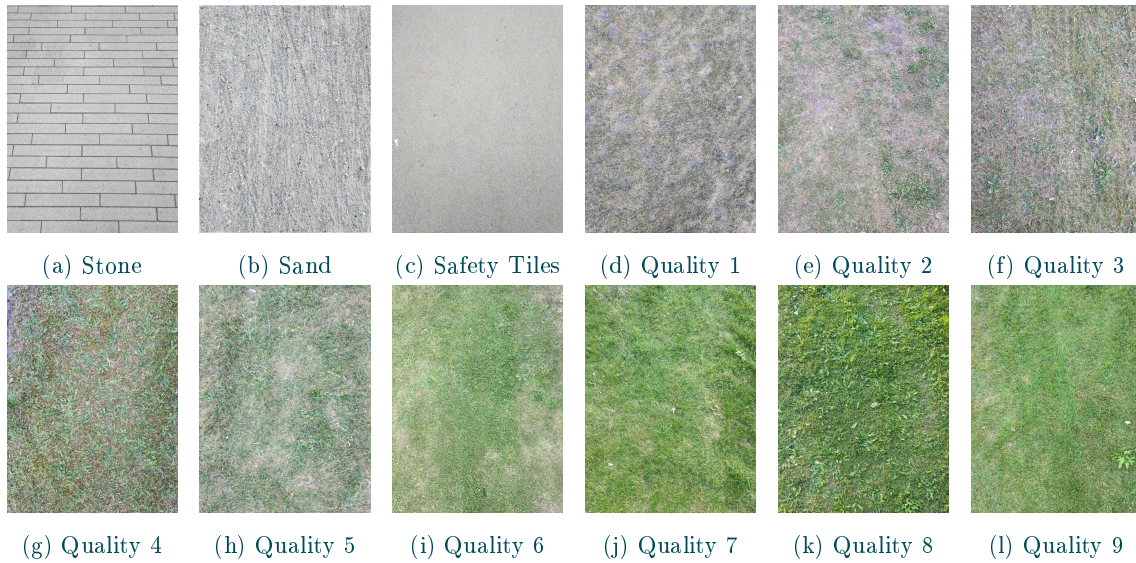


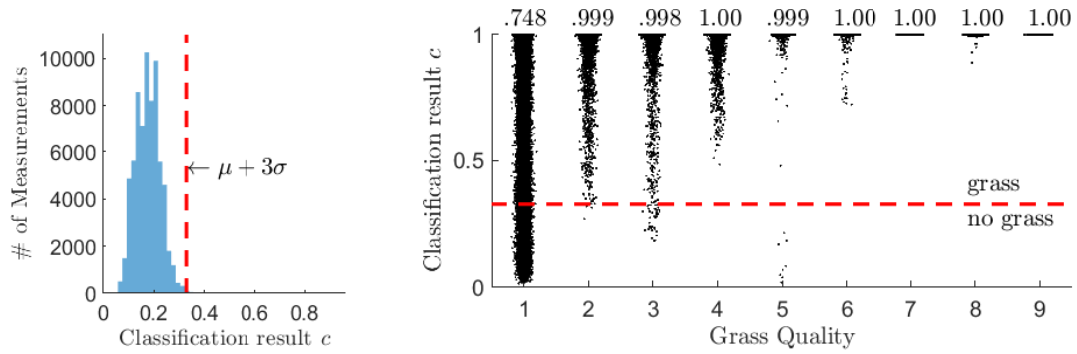
Figure 3.11: Different lawn types, sorted and classified based on the observable relative quality.

laser driven by the programmable function generator (HM8131.2 from *HAMEG*). The PT can easily reach the in the data sheet specified response (rise and fall) times of  $t_r = t_f = 10 \mu s$ . Hence, the PT is able to detect signals with a frequency up to  $f = 50 kHz$  without loss of sensitivity. Second, the saturation illuminance of the sensor was evaluated. This is mainly limited by the maximum input current to the *VSOP98260*. The maximum utilization was achieved under direct light on the sensor at  $3000 lx$ , which with a surface Albedo of 0.25 corresponds to a solar radiation of about  $12000 lx$ . This is consistent with the previous calculations in Subsection 3.5.2.

### 3.5.4 Evaluation

The sensor was tested in 9 different garden environments, which have been sorted according to their visual grass quality, and on 3 different non-grass environments, see Figure 3.11. In total 280,579 samples were collected. In addition, the sensor was tested on a mobile robot where an exemplary garden was scanned.

**Statistical Analysis in different Garden Environments** Classification samples in 9 different lawn environments were collected by attaching the sensor onto an autonomous lawn mower. For each lawn type approximately 23,500 samples were collected. In addition, data samples for 3 non-grass environments were recorded in order to analyze sensor noise. The non-grass as well as the grass environments are shown in Figure 3.11, where they are classified according to their visual grass quality. Here, a grass quality of 1 represents the lowest possible grass quality and 9 the highest.



(a) The classification results for the non-lawn environments with a mean and standard deviation of  $\mu = 0.177$  and  $\sigma = 0.051$ . The  $\mu + 3\sigma$  decision line for grass detection is shown in red. (b) The classification results  $c$  for the 9 different lawn environments, where the environments are sorted based on their visual grass quality with 1 being the lowest and 9 the highest grass quality. Above each lawn category the amount of correctly classified measurements is shown.

Figure 3.12: Statistical evaluation of the chlorophyll fluorescence sensor on 9 different lawn and 3 different non-lawn environments, Figure 3.11, given the classification results  $c$  defined in Equation (3.11). The sensor shows stable and reliable classification results with approximately 100% accuracy for different lawn types, except for the lawn with the lowest quality, whereas even for such grass an accuracy of approximately 75% could be reached.

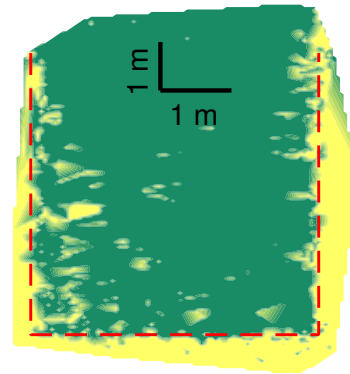
The sensor noise, analyzed by evaluating the sensor results collected on the non-lawn environments, is approximated as a normal distribution. It has a mean classification result of  $\mu \approx 0.177$  and a standard deviation of  $\sigma \approx 0.051$ . Defining a positive classification result (grass detected) as  $c > \mu + 3\sigma$  results in exclusion of nearly all false positive results, since 99.73% of all errors are within the  $\mu \pm 3\sigma$  region. In Figure 3.12a, a histogram of the classification values  $c$  for the non-grass measurements is shown with which the mean and standard deviation for the sensor noise was calculated.

Based on the  $\mu + 3\sigma$  decision boundary, it can be decided whether grass has been detected or not. In Figure 3.12b, the classification results  $c$  for the different lawn environments are shown, where the grass-quality values are inflated with some random noise for better readability. Above each lawn type, the classification accuracy is added, thus the amount of correctly classified samples, which were already shown in Figure 3.2a. All of the different lawn types reach approximately 100% classification accuracy except the lawn with the lowest quality, which still reaches a classification accuracy of approximately 75%.

**Mowing Area Detection** The proposed chlorophyll fluorescence sensor was additionally tested in a realistic garden environment deployed on an autonomous lawn mower. For



(a) The section of the garden environment from the bird's eye perspective.



(b) The interpolated sensor decision results where green signals grass detection and yellow non-grass detection.

Figure 3.13: Evaluation results for the chlorophyll fluorescence sensor under real garden conditions.

the localization of the lawn mower the real-time locating system (RTLS) MDEK1001 from *Decawave* was used. In Figure 3.13, the relative classification results  $c$  are shown. Figure 3.13a shows the evaluated section of the garden environment from the bird's eye perspective and Figure 3.13b the interpolated sensor measurements. The proposed chlorophyll fluorescence sensor reliably detects grass and thus the working space for the autonomous lawn mower.

### 3.6 Plant Classification

In the previous section, it was shown how chlorophyll fluorescence can be exploited for efficiently detecting the mowing area for autonomous lawn mowers. Now, in a further step, it is investigated whether the chlorophyll fluorescence response can also be used for plant classification. This could allow a more advanced automation of gardening tasks, e.g., through automated weed removal or scarification. For this, the first step is to acquire and preprocess chlorophyll fluorescence spectral data for different common plants using a standard spectrometer. Based on the acquired data set, a deep learning classifier is introduced which proves the general feasibility and serves as a benchmark for a low-cost design. This low-cost design utilizes consumer PTs instead of a spectrometer to record the fluorescence response. This low-cost design then enables plant classification for domestic garden robots but reduces the classification accuracy.

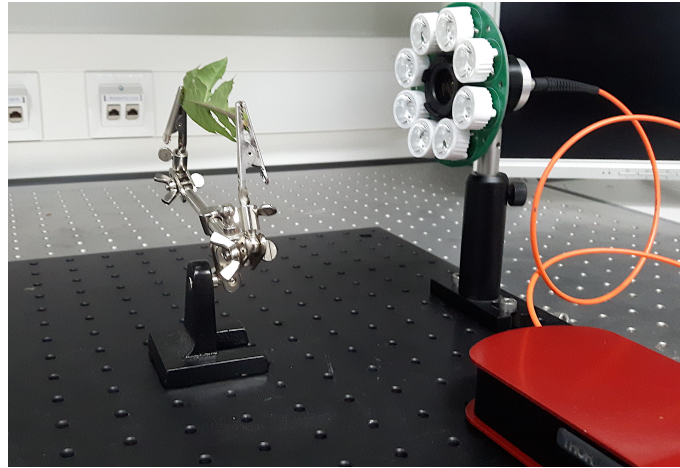


Figure 3.14: Experimental setup for the data acquisition.

### 3.6.1 Data Acquisition & Preprocessing

The spectral data required for testing the plant classification approach was recorded using the *CCS200 - Compact Spectrometer* (Thorlabs, Inc.) which has a measurement range from 200 nm to 1000 nm. The recorded spectral data, consisting of 3648 measurement points uniformly distributed from 194.64 nm to 1013.72 nm, represent the measured relative light by the spectrometer. As shown in Figure 3.4a, the absorption spectra of chlorophyll a or b are in the range of 350 nm to 500 nm. Hence, for excitation different standard LEDs with light emission peaks at {375, 385, 400, 415, 432, 445, 460} nm were used. For the data acquisition, the spectrometers fiber was connected with an optical lens, which was placed 10 cm away from the analyzed plant. Around the optical lens, the excitation LEDs were ordered in a cyclic pattern. Both, the optical lens and the excitation LEDs were focused on the same reference point 10 cm away. The recordings were collected in the dark, thus without external illumination. In Figure 3.14, the experimental setup is shown. A measurement is defined as recording of the spectral data for each excitation LED separately plus recording the emission spectrum without excitation. For the classification data set, four different plant species and lawn grass were used as test objects. From each of the five plant species, 100 leaves were taken and each leaf was measured 5 times which resulted in a total number of 2500 measurements over the 5 plant species. The leaves taken have been selected at random from different healthy plants of the same species in order to get a representation of a typical population. In Table 3.2, the species and the number of data samples collected are summarized.

To ensure that the learning procedure does not inherit information such as light intensity, the collected data is rescaled using the min-max normalization

$$y' = \frac{y - \min(y)}{\max(y) - \min(y)}. \quad (3.12)$$

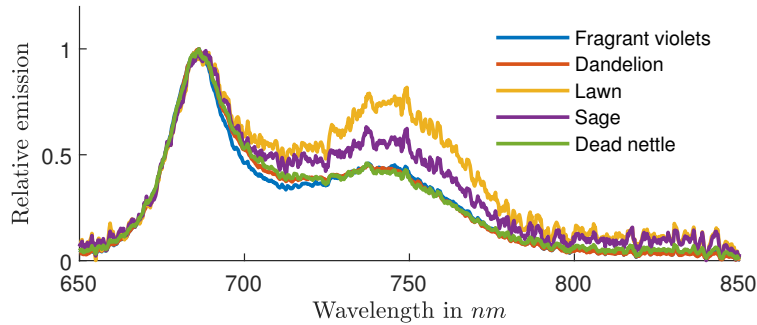


Figure 3.15: Example measurements of the chlorophyll fluorescence response for the five different plants species. The spectral data are cropped and normalized, Equation (3.12), and, for better readability, smoothed with a moving average filter with a span of 5. However, sensor noise is still visible due to a low signal-to-noise ratio.

Additionally, group numbers were assigned to the measurements such that measurements taken from the same leaf are clustered. This allows to ensure that there are no measurements from the same leaf in both the training and the test set. Examining preprocessed example measurements of the chlorophyll fluorescence response for the five different plant species, Figure 3.15, differences especially strong between lawn, sage and the rest are directly observable.

### 3.6.2 Deep Learning Classification Approach - Benchmark

In order to distinguish the different plant species based only on their individual fluorescence spectrum, deep learning [117] as a well-established but computationally expensive method is applied to the classification problem. In other research fields, e.g., blood species identification [118] or classification of different bacteria based on Raman spectra [119], convolutional neural networks recently showed promising performance on spectral data.

Table 3.2: Plant species for classification, where lawn consists of different grass species depending on the type of lawn.

Latin Name	English Name	Data
<i>Viola odorata</i>	Fragrant violets	$5 \times 100$
<i>Taraxacum</i>	Dandelion	$5 \times 100$
Gramina	Lawn	$5 \times 100$
<i>Salvia</i>	Sage	$5 \times 100$
<i>Lamium</i>	Dead nettle	$5 \times 100$

Table 3.3: Hyper-parameter values that were evaluated during architecture optimization using grid search.

Parameter	Examined values
# convolutional blocks	1, 2, 3, 4
# convolutional layers per block	1, 2, 3, 4, 5
# filters in convolutional layers	8, 16, 32, 64
kernel size in convolutional layer	3, 5, 7
# fully connected layers	1, 2, 3
# neurons per fully connected layer	10, 50, 100, 200

The general idea is to learn sufficient features implicitly and completely data-driven from the raw spectral input. Hence, a VGG-like architecture is utilized consisting of convolutional blocks with pooling layers ( $2 \times 2$ ) in between, followed by fully connected layers [120]. The one-dimensional convolutions are applied on the spectral domain while the multiple spectra that were acquired with different LEDs are integrated as channels. As activation functions, the Rectified Linear Unit (ReLU) function is used throughout the network except for the last layer where *softmax* activation functions are used for the classification.

The deep learning architecture was optimized by applying a grid search on specific values for the following hyper-parameters: the number of convolutional blocks, the number of convolutional layers within each block, the number of filters in each convolutional layer, the kernel size in each convolutional layer, the number of fully connected layers as well as the number of neurons in each fully connected layer. The hyper-parameter values that were examined in this study can be found in Table 3.3. For each combination of parameters, the corresponding architecture was trained on 72% of the data with 8% for validation and tested on the remaining 20%. To ensure a fair comparison, the split was the same for all combinations. Furthermore class balancing was applied to the training set utilizing oversampling of the minority classes. In total, 2,880 different architectures were evaluated. For training, the *adam* optimizer [121] and a batch size of 32 were used. The maximum number of epochs was 100, but training was stopped if the validation loss rose constantly over ten epochs.

The generalization capability of the best architectures were evaluated by performing a Monte-Carlo cross validation with  $N = 100$  runs. In each run, the data set was randomly divided into a training (72%), validation (8%) and test set (20%) and an evaluation was performed. As the results of the first experiment might be due to the specific train-test-split, the cross validation was not only applied to the best architecture, but to the five most accurate ones (see Table 3.4).

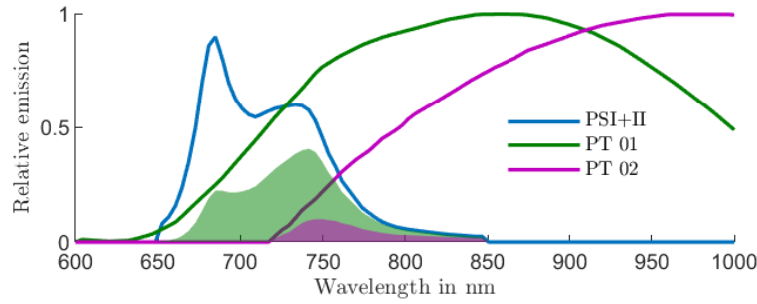


Figure 3.16: Feature extraction given two exemplary PTs. The generated features are based on the integral given in Equation (3.13) and displayed as the areas beneath the curves.

### 3.6.3 Low-cost system

For the low-cost approach, a substitution of the spectrometer with consumer PTs is performed in order to test whether sufficient accurate classification results can be achieved. A phototransistor uses the photoelectric effect to generate a photocurrent which is amplified in the transistor with an amplification factor to the collector current. The strength of the photoelectric effect depends on the strength and the spectrum of the incoming light with respect to the relative sensitivity of the phototransistor. For this purpose, the response of various different PTs are simulated using the recorded spectral data set. This leads to a generation of individual features, one for each PT, which can be used in computationally efficient machine learning techniques, e.g., Random Forest.

**PT Simulation & Feature Generation** In order to simulate the PTs for feature generation, the in the data sheets recorded relative sensitivities are used:

Let  $h_i(\lambda)$  be the function for the spectral sensitivity for a certain PT  $i$  and  $g_j(\lambda)$  the relative spectrum generated through stimulation by the LED  $j$ , where  $\lambda$  is the wavelength. A feature  $f_{ij}$ , which can be used for classification, is then given by

$$f_{ij} = \int_{\lambda} h_i(\hat{\lambda}) \cdot g_j(\hat{\lambda}) d\hat{\lambda}. \quad (3.13)$$

The feature  $f_{ij}$  simulates the analog signal of the PT  $i$  given the relative emission spectrum induced by the LED  $j$ . Since the spectral data consists of distinctive data points, Equation (3.13) turns into the sum

$$f_{ij} = \frac{1}{N} \sum_{k=1}^N h_i(\lambda_k) \cdot g_j(\lambda_k), \quad (3.14)$$

where  $N$  is the total number of data points. In Figure 3.16, two different spectral sensitivities together with the PSI and PSII spectra from Figure 3.4b are shown. The generated feature values are the marked areas beneath the curves.

Table 3.4: Classification results for the grid search over the in Table 3.3 noted hyper-parameters. The table shows the five best and worst results, respectively.

ID	Accuracy	conv.Blocks	conv.Layers	Filters	Kernel Size	dLayer	dLayerNodes
1	0.992	2	1	64	7	1	100
2	0.99	1	3	32	7	1	100
3	0.99	1	3	32	7	2	200
4	0.99	1	5	32	5	1	200
5	0.99	2	1	32	5	1	200
...	...	...	...	...	...	...	...
2876	0.2	1	2	32	3	3	10
2877	0.2	1	2	8	5	1	10
2878	0.2	1	1	64	7	1	10
2879	0.2	1	1	64	5	1	10
2880	0.176	4	4	64	5	3	50

**Classifier** Due to the small number of features, computationally efficient machine learning techniques can be used, which are applicable even on small microprocessors. Thus, a random forest classifier [122] is trained to estimate the right class labels based on the feature values given by the simulated PTs. Therefore, 200 tree estimators are used to form the random forest which are constructed using the *gini-coefficient*. The implementation is done using *scikit-learn* [123].

### 3.6.4 Evaluation

To evaluate the potential of plant classification using active chlorophyll fluorescence sensing, first a deep learning approach utilizing the complete reflectance spectrum and second, a low-cost approach based on consumer electronics are tested. The deep learning approach thereby serves as benchmark for the low-cost approach to better classify the results. While the benchmark approach showed accurate classification results with up to 99.2% classification accuracy, the low-cost approach suffers from the reduced features by simulating the PTs. Still, classification accuracies of around 70% for 5-plant-classification and accuracies of around 90% for a grass-non grass classification can be reached.

**Deep Learning Approach** As a first step, the classification results for different system architectures were investigated by applying a grid search over the in Table 3.3 noted hyper-parameters, hence in total 2,880 different net architectures were analyzed. In Table 3.4, the results for the five best and worst architectures are shown, respectively. The five best architectures provide very high accuracies of at least 99% right classifications. The results indicate that a higher number of neurons in the fully connected layers has a positive impact on the network's performance. However, such a direct relationship was not observed for

Fragrant Violets	0.982 ±0.02	0.002 ±0.006	0.0 ±0.001	0.011 ±0.016	0.005 ±0.009
Dandelion	0.002 ±0.007	0.925 ±0.065	0.005 ±0.012	0.001 ±0.003	0.067 ±0.062
Lawn	0.0 ±0.003	0.003 ±0.007	0.968 ±0.033	0.015 ±0.023	0.015 ±0.018
Sage	0.002 ±0.006	0.001 ±0.003	0.011 ±0.019	0.984 ±0.022	0.002 ±0.006
Dead Nettle	0.015 ±0.019	0.06 ±0.06	0.013 ±0.017	0.002 ±0.006	0.91 ±0.079
	Fragrant Violets	Dandelion	Lawn	Sage	Dead Nettle

Figure 3.17: Mean classification result for the best architecture over 100 Monte-Carlo cross validation iterations. The architecture consisted of two convolutional blocks of one convolutional layer with 32 filters of size 5, followed by one fully connected layer with 200 neurons.

the other hyper-parameters, highlighting their complex interplay and the importance of a vast hyper-parameter search as presented here.

