Interner Bericht

DLR-IB-FT-BS-2022-200

Simulation und Evaluierung von Pfadplanungsalgorithmen für den Flug von unbemannten Luftfahrzeugen in Innenräumen

Hochschulschrift

Winiecki, Joel

Deutsches Zentrum für Luft- und Raumfahrt

Institut für Flugsystemtechnik
Braunschweig

Deutsches Zentrum
DLR für Luft- und Raumfahrt

Institutsbericht
DLR-IB-FT-BS-2022-200

Simulation und Evaluierung von Pfadplanungsalgorithmen für den Flug von unbemannten Luftfahrzeugen in Innenräumen

Winiecki, Joel

Institut für Flugsystemtechnik
Braunschweig

106 Seiten
032 Abbildungen
009 Tabellen
054 Referenzen

Deutsches Zentrum für Luft- und Raumfahrt e.V.
Institut für Flugsystemtechnik
Abteilung Unbemannte Luftfahrzeuge

Stufe der Zugänglichkeit: I, Allgemein zugänglich: Der Interne Bericht wird elektronisch ohne Einschränkungen in ELIB abgelegt. Falls vorhanden, ist je ein gedrucktes Exemplar an die zuständige Standortbibliothek und an das zentrale Archiv abzugeben.

Braunschweig, den 02.02.2024

[redacted]

Institutsdirektor: Prof. Dr.-Ing. S. Levedag

Abteilungsleitung: Johann Dauer

Betreuer:in: Nikolaus Ammann

Verfasser:in: Joel Winiecki

Simulation und Evaluierung von Pfadplanungsalgorithmen für den Flug von unbemannten Luftfahrzeugen in Innenräumen

Bachelorarbeit

für die Prüfung zum

Bachelor of Science

des Studiengangs Informationstechnik

an der Dualen Hochschule Baden-Württemberg Mannheim

von

Joel Winiecki

30. August 2022

Bearbeitungszeitraum	07.06.22 - 30.08.22
Matrikelnummer, Kurs	1123227, TINF19IT1
Ausbildungsfirma	Deutsches Zentrum für Luft- und Raumfahrt e.V. , Braunschweig
Betrieblicher Betreuer	Nikolaus Alexander Ammann
Gutachter der Dualen Hochschule	Prof. Holger Gerhards

Erklärung

Ich versichere hiermit, dass ich meine Bachelorarbeit mit dem Thema: *Simulation und Evaluierung von Pfadplanungsalgorithmen für den Flug von unbemannten Luftfahrzeugen in Innenräumen* selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Braunschweig, 30. August 2022

Joel Winiecki

Kurzfassung

Es gibt viele Pfadplanungsalgorithmen für den Flug von unbemannten Luftfahrzeugen. Bei der Entwicklung von derartigen Algorithmen wird der Einsatz in Innenräumen häufig nicht berücksichtigt. Dadurch wird die Technologie auf große Freiräume mit wenigen Hindernissen begrenzt. Um das Anwendungsspektrum auf hindernisreiche Innenräume zu erweitern, wird die Leistung von Pfadplanungsalgorithmen in derartigen Umgebungen in der vorliegenden Arbeit evaluiert. Hierfür wird eine Metrik bestehend aus acht Kriterien entwickelt und an vier Pfadplanungsalgorithmen angewandt. Die einzelnen Kriterien der Metrik werden anhand von realitätsnahen Simulationen bewertet. Basierend auf einer gewichteten Auswertung der Evaluation wird der Pfadplanungsalgorithmus mit der besten Eignung für den Einsatz in Innenräumen identifiziert.

Abstract

There are many path planning algorithms for the flight of unmanned aerial vehicles. During the development of such algorithms, indoor use is often not considered. Therefore, the technology is limited to large open spaces with few obstacles. To extend the range of applications to obstacle-rich indoor environments, the performance of path planning algorithms in such environments is evaluated in this thesis. For this purpose, a metric consisting of eight criteria is developed and applied to four path planning algorithms. The individual criteria of the metric are evaluated using realistic simulations. Based on a weighted assessment of the evaluation, the path planning algorithm with the best suitability for indoor use is identified.

Inhaltsverzeichnis

Abkürzungsverzeichnis **VI**

Abbildungsverzeichnis **VII**

Tabellenverzeichnis **X**

Listings **XI**

1 Einleitung **1**
1.1 Motivation 2
1.2 Ziele und Anforderungen 3
1.3 Struktur der Arbeit 5
1.4 Anmerkungen zum Lesen 6

2 Theoretische Grundlagen **7**
2.1 Terminologie und Definitionen 7
2.2 Unterscheidung von Pfadplanungsalgorithmen 9
2.2.1 Heuristik-basierte Suchalgorithmen 10
2.2.2 Stichproben-basierte Suchalgorithmen 10
2.2.3 Suchalgorithmen basierend auf der Potentialfeldmethode 11
2.2.4 Von der Biologie inspirierte Suchalgorithmen 11
2.2.5 Fusion von Suchalgorithmen 12
2.3 Forschungsarbeiten im Bereich der Pfadplanung in Innenräumen für **U**nmanned **A**erial **S**ystem (UAS) 13
2.3.1 GeoSOT-basierte Pfadplanungsalgorithmus 13
2.3.2 Pfadplanung mittels maschinellem Lernen 14
2.3.3 Fusionsbasierte Ansatz für Pfadplanung 15
2.3.4 Bedeutung für die vorliegende Arbeit 16
2.4 Evaluation von Pfadplanungsalgorithmen 18

3 Technische Grundlagen **20**
3.1 Übersicht verwendeter Hardwarekomponenten 20

3.2 UAV-Kommunikation und Steuerung innerhalb Simulationen 20
3.2.1 Gazebo-Simulator 22
3.2.2 QGroundControl-Bodenstation 23
3.2.3 **R**obot **O**perating **S**ystem (ROS)-**A**pplication **P**rogramming **I**nterface (API) 24
3.2.4 Kommunikationsprotokolle 25
3.2.5 Übersicht verwendeter Softwarekomponenten 26
3.3 Behandelte Pfadplaner 26
3.3.1 Grundlagen der PX4-Avoidance-Pfadplaner 27
3.3.2 Grundlagen des CERBERUS-Pfadplaners 29
3.3.3 Grundlagen des MB-Pfadplaners 31