Furthermore, the generalization performances of the five best networks were analyzed by applying a Monte-Carlo cross validation to each of them. The best architecture after the Monte-Carlo cross validation consisted of two convolutional blocks of one convolutional layer with 32 filters of size 5, followed by one fully connected layer with 200 neurons. In Figure 3.17, the confusion matrix over all runs of the cross validation is shown for the best performing architecture, given the mean and the variance over all 100 runs. The approach allows for a very accurate classification of fragrant violets, lawn and sage, whereas dandelion and dead nettle are harder to distinguish. However, even for the harder to distinguish plant species classification accuracies over 90 % were reached. Over all plants, the network achieves an accuracy of 95.38%. The architecture with the lowest accuracy (one convolutional block, five convolutional layers with 32 filters of size 5, one fully connected layer with 200 neurons) achieved a mean accuracy over all plants of 89.32%. In this case, the discrimination of dandelion and dead nettle provided insufficient results and reduced the mean accuracy significantly.

**Low-cost Approach** For the concept of a low-cost plant classification sensor system based on simple consumer LEDs and PTs, different combinations of LEDs for the chlorophyll fluorescence excitation and different combinations of PTs for the chlorophyll fluorescence detection were tested. Therefore, the recorded spectral data were used to generate classification features given the spectral sensitivity of 11 different consumer PTs. In total,  $7\text{LEDs} \times 11\text{PTs} = 77$  features were generated using Equation (3.14) for each PT/LED combination. The detailed spectral data for the PTs and LEDs can be found in the appendix, Section E. In order to determine the accuracy of a plant classification algorithm

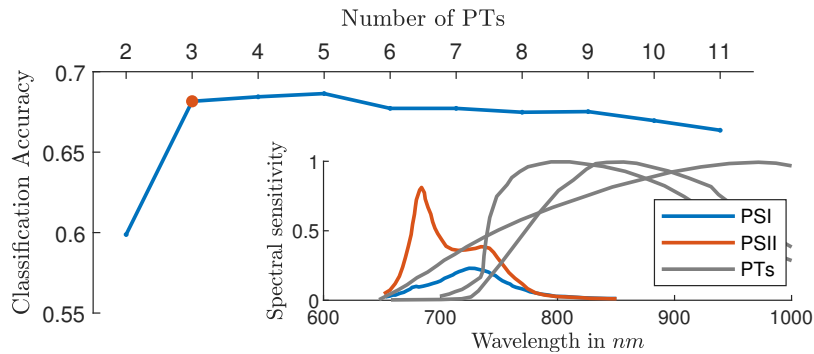


Figure 3.18: Classification accuracies with respect to the number of used PTs. Always the classification accuracy for the best combination of PTs found is shown. For the feature generation, the data from the  $432\text{ nm}$  LED have been used. In the lower right corner, a sketch of the PSI, the PSII spectra as well as the spectral sensitivities for the best combination of three PTs (red dot) are shown. For this combination, the classification accuracy is 68.16 %.

based on those features, different feature combinations were tested. Therefore, for each feature combination a random forest with 200 single decision trees, the *gini-coefficient* for split decisions and 5-fold cross validation were trained.

In order to find suitable combinations of PTs, all combinations starting with two PTs and ending with all available PTs were tested using the spectral data recorded with the  $432\text{ nm}$  LED. In Figure 3.18, the classification accuracies are shown together with the spectral sensitivities for the best triple combination of PTs. The results show, that the classification accuracy increases rapidly from around 60 % using two PTs to around 68 % using three PTs. Afterwards it stagnates and even decreases slightly as more PTs are used.

Using the best PT combinations for 3, 4, 5 PTs, the classification accuracies with respect to the number of used LEDs were investigated, Figure 3.19. As seen before, the classification accuracy using simply one LED already reaches approximately 68 % for each of the PT combinations. Increasing the number of LEDs, thus the number of recorded spectra used for feature extraction, only slightly increases classification accuracy. The best result, 71.52 %, is obtained for 3 PTs and 5 LEDs, where the 3 PTs have the spectral sensitivities as shown in the lower right of Figure 3.18 and the LEDs the peak wavelengths  $\{375, 400, 415, 432, 445\}\text{ nm}$ .

Given the best combination from above, 3 PTs and 5 LEDs, a Monte-Carlo cross validation with a 80 % – 20 % train-test split and  $N = 100$  iterations was evaluated in order to test robustness of the low-cost classification approach. The mean and standard deviation for the classification results presented as a confusion matrix are shown in Figure 3.20. The

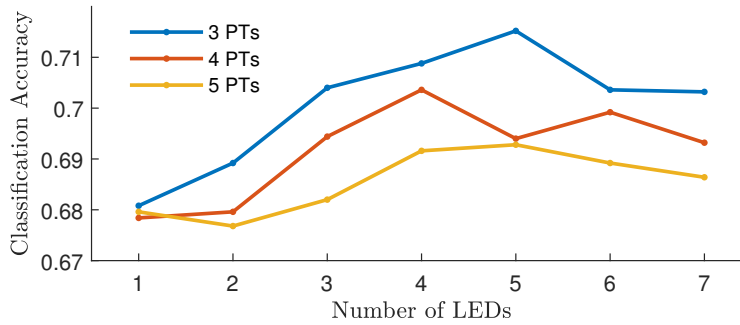


Figure 3.19: Classification accuracies with respect to the number of used LEDs given the best combinations for 3, 4, 5 PTs. The plot shows always the best combination of LEDs chosen from a pool of seven LEDs with wavelengths  $\{375, 385, 400, 415, 432, 445, 460\}$  nm.

Fragrant Violets	0.8 $\pm 0.063$	0.041 $\pm 0.025$	0.003 $\pm 0.006$	0.102 $\pm 0.045$	0.054 $\pm 0.027$
Dandelion	0.077 $\pm 0.034$	0.593 $\pm 0.078$	0.009 $\pm 0.008$	0.042 $\pm 0.023$	0.279 $\pm 0.071$
Lawn	0.006 $\pm 0.007$	0.005 $\pm 0.006$	0.859 $\pm 0.05$	0.094 $\pm 0.034$	0.036 $\pm 0.025$
Sage	0.093 $\pm 0.041$	0.006 $\pm 0.009$	0.116 $\pm 0.04$	0.711 $\pm 0.056$	0.073 $\pm 0.032$
Dead Nettle	0.088 $\pm 0.05$	0.231 $\pm 0.054$	0.031 $\pm 0.02$	0.09 $\pm 0.046$	0.559 $\pm 0.072$
	Fragrant Violets	Dandelion	Lawn	Sage	Dead Nettle

Figure 3.20: Mean classification result and variance for 100 Monte-Carlo cross validation iterations. The three best PTs and five best LEDs have been used.

classifier allows for a good detection of lawn and fragrant violets where it has its problems distinguishing between dandelion and dead nettle. These results confirm what had been seen with the deep learning approach, where also dandelion and dead nettle were proven hard to distinguish. The overall mean classification accuracy for the low cost approach is 70.43 %.

Lastly, a classifier which distinguishes only between grass or non-grass was tested. For the setting with 3 PTs and 5 LEDs a classification accuracy of 90 % was achieved using 5-fold cross validation. Testing a minimal setup given the PT combination from Figure 3.18 and using only the blue LED with 432 nm resulted also into a 90 % classification accuracy. In order to ensure the balance between grass and non-grass samples, the grass samples were copied and added such that in total 4,000 samples for training and testing were used.

## 3.7 Discussion

In the previous sections, Section 3.5 and Section 3.6, methods for the detection and classification of the working environment for autonomous lawn mowers were proposed. In the following, application limitations and implications for the proposed methods are discussed in detail.

### 3.7.1 Working Area Detection

The proposed chlorophyll fluorescence sensor is able to reliably detect grass and thus classify the working area for autonomous lawn mowers efficiently, Subsection 3.5.4. However, there are three main issues which have to be discussed: the limitation due to high sun radiation, the measurement noise of the sensor (Figure 3.12a) and possible fields of application for autonomous lawn mowers and beyond.

**Limitation due to High Sun Radiation** The limitation due to high sun radiation comes from the choice of sensor components and the design itself. Possible workarounds are to shade the grass sensor, add a resistor bridge to discharge a part of the current or choose other electronic components with larger resolvable currents. Shading the sensor might be not always possible, especially if the sensor is placed at the front of the autonomous lawn mower where bumper sensors are also often situated. Adding a resistor bridge to discharge a part of the current leads to better performance under high sun radiation but due to the current discharging, problems for low sun radiation, e.g., measuring at night, appear. During the development of the sensor various electronic components were tested, such as different LEDs and PTs. Keeping in mind the low-cost approach, the proposed design fulfills the requirements for autonomous lawn mowers as well as it is necessary.

**Sensor Measurement Noise** Following the physical principle of chlorophyll fluorescence, an ideal sensor should not detect anything if sensing non-grass (non-chlorophyll) environments. However, there is some measurement noise caused by several error sources, e.g., non-ideal emission and absorption spectra of the LED and the PT or noise of the individual electronic components due to voltage or current fluctuations. These errors were reduced by, among other things, continuously improving the design of the Printed Circuit Board (PCB). As demonstrated in Subsection 3.5.4 the current sensor design is able to reliably detect the lawn and thus the working area for the autonomous lawn mower.

**Fields of Application** As introduced, the chlorophyll sensor was developed as a mowing area detector for autonomous lawn mowers. For this sensor approach to work, the lawn environment must be clearly demarcated, for example by lawn edging stones. Additionally, the lawn should be of reasonable quality to ensure good detection. If these two conditions are fulfilled, the sensor developed here can be used efficiently for area detection. Moreover, the general sensor concept is not restricted to this field of application. Rather, all areas

in which chlorophyll fluorescence plays a role can be considered as possible fields of application, e.g., agriculture or gardening in general. A detailed outlook into future sensor developments is given in Section 3.8.1.

### 3.7.2 Plant Classification

The introduced low-cost approach for plant classification utilizing active chlorophyll fluorescence sensing showed promising results, Subsection 3.6.4. In the following, some insights are discussed in order to point out the limitations of the here performed method analysis.

**Plant Selection** The in Table 3.2 listed plants have been selected as proxies for plants/weeds found in standard garden environments, recognizing that these represent only a small sample of possible garden plants. However, for many gardening applications, e.g. lawn care, a distinction between grass and other plants/weeds might be sufficient. Also, in the future the data base will be enlarged including additional plants/weeds.

**Phototransistor Selection** It may seem strange that only PTs from the near IR-spectrum were tested for the low-cost approach. The reason for this is that most consumer PTs either cover the near-IR spectrum (mostly used in remote controls) or the complete visible spectrum. Since PTs which cover the visible spectrum also register the used LED, they would not give appropriate information for the classification procedure. Hence, only PTs covering the near-IR region are usable for the proposed low-cost approach.

**Influence of Light Conditions** It is important to note that the presented dataset was collected under ideal conditions as only one leaf of one specific plant was present in the sensor's field of view and the measurement was performed in a darkened environment. The transferability of the proposed approach to realistic conditions with ambient light and possibly several leaves from multiple plants in the field of view is an interesting research question and is addressed in Subsection 3.8.1.

## 3.8 Conclusion

In this chapter, an active low-cost chlorophyll sensor has been introduced for efficiently detecting the working area for autonomous lawn mowers. The proposed sensor concept is built up on standard consumer electronics and provides accurate lawn detection results. Such a sensory system can replace the current bounding wire systems, which leads to a reduction in installation and maintenance costs. Furthermore, it allows the use of autonomous lawn mowers on any enclosed lawn area, thus they gain enhanced mobility.

By adapting the sensory system, additional areas of application can also be opened up. One example here is plant stress recognition, which is of high importance in modern agriculture and can be measured by exploiting the “Kautsky-Effect” as demonstrated in [109].

Such plant stress measuring systems were introduced in [124] or [125]. However, these systems require an enclosure around the plant for executing the measurements. To overcome this restriction, the here presented active sensing approach can be effective to filter the sunlight by fast pulsing.

Another example for an additional area of application is plant classification which has been discussed and analyzed in Subsection 3.6. There, two different plant classification approaches based on active chlorophyll fluorescence sensing have been introduced. The first makes use of deep learning techniques to take advantage of the whole chlorophyll fluorescence emission spectrum and serves as benchmark. The latter investigates the low-cost approach based on the proposed sensor design using consumer LEDs and PTs for excitation and absorption of the chlorophyll fluorescence, respectively. For the benchmark, an accuracy of approximately 95 % could be reached classifying five different plant species. This demonstrated the general functionality of plant classification utilizing the chlorophyll fluorescence spectra. The low-cost approach is still capable of reaching classification accuracies around 70 %. For most applications in the domestic area, that might already be enough. Considering that in most cases only a differentiation between one plant type (e.g. lawn) and others is needed, a binary classification between lawn and other has reached an accuracy of approximately 90 %. This shows that the presented method might be the basis for affordable plant classification sensor design utilizing machine learning in a wide range of applications.

### 3.8.1 Detailed Future Work

Towards the development of cost-efficient plant classification sensors, the next step can be to evaluate the concept, which have been so far tested in a laboratory, ideal setting, under real-world conditions. Therefore, the measurement system consisting of a spectrometer and the LEDs can be mounted on to a mobile robot to measure large outdoor areas, e.g., harvest fields or lawns, collecting a vast amount of sample data. This allows for testing the classification approach for transferability to realistic conditions with ambient light and possibly several leaves from multiple plants in the field of view. In a final step, a new low-cost sensory system for plant classification could be designed based on the sensor concept introduced here which utilizes consumer LEDs and PTs.

## 3.9 Acknowledgments

The part of this chapter concerning the chlorophyll sensor design is based on the journal paper [126] and the conference paper [127] which was written by Nils Rottmann (NR), Ralf Bruder (RB), Achim Schweikard (AS) and Elmar Rueckert (ER). Together, NR and RB developed the basic ideas and the general sensor concept. The calculations for the design of the sensor were conducted by NR, who also implemented the algorithms and performed the experiments. The hardware design was made by RB with the support of NR. The paper

was written by NR with highly appreciated assistance from RB, AS and ER. ER greatly improved the paper by contributing important ideas on how to statistically evaluate the sensor performance.

The part of this chapter concerning the plant classification is based on the journal paper [128] written by Nils Rottmann (NR), Jannis Hagenah (JH), Jan-Philip Klose (JK), Ralf Bruder (RB), Floris Ernst (FE) and Elmar Rueckert (ER). The basic idea was developed by NR with valuable input from RB. The dataset was collected by JK with the support of NR. A first implementation of the basic classification idea was done by JK. The algorithms and evaluations were significantly improved by NR and JH, where NR focused on the low-cost approach and JH on the deep learning approach. The paper was written by NR and JH, where JH mainly focused on the deep learning parts. FE and ER gave valuable feedback as well as ideas for efficiently presenting the classification results.

## 4 Robot Control and Monitoring – ROS-Mobile

In this chapter, ROS-Mobile, an Android application for the Robot Operating System, is introduced. This application facilitates the control and monitoring of mobile robots in outdoor environments by allowing the usage of mobile devices, e.g., smartphones or tablets.

### Contents

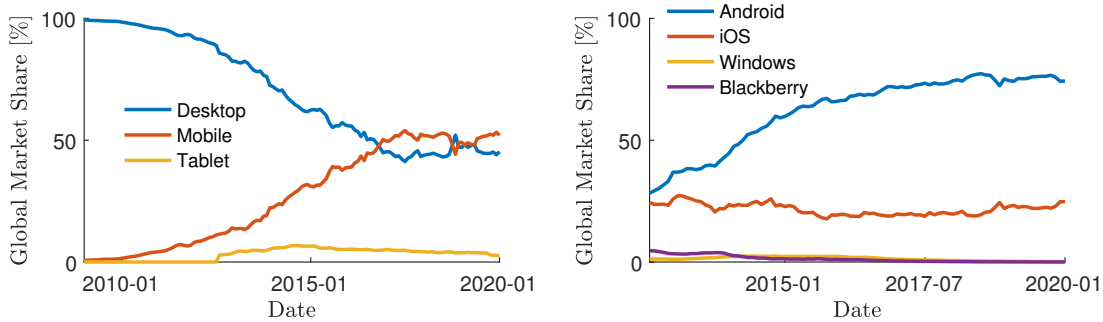
---

<b>4.1</b>	<b>Introduction</b>	<b>77</b>
<b>4.2</b>	<b>Related Work</b>	<b>79</b>
<b>4.3</b>	<b>Scientific Contributions</b>	<b>80</b>
<b>4.4</b>	<b>ROS-Basics</b>	<b>80</b>
4.4.1	ROS Components	81
4.4.2	ROS Code Hierarchy	82
<b>4.5</b>	<b>Mobile Application Design</b>	<b>82</b>
4.5.1	Software Architecture	82
4.5.2	Functionality and Extensibility	83
<b>4.6</b>	<b>Use-Case Scenarios</b>	<b>84</b>
4.6.1	Demo Example	84
4.6.2	Application Scenarios	85
<b>4.7</b>	<b>User Acceptance</b>	<b>86</b>
<b>4.8</b>	<b>Conclusion</b>	<b>87</b>
4.8.1	Detailed Future Work	87
<b>4.9</b>	<b>Acknowledgements</b>	<b>88</b>

---

### 4.1 Introduction

In the previous chapters, intelligent navigation and planning strategies for low-cost robots in closed environments as well as the sensor design and development for those robots have been discussed. Still missing is a discussion of the basic control and monitoring architecture of the robot. For the robot used throughout this thesis, this is realized with the ROS [129]



(a) The global market share for Desktop, Mobile and Tablet devices.

(b) The global market share for different operating system for mobile devices.

Figure 4.1: Global market shares for (a) Desktop, Mobile and Tablet devices and (b) different operating systems. Data retrieved from StatCounter Global Stats, accessed on 02/04/2020. <https://gs.statcounter.com/>

framework, which is a widely used framework at universities and companies for control and navigation of robotic systems. For example, it can be used for controlling an Unmanned Aerial Vehicle (UAV) using Motion Predictive Control (MPC) methods [130] or to map the environment with mobile robots using 2D SLAM techniques [131]. A common tool for controlling and monitoring robotic applications within ROS is ROS Visualization (rviz)<sup>1</sup> which comes with a full-desktop installation but is not available for mobile devices. However, for controlling and monitoring mobile robots in outdoor environments as proposed within this thesis, a tool like rviz for mobile devices would strongly facilitate the interaction with such robotic systems. Considering also the steadily increasing market share of mobile devices and tablets (Figure 4.1a) underlines the need of such a mobile application. To allow for a widest possible range of application scenarios, additional functionalities not available in rviz but provided by other ROS packages, e.g., teleoperational control should also be included. This enables the user to efficiently control and monitor mobile robots in outdoor environments with easy to handle mobile devices, e.g., smartphones

In this chapter, intuitive controlling and monitoring techniques for autonomous systems are discussed (Q3 in Subsection 1.2). As a result, ROS-Mobile is introduced, a ROS application for mobile devices. The chapter begins with an overview of related work, Section 4.2, and a short summary of the scientific contributions, Section 4.3. This is followed by a short introduction to the ROS basics, Section 4.4. In Section 4.5, the software architecture as well as the application functionalities are introduced. Furthermore, the extensibility of the application is highlighted. Afterwards, different use-case scenarios are presented (Section 4.6), including a demo example and different application scenarios. Lastly, the user acceptance is analyzed, Section 4.7, and the chapter is concluded, Section 4.8.

<sup>1</sup><http://wiki.ros.org/rviz>

Table 4.1: Comparison of different ROS applications currently available for Android.

Application	Features
ROS Control	Control & Visualization
AuTURBO	Teleoperation
ROS Image Viewer	Image View
iviz	Visualization & control
<b>ROS-Mobile</b>	Debug, Control & Visualization

## 4.2 Related Work

There have been different attempts and studies concerning remote control and monitoring of robots via mobile devices. For example, in [132] a low cost robot framework, controlled by two different Android applications based on ROSJava<sup>2</sup>, was introduced. In [133], a remote control application for mobile robots built from Lego Mindstorms NXT kit was proposed. However, these applications are limited to specific robotic system (e.g. Lego Mindstorms NXT kit) or a specific robot configuration. Moreover, ROSJava is limited to older Android versions and programming architectures, especially with the nowadays recommended Model View ViewModel (MVVM) architecture, which will be introduced later. For a robotic system to be fully operational, a ROS application for mobile devices should fulfill certain requirements. These requirements are (1) a high customizability and extensibility, (2) the possibility to both monitor and control the robot and (3) smooth integration into the device workflow to avoid interference with other running applications. Currently, there are different ROS applications for Android available for download, Table 4.1, which all suffer from different shortcomings considering the above requirements:

*ROS Control*<sup>3</sup>, developed in 2016, allows for example map view, remote control or laser scan visualization. However, it is based on outdated Android versions and its architecture is not MVVM conform which results in interference with the mobile device workflow. *ROS Teleop Controller AuTURBO*<sup>4</sup> is mainly designed for control and display data supported by the robots from the TurtleBot family<sup>5</sup>. *ROS Image Viewer* only allows to display images received from a ROS framework. A recently published mobile application for ROS is *iviz* [134], which is based on the commercial Unity engine<sup>6</sup> and written in C#. This allows targeting different Operating System (OS), e.g., iOS or Android. Currently, *iviz* allows for visualization of robot models and 3D point clouds and provides a joystick for control inputs. To add additional features, modules have to be implemented and merged into Unity.

<sup>2</sup><https://github.com/rosjava>

<sup>3</sup><https://github.com/mtbii/RobotCA>

<sup>4</sup><https://github.com/AuTURBO/ros-app-tb3-voiceorder>

<sup>5</sup><https://www.turtlebot.com>

<sup>6</sup><https://unity.com/>

Table 4.2: Software Information (Status 05/05/2021)

Current code version	1.2
Permanent link to repository	<a href="https://github.com/ROS-Mobile/ROS-Mobile-Android">https://github.com/ROS-Mobile/ROS-Mobile-Android</a>
Legal Code License	The MIT License (MIT)
Code versioning system	git
Software languages	Java, XML, Gradle
Compilation Requirements	Linux or Windows with Android Studio 3.6.1 or higher
Developer Documentation	<a href="https://github.com/ROS-Mobile/ROS-Mobile-Android/wiki">https://github.com/ROS-Mobile/ROS-Mobile-Android/wiki</a>
User and developer support	rosmobile.info@gmail.com

### 4.3 Scientific Contributions

In this chapter, a highly customizable ROS application for Android operated devices for remote control and monitoring of mobile robotic systems is introduced. The application is based on the Android OS as it is the most widely used OS for mobile devices, Figure 4.1b. Further, it is based on the MVVM architecture which comes with simple testing structures and extensibility. In order to use the application, only a mobile device, e.g., a smartphone or a tablet, with Android Version 5.0 (Lollipop) or higher is required. A wireless network connection between the ROS master and the mobile device has to be established. The current version details and requirements of the application are summarized in Table 4.2. In less than a year, the application is already wide spread within the ROS community as discussed in Section 4.7 based on statistics collected with the Google Play Console. This further underlines the need of simplified robot control and visualization with mobile devices.

### 4.4 ROS-Basics

The Robot Operating System (ROS) is a peer-to-peer middleware package for robotic systems. It is currently maintained and managed by the non-profit organisation Open Source Robotics Foundation (OSRF). The basic idea behind ROS is to allow for easy hardware abstraction and extensive code reuse within robotic projects. Therefore, all major functionalities are broken up into pieces, called *nodes*, which communicate with each other using internal ROS *messages*. For example, one node might be responsible for collecting camera data and provides (publishes) the data to the ROS system whereas another node might be responsible for map generation and requests (subscribes) those camera data. This publish-subscribe messaging pattern allows each node to subscribe or publish to certain message classes, called *topics*, without knowledge of which other nodes, if any, there are. To generate a peer-to-peer network, the nodes are matched by the ROS *master* based on the chosen published/subscribed topics. An example of a typical ROS network is shown in Figure 4.2. Here, a sensor node provides sensory information which

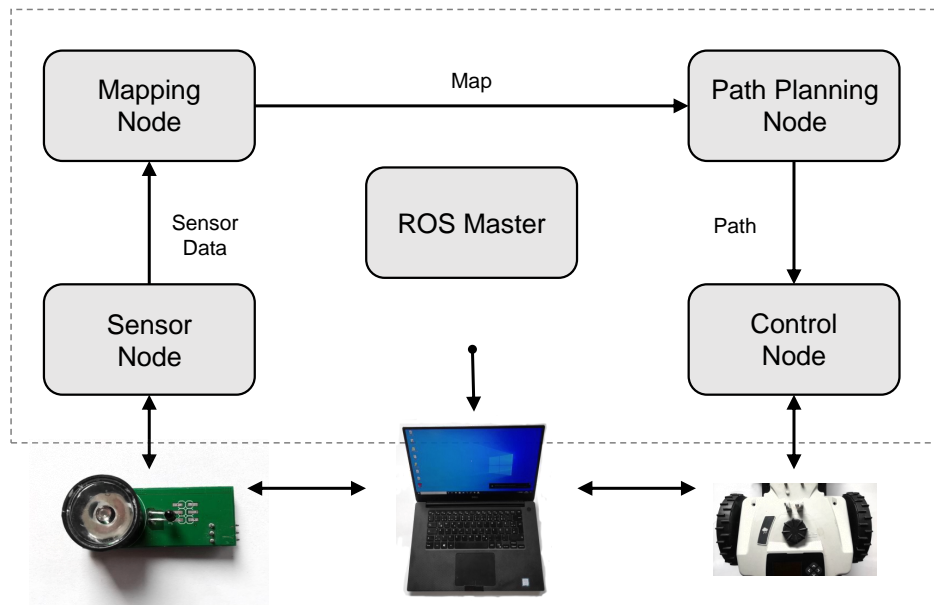


Figure 4.2: ROS system with a sensor read out by a sensor node, a mapping node which used the sensor information for map generation, a planning node for task planning based on given map information and a control node for controlling the actuators of the robot.

is processed by a mapping node which provides a map of the environment. Based on the mapping information, a planning node publishes path data which are subscribed by a control node which controls the actuators of the robot. In the following, the main ROS components are described in more details as well as an outline of the principal ROS code hierarchy is given. For a more detailed introduction to ROS, it is referred to [135].

#### 4.4.1 ROS Components

The main components a ROS system is built on are *nodes*, *topics/services* and *messages*. In the following, these components are introduced in more detail.

**ROS Nodes** A *node* is an executable of the system, a process that does some computation, e.g., a Kalman Filter. Typically, all functionalities of the robotic system are divided into different modules where each module consists of one or more *nodes*. The communication between nodes are handled over *topics* or *services*. In general, a typical robotic system operated with ROS includes many nodes interacting with each other. A system partitioned into several nodes provides several benefits such as a reduction of code complexity or code reusability.

**ROS Topics** A *topic* is a kind of a named "pipeline" over which different *nodes* communicate with each other. It allows the anonymous publish/subscribe messaging pattern by enabling *nodes* to subscribe or publish to a given *topic*. Thereby, each *topic* can have multiple publisher or subscriber. The content transported via a *topic* is defined by a *message* type and is meant to be unidirectional. *Nodes* which require bi-directional communication should use *services*.

**ROS Services** *Services* are very similar to *topics* but are designed for non-unidirectional communication, especially request/reply interactions. Therefore, a *service* requires in comparison to a *node* two *message* types, one for the request and one for the reply.

**ROS Messages** A *message* is a data structure comprising typed fields. For example, a point message consists of three floating point variables  $x$ ,  $y$  and  $z$ . In general, most *messages* also contain a header which provides important information such as a time stamp or a reference frame. Moreover, *messages* can be combined to build new, more complex message structures.

#### 4.4.2 ROS Code Hierarchy

The ROS code hierarchy is built on the idea that complex projects are broken down into smaller and smaller components, such that single developers can handle the individual components. The top hierarchy level is the *repository* which contains the code from a particular development group. One level below, the *stacks* are to be arranged, which each handle a particular subject of the project. Below this, *packages* are defined as separate modules of the subject to be handled. In general, such a *package* consists of several *nodes*, the smallest components in the code hierarchy.

### 4.5 Mobile Application Design

In the following, the underlying software architecture of the ROS application is introduced. Due to this software architecture, the application allows for a wide range of functionalities, integrates smoothly into the mobile device workflow and is easily extensible, which is described in the second part of this section.

#### 4.5.1 Software Architecture

The basic architecture that the introduced ROS application is based on is called *MVVM*, Figure 4.3. The underlying principle is the *Separation of Concerns* [136]. This means that each *concern* should be addressed by a different *section* of the program. For example, a User Interface (UI) is required to enable user interaction and ROS nodes are required to subscribe or publish to certain ROS topics. The information produced by the user via the UI and the data received or sent via the ROS nodes are thereby the *concerns* of the program which have to be handled separately. Moreover, Android applications in

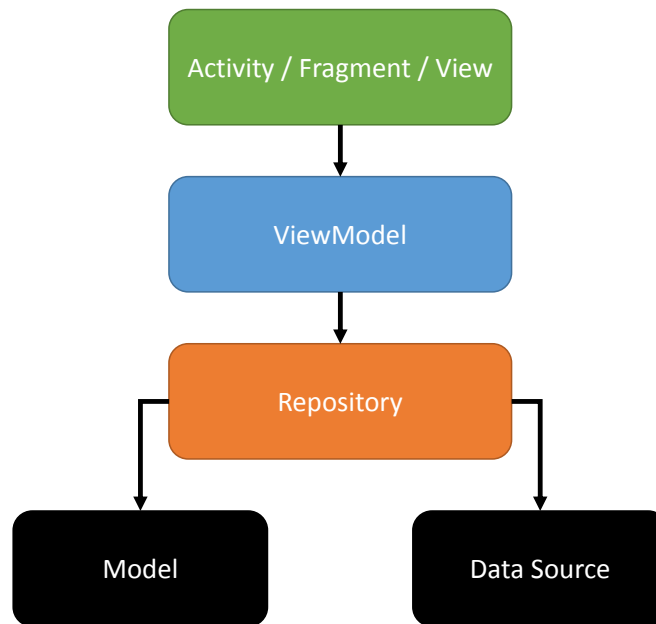


Figure 4.3: The general MVVM designing concept for mobile applications.

comparison to desktop applications in general do not have a single entry point; instead, they consist of multiple components, such as Activities or Fragments. These application components are then used by the Android OS for integration into the device workflow. Hence, the application has to be able to handle app hopping behavior correctly. To do so, the *ViewModel* class, provided by the Android OS designed for managing UI related data, is used as a connection between the model/data storage system and the UI. The workflow is as follows: Required data is provided by a model/data storage system and is requested by the *ViewModel* object. This *ViewModel* object is directly assigned to a UI Controller (e.g. Activity, Fragment), which requests the data from the *ViewModel* object and let the user interact with them. Thus, the UI controller is decoupled from the data management system and can be destroyed and rebuilt during runtime by the OS to enhance performance. In order to implement such a type of architecture, the ROSJava code, which was designed for older Android versions and architectures, had to be modified according to the requirements of the MVVM architecture.

#### 4.5.2 Functionality and Extensibility

The introduced ROS Android application enables users to easily control and monitor ROS operated systems with mobile devices. Therefore, the application establishes a ROS connection via WiFi with the system. This allows to control the robot by publishing ROS

messages (e.g. velocity commands), visualize published measurement results (e.g. map data) or analyze the system's behavior. Moreover, an additional control possibility is given by the included SSH client, which allows direct access to the operating system. Currently, the following modules are implemented:

- Debugger: A module which can display any ROS message, similar to *rostopic echo*
- Joystick: A module to teleoperate a robot
- GridMap: A module which displays grid maps
- Camera: A module which displays camera data
- GPS: A module which displays the current position of the robot in an *openstreetmap*<sup>7</sup> based on GPS data

During the development of ROS-Mobile, a lot of effort was put into making the application easily extensible. This allows members of the ROS community to adjust and extend the application to their own needs by redefining or adding new modules. Therefore, a step-by-step guide on how to create new modules<sup>8</sup> was written. In addition, easy extensibility ensures easier maintenance of the software and thus better support.

## 4.6 Use-Case Scenarios

ROS-Mobile was developed within this thesis to provide an easy way to monitor and control the autonomous lawn mower in outdoor environments. However, since easy expandability and customizability were the main focus of the development, the application can be used for various robotic controlling and monitoring scenarios. Here, a demo example showing the mapping procedure of an apartment environment is used to demonstrate the functionality of the application. Further, different use-case scenarios for ROS-Mobile currently handled within the author's institutional affiliation are introduced.

### 4.6.1 Demo Example

The aim of the demo example task is to build a map of an apartment environment using a differential drive robot. In a first step, the robot and the ROS system are started and the Android powered mobile device is connected via WiFi to the robot. The correct IP address is defined in the *MASTER* configuration tab, Figure 4.4b, and the IP address for the mobile device which is related to the robot's WiFi is chosen from the drop down menu. After hitting the connect button, it is now possible to communicate between the ROS system and the mobile device. For an efficient communication, desired widgets/nodes have

---

<sup>7</sup><https://www.openstreetmap.de/>

<sup>8</sup><https://github.com/ROS-Mobile/ROS-Mobile-Android/wiki>

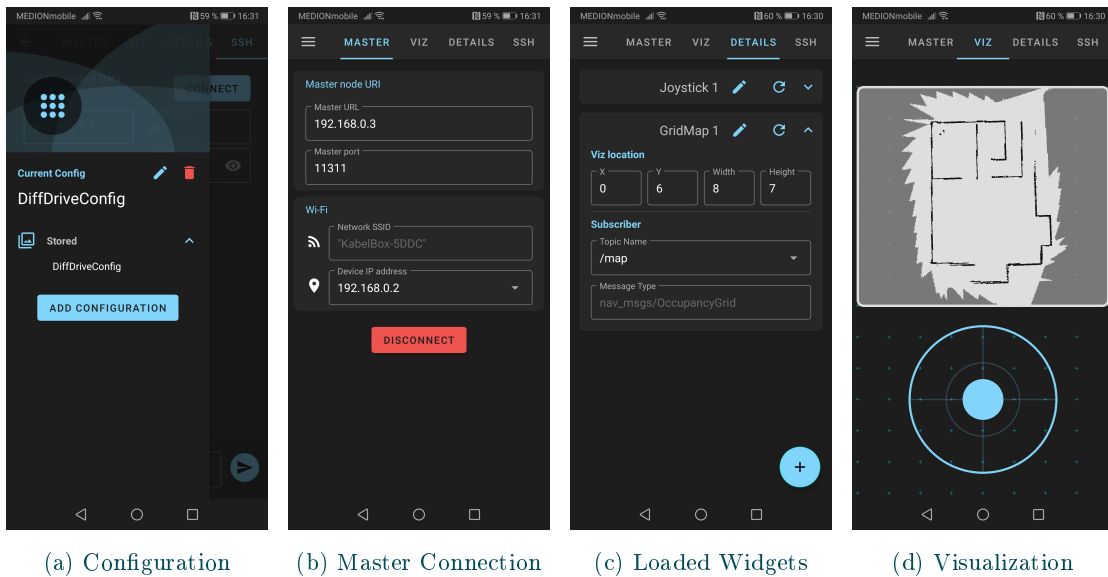


Figure 4.4: Illustrative example for the mapping of an apartment environment and displaying the mapping result with ROS-Mobile. From left to right: (a) Stored configuration files which can be loaded and/or saved for different robotic systems, (b) master tab for connecting to the ROS master which requires the robot’s IP address, (c) details tab where required widgets/nodes are defined and initialized, (d) visualization tab for either control or monitoring the robotic system.

to be chosen and configured in the *DETAILS* tab, Figure 4.4c. For this example, a joystick widget for steering the robot sending *geometry\_msgs/Twist* messages to the ROS system and a grid map widget subscribing to *nav\_msgs/OccupancyGrid* messages for visualization of a recorded occupancy grid map are chosen. The widgets can be adapted by defining the size and location in the *VIZ* tab, Figure 4.4d. In this tab, the recorded occupancy grid map is displayed as well as the joystick. This visualization can be used now to control and monitor the robot. For deeper insights, it is referred to the short use-case video<sup>9</sup> or to the ROS-Mobile introduction video<sup>10</sup>.

#### 4.6.2 Application Scenarios

In addition to the use-case of an autonomous lawn mower, as mainly considered in this thesis, different other application scenarios occur within the author’s institute. For example, controlling and monitoring of different mobile robotic platforms, such as rescue robots, service robots or agriculture robots, Figure 4.5. Rescue robots are remotely controlled and their data evaluated to monitor water pollution and/or flooding. Service robots

<sup>9</sup><https://youtu.be/p81xuGb6xNQ>

<sup>10</sup><https://youtu.be/T0HrEc0-0x0>



Figure 4.5: Mobile robotic platforms for which ROS-Mobile is currently used within the author’s institute. From left to right: (a) intelligent lawn mower with low-cost sensors for domestic purpose, (b) autonomous surface vehicle (ASV) for swarm approaches to monitor water contamination, (c) Loomo from Segway-Ninebot for person guidance in hospitals.

are tested to guide persons through hospitals and deliver files and/or medicine. Agriculture robots are developed to enable autonomous farming for small, local companies. Finally, for the autonomous lawn mower used throughout this thesis, ROS-Mobile is used to remotely control the robot and monitor mapping and path planning results.

## 4.7 User Acceptance

In order to analyze how well ROS-Mobile has been accepted by the ROS community, the user data provided by the Google Play Console (Status: 27. April 2021) are taken into account. In Figure 4.6, the accumulated number of new users and the active users with currently installed application since the application release (22. May 2020) are shown. The ROS-Mobile application shows a steadily increasing growth rate with currently 2734 acquired users and 949 active users. That reflects a high interest and an urgent need for a manageable mobile application for ROS. Moreover, the open source repository of the project has attracted multiple users to adjust the application to their needs (38 forks) and contribute to the master, either by debugging the application and/or by writing new features. Various users also asked for considering new features for implementation which are currently not provided, for example point cloud visualization. Overall, this shows a huge interest and a need to further maintain and develop ROS-Mobile. The most common features the ROS community asks for are: ROS2 integration and 3D visualization. These modules are currently being incorporated into the application, such that they will be provided in the near future.

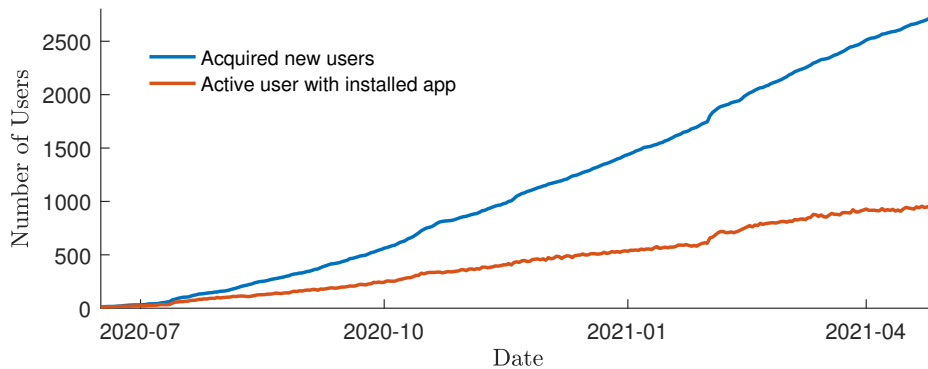


Figure 4.6: The accumulated number of new users since the app release (22. May 2020) and the active users with currently installed application.

## 4.8 Conclusion

In this chapter, the development of ROS-Mobile, a mobile application for Android powered devices, has been introduced and evaluated. The main goals for ROS-Mobile are simple design and easy extensibility/customizability to allow a wide range of users to interact with and contribute to the application. Moreover, the application should be seamlessly integrated into the workflow of mobile devices and provide debug, monitoring and controlling features. In comparison to other available applications, this was reached by using a MVVM architecture and providing basic features in combination with a detailed description how to add new modules. However, there are open tasks like ROS2 integration and/or improved user interaction feedback which should be included in the near future based on user reviews.

Despite these open tasks, the ROS community accepted the application well which is reflected in the rapidly increasing number of users and also in the huge interest of contributing to the development of the application. Thus, ROS-Mobile is a great contribution for the robotic community and has the potential to steadily increase its usability by including progressively more functionalities.

Within this thesis, ROS-Mobile fulfills its purpose by enabling simple control and monitoring of the autonomous lawn mower in outdoor environments. For example, by testing the introduced mapping and path planning approaches in realistic environments or by recording measurement data for the sensor evaluation.

### 4.8.1 Detailed Future Work

As already mentioned above, there are still open improvements for ROS-Mobile which should be included to the application in the near future. In detail to mention are:

- (1) ROS2 integration to allow the usage of the application with ROS2 operated systems,

(2) 2D/3D map view with multiple layers to allow more efficient planning and monitoring and (3) cloud configuration storage to allow the sharing of predefined configurations. Additionally to these main points, there are a lot of suggestions from the community to further improve ROS-Mobile such as being able to display the robots battery level, improve user experience and many more.

## 4.9 Acknowledgements

This chapter is based on the journal paper [137] which has been submitted to SoftwareX and is currently under review. An updated version including the user acceptance evaluation has been submitted as conference paper to IROS 2021 [138]. Both, the journal as well as the conference paper were written by Nils Rottmann (NR), Nico Studt (NS), Floris Ernst (FE) and Elmar Rueckert (ER). NR developed the basic idea and, together with NS, the general application concept and architecture. NR and NS together wrote the software and still maintain the application. Use-case examples were performed with assistance from several other people. Special thanks here to Ralf Bruder, Lars Schilling and Robin Denz. The paper was written by NR with highly appreciated assistance from NS, FE and ER. FE and ER further greatly improved the application design and the application evaluation concepts.

## 5 Conclusion

In this chapter, a summary over all topics addressed throughout this thesis is given. Afterwards, the research questions introduced in Subsection 1.2 are concluded and future work is proposed.

### Contents

---

<b>5.1 Summary</b> . . . . .	<b>89</b>
<b>5.2 Conclusion on the Guiding Research Questions</b> . . . . .	<b>90</b>
<b>5.3 Future Work</b> . . . . .	<b>92</b>

---

### 5.1 Summary

In this thesis, smart sensors and intelligent navigation and planning strategies for low-cost lawn care systems were developed. It was shown that efficient work management is possible for such low-cost autonomous robots with only limited sensing capabilities, e.g., in-/outside area detection. Additionally, a ROS application for mobile devices was developed to control and monitor ROS operated autonomous systems in outdoor environments.

In detail, Chapter 2 discussed optimal planning and navigation strategies for low-cost robots with limited sensing capabilities. Here, the focus was on differential drive robots with only in-/outside area detectors as available with common lawn mowers. In a first step, a mapping strategy for closed environments was developed which allows the generation of a map estimate of the environment. Such a map of the environment is essential for efficient planning and navigation which allows then in a second step the development of an intelligent path planning strategy for covering the complete workspace. There, a neural network approach in combination with a probabilistic coverage map was introduced to cope with the high pose uncertainties during path execution. The proposed approach allows for an efficient complete coverage of the workspace. However, the proposed methods require the adjustment of certain hyper-parameters which can be crucial for efficient method execution. Therefore, an optimization procedure for exemplary learning the hyper-parameters for the proposed mapping procedure was introduced. The research results were presented or are submitted to be presented at the European Conference on Mobile Robotics (ECMR) 2019/2021 and at the 12th IROS 2020 Workshop on Planning, Perception and Navigation for Intelligent Vehicles.