4 Umsetzung **34**
4.1 Auswahl, Analyse und Modifikation quelloffener Implementierungen von Pfadplanern 34
4.1.1 Auswahl quelloffener Implementierungen von Pfadplanern 35
4.1.2 Funktionalitäten und Schnittstellen des lokalen PX4-Pfadplaners . . 37
4.1.3 Funktionalitäten und Schnittstellen des globalen PX4-Pfadplaners . 40
4.1.4 Funktionalitäten und Schnittstellen des explorierenden CERBERUS-Pfadplaners 42
4.1.5 Funktionalitäten und Schnittstellen des explorierenden MB-Pfadplaners 44
4.1.6 Zusammenfassung der ermittelten Gemeinsamkeiten und Unterschiede der untersuchten Pfadplaner 45
4.2 Kriterien der Metrik für die Bewertung von Pfadplanungsalgorithmen . . . 47
4.2.1 Überlegungen bezüglich der Erstellung einer Metrik 47
4.2.2 Entwurf der Kriterien 49
4.3 Konzeptionierung der Methoden für die Erfassung von Kriterien 56
4.3.1 Erfassung des diskreten Kriteriums „Höhenlimitierung“ 57
4.3.2 Erfassung des diskreten Kriteriums „**dreid**imensional (3D)-Pfadplaner“ 57
4.3.3 Erfassung des diskreten Kriteriums „Kontinuierlicher Pfadplaner“ . 58
4.3.4 Erfassung des diskreten Kriteriums „Zielpositionsvorgabe“ 59
4.3.5 Erfassung des stetigen Kriteriums „Zeitkosten“ 59
4.3.6 Erfassung des stetigen Kriteriums „Minimaler Freiraum“ 60
4.3.7 Erfassung des stetigen Kriteriums „Durchschnittliche Prozessorauslastung“ 61
4.3.8 Erfassung des stetigen Kriteriums „Maximaler Speicheraufwand“ . . 63
4.4 Evaluation der Pfadplaner 64
4.4.1 Evaluation des Kriteriums „Höhenlimitierung“ 65
4.4.2 Evaluation des Kriteriums „3D-Pfadplaner“ 65

4.4.3 Evaluation des Kriteriums „Kontinuierlicher Pfadplaner“ 66
4.4.4 Evaluation des Kriteriums „Zielpositionsvorgabe“ 67
4.4.5 Evaluation des Kriteriums „Zeitkosten“ 67
4.4.6 Evaluation des Kriteriums „Minimaler Freiraum“ 68
4.4.7 Evaluation des Kriteriums „Durchschnittliche Prozessorauslastung“ 69
4.4.8 Evaluation des Kriteriums „Maximaler Speicheraufwand“ 73
4.4.9 Beobachtungen bei der Evaluation 77
4.5 Anwendungsspezifische Auswertung der Evaluation 77

5 Ergebnisse und Diskussion **81**
5.1 Zusammenfassung anhand der Anforderungstabelle 82
5.2 Diskussion der Ergebnisse 83
5.2.1 Vergleichbarkeit und Vollständigkeit der Kriterien 84
5.2.2 Erweiterbarkeit der Untersuchungen mit neuen Daten 85
5.2.3 Aussagekraft der Kriterien unter Berücksichtigung des Wertebereiches 85
5.2.4 Übertragbarkeit auf die Realität 86

6 Ausblick **87**

Danksagung **I**

Literatur **II**

Abkürzungsverzeichnis

UAV	**U**nmanned **A**erial **V**ehicle
UAS	**U**nmanned **A**erial **S**ystem
API	**A**pplication **P**rogramming **I**nterface
LiDAR	**Li**ght **D**etection **A**nd **R**anging
ROS	**R**obot **O**perating **S**ystem
FoV	**F**ield **o**f **V**iew
2D	**zweid**imensional
3D	**dreid**imensional
GUI	**G**raphical **U**ser **I**nterface
ML	**M**achine **L**earning
GNC	**G**uidance, **N**avigation and **Control**
DARPA	**D**efense **A**dvanced **R**esearch **P**rojects **A**gency
IMU	**I**nertia **M**easurement **U**nit
SLAM	**S**imultaneous **L**ocalization **a**nd **M**apping
PRM	**Pr**obabilistic **R**oadmap
RRT	**R**apidly exploring **R**andom **T**ree
APF	**A**rtificial **P**otential **F**ield
VFH	**V**ector **F**ield **H**istogram
GeoSOT	**Geo**graphic Coordinate Subdividing Grid with **O**ne Dimension Integral Coding on 2^n **T**ree
SA	**S**imulated **A**nnealing
A*	**A star**
BiA*	**Bi**directional **A star**
BiAPF	Bidirectional **A**rtificial **P**otential **F**ield
BiRRT	**Bi**directional **R**apidly exploring **R**andom **T**ree
RRG	**R**apidly Exploring **R**andom **G**raphs
SAR	**S**earch **a**nd **R**escue
CPU	**C**entral **P**rocessing **U**nit
GPU	**G**raphics **P**rocessing **U**nit
RAM	**R**andom **A**ccess **M**emory
SITL	**S**oftware **I**n **T**he **L**oop
TCP	**T**ransmission **C**ontrol **P**rotocol
UDP	**U**ser **D**atagram **P**rotocol
MAVLink	**M**icro **A**ir **V**ehicle **Link**
RTF	**R**eal **T**ime **F**actor

Abbildungsverzeichnis

2.1 Ausschnitt der Veranschaulichung von dem *world space* entnommen aus Goerzen, C. et al. [1] 8
2.2 Veranschaulichung von einer von Han, B. et al. [2] genutzten Umgebung für die Erprobung des vorgestellten Pfadplanungsalgorithmus 14
2.3 Ausschnitt der Veranschaulichung von den Umgebungen für die Simulationen von Park, J. et al. [3]. Auf der linken Seite ist die Umgebung ohne Hindernisse und auf der rechten Seite die Umgebung mit Hindernissen dargestellt. . . 15
2.4 Ausschnitt der Veranschaulichung von der von Rocha, L. et al. [4] genutzten Innenraumumgebung. In der Mitte der Abbildung befindet sich das **U**nmanned **A**erial **V**ehicle (UAV). 16

3.1 Übersicht der Kommunikation der **S**oftware **I**n **T**he **L**oop (SITL) Simulationsumgebung [5] 21
3.2 Gazebo-Simulation während der Verwendung eines Pfadplaners 22
3.3 Gazebo Building-Editor bei der Erstellung einer Testumgebung 23
3.4 Übersicht QGroundControl während aktiver Simulation 24
3.5 Übersicht von RViz während aktiver Simulation. Das Luftfahrzeug wird nicht visualisiert. 25
3.6 Darstellung des Iris-3Cam-UAS in Gazebo. Das Luftfahrzeug ist von der vorderen Seite zu sehen. Es sind drei Stereokameras unter den vorderen Armen des Luftfahrzeugs angebracht. 28
3.7 Ausschnitt der Visualisierung mittels RViz des lokalen Planers aus dem PX4-Avoidance Paket. Die Umgebung und Sensorwahrnehmung des Iris-3Cam-UAS werden dargestellt. Die Drohne schwebt in der Nähe von Bäumen. Die gelbe Sphäre stellt die Zielposition dar. 28
3.8 Ausschnitt der Visualisierung mittels RViz des globalen Planers aus dem PX4-Avoidance Paket. Die Drohne schwebt vor zwei Bäumen. Die pinke Sphäre stellt die Zielposition dar. Die Iris-Drohne befindet sich auf der Zielposition. Der grüne Pfad stellt den geflogenen Pfad bis zu der Ist-Position dar. Die schwarzen Blöcke stellen die OctoMap dar. 29