In Chapter 2, for the proposed navigation and planning methods, existing sensor technologies for in-/outside area detection, e.g., perimeter wire systems, were assumed as given. In Chapter 3, a novel low-cost sensor was developed which exploits the characteristics of the working environment and replaces the commonly used in-/outside area detectors. For this purpose, an active optical sensor was developed which stimulates the chlorophyll fluorescence of the grass and captures the response. In this way, the working environment can be swept without the need for additional instruments, e.g., bounding wires. Additionally, plant classification was investigated based on the same idea stimulating the chlorophyll fluorescence and capturing the response. Such a classification is able to expand the range of application for lawn mowers, e.g., by enabling weed detection. The research results were presented at the international conference IEEE Sensors 2020 and are published or under review for publication in the IEEE Sensors journal.

In Chapter 4, the problem of controlling and monitoring a mobile robot in outdoor environments was addressed. Such a control/monitor panel must be easy to handle and intuitive to use. Thus, mobile devices, e.g., smartphones or tablets, are the natural choice here. A ROS application for Android powered devices was developed which enables the user to intuitively control and monitor mobile robots. Open source code was published on GitHub<sup>1</sup> and the application was made available via the Google Play Store. Additionally, the research results are submitted to the International Conference on Intelligent Robots and Systems (IROS) 2021 and to the SoftwareX journal.

## 5.2 Conclusion on the Guiding Research Questions

In Section 1.2, the research questions were defined which led through this thesis. In the following, the contributions and limitations of the methods developed to answer these research questions are discussed.

### **Q1.a: Is it possible to develop planning and navigation strategies for autonomous systems with only limited sensing capabilities?**

In general yes, it is possible to develop planning and navigation strategies for autonomous systems with only limited sensing capabilities, e.g., in-/outside area detectors, as shown in Chapter 2. However, the here proposed methods for map estimation and complete coverage path planning require both the adjustment of method specific hyper-parameters which crucially effect the performs of the methods. Especially for the mapping procedure badly chosen hyper-parameter might result in bad or failed map estimates. Moreover, the accuracy of the generated map estimates as well as the performance of the proposed path planning method drops with increasing odometry error. Nevertheless, the proposed methods enable autonomous robots with only in-/outside area detectors to avoid the commonly

---

<sup>1</sup><https://github.com/ROS-Mobile/ROS-Mobile-Android>

used random walk strategy. Thereby, the methods become better especially with increasing map complexity. For the here investigated autonomous lawn mower, improvements of approximately 50 % for the traveled distances compared to a random walk strategy could be shown.

### **Q1.b: Can reinforcement learning be used for hyper-parameter learning to enable true autonomy for lawn mower?**

This question arises while adapting the hyper-parameters for the proposed mapping procedure. The hyper-parameters are sensitive to the problem and thus have to be adjusted accordingly. Therefore, cost functions dependent on the hyper-parameters were proposed which can be optimized in order to find sufficient accurate hyper-parameters. In general, the proposed optimization method led to sufficient accurate hyper-parameters which resulted into accurate map estimates. However, in some cases the hyper-parameters found by the optimization procedure did not lead to accurate map estimates. This can be attributed to the fact that the cost functions are highly non-linear and the optimization procedure can get stuck into local minima. Nevertheless, learning the hyper-parameters is the first step towards true autonomy for lawn mowers, even though the optimization process does not provide the optimal solution in every case. Also, since the cost functions use the fact that the boundary of an enclosed environment has a constant circumference, they can potentially be adopted for learning the hyper-parameters of other mapping or SLAM algorithms. This holds especially for robots operating in indoor environments.

### **Q2.a: How to efficiently exploit characteristics of the working environment for navigation and planning for autonomous lawn mowers?**

For autonomous lawn mowers, the working environment consists of grass surrounded by an enclosing line, e.g., lawn edging stones. Here, a sensor was developed that exploits the chlorophyll fluorescence properties of the grass and is thus able to replace the conventional boundary wire sensors. Due to the low-cost design of the proposed chlorophyll sensor, this leads to a reduction in acquisition and operating costs. Moreover, with the in Chapter 2 mentioned methods, autonomous lawn mowers are able to efficiently navigate and operate within their working environment utilizing only the proposed chlorophyll sensor for in-/outside area detection. Limitations of the sensor are its light saturation, which in the current design is at approximately 12000 lx, and that the sensor can not distinguish between grass and other plants.

### **Q2.b: Can machine learning approaches be used to learn to distinguish different plant species based on their chlorophyll fluorescence response?**

In general yes, machine learning can be used to distinguish between different plant species only based on their chlorophyll fluorescence emission. However, in order to achieve accuracies over 90 % for a five plant classification, deep learning techniques were required. A

low-cost approach instead, utilizing simulated phototransistor data based on the recorded full spectra, only led to accuracies of around 70 % but also only required computationally, less demanding machine learning techniques like random forests. Considering only a binary classification, e.g., between grass and non-grass, even the low-cost classification approach was able to reach accuracies of about 90 %. This demonstrates the general applicability for the approach in the gardening area.

### **Q3: What are intuitive controlling and monitoring techniques for autonomous robots in outdoor environments?**

First, as today's most intuitive systems to operate are mobile devices, e.g., smartphones or tablets, autonomous systems should be controlled/monitored with such devices. Second, ROS is widely used in many robotic applications and provides a large number of control, planning and monitoring tools. Therefore, a mobile application (ROS-Mobile) was developed which allows to control and monitor ROS operated autonomous systems with mobile devices. This application (Release: 22. May 2020) is already widely spread within the robotics community with currently 2734 acquired and 949 active users (Status: 27. April 2021) which illustrates the need for such an application.

## **5.3 Future Work**

Promising future work could be the development of a novel smart mobile robot for lawn care equipped with multiple of the developed chlorophyll fluorescence sensors. Further, the robot should be tested using the developed mapping and planning algorithms in challenging complex garden environments. Finally, the plant classification approach can be further developed and a first plant classification sensor prototype could be built and tested on the lawn mower. With the additional information available through the plant classification, the lawn mower can then be situated within a smart garden setup, where it communicates with other garden robots, e.g., weed removing robots.

For more details on these research directions, see the individual subsections to future works in the three research chapters (i.e., Subsections 2.8.1, 3.8.1 and 4.8.1).

## Appendix

In this chapter, further implementations details are presented to enable the reader to reproduce the results of this thesis. Implementations in Matlab and C++ can be found in open access GitHub repositories on the author's GitHub page<sup>2</sup>.

### Contents

---

<b>A</b>	<b>Wall Following Algorithm/Controller</b>	<b>93</b>
<b>B</b>	<b>Bayesian Optimization</b>	<b>95</b>
<b>C</b>	<b>Optimal traveled Distance</b>	<b>96</b>
<b>D</b>	<b>Sensor Circuit Diagrams</b>	<b>97</b>
<b>E</b>	<b>PT &amp; LED Spectra</b>	<b>100</b>

---

## A Wall Following Algorithm/Controller

For mapping and relocalization with only in-/outside area detection sensors, a wall following method is required since only at the boundary valid sensor data for those tasks can be generated. Therefore, a control algorithm is proposed which steers the robot along the boundary in wiggly lines. This steering pattern increases the scanning area and allows to navigate the robot along the boundary line even if the binary measurements are corrupted by noise. The approach proposed here is based on [139].

Let  $s$  be the sensor measurement with  $s = 1$  being an in-field and  $s = 0$  being an outside-field detection, then a running average  $\mu_d$  can be calculated using exponential smoothing

$$\mu_d \leftarrow a_\mu \mu_d + (1 - a_\mu) s, \quad (\text{A.1})$$

where  $a_\mu$  defines the update rate of the mean  $\mu_d$ . The initial value for the running average can for example be set to  $\mu_d = 1.0$  under the assumption that the robot starts within the field. By following the boundary line, the desired running average should be  $\mu_d = 0.5$ , whereas the difference between the actual running average and the desired one is given as

$$d = 2 \left( \frac{1}{2} - \mu_d \right). \quad (\text{A.2})$$

---

<sup>2</sup><https://github.com/NRottmann>

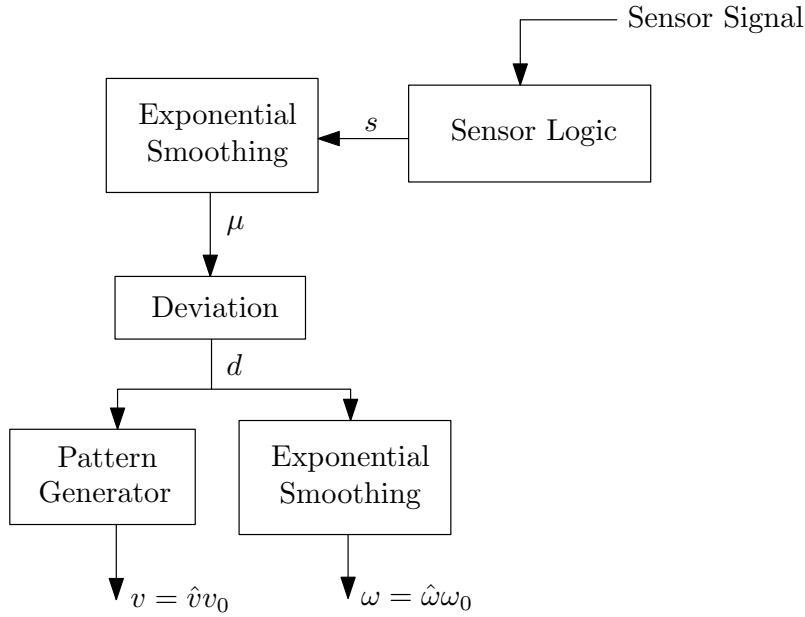


Figure A.1: Wall Follower: Control method blocks representing the wall following algorithm.

The difference is scaled onto the range  $[-1, 1]$  for further use in the control algorithm. Based on the difference  $d$  between desired and actual average measurement value, the relative linear velocity  $\hat{v}$  can be updated as

$$\hat{v} \leftarrow a_v \hat{v} + (1 - a_v)(1 - |d|). \quad (\text{A.3})$$

Here,  $a_v$  defines an update rate and  $|d|$  the absolute value of  $d$ . The relative velocity increases the longer and better the desired mean of  $\mu_d = 0.5$  is reached. Thereby, the relative velocity is always in the range of  $\hat{v} \in [0, 1]$ . Additionally, the relative angular velocity can be determined using the deviation  $d$  and a stabilizing term  $\cos\left(2\pi\frac{c}{M}\right)$  to ensure robustness against noise corrupted measurements

$$\hat{\omega} = \frac{1}{2} \left( d + \cos\left(2\pi\frac{k}{K}\right) \right). \quad (\text{A.4})$$

Here,  $k$  is a counter variable and  $K$  the counter divider. Thus,  $K$  should be chosen accordingly to the frequency of the operating system. Finally, the relative velocities can be transformed to the desired velocities by multiplication with the maximum allowed velocities for the given robotic system,  $v_0$  and  $\omega_0$ ,

$$v = \hat{v}v_0, \quad \omega = \hat{\omega}\omega_0. \quad (\text{A.5})$$

A diagram of the controller can be found in Figure A.1.

## B Bayesian Optimization

Bayesian Optimization (BO) is used as global optimizer for the cost functions introduced in Subsection 2.5.3. For optimizing these cost functions, any global optimizer could be used but BO is efficient if only few hyper-parameters have to be optimized, as it is the case for the proposed cost-functions. Hence, in the following BO is introduced based on [140]. For further details it is referred to [141].

The main concept of BO is to build a model for a given system based on the so far observed data  $D = \{X, \mathbf{y}\}$ . The model describes a transformation from a given data point  $\mathbf{x} \in X$  to a scalar value  $y \in \mathbf{y}$ , e.g., from the parameter vector  $\boldsymbol{\theta}$  to the return  $J(\boldsymbol{\theta})$ . Such model can either be parametrized or non-parametrized and is used for choosing the next query point by evaluating an acquisition function  $\alpha(D)$ . Here, the non-parametric Gaussian Process (GP) is used for modeling the unknown system which is a state-of-the-art model learning or regression approach [142, 143] that was successfully used for learning inverse dynamics models in robotic applications [144, 145]. For comprehensive discussions it is referred to [143, 146].

GPs represent a distribution over a partial observed system in the form of

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}_* \end{bmatrix} \sim N \left( \begin{bmatrix} \mathbf{m}(X) \\ \mathbf{m}(X_*) \end{bmatrix}, \begin{bmatrix} \mathbf{K}(X, X) & \mathbf{K}(X, X_*) \\ \mathbf{K}(X_*, X) & \mathbf{K}(X_*, X_*) \end{bmatrix} \right), \quad (\text{B.1})$$

where  $D = \{X, \mathbf{y}\}$  are the so far observed data points and  $D_* = \{X_*, \mathbf{y}_*\}$  the query points. This representation is fully defined by the mean  $\mathbf{m}$  and the covariance  $\mathbf{K}$ . The mean function is here defined as  $m(\mathbf{x}) = 0$  and for the covariance function a *Matérn kernel*

$$k(d) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left( \sqrt{2\nu} \frac{d}{\rho} \right)^\nu K_\nu \left( \sqrt{2\nu} \frac{d}{\rho} \right)$$

is used with  $d = \|\mathbf{x} - \mathbf{x}_*\|$ . Here,  $\Gamma$  is the gamma function,  $K_\nu$  the modified Bessel function and  $\rho$  and  $\nu$  positive parameters. The *Matérn kernel* is a generalization of the *squared-exponential kernel* where the additional parameter  $\nu$  controls the smoothness of the resulting function. The smoothing parameter can be beneficial for learning local models. The hyper-parameters can be optimized by maximizing the marginal likelihood [143].

Predictions for a query points  $D_* = \{\mathbf{x}_*, y_*\}$  can then be determined as

$$\begin{aligned} \mathbb{E}(y_* | \mathbf{y}, X, \mathbf{x}_*) &= \mu(\mathbf{x}_*) = \mathbf{m}_* + \mathbf{K}_*^\top \mathbf{K}^{-1} (\mathbf{y} - \mathbf{m}) \\ \text{var}(y_* | \mathbf{y}, X, \mathbf{x}_*) &= \sigma(\mathbf{x}_*) = \mathbf{K}_{**} - \mathbf{K}_*^\top \mathbf{K}^{-1} \mathbf{K}_*. \end{aligned} \quad (\text{B.2})$$

The predictions are then used for choosing the next model evaluation point  $\mathbf{x}_n$  based on the acquisition function  $\alpha(\mathbf{x}; D)$ . Therefore, EI [63] is used which considers the amount of

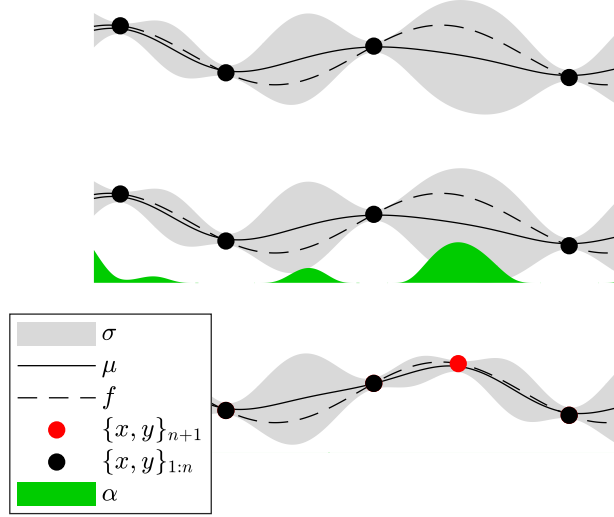


Figure B.1: Illustration of the Bayesian Optimization Process. Based on the measurements  $D = \{x_{1:n}, y_{1:n}\}$  the mean  $\mu$  and variance  $\sigma$  are estimated for sampled position in a defined area. The acquisition function  $\alpha(x; D)$  is calculated as in Equation (B.3) and the position of the maximum value  $\max_x \alpha(x; D)$  is chosen as next evaluation point  $x_{n+1}$ . For comparison the true function  $f$  is depicted as dashed line.

improvement

$$\begin{aligned} \alpha(\mathbf{x}; D) &= (\mu(\mathbf{x}) - \tau) \Phi \left( \frac{\mu(\mathbf{x}) - \tau}{\sigma(\mathbf{x})} \right) \\ &+ \sigma(\mathbf{x}) \phi \left( \frac{\mu(\mathbf{x}) - \tau}{\sigma(\mathbf{x})} \right), \end{aligned} \tag{B.3}$$

where  $\tau$  is the so far best measured value  $\max(\mathbf{y})$ ,  $\Phi$  the standard normal cumulative distribution function and  $\phi$  the standard normal probability density function. Samples, distributed over the area of interest, are evaluated and the best point is chosen for evaluation based on the acquisition function values. In Figure B.1 the BO process is illustrated.

## C Optimal traveled Distance

The optimal traveled distance, assuming the centers of each grid cell should be traversed, can be determined as the number of cells visited multiplied by the length between two grid cell centers, thus the reciprocal of the resolution of the grid. Given a grid map with an area to cover  $A$  and a resolution  $r$ , the total number of cells to cover is

$$\text{cells} = Ar^2 \tag{C.1}$$

and the distance required to traverse over all cell centers, thus to reach 100% coverage, is then

$$T_{opt} = \frac{1}{r} \text{cells} = Ar. \tag{C.2}$$

For this calculation, it is assumed that it is possible to find a path along all cell centers without traversing diagonal or revisiting cells. This assumption might not hold always and thus the calculated distance reflects a lower bound for the optimal traveled distance along all cell centers.

### D Sensor Circuit Diagrams

The core of the chlorophyll fluorescence sensor introduced in Section 3.5 is the micro-processor *ATMEGA32U4*. In Figure D.1, the electrical circuit for the processor within the chlorophyll fluorescence sensor is shown. The *ATMEGA32U4* has multiple pins for controlling and monitoring the sensor components. In the following, the most important sensor component circuits, such as the LED control circuit, are highlighted in more detail.

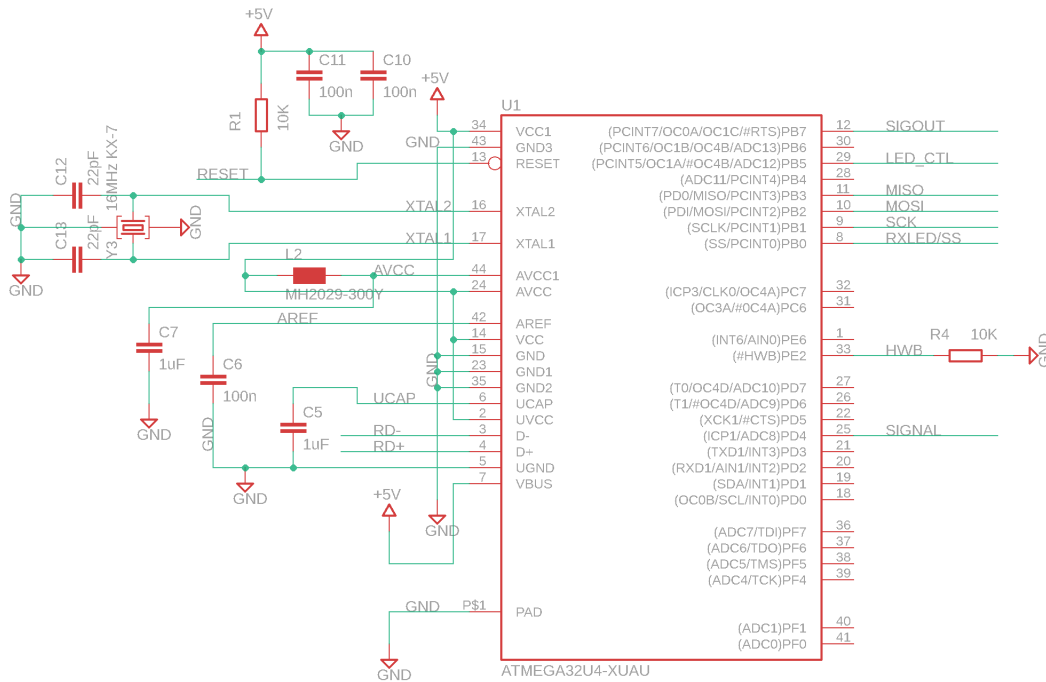


Figure D.1: Processor circuit.

**LED control circuit** The LED used for excitation of the chlorophyll fluorescence is controlled via a current control loop to ensure a constant current flow through the LED, Figure D.2. Therefore, two Bipolar Junction Transistors (BJTs) are used in combination with a resistor, which determines the current flow through the LED. If the LED control signal is turned on, current flows through the first BJT (T1) base,  $I_{B,1}$ , and allows a current flow from the collector

$$I_{C,1} = \beta I_{B,1}, \quad (\text{D.1})$$

where  $\beta$  is usually in the range of 100. The current through the resistor is then the sum of the collector and the base current,  $I_{C,1} + I_{B,1} \approx I_{C,1}$ . This current increases until a voltage of  $0.7\text{ V}$  is reached over the resistor, since this induces also a base-emitter voltage over the second BJT of  $V_{BE} = 0.7\text{ V}$ . This activates the second BJT such that excess current can flow over this BJT and does not increase the collector current according to Equation (D.1). Hence, the current through the LED is controlled at

$$I_{\text{LED}} = I_{C,1} \approx \frac{0.7\text{ V}}{R}. \quad (\text{D.2})$$

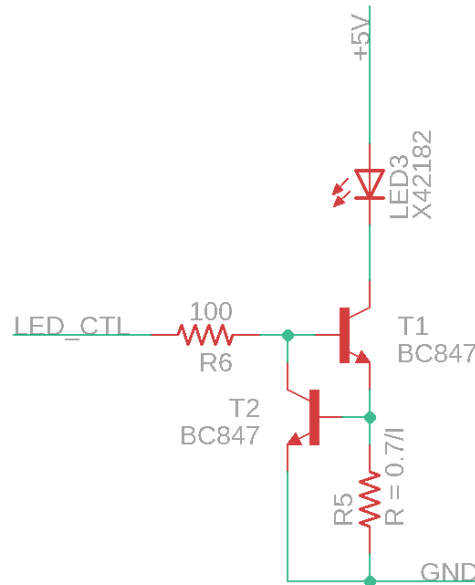


Figure D.2: LED circuit.

**PT monitoring circuit** The chlorophyll fluorescence is measured with a phototransistor which is connected to a transimpedance amplifier, an additional amplifier and a band pass filter, Figure D.3. All of these components are included within one element, the *VSOP98260*. The phototransistor provides a current based on the absorbed light energy which is then transformed to a voltage by the transimpedance amplifier. The voltage is

then further amplified and the signals is band pass filtered around 38 KHz. The processed signal is then forwarded to the microprocessor for further analysis.

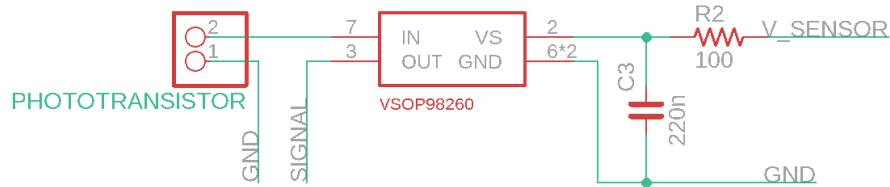


Figure D.3: PT circuit.

**Power supply circuit** In order to stabilize the operating voltage of the phototransistor circuit, a power supply circuit with large power supply rejection ratio has been inserted to the sensor design, Figure D.4. This prevents supply voltage dips caused by pulsing of the LED.

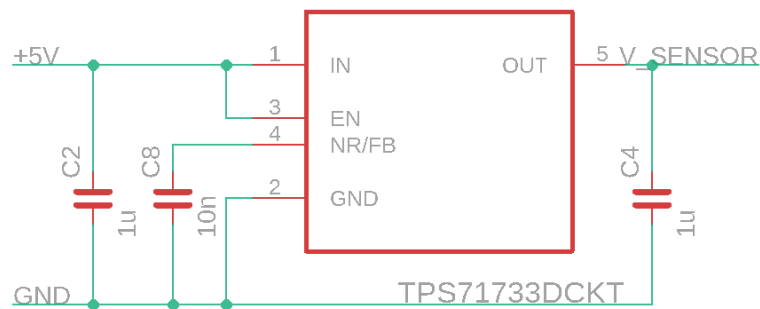


Figure D.4: Power supply circuit.

**Output signal circuit** In order to receive the sensor signal produced from the proposed chlorophyll fluorescence sensor, either a simple three pin connection with ground, power and signals can be used, Figure D.6, or a USB connection, Figure D.5, with which the sensor can also be programmed.

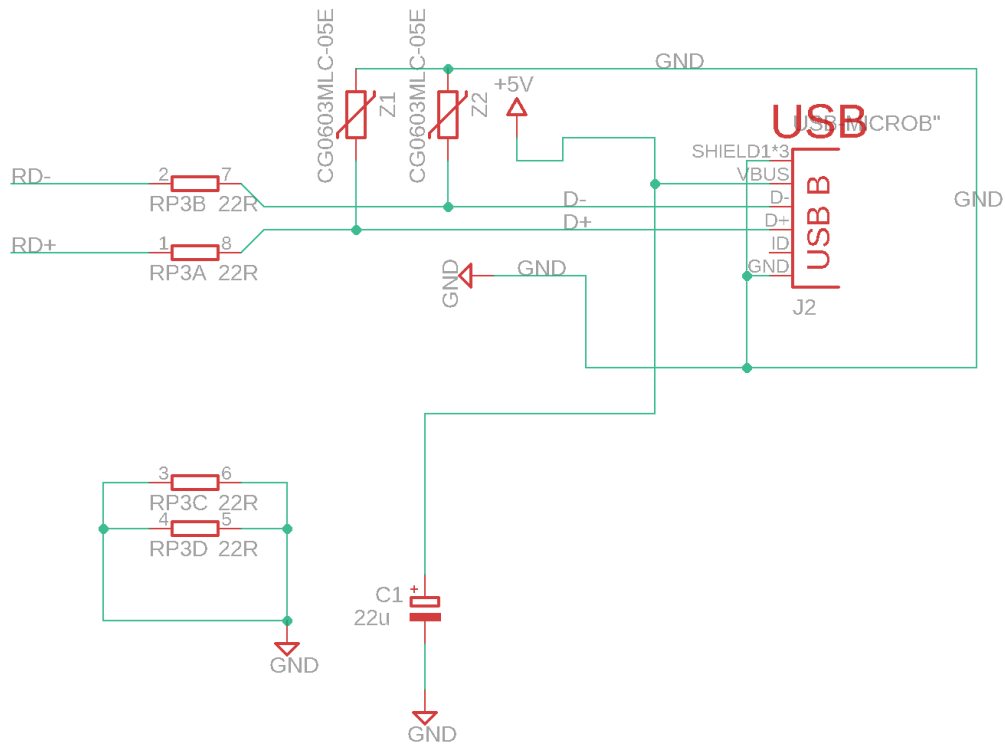


Figure D.5: USB port circuit.

## E PT & LED Spectra

In order to analyze which different PT and LED combinations can be used for the low-cost plant classification approach from Section 3.6, 11 different PTs and 7 different LEDs were tested. In Figure E.2, the relative spectral sensitivities for the PTs are shown and in Figure E.2 the relative spectral emission curves for the LEDs. To better classify the data, the chlorophyll a and b emission and absorption spectra are included as well. Additionally, in Table E.1 all PTs are listed given their label, the producing company, the peak sensitivity wavelength and the response time (rise and fall).

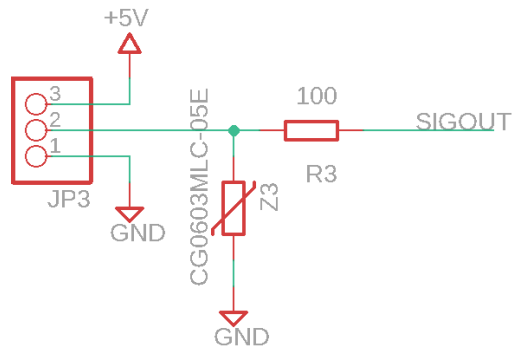


Figure D.6: Analog output circuit.

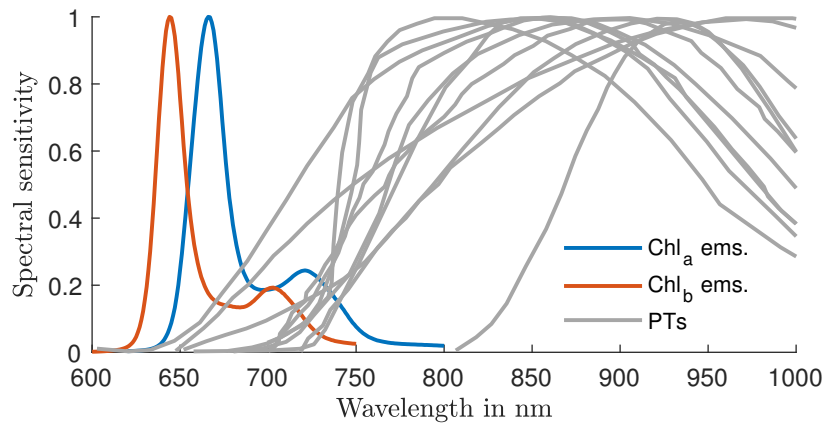


Figure E.1: Relative sensitivities of the investigated PTs for plant classification.

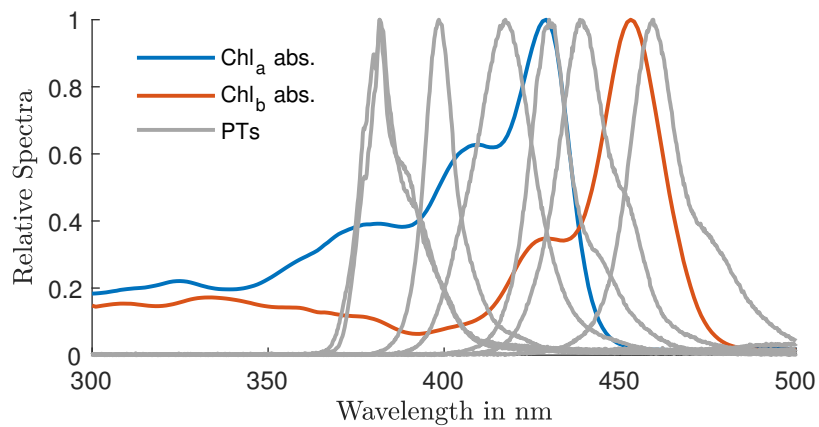


Figure E.2: Relative emission spectra of the investigated LEDs for plant classification.

Table E.1: Phototransistor selection analyzed for plant classification.

Label	Company	Peak Sensitivity Wavelength	Rise Time	Fall Time
PT480FE0000F	SHARP	860 nm	3 $\mu$ s	3.5 $\mu$ s
SDP8106	Honeywell	890 nm	75 $\mu$ s	75 $\mu$ s
PT204-6B	Everlight	940 nm	15 $\mu$ s	15 $\mu$ s
PT5529B/L2/H2-F	Everlight	940 nm	15 $\mu$ s	15 $\mu$ s
RPT-34PB3F	ROHM	800 nm	10 $\mu$ s	10 $\mu$ s
SFH 309 FA	OSRAM	950 nm	9 $\mu$ s	9 $\mu$ s
SFH 313 FA	OSRAM	950 nm	14 $\mu$ s	14 $\mu$ s
WP3DP3BT/BD-P22	Kingbright	940 nm	15 $\mu$ s	15 $\mu$ s
PNA1601M	PANASONIC	850 nm	4 $\mu$ s	4 $\mu$ s
OP599	TT Electronics	890 nm	-	-
OP522	TT Electronics	935 nm	-	-

## List of own Publications

### Conference Publications

- [1] **Nils Rottmann**, Ralf Bruder, Achim Schweikard and Elmar Rueckert. ‘Exploiting Chlorophyll Fluorescence for building robust low-cost Mowing Area Detectors’. In: *2020 IEEE SENSORS*. 2020, pp. 1–4. DOI: [10.1109/SENSORS47125.2020.9278858](https://doi.org/10.1109/SENSORS47125.2020.9278858). published.
- [2] **Nils Rottmann**, Tjaša Kunavar, Jan Babič, Jan Peters and Elmar Rueckert. ‘Learning Hierarchical Acquisition Functions for Bayesian Optimization’. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020, pp. 5490–5496. DOI: [10.1109/IROS45743.2020.9341335](https://doi.org/10.1109/IROS45743.2020.9341335). published.
- [3] **Nils Rottmann**, Ralf Bruder, Achim Schweikard and Elmar Rueckert. ‘Loop Closure Detection in Closed Environments’. In: *2019 European Conference on Mobile Robots (ECMR)*. IEEE, 2019, pp. 1–8. DOI: [10.1109/ECMR.2019.8870938](https://doi.org/10.1109/ECMR.2019.8870938). published.
- [4] **Nils Rottmann**, Ralf Bruder, Achim Schweikard and Elmar Rueckert. ‘Cataglyphis Ant Navigation Strategies Solve the Global Localization Problem in Robots with Binary Sensors’. In: *Proceedings of the 12th International Joint Conference on Biomedical Engineering Systems and Technologies - Volume 4: BIOSIGNALS, INSTICC*. SciTePress, 2019, pp. 214–223. ISBN: 978-989-758-353-7. DOI: [10.5220/0007556102140223](https://doi.org/10.5220/0007556102140223). published.
- [5] Honghu Xue, Sven Boettger, **Nils Rottmann**, Harit Pandya, Ralf Bruder, Gerhard Neumann, Achim Schweikard and Elmar Rueckert. ‘Sample-Efficient Covariance Matrix Adaptation Evolutional Strategy via Simulated Rollouts in Neural Networks’. In: *International Conference on Advances in Signal Processing and Artificial Intelligence (ASPAI’ 2020)*. 30th June 2020. URL: <https://ai-lab.science/wp/ASPAI2020Xue.pdf,%20Article%20File>. published.

### Conference Under Review

- [6] **Nils Rottmann**, Nico Studt, Floris Ernst and Elmar Rueckert. ‘Introduction to ROS-Mobile: Visualization and Control for the Robot Operating System on Android powered Devices’. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021, pp. 1–5. under review.
- [7] **Nils Rottmann**, Robin Denz, Ralf Bruder and Elmar Rueckert. ‘Probabilistic Complete Coverage Path Planning for low-cost Systems’. In: *2021 European Conference on Mobile Robots (ECMR)*. 2021. under review.

- [8] Robin Denz, Rabia Demirci, Mehmet Ege Cansev, Adna Blied, Philipp Beckerle, Elmar Rueckert and **Nils Rottmann**. ‘A high-accuracy, low-budget Sensor Glove for Trajectory Model Learning’. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021, pp. 1–7. under review.

### Journal Publications

- [9] **Nils Rottmann**, Ralf Bruder, Achim Schweikard and Elmar Rueckert. ‘A Novel Chlorophyll Fluorescence-Based Approach for Mowing Area Classification’. In: *IEEE Sensors Journal* 21.4 (2021), pp. 4500–4508. DOI: [10.1109/JSEN.2020.3032722](https://doi.org/10.1109/JSEN.2020.3032722). published.
- [10] Mehmet Ege Cansev, Honghu Xue, **Nils Rottmann**, Adna Blied, Luke E. Miller, Elmar Rueckert and Philipp Beckerle. ‘Interactive Human-Robot Skill Transfer: A Review of Learning Methods and User Experience’. In: *Advanced Intelligent Systems* (10th Mar. 2021), pp. 1–28. DOI: [10.1002/aisy.202000247](https://doi.org/10.1002/aisy.202000247). published.

### Journal Under Review

- [11] **Nils Rottmann**, Jannis Hagenah, Jan-Philip Klose, Ralf Bruder, Floris Ernst and Elmar Rueckert. ‘Plant Classification for low-cost Systems based on active Chlorophyll Fluorescence Stimulation: A case-study’. In: *IEEE Sensors Journal* (). under review.
- [12] **Nils Rottmann**, Nico Studt, Floris Ernst and Elmar Rueckert. ‘ROS-Mobile: An Android application for the Robot Operating System’. In: *SoftwareX* (). arXiv: [2011.02781](https://arxiv.org/abs/2011.02781). under review.

### Workshop Publications

- [13] **Nils Rottmann**, Ralf Bruder, Honghu Xue, Achim Schweikard and Elmar Rueckert. ‘Parameter Optimization for Loop Closure Detection in Closed Environments’. In: *12th IROS20 Workshop on Planning, Perception and Navigation for Intelligent Vehicles*. 25th Oct. 2020, pp. 1–8. URL: <https://arxiv.org/pdf/2011.06286.pdf>. published.

# List of Figures

<b>Introduction</b>	<b>1</b>
1.1 Growth of robot stocks and sales in the industrial and domestic area. . . . .	2
1.2 Example for domestic robot types. . . . .	3
<b>Navigation &amp; Planning</b>	<b>7</b>
2.1 General model for a differential drive robot. . . . .	13
2.2 Odometry model for a differential drive robot. . . . .	16
2.3 Cumulative distribution for the empirical model error. . . . .	18
2.4 Pose graph example. . . . .	19
2.5 Path segmentation. . . . .	21
2.6 Transformation of the dominant points to pose representations. . . . .	22
2.7 Example of a piecewise linear orientation function. . . . .	23
2.8 Comparison between shapes of neighboring vertices of a pose graph. . . . .	24
2.9 Exemplary optimization procedure for map estimation. . . . .	25
2.10 Effects of hyper-parameters on loop closure detection. . . . .	27
2.11 Example of the cost function for hyper-parameter learning. . . . .	28
2.12 Exemplary map comparison procedure. . . . .	29
2.13 Map estimate for a real courtyard environment. . . . .	30
2.14 Map estimate for a typical private lawn. . . . .	31
2.15 Evaluation of the mapping approach in different simulation scenarios. . . . .	32
2.16 Exemplary mapping results for different simulated map environments. . . . .	33
2.17 Dynamical system for the path planning approach . . . . .	34
2.18 Illustration of different coverage models. . . . .	35
2.19 Neural network schematic illustrating neighborhood connections. . . . .	36
2.20 Neural activities for different coverage rates. . . . .	37
2.21 Example path planning path. . . . .	38
2.22 Different simulation scenarios for evaluation. . . . .	39
2.23 Evaluation of the CCPP approach in different simulation scenarios. . . . .	40
2.24 Correlation between true and estimated coverage. . . . .	41
2.25 Method performance for the CCPP approach. . . . .	42
<b>Sensor Development for Working Area Detection and Classification</b>	<b>47</b>
3.1 Plant species used for testing the proposed plant classification approach. . . . .	49

3.2	The chlorophyll fluorescence sensor and the respective classification accuracy.	50
3.3	Example diagram for chlorophyll fluorescence.	52
3.4	Chlorophyll a,b absorption and emission spectra.	53
3.5	The “Kautsky-Effect” and the chlorophyll lifetime.	54
3.6	Examples of different normalized absorption and emission spectra.	55
3.7	Spectral sensitivity and collector current of the RPT-37PB3F.	56
3.8	The signal path for the proposed low cost sensor.	57
3.9	A sketch of the sensor emission (LED) and absorption (PT) apertures.	58
3.10	Relative light emission and Bode plot for the used high-power LED.	61
3.11	Different lawn types, classified based on the observable relative quality.	62
3.12	Evaluation of the classification accuracy of the chlorophyll fluorescence sensor.	63
3.13	Evaluation of the chlorophyll fluorescence sensor under real garden conditions.	64
3.14	Experimental setup for the data acquisition for plant classification.	65
3.15	Example measurements of the chlorophyll fluorescence response.	66
3.16	Feature extraction given two exemplary PTs.	68
3.17	Mean classification result for the best CNN architecture.	70
3.18	Classification accuracies with respect to the number of used PTs.	71
3.19	Classification accuracies with respect to the number of used LEDs.	72
3.20	Mean classification result and variance for the low-cost approach.	72
<b>Robot Control and Monitoring – ROS-Mobile</b>		<b>77</b>
4.1	Global market shares for Desktop, Mobile and Tablet devices.	78
4.2	An exemplary ROS system.	81
4.3	The general MVVM designing concept for mobile applications.	83
4.4	Illustrative example for the mapping of an apartment environment.	85
4.5	Example mobile robotic platforms for which ROS-Mobile can be used.	86
4.6	User statistics since application release.	87
<b>Appendix</b>		<b>93</b>
A.1	Wall following method.	94
B.1	Illustration of the Bayesian Optimization Process.	96
D.1	Processor circuit.	97
D.2	LED circuit.	98
D.3	PT circuit.	99
D.4	Power supply circuit.	99
D.5	USB port circuit.	100
D.6	Analog output circuit.	101
E.1	Relative sensitivities of the investigated PTs for plant classification.	101
E.2	Relative emission spectra of the investigated LEDs for plant classification.	101

# List of Tables

<b>Navigation &amp; Planning</b>	<b>7</b>
2.1 Estimated motion model parameters and measured robot parameters. . . . .	17
2.2 Map generation method performance. . . . .	31
2.3 CCPP method parameters . . . . .	40
2.4 CCPP method performance. . . . .	41
<b>Sensor Development for Working Area Detection and Classification</b>	<b>47</b>
3.1 Important parameters for the analysis of the signal path. . . . .	60
3.2 Plant species for classification. . . . .	66
3.3 Evaluated hyper-parameter values during architecture optimization. . . . .	67
3.4 Classification results for CNN. . . . .	69
<b>Robot Control and Monitoring – ROS-Mobile</b>	<b>77</b>
4.1 Comparison of different ROS applications currently available for Android. . .	79
4.2 Software Information . . . . .	80
<b>Appendix</b>	<b>93</b>
E.1 Phototransistor selection analyzed for plant classification. . . . .	102



## Acronyms

<b>AC</b>	Alternating Current
<b>BCD</b>	Boustrophedon Cell Decomposition
<b>BJT</b>	Bipolar Junction Transistor
<b>BO</b>	Bayesian Optimization
<b>CCPP</b>	Complete Coverage Path Planning
<b>DC</b>	Direct Current
<b>EI</b>	Expected Improvement
<b>EKF</b>	Extended Kalman Filter
<b>EM</b>	Expectation-Maximization
<b>GMM</b>	Gaussian Mixture Model
<b>GP</b>	Gaussian Process
<b>LED</b>	Light-Emitting Diode
<b>LiDAR</b>	Light Detection And Ranging
<b>MCL</b>	Monte Carlo Localization
<b>MLE</b>	Maximum Likelihood Estimation
<b>MPC</b>	Motion Predictive Control
<b>MVVM</b>	Model View ViewModel
<b>OS</b>	Operating System
<b>OSRF</b>	Open Source Robotics Foundation
<b>PCB</b>	Printed Circuit Board
<b>PGO</b>	Pose Graph Optimization
<b>PSI</b>	Photosystem I