3.9 Ausschnitt einer in Gazebo geöffneten Welt. Diese Umgebung wurde bei der **D**efense **A**dvanced **R**esearch **P**rojects **A**gency (DARPA)-*Subterranean Challenge* für das Testen von Algorithmen verwendet [6]. Zu sehen ist ein Überblick eines Tunnelsystems. 30
3.10 Ausschnitt aus Gazebo bei aktivem CERBERUS-Pfadplaner. In der Mitte der Abbildung ist das UAS sichtbar. Die Umgebung beinhaltet verschiedene Hindernisse. 31
3.11 Ausschnitt aus RViz bei aktivem CERBERUS-Pfadplaner. Sichtbar sind erkannte Hindernisse, welche als farbige Quadrate in RViz dargestellt werden. Die einzelnen Tunnel in des Tunnelsystems sind sichtbar. 32
3.12 Ausschnitt aus Gazebo bei aktivem MB-Pfadplaner. In der Mitte der Abbildung ist das UAS sichtbar. Die Umgebung besteht aus einem Boden und mehreren Wänden. 32
3.13 Ausschnitt aus RViz bei aktivem MB-Pfadplaner. Sichtbar sind erkannte Hindernisse, welche farbig dargestellt werden. Das UAV wird mit drei Koordinatenachsen sowie durch ein Quader dargestellt. Der geflogene Pfad wird mithilfe von orange-farbigen Zylindern visualisiert. 33

4.1 Ausschnitt des lokalen PX4-Planers in RViz. Das UAS führte eine kreisende Bewegung über der gelb markierten Zielposition aus. Die verfolgte Pfad ist grün dargestellt und der geplante Pfad ist rot dargestellt. 38
4.2 Darstellung des Iris-**Li**ght **D**etection **A**nd **R**anging (LiDAR) Luftfahrzeuges. Der Zylinder stellt den LiDAR-Sensor dar. 39
4.3 Darstellung der relevanten Strecken des Kriteriums „Minimaler Freiraum“ am Beispiel des Luftfahrzeugs Iris-LiDAR in einer Gazebo-Simulation. Der Durchgang zwischen zwei Wänden wird mit der Variable $S_{passierbar}$ angegeben. Die Breite des Luftfahrzeugs wird als $S_{BreiteVehikel}$ angegeben und wird durch die Kollisionsbox des Luftfahrzeugs verdeutlicht. 54
4.4 Genutzte Gazebo-Umgebung für die Erfassung des Kriteriums „3D-Pfadplaner“. Die Fläche des Grundrisses ist $10m \cdot 5m$. Die vorderen Wände sind transparent, um die Visualisierung des Innenraumes zu ermöglichen. Dies hat keine Auswirkungen auf die Simulation. 58
4.5 Genutzte Gazebo-Umgebung für die Erfassung des Kriteriums „Zeitkosten“. Die schwarzen Pfeile zeigen den Pfad des Luftfahrzeugs durch das Erdgeschoss. Diese befinden sich immer im Sichtbereich des UAS. Der lila Pfeil zeigt den direkten Pfad durch die bereits bekannte Umgebung. Die blauen Pfeile zeigen den aufsteigenden Pfad entlang der Treppe an. Der grüne Pfeil zeigt den Pfad im ersten Obergeschoss an. Die Decke ist aus Gründen der Visualisierung transparent. 60

4.6 Genutzte Gazebo-Umgebung für die Erfassung des Kriteriums „Minimaler Freiraum“. Der schwarze Pfeil zeigt den Pfad, der durch das Luftfahrzeug passiert werden muss, um das Kriterium zu erfüllen. Die Verengung hat eine Breite von $60cm$. Die Decke ist aus Gründen der Visualisierung transparent. 61
4.7 Genutzte Gazebo-Umgebung für die Evaluation des Kriteriums „Durchschnittliche Prozessorauslastung“. Die Pfeile geben den ungefähren antizipierten Pfad, der von dem jeweiligen Luftfahrzeug verfolgt werden soll, an. 63
4.8 Die Prozessorauslastung bei der Ausführung des lokalen PX4-Pfadplaners in der in Abbildung 4.7 dargestellten Gazebo-Umgebung. Die gestrichelte Linie visualisiert die durchschnittliche Prozessorauslastung. 71
4.9 Die Prozessorauslastung bei der Ausführung des globalen PX4-Pfadplaners in der in Abbildung 4.7 dargestellten Gazebo-Umgebung. Die gestrichelte Linie visualisiert die durchschnittliche Prozessorauslastung. 71
4.10 Die Prozessorauslastung bei der Ausführung des globalen CERBERUS-Pfadplaners in der in Abbildung 4.7 dargestellten Gazebo-Umgebung. Die gestrichelte Linie visualisiert die durchschnittliche Prozessorauslastung. . . 72
4.11 Die Prozessorauslastung bei der Ausführung des globalen MB-Pfadplaners in der in Abbildung 4.7 dargestellten Gazebo-Umgebung. Die gestrichelte Linie visualisiert die durchschnittliche Prozessorauslastung. 72
4.12 Der genutzte Arbeitsspeicher bei der Ausführung des lokalen PX4-Pfadplaners auf der in Abbildung 4.7 dargestellten Gazebo-Umgebung. Die gestrichelte Linie visualisiert den maximal genutzten Arbeitsspeicher. 75
4.13 Der genutzte Arbeitsspeicher bei der Ausführung des globalen PX4-Pfadplaners auf der in Abbildung 4.7 dargestellten Gazebo-Umgebung. Die gestrichelte Linie visualisiert den maximal genutzten Arbeitsspeicher. 75
4.14 Der genutzte Arbeitsspeicher bei der Ausführung des globalen CERBERUS-Pfadplaners auf der in Abbildung 4.7 dargestellten Gazebo-Umgebung. Die gestrichelte Linie visualisiert den maximal genutzten Arbeitsspeicher. . . . 76
4.15 Der genutzte Arbeitsspeicher bei der Ausführung des globalen MB-Pfadplaners auf der in Abbildung 4.7 dargestellten Gazebo-Umgebung. Die gestrichelte Linie visualisiert den maximal genutzten Arbeitsspeicher. 76

Tabellenverzeichnis

3.1 Übersicht verwendeter Computerkomponenten für die Simulationen der vorliegenden Arbeit 21
3.2 Übersicht verwendeter Softwarekomponenten für die Simulationen der behandelten Pfadplaner 26