**PSII** Photosystem II

**PT** Phototransistor

**ReLU** Rectified Linear Unit

**RL** Reinforcement Learning

**ROS** Robot Operating System

**rviz** ROS Visualization

**SLAM** Simultaneous Localization and Mapping

**SPI** Serial Peripheral Interface

**TSP** Traveling Salesman Problem

**UAV** Unmanned Aerial Vehicle

**UI** User Interface

**USB** Universal Serial Bus

# Nomenclature

## Chapter 2: Navigation & Planning

$\alpha_i$	Noise parameters for the kinematic motion model
$\xi_{ij}$	Relative measurement between two poses $i$ and $j$
$\mathbf{A}$	Pose graph incident matrix
$\mathbf{C}$	Loop closure comparison matrix
$\mathbf{P}_{ij}$	Noise covariance matrix for the relative measurement $\xi_{ij}$
$\Delta\varphi$	Angular movement of the robot
$\Delta A$	Difference between groundtruth and map estimate
$\Delta s$	Translational movement of the robot
$\delta_{\text{var}}$	Robot's movement for the odometry motion model
$\epsilon_{\text{var}}$	Zero mean Gaussian noise variable, e.g., for translational velocity
$\gamma_i$	Parameters for the covariance matrix for the loop closing constraints
$\mathcal{B}$	Body coordinate frame
$\mathcal{C}$	Probability occupancy grid map for coverage estimates
$\mathcal{M}$	Binary occupancy grid map of the environment
$\mathcal{W}$	World coordinate frame
$\sigma_{\text{max}}$	Upper limit after which relocalization starts
$\sigma_{\text{min}}$	Lower limit after which path execution starts
$A$	Passive decay rate
$a_i$	Noise parameters for the odometry motion model
$B$	Upper bound of the neural activity
$c_{\text{max}}$	Maximum comparison error for loop closure detection

$c_{i,t}$	Coverage probability of a cell $i$ at time $t$ to be covered
$D$	Lower bound of the neural activity
$d_{\text{effector}}$	Diameter of the robot's effector
$d_r, d_l$	Diameter of the right and left wheel
$DP$	Set of dominant points which represent the pruned odometry data
$E$	External input accelerator
$e_{\text{max}}$	Maximum pruning error
$I_i$	External input to a neuron $i$
$L$	Half of the robot's wheel distance
$L_{\text{min}}$	Minimum pruning length
$L_{\text{NH}}$	Neighborhood region parameter
$l_r, l_l$	Unrolled distance of the right and left wheel
$n_{\text{Hall}}$	Total number of ticks per complete rotation
$n_r, n_l$	Number of ticks recorded by the right and left wheel encoder
$N_{\text{thr}}$	Threshold for particle resampling
$q_i$	Neural activity of a neuron $i$
$SP$	Loop closing pair of poses from the pose graph
$T$	Traveled path distance after complete coverage path execution
$U$	Circumference of the closed environment
$u$	Path distance between poses of a loop closing pair within a pose graph
$v_r, v_l$	Translational velocity of the right and left wheel

### Chapter 3: Working Area Detection and Classification

$\alpha$	Surface Albedo
$\bar{h}_{\text{PT}}$	Mean distance between the PT and the illuminated area
$\beta_{\text{LED}}$	Lens angle of the LED
$\gamma$	Re-emission magnitude for the chlorophyll fluorescence
$\Phi_{\text{Chl}}$	Luminous flux from chlorophyll fluorescence response of the grass

$\Phi_{\text{LED}}$	Luminous flux of the LED under full power
$\Phi_{\text{PT,AC}}$	Luminous flux received by the PT, modulated AC signal
$\Phi_{\text{PT,DC}}$	Luminous flux received by the PT, sunlight DC signal
$\varphi$	Part of scattered light from the lawn surface received by the PT
$A_{\text{LED}}$	Illuminated area by the LED
$A_{\text{PT,act}}$	Active measurement surface of the PT
$A_{\text{PT}}$	Scanned area by the PT
$E_{\text{Sun}}$	Illuminance of the sun with $10^5 lx$ by summer and clear sky
$f_{\text{Chl}}(\lambda)$	Normalized emission spectrum of the chlorophyll fluorescence
$f_{\text{Grass}}(\lambda)$	Normalized reflection spectrum of grass
$f_{\text{LED}}$	LED pulsing frequency
$f_{\text{LED}}(\lambda)$	Normalized emission spectrum of the LED
$f_{\text{sensor}}$	Classification frequency of the sensor
$f_{ij}$	A feature for plant classification with PT $i$ and LED $j$
$g_j(\lambda)$	Relative spectrum of a plant by LED $j$
$g_{\text{Chl}}(\lambda)$	Relative absorption spectrum of chlorophyll a
$g_{\text{PT}}(\lambda)$	Relative spectral sensitivity of the PT
$h_i(\lambda)$	Spectral sensitivity for PT $i$
$h_{\text{LED}}$	Distance from the sensor to the ground
$I_{C,AC}$	Collector current of the PT, AC signal
$I_{C,DC}$	Collector current of the PT, DC signal
$t_r, t_f$	Rise and fall time for PTs



## Bibliography

- [1] Karel Capek. *R.U.R. (Rossum's universal robots)*. Dover Publications Inc., 2001. ISBN: 9780486419268.
- [2] Isaac Asimov. 'Runaround'. In: *Astounding Science Fiction* 29.1 (1942), pp. 94–103.
- [3] Irati Zamalloa, Risto Kojcev, Alejandro Hernández, Inigo Muguruza, Lander Usategui, Asier Bilbao and Víctor Mayoral. 'Dissecting robotics-historical overview and future perspectives'. In: *arXiv preprint* (2017). arXiv: 1704.08617.
- [4] A. Gasparetto and L. Scalera. 'From the Unimate to the Delta Robot: The Early Decades of Industrial Robotics'. In: *Explorations in the History and Heritage of Machines and Mechanisms*. Ed. by Baichun Zhang and Marco Ceccarelli. Cham: Springer International Publishing, 2019, pp. 284–295. ISBN: 978-3-030-03538-9. DOI: 10.1007/978-3-030-03538-9\_23.
- [5] Erwin Prassler, Mario E Munich, Paolo Pirjanian and Kazuhiro Kosuge. 'Domestic robotics'. In: *Springer handbook of robotics*. Springer, 2016, pp. 1729–1758. ISBN: 978-3-319-32552-1. DOI: 10.1007/978-3-319-32552-1\_65.
- [6] Michael Montemerlo, Sebastian Thrun, Daphne Koller, Ben Wegbreit et al. 'FastSLAM: A factored solution to the simultaneous localization and mapping problem'. In: *In Proceedings of the AAAI National Conference on Artificial Intelligence*. AAAI, 2003, pp. 593–598. URL: <http://www.aaai.org/Papers/AAAI/2002/AAAI02-089.pdf>.
- [7] Giorgio Grisetti, Rainer Kümmerle, Cyrill Stachniss and Wolfram Burgard. 'A Tutorial on Graph-Based SLAM'. In: *IEEE Intelligent Transportation Systems Magazine* 2.4 (2010), pp. 31–43. DOI: 10.1109/MITS.2010.939925.
- [8] C. Brenneke, O. Wulf and B. Wagner. 'Using 3D laser range data for SLAM in outdoor environments'. In: *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*. Vol. 1. 2003, 188–193 vol.1. DOI: 10.1109/IROS.2003.1250626.
- [9] Javier Civera, Dorian Gálvez-López, L. Riazuelo, Juan D. Tardós and J. M. M. Montiel. 'Towards semantic SLAM using a monocular camera'. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2011, pp. 1277–1284. DOI: 10.1109/IROS.2011.6094648.

- [10] Kurt Konolige, Motilal Agrawal, Robert C. Bolles, Cregg Cowan, Martin Fischler and Brian Gerkey. ‘Outdoor Mapping and Navigation Using Stereo Vision’. In: *Experimental Robotics: The 10th International Symposium on Experimental Robotics*. Berlin, Heidelberg: Springer, 2008, pp. 179–190. ISBN: 978-3-540-77457-0. DOI: 10.1007/978-3-540-77457-0\_17.
- [11] Nikolas Engelhard, Felix Endres, Jürgen Hess, Jürgen Sturm and Wolfram Burgard. ‘Real-time 3D visual SLAM with a hand-held RGB-D camera’. In: *Proc. of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum, Vasteras, Sweden*. Vol. 180. 2011, pp. 1–15. URL: [https://vision.in.tum.de/\\_media/spezial/bib/engelhard11euron.pdf](https://vision.in.tum.de/_media/spezial/bib/engelhard11euron.pdf).
- [12] Tim Bailey, Juan Nieto, Jose Guivant, Michael Stevens and Eduardo Nebot. ‘Consistency of the EKF-SLAM Algorithm’. In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2006, pp. 3562–3568. DOI: 10.1109/IR0S.2006.281644.
- [13] K.R. Beevers and W.H. Huang. ‘SLAM with sparse sensing’. In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*. 2006, pp. 2285–2290. DOI: 10.1109/ROBOT.2006.1642043.
- [14] Young-Ho Choi, Tae-Kyeong Lee and Se-Young Oh. ‘A line feature based SLAM with low grade range sensors using geometric constraints and active exploration for mobile robot’. In: *Autonomous Robots* 24.1 (2008), pp. 13–27. DOI: 10.1007/s10514-007-9050-y.
- [15] Ozan Ozisik and Sirma Yavuz. ‘Simultaneous Localization and Mapping with Limited Sensing Using Extended Kalman Filter and Hough Transform’. In: *Tehnički vjesnik/Technical Gazette* 23 (Dec. 2016). DOI: 10.17559/TV-20150830235942.
- [16] Ying Zhang, Juan Liu, Gabriel Hoffmann, Mark Quilling, Kenneth Payne, Prasanta Bose and Andrew Zimdars. ‘Real-time indoor mapping for mobile robots with limited sensing’. In: *The 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2010)*. 2010, pp. 636–641. DOI: 10.1109/MASS.2010.5663778.
- [17] Nils Einecke, Jörg Deigmöller, Keiji Muro and Mathias Franzius. ‘Boundary Wire Mapping on Autonomous Lawn Mowers’. In: *Field and Service Robotics*. Cham: Springer International Publishing, 2018, pp. 351–365. ISBN: 978-3-319-67361-5.
- [18] Zerong Su, Xuefeng Zhou, Taobo Cheng, Hong Zhang, Baolai Xu and Weinan Chen. ‘Global localization of a mobile robot using lidar and visual features’. In: *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. 2017, pp. 2377–2383. DOI: 10.1109/ROBIO.2017.8324775.
- [19] Liqian Feng, Sheng Bi, Min Dong, Fating Hong, Yuhong Liang, Qinjie Lin and Yunda Liu. ‘A Global Localization System for Mobile Robot Using LIDAR Sensor’. In: *2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*. 2017, pp. 478–483. DOI: 10.1109/CYBER.2017.8446067.

- [20] Seigo Ito, Felix Endres, Markus Kuderer, Gian Diego Tipaldi, Cyrill Stachniss and Wolfram Burgard. ‘W-RGB-D: Floor-plan-based indoor global localization using a depth camera and WiFi’. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. 2014, pp. 417–422. DOI: [10.1109/ICRA.2014.6906890](https://doi.org/10.1109/ICRA.2014.6906890).
- [21] Ben Tribelhorn and Zachary Dodds. ‘Evaluating the Roomba: A low-cost, ubiquitous platform for robotics research and education’. In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. 2007, pp. 1393–1399. DOI: [10.1109/ROBOT.2007.363179](https://doi.org/10.1109/ROBOT.2007.363179).
- [22] Jason M. O’Kane and Steven M. LaValle. ‘Localization With Limited Sensing’. In: *IEEE Transactions on Robotics* 23.4 (2007), pp. 704–716. DOI: [10.1109/TR0.2007.900636](https://doi.org/10.1109/TR0.2007.900636).
- [23] Lawrence H. Erickson, Joseph Knuth, Jason M. O’Kane and Steven M. LaValle. ‘Probabilistic localization with a blind robot’. In: *2008 IEEE International Conference on Robotics and Automation*. 2008, pp. 1821–1827. DOI: [10.1109/ROBOT.2008.4543472](https://doi.org/10.1109/ROBOT.2008.4543472).
- [24] Demetris Stavrou and Christos Panayiotou. ‘Localization of a simple robot with low computational power using a single short range sensor’. In: *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. 2012, pp. 729–734. DOI: [10.1109/ROBIO.2012.6491054](https://doi.org/10.1109/ROBIO.2012.6491054).
- [25] F. Dellaert, D. Fox, W. Burgard and S. Thrun. ‘Monte Carlo localization for mobile robots’. In: *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*. Vol. 2. 1999, 1322–1328 vol.2. DOI: [10.1109/ROBOT.1999.772544](https://doi.org/10.1109/ROBOT.1999.772544).
- [26] V.J. Lumelsky, S. Mukhopadhyay and K. Sun. ‘Dynamic path planning in sensor-based terrain acquisition’. In: *IEEE Transactions on Robotics and Automation* 6.4 (1990), pp. 462–472. DOI: [10.1109/70.59357](https://doi.org/10.1109/70.59357).
- [27] Howie Choset and Philippe Pignon. ‘Coverage Path Planning: The Boustrophedon Cellular Decomposition’. In: *Field and Service Robotics*. London: Springer, 1998, pp. 203–209. ISBN: 978-1-4471-1273-0.
- [28] S.C. Wong and B.A. MacDonald. ‘A topological coverage algorithm for mobile robots’. In: *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*. Vol. 2. 2003, 1685–1690 vol.2. DOI: [10.1109/IROS.2003.1248886](https://doi.org/10.1109/IROS.2003.1248886).
- [29] H. Moravec and A. Elfes. ‘High resolution maps from wide angle sonar’. In: *Proceedings. 1985 IEEE International Conference on Robotics and Automation*. Vol. 2. 1985, pp. 116–121. DOI: [10.1109/ROBOT.1985.1087316](https://doi.org/10.1109/ROBOT.1985.1087316).
- [30] Alexander Zelinsky, Ray A Jarvis, JC Byrne, Shinichi Yuta et al. ‘Planning paths of complete coverage of an unstructured environment by a mobile robot’. In: *Proceedings of international conference on advanced robotics*. Vol. 13. 1993, pp. 533–538. URL: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.53.7617&rep=rep1&type=pdf>.

- [31] Enric Galceran and Marc Carreras. ‘A survey on coverage path planning for robotics’. In: *Robotics and Autonomous Systems* 61.12 (2013), pp. 1258–1276. ISSN: 0921-8890. DOI: 10.1016/j.robot.2013.09.004.
- [32] Michel Taïx, Philippe Souères, Helene Frayssinet and Lionel Cordesses. ‘Path Planning for Complete Coverage with Agricultural Machines’. In: *Field and Service Robotics: Recent Advances in Reserch and Applications*. Berlin, Heidelberg: Springer, 2006, pp. 549–558. ISBN: 978-3-540-32854-4. DOI: 10.1007/10991459\_53. URL: 10.1007/10991459\_53.
- [33] Iwan Ulrich, Francesco Mondada and J.-D. Nicoud. ‘Autonomous vacuum cleaner’. In: *Robotics and Autonomous Systems* 19.3 (1997). Intelligent Robotic Systems SIRS’95, pp. 233–245. ISSN: 0921-8890. DOI: 10.1016/S0921-8890(96)00053-X.
- [34] Jürgen Hess, Maximilian Beinhofer, Daniel Kuhner, Philipp Ruchti and Wolfram Burgard. ‘Poisson-driven dirt maps for efficient robot cleaning’. In: *2013 IEEE International Conference on Robotics and Automation*. 2013, pp. 2245–2250. DOI: 10.1109/ICRA.2013.6630880.
- [35] Jürgen Hess, Maximilian Beinhofer and Wolfram Burgard. ‘A probabilistic approach to high-confidence cleaning guarantees for low-cost cleaning robots’. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. 2014, pp. 5600–5605. DOI: 10.1109/ICRA.2014.6907682.
- [36] Ping-Min Hsu and Chun-Liang Lin. ‘Optimal planner for lawn mowers’. In: *2010 IEEE 9th International Conference on Cyberntic Intelligent Systems*. 2010, pp. 1–7. DOI: 10.1109/UKRICIS.2010.5898126.
- [37] Z.J. Butler, A.A. Rizzi and R.L. Hollis. ‘Contact sensor-based coverage of rectilinear environments’. In: *Proceedings of the 1999 IEEE International Symposium on Intelligent Control Intelligent Systems and Semiotics (Cat. No.99CH37014)*. 1999, pp. 266–271. DOI: 10.1109/ISIC.1999.796666.
- [38] G. Grisettiyz, C. Stachniss and W. Burgard. ‘Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling’. In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. 2005, pp. 2432–2437. DOI: 10.1109/ROBOT.2005.1570477.
- [39] Giorgio Grisetti, Cyrill Stachniss and Wolfram Burgard. ‘Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters’. In: *IEEE Transactions on Robotics* 23.1 (2007), pp. 34–46. DOI: 10.1109/TR0.2006.889486.
- [40] Mathieu Labbé and François Michaud. ‘Online global loop closure detection for large-scale multi-session graph-based SLAM’. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2014, pp. 2661–2666. DOI: 10.1109/IR0S.2014.6942926.
- [41] Mathieu Labbé and François Michaud. ‘RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation’. In: *Journal of Field Robotics* 36.2 (2019), pp. 416–446. DOI: 10.1002/rob.21831.

- [42] Yassin Abdelrasoul, Abu Bakar Sayuti HM Saman and Patrick Sebastian. ‘A quantitative study of tuning ROS gmapping parameters and their effect on performing indoor 2D SLAM’. In: *2016 2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA)*. 2016, pp. 1–6. DOI: [10.1109/ROMA.2016.7847825](https://doi.org/10.1109/ROMA.2016.7847825).
- [43] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. Vol. 2. MIT press Cambridge, 2018. ISBN: 9780262039246.
- [44] Jasper Snoek, Hugo Larochelle and Ryan P. Adams. ‘Practical Bayesian Optimization of Machine Learning Algorithms’. In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2. NIPS’12*. Lake Tahoe, Nevada: Curran Associates Inc., 2012, pp. 2951–2959. arXiv: [1206.2944](https://arxiv.org/abs/1206.2944).
- [45] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams and Nando de Freitas. ‘Taking the Human Out of the Loop: A Review of Bayesian Optimization’. In: *Proceedings of the IEEE* 104.1 (2016), pp. 148–175. DOI: [10.1109/JPR0C.2015.2494218](https://doi.org/10.1109/JPR0C.2015.2494218).
- [46] S.X. Yang and C. Luo. ‘A neural network approach to complete coverage path planning’. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 34.1 (2004), pp. 718–724. DOI: [10.1109/TSMCB.2003.811769](https://doi.org/10.1109/TSMCB.2003.811769).
- [47] Chaomin Luo and Simon X. Yang. ‘A Bioinspired Neural Network for Real-Time Concurrent Map Building and Complete Coverage Robot Navigation in Unknown Environments’. In: *IEEE Transactions on Neural Networks* 19.7 (2008), pp. 1279–1298. DOI: [10.1109/TNN.2008.2000394](https://doi.org/10.1109/TNN.2008.2000394).
- [48] Sebastian Thrun, Wolfram Burgard and Dieter Fox. *Probabilistic Robotics*. MIT press Cambridge, 2005. ISBN: 978-0-262-20162-9.
- [49] Richard J Rossi. *Mathematical Statistics: An introduction to likelihood based inference*. John Wiley & Sons, 2018. ISBN: 978-1-118-77104-4.
- [50] Frank J. Massey Jr. ‘The Kolmogorov-Smirnov Test for Goodness of Fit’. In: *Journal of the American Statistical Association* 46.253 (1951), pp. 68–78. DOI: [10.1080/01621459.1951.10500769](https://doi.org/10.1080/01621459.1951.10500769).
- [51] Virginia Torczon. ‘On the Convergence of Pattern Search Algorithms’. In: *SIAM Journal on Optimization* 7.1 (1997), pp. 1–25. DOI: [10.1137/S1052623493250780](https://doi.org/10.1137/S1052623493250780).
- [52] Rico Klinckenberg. ‘Localization and Control for Trajectory Tracking for Autonomous Lawn Mowers’. Bachelorthesis. University of Luebeck, 2018.
- [53] Feng Lu and Evangelos Milios. ‘Globally consistent range scan alignment for environment mapping’. In: *Autonomous robots* 4.4 (1997), pp. 333–349. DOI: [10.1023/A:1008854305733](https://doi.org/10.1023/A:1008854305733).
- [54] Giorgio Grisetti, Cyrill Stachniss and Wolfram Burgard. ‘Nonlinear Constraint Network Optimization for Efficient Map Learning’. In: *IEEE Transactions on Intelligent Transportation Systems* 10.3 (2009), pp. 428–439. DOI: [10.1109/TITS.2009.2026444](https://doi.org/10.1109/TITS.2009.2026444).

- [55] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige and Wolfram Burgard. ‘G2o: A general framework for graph optimization’. In: *2011 IEEE International Conference on Robotics and Automation*. 2011, pp. 3607–3613. DOI: [10.1109/ICRA.2011.5979949](https://doi.org/10.1109/ICRA.2011.5979949).
- [56] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J Leonard and Frank Dellaert. ‘iSAM2: Incremental smoothing and mapping using the Bayes tree’. In: *The International Journal of Robotics Research* 31.2 (2012), pp. 216–235. DOI: [10.1177/0278364911430419](https://doi.org/10.1177/0278364911430419).
- [57] Luca Carlone, Rosario Aragues, José A. Castellanos and Basilio Bona. ‘A fast and accurate approximation for planar pose graph optimization’. In: *The International Journal of Robotics Research* 33.7 (2014), pp. 965–987. DOI: [10.1177/0278364914523689](https://doi.org/10.1177/0278364914523689).
- [58] E. Olson, J. Leonard and S. Teller. ‘Fast iterative alignment of pose graphs with poor initial estimates’. In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*. 2006, pp. 2262–2269. DOI: [10.1109/ROBOT.2006.1642040](https://doi.org/10.1109/ROBOT.2006.1642040).
- [59] Per Bergström and Ove Edlund. ‘Robust registration of point sets using iteratively re-weighted least squares’. In: *Computational optimization and applications* 58.3 (2014), pp. 543–561. DOI: [10.1007/s10589-014-9643-2](https://doi.org/10.1007/s10589-014-9643-2).
- [60] Douglas Reynolds. ‘Gaussian Mixture Models’. In: *Encyclopedia of Biometrics*. Boston, MA: Springer US, 2015, pp. 827–832. ISBN: 978-1-4899-7488-4. DOI: [10.1007/978-1-4899-7488-4\\_196](https://doi.org/10.1007/978-1-4899-7488-4_196).
- [61] A. P. Dempster, N. M. Laird and D. B. Rubin. ‘Maximum Likelihood from Incomplete Data Via the EM Algorithm’. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (1977), pp. 1–22. DOI: [10.1111/j.2517-6161.1977.tb01600.x](https://doi.org/10.1111/j.2517-6161.1977.tb01600.x).
- [62] Geoffrey J McLachlan, Kaye E Basford and Marcel Dekker. ‘Mixture models: Inference and applications to clustering’. In: *Journal of the American Statistical Association* 84 (1988), pp. 337–338. DOI: [10.2307/2289892](https://doi.org/10.2307/2289892).
- [63] J. Mockus, Vytautas Tiesis and Antanas Zilinskas. ‘The application of Bayesian methods for seeking the extremum’. In: *Towards Global Optimization*. Vol. 2. Sept. 2014, pp. 117–129. ISBN: 0-444-85171-2.
- [64] Stephen Grossberg. ‘Nonlinear neural networks: Principles, mechanisms, and architectures’. In: *Neural Networks* 1.1 (1988), pp. 17–61. ISSN: 0893-6080. DOI: [10.1016/0893-6080\(88\)90021-4](https://doi.org/10.1016/0893-6080(88)90021-4).
- [65] A. L. Hodgkin and A. F. Huxley. ‘A quantitative description of membrane current and its application to conduction and excitation in nerve’. In: *The Journal of Physiology* 117.4 (1952), pp. 500–544. DOI: [10.1113/jphysiol.1952.sp004764](https://doi.org/10.1113/jphysiol.1952.sp004764).

- [66] Wolfram Burgard, Cyrill Stachniss, Giorgio Grisetti, Bastian Steder, Rainer Kümmerle, Christian Dornhege, Michael Ruhnke, Alexander Kleiner and Juan D. Tardös. ‘A comparison of SLAM algorithms based on a graph of relations’. In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2009, pp. 2089–2095. DOI: 10.1109/IRoS.2009.5354691.
- [67] Nils Rottmann, Ralf Bruder, Achim Schweikard and Elmar Rueckert. ‘Loop Closure Detection in Closed Environments’. In: *2019 European Conference on Mobile Robots (ECMR)*. IEEE, 2019, pp. 1–8. DOI: 10.1109/ECMR.2019.8870938. published.
- [68] Nils Rottmann, Ralf Bruder, Honghu Xue, Achim Schweikard and Elmar Rueckert. ‘Parameter Optimization for Loop Closure Detection in Closed Environments’. In: *12th IROS20 Workshop on Planning, Perception and Navigation for Intelligent Vehicles*. 25th Oct. 2020, pp. 1–8. URL: <https://arxiv.org/pdf/2011.06286.pdf>. published.
- [69] Nils Rottmann, Robin Denz, Ralf Bruder and Elmar Rueckert. ‘Probabilistic Complete Coverage Path Planning for low-cost Systems’. In: *2021 European Conference on Mobile Robots (ECMR)*. 2021. under review.
- [70] Junho Yang, Soon-Jo Chung, Seth Hutchinson, David Johnson and Michio Kise. ‘Vision-based localization and mapping for an autonomous mower’. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2013, pp. 3655–3662. DOI: 10.1109/IRoS.2013.6696878.
- [71] Tomoya Fukukawa, Kosuke Sekiyama, Yasuhisa Hasegawa and Toshio Fukuda. ‘Vision-Based Mowing Boundary Detection Algorithm for an Autonomous Lawn Mower’. In: *Journal of Advanced Computational Intelligence and Intelligent Informatics* 20.1 (2016), pp. 49–56. DOI: 10.20965/jaciii.2016.p0049.
- [72] Fabrizio Bernini. ‘Lawn-mower with sensor’. US Patent 7,613,552. 2009.
- [73] Ukrit Watchareeruetai, Yoshinori Takeuchi, Tetsuya Matsumoto, Hiroaki Kudo and Noboru Ohnishi. ‘Computer vision based methods for detecting weeds in lawns’. In: *Machine Vision and Applications* 17.5 (2006), pp. 287–296. DOI: 10.1007/s00138-006-0039-x.
- [74] Alexander Schepelmann, Richard E. Hudson, Frank L. Merat and Roger D. Quinn. ‘Visual segmentation of lawn grass for a mobile robotic lawnmower’. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2010, pp. 734–739. DOI: 10.1109/IRoS.2010.5650430.
- [75] Aaron D. Smith, H. Jacky Chang and Edward J. Blanchard. ‘An outdoor high-accuracy local positioning system for an autonomous robotic golf greens mower’. In: *2012 IEEE International Conference on Robotics and Automation*. 2012, pp. 2633–2639. DOI: 10.1109/ICRA.2012.6224990.
- [76] S Lawrence Bellinger. ‘Self-propelled random motion lawnmower’. US Patent 3,570,227. 1971.

- [77] Ismael Moya and Zoran G. Cerovic. 'Remote Sensing of Chlorophyll Fluorescence: Instrumentation and Analysis'. In: *Chlorophyll a Fluorescence: A Signature of Photosynthesis*. Dordrecht: Springer Netherlands, 2004, pp. 429–445. ISBN: 978-1-4020-3218-9. DOI: [10.1007/978-1-4020-3218-9\\_16](https://doi.org/10.1007/978-1-4020-3218-9_16).
- [78] F Castagnoli, G Cecchi, L Pantani, I Pippi, B Radicati and P Mazzinghi. 'A Fluorescence LIDAR For Land And Sea Remote Sensing'. In: *Laser Radar Technology and Applications I*. Vol. 0663. International Society for Optics and Photonics. SPIE, 1986, pp. 212–216. DOI: [10.1117/12.938676](https://doi.org/10.1117/12.938676).
- [79] Jian Yang, Wei Gong, Shuo Shi, Lin Du, Bo Zhu, Jia Sun and Shalei Song. 'Excitation Wavelength Analysis of Laser-Induced Fluorescence LiDAR for Identifying Plant Species'. In: *IEEE Geoscience and Remote Sensing Letters* 13.7 (2016), pp. 977–981. DOI: [10.1109/LGRS.2016.2557484](https://doi.org/10.1109/LGRS.2016.2557484).
- [80] Guy Samson, Nicolas Tremblay, A Dudelzak, S. Babichenko, L Dextraze and J Wollring. 'Nutrient stress of corn plants: Early detection and discrimination using a compact multi-wavelength fluorescent lidar'. In: *EARSEL eProceedings* 1 (Jan. 2001), pp. 214–223. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.484.3413&rep=rep1&type=pdf>.
- [81] James A. Plascyk. 'The MK II Fraunhofer Line Discriminator (FLD-II) for Airborne and Orbital Remote Sensing of Solar-Stimulated Luminescence'. In: *Optical Engineering* 14.4 (1975). DOI: [10.1117/12.7971842](https://doi.org/10.1117/12.7971842).
- [82] I. Moya, L. Camenen, G. Latouche, C. Mauxion, S. Evain and Z. G. Cerovic. 'An Instrument for the Measurement of Sunlight Excited Plant Fluorescence'. In: *Photosynthesis: Mechanisms and Effects: Volume I–V: Proceedings of the XIth International Congress on Photosynthesis, Budapest, Hungary, August 17–22, 1998*. Dordrecht: Springer Netherlands, 1998, pp. 4265–4270. ISBN: 978-94-011-3953-3. DOI: [10.1007/978-94-011-3953-3\\_986](https://doi.org/10.1007/978-94-011-3953-3_986).
- [83] Paul L. Kebabian, Arnold F. Theisen, Spiros Kallelis and Andrew Freedman. 'A passive two-band sensor of sunlight-excited plant fluorescence'. In: *Review of Scientific Instruments* 70.11 (1999), pp. 4386–4393. DOI: [10.1063/1.1150083](https://doi.org/10.1063/1.1150083).
- [84] J.A. Gamon, J. Peñuelas and C.B. Field. 'A narrow-waveband spectral index that tracks diurnal changes in photosynthetic efficiency'. In: *Remote Sensing of Environment* 41.1 (1992), pp. 35–44. ISSN: 0034-4257. DOI: [https://doi.org/10.1016/0034-4257\(92\)90059-S](https://doi.org/10.1016/0034-4257(92)90059-S).
- [85] M. Meroni, M. Rossini, L. Guanter, L. Alonso, U. Rascher, R. Colombo and J. Moreno. 'Remote sensing of solar-induced chlorophyll fluorescence: Review of methods and applications'. In: *Remote Sensing of Environment* 113.10 (2009), pp. 2037–2051. ISSN: 0034-4257. DOI: <https://doi.org/10.1016/j.rse.2009.05.003>.
- [86] Tiit Kutser. 'Quantitative detection of chlorophyll in cyanobacterial blooms by satellite remote sensing'. In: *Limnology and Oceanography* 49.6 (2004), pp. 2179–2189. DOI: <https://doi.org/10.4319/lo.2004.49.6.2179>.

- [87] J. Joiner, Y. Yoshida, A. P. Vasilkov, Y. Yoshida, L. A. Corp and E. M. Middleton. ‘First observations of global and seasonal terrestrial chlorophyll fluorescence from space’. In: *Biogeosciences* 8.3 (2011), pp. 637–651. DOI: [10.5194/bg-8-637-2011](https://doi.org/10.5194/bg-8-637-2011).
- [88] G. Saleem, M. Akhtar, N. Ahmed and W.S. Qureshi. ‘Automated analysis of visual leaf shape features for plant classification’. In: *Computers and Electronics in Agriculture* 157 (2019), pp. 270–280. ISSN: 0168-1699. DOI: <https://doi.org/10.1016/j.compag.2018.12.038>.
- [89] Stephen Gang Wu, Forrest Sheng Bao, Eric You Xu, Yu-Xuan Wang, Yi-Fan Chang and Qiao-Liang Xiang. ‘A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network’. In: *2007 IEEE International Symposium on Signal Processing and Information Technology*. 2007, pp. 11–16. DOI: [10.1109/ISSPIT.2007.4458016](https://doi.org/10.1109/ISSPIT.2007.4458016).
- [90] Esraa Elhariri, Nashwa El-Bendary and Aboul Ella Hassanien. ‘Plant classification system based on leaf features’. In: *2014 9th International Conference on Computer Engineering Systems (ICCES)*. 2014, pp. 271–276. DOI: [10.1109/ICCES.2014.7030971](https://doi.org/10.1109/ICCES.2014.7030971).
- [91] Sue Han Lee, Chee Seng Chan, Paul Wilkin and Paolo Remagnino. ‘Deep-plant: Plant identification with convolutional neural networks’. In: *2015 IEEE International Conference on Image Processing (ICIP)*. 2015, pp. 452–456. DOI: [10.1109/ICIP.2015.7350839](https://doi.org/10.1109/ICIP.2015.7350839).
- [92] Alexis Joly, Hervé Goëau, Christophe Botella, Hervé Glotin, Pierre Bonnet, Willem-Pier Vellinga, Robert Planqué and Henning Müller. ‘Overview of LifeCLEF 2018: A Large-Scale Evaluation of Species Identification and Recommendation Algorithms in the Era of AI’. In: *Experimental IR Meets Multilinguality, Multimodality, and Interaction*. Cham: Springer International Publishing, 2018, pp. 247–266. ISBN: 978-3-319-98932-7.
- [93] Dimitrios Moshou, Els Vrindts, Bart De Ketelaere, Josse De Baerdemaeker and Herman Ramon. ‘A neural network based plant classifier’. In: *Computers and Electronics in Agriculture* 31.1 (2001), pp. 5–16. ISSN: 0168-1699. DOI: [https://doi.org/10.1016/S0168-1699\(00\)00170-8](https://doi.org/10.1016/S0168-1699(00)00170-8).
- [94] Xanthoula-Eirini Pantazi, Dimitrios Moshou and Cedric Bravo. ‘Active learning system for weed species recognition based on hyperspectral sensing’. In: *Biosystems Engineering* 146 (2016). Special Issue: Advances in Robotic Agriculture for Crops, pp. 193–202. ISSN: 1537-5110. DOI: [10.1016/j.biosystemseng.2016.01.014](https://doi.org/10.1016/j.biosystemseng.2016.01.014).
- [95] Saman Akbarzadeh, Arie Paap, Selam Ahderom, Beniamin Apopei and Kamal Alameh. ‘Plant discrimination by Support Vector Machine classifier based on spectral reflectance’. In: *Computers and Electronics in Agriculture* 148 (2018), pp. 250–258. ISSN: 0168-1699. DOI: <https://doi.org/10.1016/j.compag.2018.03.026>.
- [96] Krzysztof Kusnierek and Audun Korsæth. ‘Simultaneous identification of spring wheat nitrogen and water status using visible and near infrared spectra and Powered Partial Least Squares Regression’. In: *Computers and Electronics in Agriculture* 117 (2015), pp. 200–213. ISSN: 0168-1699. DOI: <https://doi.org/10.1016/j.compag.2015.08.001>.

- [97] Gerassimos G. Peteinatos, Audun Korsaeath, Therese W. Berge and Roland Gerhards. ‘Using Optical Sensors to Identify Water Deprivation, Nitrogen Shortage, Weed Presence and Fungal Infection in Wheat’. In: *Agriculture* 6.2 (2016). ISSN: 2077-0472. DOI: [10.3390/agriculture6020024](https://doi.org/10.3390/agriculture6020024).
- [98] Mika Keränen, Eva-Mari Aro, Esa Tyystjärvi and Olli Nevalainen. ‘Automatic plant identification with chlorophyll fluorescence fingerprinting’. In: *Precision agriculture* 4.1 (2003), pp. 53–67. DOI: [10.1023/A:1021863005378](https://doi.org/10.1023/A:1021863005378).
- [99] C.M Codrea, T Aittokallio, M Keränen, E Tyystjärvi and O.S Nevalainen. ‘Feature learning with a genetic algorithm for fluorescence fingerprinting of plant species’. In: *Pattern Recognition Letters* 24.15 (2003), pp. 2663–2673. ISSN: 0167-8655. DOI: [10.1016/S0167-8655\(03\)00109-0](https://doi.org/10.1016/S0167-8655(03)00109-0).
- [100] Malcolm Davidson, Michael Berger, Ismaël Moya, Jose Moreno, Tuomas Laurila, Marc-Phillipe Stoll and John Miller. ‘Mapping photosynthesis from space - A new vegetation-fluorescence technique’. In: *Esa Bulletin-european Space Agency - ESA BULL-EUR SPACE AGENCY* 116 (Nov. 2003), pp. 34–37.
- [101] Kate Maxwell and Giles N. Johnson. ‘Chlorophyll fluorescence—a practical guide’. In: *Journal of Experimental Botany* 51.345 (Apr. 2000), pp. 659–668. ISSN: 0022-0957. DOI: [10.1093/jexbot/51.345.659](https://doi.org/10.1093/jexbot/51.345.659).
- [102] Hazem M Kalaji, Gert Schansker, Marian Brestic, Filippo Bussotti, Angeles Calatayud, Lorenzo Ferroni, Vasilij Goltsev, Lucia Guidi, Anjana Jajoo, Pengmin Li et al. ‘Frequently asked questions about chlorophyll fluorescence, the sequel’. In: *Photosynthesis Research* 132.1 (2017), pp. 13–66. DOI: [10.1007/s11120-016-0318-y](https://doi.org/10.1007/s11120-016-0318-y).
- [103] Masahiko Taniguchi, Hai Du and Jonathan S. Lindsey. ‘PhotochemCAD 3: Diverse Modules for Photophysical Calculations with Multiple Spectral Databases’. In: *Photochemistry and Photobiology* 94.2 (2018), pp. 277–289. DOI: [10.1111/php.12862](https://doi.org/10.1111/php.12862).
- [104] Masahiko Taniguchi and Jonathan S. Lindsey. ‘Database of Absorption and Fluorescence Spectra of >300 Common Compounds for use in PhotochemCAD’. In: *Photochemistry and Photobiology* 94.2 (2018), pp. 290–327. DOI: [10.1111/php.12860](https://doi.org/10.1111/php.12860).
- [105] Fabrice Franck, Philippe Juneau and Radovan Popovic. ‘Resolution of the Photosystem I and Photosystem II contributions to chlorophyll fluorescence of intact leaves at room temperature’. In: *Biochimica et Biophysica Acta (BBA) - Bioenergetics* 1556.2 (2002), pp. 239–246. ISSN: 0005-2728. DOI: [10.1016/S0005-2728\(02\)00366-3](https://doi.org/10.1016/S0005-2728(02)00366-3).
- [106] H Kautsky and A Hirsch. ‘Neue Versuche zur Kohlensäureassimilation’. In: *Die Naturwissenschaften* 19 (1931), pp. 964–964. DOI: [10.1007/BF01516164](https://doi.org/10.1007/BF01516164).
- [107] Joseph R. Lakowicz. ‘Quenching of Fluorescence’. In: *Principles of Fluorescence Spectroscopy*. Boston, MA: Springer US, 1983, pp. 257–301. ISBN: 978-1-4615-7658-7. DOI: [10.1007/978-1-4615-7658-7\\_9](https://doi.org/10.1007/978-1-4615-7658-7_9).

- [108] J. F. Allen, J. A. Raven, A. W. Rutherford and P. Faller. 'Photosystem II: evolutionary perspectives'. In: *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences* 358.1429 (2003), pp. 245–253. DOI: [10.1098/rstb.2002.1186](https://doi.org/10.1098/rstb.2002.1186).
- [109] U Schreiber and Wo Bilger. 'Rapid assessment of stress effects on plant leaves by chlorophyll fluorescence measurements'. In: *Plant response to stress*. Springer, 1987, pp. 27–53. DOI: [10.1007/s00425-011-1366-3](https://doi.org/10.1007/s00425-011-1366-3).
- [110] David Mauzerall. 'Light-Induced Fluorescence Changes in Chlorella, and the Primary Photoreactions for the Production of Oxygen'. In: *Proceedings of the National Academy of Sciences* 69.6 (1972), pp. 1358–1362. ISSN: 0027-8424. DOI: [10.1073/pnas.69.6.1358](https://doi.org/10.1073/pnas.69.6.1358).
- [111] Govindjee Govindjee. 'Sixty-three years since Kautsky: Chlorophyll a fluorescence'. In: *Functional Plant Biology* 22 (Jan. 1995), pp. 131–160. DOI: [10.1071/PP9950131](https://doi.org/10.1071/PP9950131).
- [112] Guido Schmuck and Ismael Moya. 'Time-resolved chlorophyll fluorescence spectra of intact leaves'. In: *Remote Sensing of Environment* 47.1 (1994). Fluorescence Measurements of Vegetation, pp. 72–76. ISSN: 0034-4257. DOI: [10.1016/0034-4257\(94\)90130-9](https://doi.org/10.1016/0034-4257(94)90130-9).
- [113] F Pellegrino, A Dagen, P Sekuler and RR Alfano. 'Temperature dependence of the 735 nm fluorescence kinetics from spinach measured by picosecond laser-streak camera system'. In: *Photobiology and photobiophysics* 6 (1983). ISSN: 0165-8646.
- [114] Mathias Senoner. 'The nanosecond decay of variable chlorophyll fluorescence in leaves of higher plants'. In: *Biochimica et Biophysica Acta (BBA) - Bioenergetics* 849.3 (1986), pp. 374–380. ISSN: 0005-2728. DOI: [10.1016/0005-2728\(86\)90149-0](https://doi.org/10.1016/0005-2728(86)90149-0).
- [115] J. R. Bray, J. E. Sanger and A. L. Archer. 'The Visible Albedo of Surfaces in Central Minnesota'. In: *Ecology* 47.4 (1966), pp. 524–531. DOI: [10.2307/1933928](https://doi.org/10.2307/1933928).
- [116] RN Clark, GA Swayze, R Wise, KE Livo, TM Hoefen, RF Kokaly and SJ Sutley. 'USGS Digital Spectral Library splib05a. USGS Open-File Report 2003-395'. In: *US Geological Survey: Reston, VA, USA* (2003). DOI: [10.3133/ofr03395](https://doi.org/10.3133/ofr03395).
- [117] Yann LeCun, Yoshua Bengio and Geoffrey Hinton. 'Deep learning'. In: *nature* 521.7553 (2015), pp. 436–444. DOI: [10.1038/nature14539](https://doi.org/10.1038/nature14539).
- [118] Shan Huang, Peng Wang, Yubing Tian, Pengli Bai, DaQing Chen, Ce Wang, JianSheng Chen, ZhaoBang Liu, Jian Zheng, WenMing Yao, JianXin Li and Jing Gao. 'Blood species identification based on deep learning analysis of Raman spectra'. In: *Biomed. Opt. Express* 10.12 (Dec. 2019), pp. 6129–6144. DOI: [10.1364/B0E.10.006129](https://doi.org/10.1364/B0E.10.006129).
- [119] Chi-Sing Ho, Neal Jean, Catherine A. Hogan, Lena Blackmon, Stefanie S. Jeffrey, Mark Holodniy, Niaz Banaei, Amr A. E. Saleh, Stefano Ermon and Jennifer Dionne. 'Rapid identification of pathogenic bacteria using Raman spectroscopy and deep learning'. In: *Nature Communications* 10.1 (2019), p. 4927. DOI: [10.1038/s41467-019-12898-9](https://doi.org/10.1038/s41467-019-12898-9).