4.1 Übersicht von funktionalen Merkmalen von den untersuchten Pfadplanern. Die Namen der Pfadplaner sind aus räumlichen Gründen verkürzt: lokaler PX4-Pfadplaner (lok. PX4), globaler PX4-Pfadplaner (glob. PX4), CERBERUS-Pfadplaner (CERBERUS) und MB-Pfadplaner (MB). Vorhandene Merkmale werden mit dem Symbol ✓ gekennzeichnet, wenn der entsprechende Pfadplaner dieses Merkmal erfüllt oder mit ✗ gekennzeichnet, wenn der entsprechende Pfadplaner das Merkmal nicht erfüllt. 45
4.2 Zusammenfassung von Kriterien für die Bewertung von Pfadplanern. Bei allen Kriterien entspricht der Wert 1 der besten Bewertung. 56
4.3 Zusammenfassung der Ergebnisse der Evaluation von allen untersuchten Pfadplaner. Die Namen der Pfadplaner sind aus räumlichen Gründen verkürzt: lokaler PX4-Pfadplaner (lok. PX4), globaler PX4-Pfadplaner (glob. PX4), CERBERUS-Pfadplaner (CERBERUS) und MB-Pfadplaner (MB). . 64
4.4 Übersicht der beobachteten Probleme der Pfadplaner für den Einsatz in Innenräumen. 79
4.5 Zusammenfassung der gewichteten Auswertung von allen untersuchten Pfadplaner. Die Namen der Pfadplaner sind aus räumlichen Gründen verkürzt: lokaler PX4-Pfadplaner (lok. PX4), globaler PX4-Pfadplaner (glob. PX4), CERBERUS-Pfadplaner (CERBERUS) und MB-Pfadplaner (MB). 80

5.1 Gesamtergebnis der Bewertung der Pfadplaner für den Einsatz in Innenräumen. Der beste erreichbare Wert ist 1 und der schlechteste erreichbare Wert ist 0. Die genutzten Gewichtungen werden in Abschnitt 4.5 vorgestellt. 81
5.2 Übersicht der Anforderungen an die vorliegende Arbeit für die Evaluation der Pfadplaner. Mit dem ✓-Symbol wird gekennzeichnet, dass die Anforderung eingehalten wurde. 82

Listings

4.1 Ausschnitt aus der getSingleCellRisk-Funktion der GlobalPlanner-Klasse. . 41
4.2 Ausschnitt aus der modifizierten getSingleCellRisk-Funktion der GlobalPlanner-Klasse. 41

1 Einleitung

Unbemannte Luftfahrzeuge[1] haben in den letzten Jahren immer mehr Aufsehen erregt [7]. Dies liegt nicht nur an den fallenden Kosten von derartigen Systemen [8] und der damit verbundenen Zugänglichkeit, sondern auch an dem wachsenden Anwendungsspektrum durch die zunehmende Automation von UAS. Dabei wird der Lebensstandard verbessert oder gar Leben gerettet. Beispielsweise arbeiten Unternehmen wie Amazon an einem System für die Lieferung von Paketen mittels autonomen Drohnen [9] mit dem Ziel, Pakete schneller an den Kunden zu liefern. Weitere Anwendungen sehen UAV-basierte Systeme für die Überwachung von Industrieanlagen [10] [11] und Inspektionen von beispielsweise Gebäuden [12] oder Bahnschienen [13] vor. Damit können hohe Kosten, die mit der Inspektion von schwer zugänglichen Orten verbunden sind, signifikant verringert werden. Zusätzlich sind Personen nicht gezwungen, sich in gefährliche Situationen zu begeben, um derartige Inspektionen durchzuführen. Nicht zuletzt werden Anstrengungen für den Einsatz von UAV in **S**earch **a**nd **R**escue (SAR)[2]-Operationen erforscht [14]. Damit soll die Wahrscheinlichkeit für die Rettung bei einer Suchaktion vergrößert und die Abhängigkeit von menschlichen Operatoren verringert werden.

Alle genannten Anwendungsfälle sind auf den Außenbereich ausgelegt. Außenumgebungen zeichnen sich dadurch aus, dass der Großteil des Raumes von einem UAV passierbar ist. Im Gegensatz zu den meisten Starrflügelflugzeugen[3] sind Multikopter[4] in der Lage, an einem Punkt im Raum im Schwebezustand zu verweilen. Starrflügelflugzeuge hingegen benötigen eine minimale Geschwindigkeit, um Auftrieb durch die Flügel zu erzeugen. Diese Beschränkung durch die Mindestgeschwindikeit ist bei Multikoptern nicht gegeben. Dadurch bieten Multikopter die Möglichkeit, in Innenumgebungen eingesetzt zu werden. Die Verfolgung von engen, kurvigen Pfaden ist mit Multikoptern möglich. Die vorliegende Arbeit beschäftigt sich mit dem Anwendungsfall von automatischen Multikoptern in Innenräumen für den Anwendungsfall SAR. Dabei handelt es sich um einen Anwendungsgebiet von UAV, welches zum Zeitpunkt der Arbeit wenig erforscht ist.

Ein integraler Bestandteil von automatischen UAV ist die Ermittlung von verfolgbaren

[1] engl. für UAS
[2] engl. für Suche und Rettung
[3] Flugzeuge, deren Tragflächen fest mit dem Rumpf des Flugzeuges verbunden sind.
[4] Flugzeuge, deren Bewegung durch nach unten gerichtete Rotoren gesteuert wird.

Pfaden basierend auf Daten der Umweltwarhnemung. Dafür werden Pfadplanungsalgorithmen benötigt. Diese Pfadplanungsalgorithmen werden in der vorliegenden Arbeit in Innenräumen anhand von 3D-Simulationen erprobt und nach einer erstellten Metrik evaluiert.

1.1 Motivation

Ein UAS hat mehrere Bestandteile, die benötigt werden, um automatische Missionen austragen zu können. Die Bestandteile werden in die Subsysteme **G**uidance, **N**avigation and **Control** (GNC)[1] eingeteilt [15]. Die Funktionalitäten der einzelnen Subsysteme werden in folgender Auflistung erklärt:

- Das Führungssystem beschäftigt sich mit Planung sowie der Vorgabe von Zielen.
- Das Navigationssystem ermittelt den Zustand des Vehikels sowie der Umgebung anhand von der genutzten Sensorik.
- Das Steuerungssystem übernimmt die Umsetzung der Zielvorgaben auf die Aktuatoren, welche die Bewegung des Vehikels bestimmen. Das Steuerungssystem arbeitet mit dem von dem Navigationssystem ermittelten Zustand des Vehikels.

Ein wichtiger Bestandteil von automatischen UAS sind die Pfadplaner[2]. Der Pfadplaner ist ein Teil des Führungssystem, da dieser räumliche Ziele vorgibt. Pfadplaner haben die Aufgabe, den bestmöglichen Pfad für die zugrundeliegende Aufgabe zu ermitteln. Alle in Kapitel 1 genannten Anwendungsbeispiele sind maßgeblich auf diese Pfadplanungsalgorithmen angewiesen. Hierbei können sich die Ziele der Pfadplaner signifikant unterscheiden. Bei SAR-Operationen soll möglichst effizient Fläche abgedeckt werden. Dementsprechend ist es wichtig, strukturiert den gesamten oder einen großen Teil des gesamten Pfades bereits am Anfang zu planen. Bei Drohnen für die Auslieferungen von Paketen hingegen kann das Ziel eine möglichst kurze Flugbahn sein. Zusätzlich kann hierbei die Hindernisvermeidung eine große Rolle spielen. Bei dem Einsatz von vielen Drohnen muss sichergestellt werden, dass diese nicht miteinander kollidieren. Daher ist ein dynamisches Verhalten für das Ausweichen von mobilen Hindernissen notwendig. Daraus ergibt sich die Anforderung für den Pfadplaner, Pfade in ausreichend kurzen Zeitspannen zu ermitteln.
In dieser Arbeit wird die Pfadplanung mit dem Fokus auf die Anwendung in Innenräumen erforscht. Die behandelte Problemstellung ist die Evaluation von bestehenden Pfadplanungsalgorithmen für UAS in Innenraumumgebungen. Dies ist ein relativ unerforschtes Anwendungsgebiet im Gegensatz zu den Anwendungen in Außenumgebungen. Dennoch

[1] engl. für Führung, Navigation und Steuerung
[2] Pfadplaner werden in dieser Arbeit auch als Pfadplanungsalgorithmen bezeichnet.

bieten moderne Entwicklungen im Bereich von Multikopter und Sensorik zunehmend die Möglichkeit, Einsatzzwecke in Innenräumen zu erschließen.
Insbesondere beschäftigt sich die der vorliegenden Arbeit übergeordnete Aufgabe mit der Identifikation von Zielen in schwer zugänglichen bzw. für den Menschen gefährlichen Situationen. Beispielsweise wird untersucht, inwiefern Ziele wie Menschen in brennenden Gebäuden effizient und gefahrlos mithilfe von UAS aufgespürt werden können. Dies soll möglichst von menschlichen Operatoren entkoppelt durchgeführt werden. Ein derartiges System bietet die Chance, die Dauer zum Erscheinen von Hilfskräften an kritischen Stellen zu verringern. Zudem müssen die Hilfskräfte selbst nicht das eigene Leben in Gefahr bringen, um leere Räumlichkeiten zu inspizieren. Demnach können verfügbare Ressourcen signifikant besser genutzt werden.
Ein derartiges System bedarf einer weitestgehend automatischen und effizienten Pfadplanung in Umgebungen mit wenig Raum für das Ausweichen von Hindernissen. Die vorliegende Arbeit beschäftigt sich mit der Pfadplanung in diesen beschränkten Umgebungen.

1.2 Ziele und Anforderungen

Ziel der Arbeit ist die Evaluation von bestehenden Pfadplanungsalgorithmen in Innenräumen. Hierbei wird der Fokus auf bestehende quelloffene Umsetzungen gelegt. Diese sollen in eine Simulation eingebunden und anhand von einer Sequenz von Tests bewertet werden. Die Anforderungen sind wie folgt definiert:

1. Die Simulation hat realitätsnah zu sein. Dies ist auf die physikalischen Effekte und die Aufnahme von Umgebungsinformationen bezogen.

2. Für die Evaluation der Pfadplanungsalgorithmen ist eine Metrik zu verwenden, welche aus objektiv messbaren Kennwerten besteht. Die Auswertung dieser Metrik soll mithilfe von einer gewichteten Matrix möglich sein.

3. Die Evaluation ist anhand von für Innenräume repräsentativen Umgebungen vorzunehmen.[1]

4. Die Leistungsanforderungen der Pfadplanungsalgorithmen sind für die Evaluation zu betrachten. Hierbei gilt es die Auslastung der **C**entral **P**rocessing **U**nit (CPU), des **R**andom **A**ccess **M**emory (RAM) sowie der **G**raphics **P**rocessing **U**nit (GPU) zu bewerten.

[1] Es kann sinnvoll sein, bestimmte Umgebungen für spezifische Schwierigkeiten von Algorithmen zu verwenden.

Im Folgenden werden die aufgeführten einzelnen Punkte näher betrachtet. Die in Punkt 1 genannten physikalischen Effekte sind auf das Flugverhalten von dem UAS bezogen. Beispielsweise ist das simulierte Vehikel von Trägheitseffekten betroffen, die eine exakte Verfolgung von Pfaden nicht erlauben. Die im gleichen Punkt genannten Umgebungsinformationen beziehen sich auf die Sensorik des UAS. Diese ist in der Realität diskret und hat einen begrenzten Operationsbereich. Beispielsweise haben Kameras eine limitierte Anzahl von Pixeln.
Die in Punkt 2 genannte Metrik kann binäre, diskrete sowie kontinuierliche Kriterien beinhalten. Es ist davon auszugehen, dass verschiedene quelloffenen Implementierungen von Pfadplanern sich stark voneinander unterschieden. Entsprechend ist davon auszugehen, dass nicht alle Kriterien der Metrik an allen Pfadplaner-Umsetzungen gleichermaßen zu ermitteln sind. Die Metrik soll daher den Fokus auf fundamentale Funktionalitäten von den Pfadplanern legen.
Für die in Punkt 3 genannte Evaluation sind Umgebungen zu verwenden, welche realitätsnahe Eigenschaften von Innenräumen widerspiegeln. Insbesondere sind die folgenden Eigenschaften von Innenräumen zu betrachten:

1. Die Entfernung von einem festen Körper ist in nahezu alle Richtungen beschränkt. Das bedeutet, das UAS ist gezwungen, in der Nähe von Hindernissen wie Wänden zu operieren.

2. Räume haben unterschiedlich viel freien Platz. Die Wände können sehr unterschiedliche Distanzen voneinander haben. Entsprechend ist der von dem UAV befliegbare Raum in verschiedenen Räumen potentiell unterschiedlich. Zudem sind Verengungen anzutreffen. Beispielsweise verringern Türen den passierbaren Raum.

3. Die Umgebungen beinhalten mehr als eine Ebene. Dies ist zu betrachten, damit ermittelt werden kann, ob der untersuchte Pfadplaner mit 3D-Umgebungen umgehen kann.

Bei den in Punkt 4 genannten Leistungsanforderungen, die für die Ausführung von den Pfadplanungsalgorithmen zu betrachten sind, ist die Methodik besonders zu beachten. Die Messungen sind stark von der genutzten Hardware abhängig. Um die Vergleichbarkeit mit zukünftigen Arbeiten zu gewährleisten, sind entsprechende Maßnahmen zu ergreifen. Insbesondere ist das Vorgehen bei den Messungen sowie die genutzte Hardware zu dokumentieren.

1.3 Struktur der Arbeit

Dieser Abschnitt stellt die Struktur sowie das Vorgehen der vorliegenden Arbeit dar. Die Arbeit strukturiert sich in fünf übergeordnete Segmente. Diese werden im Folgenden beschrieben:

- In Kapitel 2 wird der Stand der Technik beschrieben. Zu diesem Zweck werden zunächst verwendete Begriffe definiert. Hierbei wird sich an etablierter Terminologie orientiert. Nach dem Definieren der Terminologie werden zu der Thematik der Pfadplanung bisher veröffentlichte Forschungsergebnisse beschrieben und die daraus resultierenden Konsequenzen für die vorliegende Arbeit dargelegt. Für die Einordnung der Pfadplanungsansätze wird eine Kategorisierung von diskutierten Pfadplanungsalgorithmen vorgenommen. Nach der Betrachtung von Pfadplanungsalgorithmen werden Forschungsergebnisse für die Bewertung von diesen Algorithmen untersucht.