- [120] Karen Simonyan and Andrew Zisserman. ‘Very Deep Convolutional Networks for Large-Scale Image Recognition’. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 2015. arXiv: 1409.1556.
- [121] Diederik P. Kingma and Jimmy Ba. ‘Adam: A Method for Stochastic Optimization’. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 2015. arXiv: 1412.6980.
- [122] Leo Breiman. ‘Random Forests’. In: *Machine Learning* 45.1 (2001), pp. 5–32. DOI: 10.1023/A:1010933404324.
- [123] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay. ‘Scikit-learn: Machine Learning in Python’. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830. URL: <http://jmlr.org/papers/v12/pedregosa11a.html>.
- [124] Robert Lussier. *Bio-imaging and information system for scanning, detecting, diagnosing and optimizing plant health*. US Patent 7,112,806. Sept. 2006.
- [125] Robert Lussier. *Portable intelligent fluorescence and transmittance imaging spectroscopy system*. US Patent 8,249,308. Aug. 2012.
- [126] **Nils Rottmann**, Ralf Bruder, Achim Schweikard and Elmar Rueckert. ‘A Novel Chlorophyll Fluorescence-Based Approach for Mowing Area Classification’. In: *IEEE Sensors Journal* 21.4 (2021), pp. 4500–4508. DOI: 10.1109/JSEN.2020.3032722. published.
- [127] **Nils Rottmann**, Ralf Bruder, Achim Schweikard and Elmar Rueckert. ‘Exploiting Chlorophyll Fluorescence for building robust low-cost Mowing Area Detectors’. In: *2020 IEEE SENSORS*. 2020, pp. 1–4. DOI: 10.1109/SENSORS47125.2020.9278858. published.
- [128] **Nils Rottmann**, Jannis Hagenah, Jan-Philip Klose, Ralf Bruder, Floris Ernst and Elmar Rueckert. ‘Plant Classification for low-cost Systems based on active Chlorophyll Fluorescence Stimulation: A case-study’. In: *IEEE Sensors Journal* (). under review.
- [129] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler and Andrew Ng. ‘ROS: an open-source Robot Operating System’. In: vol. 3. Jan. 2009. URL: <https://www.ros.org/>.
- [130] Mina Kamel, Thomas Stastny, Kostas Alexis and Roland Siegwart. ‘Model Predictive Control for Trajectory Tracking of Unmanned Aerial Vehicles Using Robot Operating System’. In: *Robot Operating System (ROS): The Complete Reference (Volume 2)*. Ed. by Anis Koubaa. Cham: Springer International Publishing, 2017, pp. 3–39. ISBN: 978-3-319-54927-9. DOI: 10.1007/978-3-319-54927-9\_1.
- [131] João Machado Santos, David Portugal and Rui P. Rocha. ‘An evaluation of 2D SLAM techniques available in Robot Operating System’. In: (2013), pp. 1–6. DOI: 10.1109/SSRR.2013.6719348.

- [132] Joao Paulo de A. Barbosa, Fabio do P. de C. Lima, Leonardo dos S. Coutinho, Joao Paulo R. Rodrigues Leite, Jeremias Barbosa Machado, Carlos Henrique Valerio and Guilherme Sousa Bastos. ‘ROS, Android and cloud robotics: How to make a powerful low cost robot’. In: *2015 International Conference on Advanced Robotics (ICAR)*. 2015, pp. 158–163. DOI: 10.1109/ICAR.2015.7251449.
- [133] Jan Nádvořník and Pavel Smutný. ‘Remote control robot using Android mobile device’. In: *Proceedings of the 2014 15th International Carpathian Control Conference (ICCC)*. 2014, pp. 373–378. DOI: 10.1109/CarpathianCC.2014.6843630.
- [134] Antonio Zea and Uwe D. Hanebeck. ‘iviz: A ROS visualization app for mobile devices’. In: *Software Impacts* 8 (2021), p. 100057. ISSN: 2665-9638. DOI: 10.1016/j.simpa.2021.100057.
- [135] Jason M. O’Kane. *A Gentle Introduction to ROS*. Available at <http://www.cse.sc.edu/~jokane/agitr/>. Independently published, Oct. 2013. ISBN: 978-1492143239.
- [136] Walter L Hürsch and Cristina Videira Lopes. *Separation of concerns*. Tech. rep. 1995. URL: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.125.2723&rep=rep1&type=pdf>.
- [137] **Nils Rottmann**, Nico Studt, Floris Ernst and Elmar Rueckert. ‘ROS-Mobile: An Android application for the Robot Operating System’. In: *SoftwareX* (). arXiv: 2011.02781. under review.
- [138] **Nils Rottmann**, Nico Studt, Floris Ernst and Elmar Rueckert. ‘Introduction to ROS-Mobile: Visualization and Control for the Robot Operating System on Android powered Devices’. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021, pp. 1–5. under review.
- [139] **Nils Rottmann**, Ralf Bruder, Achim Schweikard and Elmar Rueckert. ‘Cataglyphis Ant Navigation Strategies Solve the Global Localization Problem in Robots with Binary Sensors’. In: *Proceedings of the 12th International Joint Conference on Biomedical Engineering Systems and Technologies - Volume 4: BIOSIGNALS*, INSTICC. SciTePress, 2019, pp. 214–223. ISBN: 978-989-758-353-7. DOI: 10.5220/0007556102140223. published.
- [140] **Nils Rottmann**, Tjaša Kunavar, Jan Babič, Jan Peters and Elmar Rueckert. ‘Learning Hierarchical Acquisition Functions for Bayesian Optimization’. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020, pp. 5490–5496. DOI: 10.1109/IR0S45743.2020.9341335. published.
- [141] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams and Nando de Freitas. ‘Taking the Human Out of the Loop: A Review of Bayesian Optimization’. In: *Proceedings of the IEEE* 104.1 (2016), pp. 148–175. DOI: 10.1109/JPR0C.2015.2494218.

- [142] C. K. I. Williams and C. E. Rasmussen. ‘Gaussian processes for regression’. In: *Advances in Neural Information Processing Systems 8*. Ed. by D. S. Touretzky, M. C. Mozer and M. E. Hasselmo. Copyright of the Massachusetts Institute of Technology Press (MIT Press). MIT, June 1996. URL: <http://publications.aston.ac.uk/id/eprint/651/>.
- [143] Carl Edward Rasmussen. ‘Gaussian Processes in Machine Learning’. In: *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003, Revised Lectures*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 63–71. ISBN: 978-3-540-28650-9. DOI: 10.1007/978-3-540-28650-9\_4.
- [144] Duy Nguyen-Tuong, Matthias Seeger and Jan Peters. ‘Model Learning with Local Gaussian Process Regression’. In: *Advanced Robotics* 23.15 (2009), pp. 2015–2034. DOI: 10.1163/016918609X12529286896877.
- [145] Roberto Calandra, Serena Ivaldi, Marc Peter Deisenroth, Elmar Rueckert and Jan Peters. ‘Learning inverse dynamics models with contacts’. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 3186–3191. DOI: 10.1109/ICRA.2015.7139638.
- [146] Duy Nguyen-Tuong and Jan Peters. ‘Model learning for robot control: a survey’. In: *Cognitive processing* 12.4 (2011), pp. 319–340. DOI: 10.1007/s10339-011-0404-1.

# Nils Rottmann

## Curriculum Vitae

Brandenbaumer Landstrasse 220  
23564 Luebeck, Schleswig-Holstein

+49 178 3196518

✉ Rottmann1990@gmail.com

📄 nrothmann.github.io

🌐 NRottmann

🌐 NRottmann



### Current Position

since 01/2018 **Research Scientist and Ph.D. candidate for Autonomous and Intelligent Systems, University of Luebeck, Germany.**  
Dynamics & Control, Path & Motion Planning, Machine Learning.

### Research Interests

Robotics Robot Dynamics, Trajectory Generation, Motion Planning, Nonlinear and Intelligent Control, Real-Time Planning and Control, SLAM, Pose Graph Optimization.

Robot Learning Reinforcement Learning, Imitation Learning, Grasping, Stochastic Optimal Control, Dynamical Movement Primitives.

Sensors Optical Sensors, Tactile Sensors, Data Evaluation, Sensor Fusion.

### Education

since 01/2018 **Ph.D. candidate in Robotics, University of Luebeck, Germany.**  
Smart Sensor, Navigation and Learning Strategies for low-cost Lawn Care Systems.

2015/04 – **M.Sc., Theoretical Mechanical Engineering, Uni. of Technology Hamburg, Germany, Grade 1.2.**  
2017/12 Focus on Intelligent and Autonomous Systems.

2015/09 – **Exchange Semester, Instituto Superior Tecnico Lisbon, Portugal.**  
2016/02 Focus on Intelligent and Autonomous Systems.

2011/10 – **B.Sc., Mechanical Engineering, Uni. of Technology Hamburg, Germany, Grade 1.5.**  
2015/03 Focus on Theoretical Mechanical Engineering.

2003/07 – **High school, Hoya, Germany, Grade 1.6.**  
2010/06 Completion with the general university entrance.

### Academic Experience

since 2018/01 **Research Scientist for Robotics, University of Luebeck, Germany.**  
Dynamics & Control, Path & Motion Planning, Machine Learning.

since 2020/02 **Project Lead ROS-Mobile, github.com/ROS-Mobile, Luebeck, Germany.**  
Development of a mobile application for the Robot Operating System (ROS).

since 2020/08 **Project Member TRAIN, train.ai-lab.science, Luebeck, Germany.**  
Robot skill learning with human feedback.

since 2018/10 **Project Member A-DRZ, Dortmund, Germany.**  
Integration of life sign modules and ASVs into a ROS system.

- 2020/07 **Summer School**, *Max Planck Institute for Intelligent Systems, Tuebingen, Germany.*  
Machine Learning Summer School.
- 2019/07 **Summer School**, *Frankfurt Institute for Advanced Studies, Germany.*  
Intrinsically Motivated Open-Ended Learning Summer School.
- since 2019/10 **Teaching Assistant**, *University of Luebeck, Germany.*  
Probabilistic Machine Learning (RO5101).
- since 2018/04 **Teaching Assistant**, *University of Luebeck, Germany.*  
Humanoid Robotics (RO5300).
- 2020/04 – **Teaching Assistant**, *University of Luebeck, Germany.*  
2020/09 Rescue Robotics (RO5801).
- 2014/08 – **Internship**, *UNESP, Ilha Solteira, Brazil.*  
2014/09 Finite Element Methods for Baja SAE outdoor vehicle.

## Industrial Experience

- 2018/09 – **Project Lead MIRANA**, *mirana.rob.uni-luebeck.de, Luebeck/Kiel, Germany.*  
2020/08 Development of an intelligent mobile system for patient guidance in cooperation with the UKSH.
- 2018/01 – **Collaboration with STIHL**, *Waiblingen, Germany.*  
2020/03 Development of low range sensors and algorithms for mobile outdoor navigation.
- 2014/11 – **Internship**, *Drägerwerk AG & Co. KGaA, Luebeck, Germany.*  
2015/03 Advanced Engineering Solutions Section, Thermodynamic studies of volume flows containing anesthetic gases.

## Miscellaneous Experience

- 2016/02- **Local CFO**, *bonding student initiative, Hamburg, Germany.*  
17/02 Administration of the finances of the association.
- 2010/07- **Civilian Service**, *Diakonie Sozialstation Hoya, Germany.*  
2011/03 Care of elderly.

## Publications

### Conference Publications

- [1] **Nils Rottmann**, Ralf Bruder, Achim Schweikard and Elmar Rueckert. Exploiting Chlorophyll Fluorescence for building robust low-cost Mowing Area Detectors. 2020, IEEE Sensors, Rotterdam. <https://ieeexplore.ieee.org/document/9278858>
- [2] **Nils Rottmann**, Tjasa Kunavar, Jan Babic, Jan Peters and Elmar Rueckert. Learning Hierarchical Acquisition Functions for Bayesian Optimization. 2020, IROS, Las Vegas. <https://ieeexplore.ieee.org/document/9341335>
- [3] Honghu Xue, Sven Boettger, **Nils Rottmann**, Harit Pandya, Ralf Bruder, Gerhard Neumann and Elmar Rueckert. Sample-Efficient Covariance Matrix Adaptation Evolutional Strategy via Simulated Rollouts in Neural Networks (ASPAI), 2020, Berlin, Germany. <https://ai-lab.science/wp/ASPAI2020Xue.pdf>
- [4] **Nils Rottmann**, Ralf Bruder, Achim Schweikard and Elmar Rueckert. Loop Closure Detection in Closed Environments. *European Conference on Mobile Robots (ECMR), 2019, Prague, Czech Republic.* <https://ieeexplore.ieee.org/abstract/document/8870938>

- [5] **Nils Rottmann**, Ralf Bruder, Achim Schweikard and Elmar Rueckert. Cataglyphis Ant Navigation Strategies Solve the Global Localization Problem in Robots with Binary Sensors. *Proceedings of the 12th International Joint Conference on Biomedical Engineering Systems and Technologies - Volume 5: BIOSIGNALS, 2019, Prague, Czech Republic*. <http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0007556102140223>

#### Conference Under Review

- [6] **Nils Rottmann**, Nico Studt, Floris Ernst and Elmar Rueckert. Introduction to ROS-Mobile: Visualization and Control for the Robot Operating System on Android powered Devices. 2021, IROS.
- [7] **Nils Rottmann**, Robin Denz, Ralf Bruder and Elmar Rueckert. Probabilistic Complete Coverage Path Planning for low-cost Systems. 2021, ECMR.
- [8] Robin Denz, Rabia Demirci, Mehmet Ege Cansev, Adna Bliet, Philipp Beckerle, Elmar Rueckert and **Nils Rottmann**. A high-accuracy, low-budget Sensor Glove for Trajectory Model Learning. 2021, IROS.

#### Journal Publications

- [9] **Nils Rottmann** Ralf Bruder, Achim Schweikard and Elmar Rueckert. A Novel Chlorophyll Fluorescence-Based Approach for Mowing Area Classification. 2021, IEEE Sensors Journal. <https://ieeexplore.ieee.org/document/9234496>
- [10] Mehmet Ege Cansev, Honghu Xue, **Nils Rottmann**, Adna Bliet, Luke E. Miller, Elmar Rueckert and Philipp Beckerle. Interactive Human-Robot Skill Transfer: A Review of Learning Methods and User Experience. 2021, Advanced Intelligent Systems.

#### Journal Publications Under Review

- [11] **Nils Rottmann**, Nico Studt, Floris Ernst and Elmar Rueckert. ROS-Mobile: An Android application for the Robot Operating System. SoftwareX.
- [12] **Nils Rottmann**, Jannis Hagenah, Jan-Philip Klose, Ralf Bruder, Floris Ernst and Elmar Rueckert. Plant Classification for low-cost Systems based on active Chlorophyll Fluorescence Stimulation: A case-study. IEEE Sensors Journal.

#### Workshop Publications

- [13] **Nils Rottmann**, Ralf Bruder, Achim Schweikard and Elmar Rueckert. Parameter Optimization for Loop Closure Detection in Closed Environments. 12th IROS Workshop on Planning, Perception, Navigation for Intelligent Vehicle, IROS 2020. <https://ai-lab.science/wp/IROSW2020Rottmann.pdf>.

#### Theses

- [14] **Nils Rottmann**. Geometric Control and Stochastic Trajectory Planning for Underwater Robotic Systems. *Master Thesis*, 2017, Technische Universität Hamburg–Harburg
- [15] **Nils Rottmann**. Development of a diving cell for experimental detection of orbital motion of water particles under the influence of gravity waves. *Bachelor Thesis*, 2014, Technische Universität Hamburg–Harburg

---

## Student Supervision

### Assistants (typ. empl. 40h/month)

- since 2020/04 Student teaching assistant for Humanoid Robotics and Probabilistic Machine Learning.
- since 2020/02 Student assistant for the ROS-Mobile project.
- since 2020/10 Student assistant for the TRAIN project.

- 2019/06 – Student assistant for the MIRANA project.  
2020/01
- 2019/10 – Student assistants for the "Our common future" project to teach pupils the basics for  
2021/01 mobile robot development.

### M.Sc. Projects

- 2020/10 – Development of a sensor glove for robot learning applications for the TRAIN project,  
2021/03 University of Luebeck.
- 2020/10 – Further development of a mobile application for the Robot Operating System (ROS),  
2021/03 University of Luebeck.

### B.Sc. Projects

- 2018/09 – Simultaneous Localization and Mapping, Speech Recognition and Keyword Detection for  
2019/03 Service Robots, University of Luebeck.

### M.Sc. Theses

- 2019/12 HIBO: Hierarchical Acquisition Functions for Bayesian Optimization, University of Luebeck.
- 2018/09 Simultaneous Localization and Mapping for Autonomous Lawn Mowers, University of  
Luebeck.
- 2018/10 Optimization of a Chlorophyll-Sensor for Mowing-Area-Detection for Autonomous Lawn  
Mowers, University of Luebeck.

### B.Sc. Theses

- 2020/08 Plant Classification based on Chlorophyll Detection for autonomous Gardening, University  
of Luebeck.
- 2019/11 Simultaneous Localization and Mapping with Room Labeling, University of Luebeck.
- 2019/11 Complete Coverage Path Planning for Low Cost Robots, University of Luebeck.
- 2019/06 Machine Learning for Plant Classification based on Chlorophyll Detection, University of  
Luebeck.
- 2018/11 Trajectory Planning for Mobile Robots for Working Area Complete Coverage under High  
Uncertainty, University of Luebeck.
- 2018/09 Development of an electromagnetic Tracking System for use in Medical Interventions,  
University of Luebeck.
- 2018/09 Localization and Control for Trajectory Tracking for Autonomous Lawn Mowers, University  
of Luebeck.

---

## Talks

- 2020/10 Parameter Optimization for Loop Closure Detection in Closed Environments. **Workshop  
Talk.** *12th IROS Workshop on Planning, Perception, Navigation for Intelligent Vehicle,  
IROS 2020, Las Vegas, USA (virtual).*
- 2020/10 Learning Hierarchical Acquisition Functions for Bayesian Optimization. **Conference Talk.**  
*IROS 2020, Las Vegas, USA (virtual).*
- 2020/10 Exploiting Chlorophyll Fluorescence for building robust low-cost Mowing Area Detectors.  
**Conference Talk.** *IEEE Sensors 2020, Rotterdam, Netherlands (virtual).*
- 2019/11 MIRANA: A Mobile Intelligent Robotic Agent for Navigation and Assistance. **Invited Talk.**  
*At the Nook 2019, Luebeck, Germany.*
- 2019/03 Think at IBM – Introduction to the MIRANA project. **Invited Talk.** *Berlin, Germany.*

2018/07 Künstliche Intelligenz – Wie Roboter lernen können. **Invited Talk.** *At the Informatik Summer Camp 2018, Luebeck, Germany.*

---

## Contributor to Research Proposals

- 2019/11 **DFG Sachbeihilfe (~322.622 EURO of 653.663 EURO)**, Granted Nov. 22nd 2019, joint proposal of Prof. Rueckert and Prof. Philipp Beckerle: Active transfer learning with neural networks through human-robot interactions, One Ph.D. position (3 years each) and one student assistant (3 years).
- 2019/04 **Robert-Bosch Stiftung: Our Common future (~25k EURO)**, proposal of Prof. Rueckert, Autonome Elektrofahrzeuge als urbane Lieferanten, future.ai-lab.science.
- 2019/03 **Verbundforschung UL UKSH ARGE (~72k EURO)**, Initiative Service Roboter im Gesundheitswesen.
- 2019/01 **Industrial collaboration with STIHL (~120k EURO)**, Development of low range sensors and algorithms for mobile outdoor navigation.

---

## Reviewing Experience

- 2021 International Conference on Robotics and Automation (ICRA).
- 2021 IEEE Robotics and Automation Letters (RA-L).
- 2020/21 International Conference on Intelligent Robots and Systems (IROS).
- 2020 Robotics and Autonomous Systems, Journal (RAS).
- 2020 Automatisierungstechnik Journal (at).
- 2019 International Conference on Machine Learning (ICML).
- 2019 European Conference on Mobile Robots (ECMR).
- 2018 Conference on Robot Learning (CoRL).

---

## Outreach Activities

- 2019/10, **Co-Organizer**, *1st and 2nd LEGO Robotic Workshop: Autonome Elektrofahrzeuge als urbane Lieferanten*, future.ai-lab.science.
- 2020/10 **Co-Organizer**, *Laboratory tour for pupils and lab demos.*

---

## Skills & Software Projects

### Software Projects

ROS-Mobile: An android application for the Robot Operating System, available since 2020/06. Currently 949 people use the application regularly and in total 2734 downloads have been counted. Written in JAVA.

### Programming and Software Skills

L<sup>A</sup>T<sub>E</sub>X, Robot Operating System (ROS), Gazebo, C/C++, JAVA, Matlab, PYTHON, V-REP, Computer-Aided Design (CAD), Finite-Element-Methods (FEM), Git, Wordpress.

### Languages

German (Mothertongue), English (Fluent), Spanish (Intermediate), Portuguese (Basics).

### Hobbies

Cycling, Swimming, Dancing (Sala, Bachata), Yoga, Computer Games.