- In Kapitel 3 werden die Methoden für die Bearbeitung der vorliegenden Arbeit beschrieben. Das Ziel ist die Reproduzierbarkeit der Resultate dieser Arbeit zu gewährleisten. Mit diesem Ziel wird die verwendete Hardware des simulierenden Computersystems sowie die verwendeten Softwares aufgelistet. Bei den Softwares wird die Funktionalität, die Verwendung in der vorliegenden Arbeit und die Versionsnummern dargelegt. Weiterhin werden die Softwares aufgeteilt in Softwares für die Simulation und untersuchte Pfadplaner-Umsetzungen. Für die Simulation werden Gazebo, PX4, QGroundControl und ROS mit Untermodulen wie RViz auf einen Ubuntu-System genutzt. Bei den Pfadplanern werden die genutzten UAS sowie die Funktionalitäten der jeweiligen Pfadplaner untersucht.

- Das darauffolgende Kapitel 4 beschäftigt sich mit der Evaluation von Pfadplanern. Hierbei werden zunächst vier quelloffene Pfadplaner ausgewählt, analysiert und modifiziert. Basierend auf der Analyse der Pfadplaner wird eine Metrik mit acht Kriterien erstellt. Für jedes Kriterium wird der Wertebereich dargelegt. Zusätzlich wird klargestellt, was erfüllt sein muss, um ein Kriterium zu erfüllen. Weiterhin werden Methoden für die Erfassung aller Kriterien entworfen. Nicht bei allen Pfadplanern ist die Erfassung eines Kriteriums nach der gleichen Methodik möglich. Daher werden, wenn es nötig ist, mehrere Methoden für die Erfassung eines Kriteriums unter Berücksichtigung der gegebenen Funktionalitäten der Pfadplaner entworfen. Als Nächstes folgt die Bewertung der Kriterien für alle Pfadplaner. Für jedes Kriterium wird der ermittelte Wert begründet dargelegt. Schließlich wird eine Gewichtung der Kriterien vorgenommen. Anhand dieser Gewichtung wird eine Auswertung

durchgeführt. Es wird klargestellt, welcher der Pfadplaner sich für den Einsatz in Innenräumen am besten eignet.

- Kapitel 5 beschreibt zunächst die im Rahmen dieser Arbeit erreichten Ergebnisse. Es wird werden die Evaluationsergebnisse der vier Pfadplaner vorgestellt und anschließend diskutiert. Insbesondere wird die Vergleichbarkeit, Erweiterbarkeit und Aussagekraft der Kriterien betrachtet. Abschließend wird die Übertragbarkeit der Ergebnisse der vorliegenden Arbeit auf die Realität diskutiert.
- Zuletzt bietet Kapitel 6 einen Ausblick darüber, welche Anwendungen mit den aktuellen Entwicklungen im Bereich der Pfadplanung bei UAS zukünftig erforscht werden können. Es wird die Optimierung der Pfadplaner, die Verbesserung der Aussagekraft der Evaluation und die Erschließung von weiteren Anwendungsgebieten betrachtet.

1.4 Anmerkungen zum Lesen

Da die genutzte Literatur nahezu ausschließlich in englischer Sprache vorliegt, werden Fachbegriffe in die deutsche Sprache übersetzt. Die in englischer Literatur häufig angetroffenen Begrifflichkeiten werden stets zusätzlich angegeben. Bei einigen Terminologien ist das Übersetzen nicht möglich, ohne die kontextbezogene Aussagekraft der Begrifflichkeiten zu verlieren. Aus diesem Grund werden bestimmte Begriffe in englischer Sprache genutzt. Weiterhin ist anzumerken, dass die in Kapitel 2 dargelegte Kategorisierung unternommen wird, um Implikationen von spezifischen Algorithmen wie beispielsweise Anwendungsfälle vereinfacht darzulegen.

2 Theoretische Grundlagen

Die im vorangegangenen Kapitel definierten Ziele und Anforderungen der vorliegenden Arbeit erfordern eine fundierte Recherche zu bisheriger Forschungen in diesem Bereich. In den folgenden Abschnitten werden Begriffe und Definitionen erläutert und anschließend der aktuelle Stand der Forschung, welcher für die vorliegende Arbeit relevant ist, vorgestellt. Hierzu wird das Kapitel in zwei Teile aufgeteilt:

- Zum einen werden Pfadplanungsalgorithmen und deren Kategorien dargelegt.
- Zum anderen werden bisherige Forschungsergebnisse für die Bewertung von diesen Algorithmen vorgestellt.

Aufbauend auf diesen Erkenntnissen wird das Vorgehen dargelegt.

2.1 Terminologie und Definitionen

Goerzen, C. et al. [1] definieren den *world space*[1] als physischen Raum, in dem ein Vehikel existiert. Ein Vehikel ist als ein bewegliches Objekt definiert. Die Autoren unterstützen die Definitionen mit einer auf zwei Dimensionen reduzierten Veranschaulichung. Ein Ausschnitt dieser Grafik wird in der Abbildung 2.1 dargestellt.
Der *world space* wird in die zwei Untermengen unterteilt: *free space*[2] und *obstacle space*[3]. Der *free space* beinhaltet die von dem Vehikel kollisionsfrei passierbaren Punkte im Raum. Der *obstacle space* beinhaltet räumliche Punkte, die nicht kollisionsfrei passierbar sind.
Ein Pfad wird als eine Kurve im Raum, die von dem Vehikel verfolgt wird, definiert. Eine Trajektorie bzw. eine (Flug-)Bahn ist ein Pfad, bei dem zusätzlich zu den räumlichen Informationen zeitliche Informationen gegeben sind. Dies bedeutet, die geplanten Zielpositionen im Raum haben bei einer Trajektorie zusätzlich einen geplanten Zeitpunkt, zu dem diese erreicht werden.
In der Abbildung 2.1 wird das Vehikel als Kreis dargestellt. Der pink hervorgehobene Pfad

[1] engl. für Weltraum; Im Kontext dieser Arbeit beschreibt dieser Begriff den Raum bzw. die Umgebung, der bzw. die für das Vehikel relevant ist.
[2] engl. für Freiraum
[3] engl. für Hindernisraum

erstreckt sich bis zum Ziel. Das Ziel ist durch eine Raute gekennzeichnet.
An der Abbildung wird deutlich, dass kein Punkt des Vehikels in dem *obstacle space* eindringt. Dadurch wird bei dem Passieren eine Kollision mit einem Hindernis vermieden. In der Literatur wird die Vermeidung von Hindernissen als *obstacle avoidance*[1] bezeichnet [16] [17]. Die von Goerzen, C. et al. [1] genutzten Definitionen decken sich mit weiterer Literatur [18] [17].

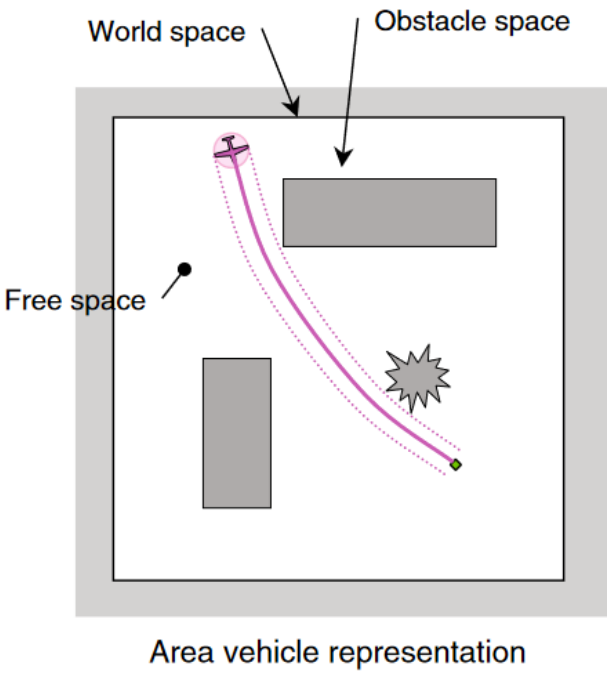


Abbildung 2.1: Ausschnitt der Veranschaulichung von dem *world space* entnommen aus Goerzen, C. et al. [1]

Anzumerken ist, dass der physische Raum in dieser Arbeit simuliert und damit nicht die reale physische Welt referenziert wird. Weiterhin ist die simulierte Welt im Gegensatz zu der Veranschaulichung in Abbildung 2.1 nicht **zweid**imensional (2D), sondern 3D.
Aus den definierten Begriffen ergibt sich, dass Pfadplanungsalgorithmen Algorithmen sind, welche einen Pfad zwischen zwei Punkten im Raum ermitteln. Bahnplanungsalgorithmen sind Pfadplanungsalgorithmen, welche zusätzlich zu den räumlichen Punkten zeitliche Informationen liefern.
Zuletzt kommt in der Literatur der Terminus *motion planning*[2] in dem Zusammenhang mit *path planning* und *trajectory planning* auf. Dieser Begriff ist ein Oberbegriff und umfasst Pfad- sowie Bahnplanung [18].
In dem folgenden Abschnitt wird der aktuelle Stand der Technik bezüglich Pfadplanungsalgorithmen bei UAS mit dem Fokus auf der Anwendung in Innenräumen vorgestellt.

[1] engl. für Hindernisvermeidung
[2] engl. für Bewegungsplanung

2.2 Unterscheidung von Pfadplanungsalgorithmen

Es gibt eine Vielzahl von Ansätzen für Pfadplanungsalgorithmen. Dieser Abschnitt führt zwei Kategorisierungen von Pfadplanungsalgorithmen basierend auf bestehender Literatur ein. Dies hat zum Ziel, im weiteren Verlauf der Arbeit die zugrundeliegenden Methodik eines betrachteten Planers durch einen Verweis auf die Kategorie vereinfacht darzulegen. Dieser Abschnitt beginnt mit einer einfachen Unterscheidung von Pfadplanern anhand von den genutzten Informationen für die Pfadplanung. Anschließend wird in Unterabschnitt 2.2.1 bis Unterabschnitt 2.2.5 eine weitere Kategorisierung anhand der zugrundeliegenden Methodik für den Prozess der Ermittlung eines Pfades eingeführt.
Eine mögliche Unterscheidung von Pfadplanern ist die Klassifizierung von *local*[1] und *global*[2] *path planner*[3]. Baumann, T. [19] beschreibt lokale Planer als Planer, welche auf Sensordaten reagieren. Derartige Algorithmen führen keine Kartierung bzw. Modellierung eines digitalen Abbildes der physischen Welt durch. Es wird zu jedem Zeitpunkt mit den aktuell zur Verfügung stehenden Sensordaten gearbeitet. Ein Vorteil von diesem Ansatz ist, dass dieser wenige Ressourcen benötigt. Jedoch ist bei diesem Ansatz die Planung des gesamten Pfades nicht möglich. Das resultiert in nicht optimalen Pfaden. Ein weiterer Nachteil ist, dass lokale Planer in lokale Minima fallen können. Lokale Minima sind Punkte im Raum, in denen die Summe der Gesamtheit aller künstlich wirkenden Kräfte Null ergibt und in einem Stillstehen des UAS resultieren. Das bedeutet, dass der Pfadplaner in einem Zustand kommen kann, in dem dieser nicht in der Lage ist, eine andere als die aktuelle Position als Ziel zu ermitteln. Daher ist der Planer in der aktuellen Position bzw. dem lokalen Minimum gefangen.
Baumann, T. beschreibt weiterhin, globale Planer arbeiten mit einer vorberechneten Karte der Umgebung oder bilden eine Karte basierend auf Sensordaten, welche während der Laufzeit aufgenommen werden. Mithilfe dieser Karte können Algorithmen wie **A star** (A*) verwendet werden, um einen Pfad zu ermitteln. Der ermittelte Pfad kann optimal sein, wenn die gesamte Umgebung bekannt ist. Ein weiterer Vorteil ist, dass der globale Ansatz weniger anfällig für den Verfall in ein lokales Minimum ist. Jedoch ist dieser Ansatz im Gegensatz zu lokalen Lösungen ressourcenaufwändiger.
In der vorliegenden Arbeit wird weiterhin zwischen informierten und uninformierten globalen Planern unterschieden. Ein globaler Planer, dem die Umgebungsinformationen bekannt sind, wird als informierter globaler Planer bezeichnet. Ein globaler Planer, der eine Karte basierend auf den von der Sensorik aufgenommenen Umgebungsinformationen zu der Laufzeit erstellt, wird als uninformierter globaler Planer bezeichnet.

[1] engl. für lokal
[2] engl. für global
[3] engl. für Pfadplaner

Die Kategorisierung von Pfadplanungsalgorithmen wird von Yu, Z. et al. [20] nach der zugrundeliegenden Methodik unternommen. Die Autoren unterscheiden zwischen folgenden Methodiken:

1. *Heuristic-based Path Search Algorithm*[1]
2. *Sample-based Search Algorithm*[2]
3. *Potential Field Method*[3]
4. *Biologically Inspired Algorithm*[4]
5. *Based on Fusion Algorithm*[5]

welche in den folgenden Abschnitten in gleicher Reihenfolge beschrieben werden.

2.2.1 Heuristik-basierte Suchalgorithmen

Heuristik-basierte Suchalgorithmen [20] sind Algorithmen wie der Djikstra Algorithmus oder der A* Algorithmus bzw. dessen Variationen wie zum Beispiel D*, Theta* und Lazy Theta*. Diese Algorithmen konstruieren eine Bewertungsfunktion für die Kosten und ermitteln mithilfe von dieser Funktion den Pfad mit den geringsten Kosten aus einem Graphen möglicher Pfade. Ein Vorteil von diesen Algorithmen ist, dass optimale Lösungen gefunden werden können. Jedoch sind diese ressourcenintensiv und damit für die Echtzeit-Anwendung auf mobilen Systemen mit begrenzten Ressourcen beschränkt geeignet.

2.2.2 Stichproben-basierte Suchalgorithmen

Stichproben-basierte Suchalgorithmen [21] nutzen kein genaues Abbild der Umgebung. Aus dem *free space* werden stichprobenartig räumliche Punkte ermittelt. Diese werden genutzt, um einen Graphen[6] von möglichen Trajektorien zu generieren. Anschließend wird eine Kollisionsprüfung der Trajektorien vorgenommen, um passierbare Trajektorien zu identifizieren. Aus dem Graphen passierbarer Trajektorien wird schließlich eine Trajektorie für die Pfadplanungsaufgabe ermittelt. Der Vorteil von diesem Ansatz ist der geringe

[1] engl. für Heuristik-basierte Suchalgorithmen
[2] engl. für Stichproben-basierte Suchalgorithmen
[3] engl. für Potentialfeldmethode
[4] engl. für von der Biologie inspirierte Algorithmen
[5] engl. für Fusionsalgorithmen
[6] im engl. als *roadmap* bezeichnet

rechnerische Aufwand. Dies liegt daran, dass keine Hindernisse aus verfügbaren Sensordaten rekonstruiert werden. Ein weiterer Vorteil ist, dass die Wahrscheinlichkeit dafür, dass der Algorithmus kein Ergebnis liefert, gegen Null konvergiert, wenn sich die Anzahl der Stichproben der Unendlichkeit nähert. Die Voraussetzung ist, es existiert ein Ergebnis. Als Nachteil ist zu nennen, dass die mit diesen Methoden ermittelten Pfade zufällig sind und daher keine optimalen Pfade garantieren [21]. Beispiele für Stichproben-basierte Suchalgorithmen sind **Pr**obabilistic **R**oadmap (PRM) und **R**apidly exploring **R**andom **T**ree (RRT) [20]. Davon hat RRT den Vorteil, dass nicht holonomische Beschränkungen betrachtet werden können. Holonomische Beschränkungen erlauben die Reduktion von Freiheitsgraden aufgrund von vorherrschenden Rahmenbedingungen in einem System. Dies ist beispielsweise dann der Fall, wenn aus zwei Koordinaten in einem dreidimensionalen Koordinatensystem die dritte Koordinate aufgrund von einer Abhängigkeit bzw. Beschränkung ermittelt werden kann. Das resultiert in einer Reduktion der Komplexität des Systems.
Da RRT nicht-holonomische Beschränkungen erlaubt, können komplexe kinematische Beschränkungen vermieden werden. Dies resultiert in einer vereinfachten Pfadplanung. Der PRM-Algorithmus hingegen kann schnelle Resultate liefern, jedoch müssen diese nicht optimal sein [22]. Die Laufzeit von einem PRM-basierten Pfadplaner hängt maßgeblich von dem darauf aufbauenden Algorithmus für die Pfadplanung, welche die Knoten des probabilistischen Graphen verbindet, ab [23].

2.2.3 Suchalgorithmen basierend auf der Potentialfeldmethode

Die Potentialfeldmethode [20] arbeitet mit virtuellen künstlichen Potentialfeldern. Hindernissen wird eine abstoßende Kraft und dem Ziel eine anziehende Kraft verliehen. Anschließend wird die absteigende Richtung der Potentialfunktion gesucht. Diese Richtung ist mit der des Ziels identisch und vermeidet Hindernisse. Vorteil dieser Methode der Pfadplanung ist die gute Echtzeit-Performance. Zudem sind **A**rtificial **P**otential **F**ield (APF)-Algorithmen in der Lage auf dynamische Hindernisse zu reagieren [24]. Nachteil ist die starke Anfälligkeit für das Feststecken in einem lokalen Minimum.

2.2.4 Von der Biologie inspirierte Suchalgorithmen

Von der Biologie inspirierte Suchalgorithmen werden von Yu, Z. et al. [20] in zwei weitere Unterkategorien geteilt: *evolutional algorithms*[1] und *neural networks*[2]. Beide Kategorien

[1] engl. für evolutionäre Algorithmen
[2] engl. für neuronale Netzwerke

von Pfadplanungsmethoden konvergieren zum Ziel ohne die Erstellung von komplexen Abbildern der Umgebung. Vorteil dieser Methoden ist die Möglichkeit der Handhabung von komplexen Beschränkungen. Zudem können diese Algorithmen auch bei großen räumlichen Ausmessungen eingesetzt werden. Ein Nachteil von derartigen Methoden ist, dass diese dazu tendieren, frühzeitig zu konvergieren. Wenn dies passiert, wird keine optimale Lösung gefunden.

2.2.5 Fusion von Suchalgorithmen

Die letzte von den Autoren Yu, Z. et al. [20] vorgestellte Kategorie ist der *fusion algorithm*[1]. Ein in der Literatur vorzufindendes Synonym für diese Kategorie von Planern ist *hybrid path planner*[2] [25]. Auch hier wird in zwei Unterkategorien geteilt:

1. Zum einen gibt es Fusionsalgorithmen, die eine Menge von Pfadplanungsalgorithmen verwenden, um kooperativ die beste Lösung zu ermitteln.

2. Zum anderen gibt es Fusionsalgorithmen, die eine Menge von Pfadplanungsalgorithmen sequentiell miteinander verknüpfen.

So kann beispielsweise der APF-Algorithmus mit dem A*-Algorithmus verknüpft werden. Der APF-Algorithmus übernimmt die globale Pfadplanung und der A*-Algorithmus die lokale Pfadplanung. Eine umgekehrte Kombination von der Fusion der beiden Algorithmen wird von Ju, C. / Luo, Q. / Yan, X. [24] vorgestellt. Bei dem vorgeschlagenen Fusionsalgorithmus wird der APF-Algorithmus für die lokale und der A*-Algorithmus für globale Pfadplanung genutzt. Den Autoren gelingt mittels dieser Fusion eine Verkürzung der Pfade im Vergleich zum APF-Algorithmus. Gleichzeitig ist der Fusionsalgorithmus in der Lage auf dynamische Hindernisse zu reagieren, was mit dem A*-Algorithmus nicht ohne eine Neuberechnung des Pfades möglich ist.
Anzumerken ist, dass die vorgestellte Kategorisierung nicht allgemeingültig ist. Es existieren jedoch Überschneidungen mit anderen Kategorisierungsvorschlägen. Beispielsweise unterscheidet Han, B. et al. [2] zwischen *biological intelligence algorithm*[3], *neural network* und *search algorithm*[4]. Hierbei ist die Kategorie der Suchalgorithmen eine Verallgemeinerung der zuvor genannten Kategorie der Heuristik-basierten Suchalgorithmen.
Aus den vorgestellten Ansätzen für Pfadplanung wird deutlich, dass jeder Ansatz mit eigenen Vor- und Nachteilen verbunden ist. Es gibt keinen Pfadplanungsalgorithmus, der für alle Szenarien anwendbar ist. In dem folgenden Kapitel werden Ansätze für Pfadplanung

[1] engl. für Fusionsalgorithmus
[2] engl. für hybride Pfadplaner
[3] engl. für biologische Intelligenz Algorithmus
[4] engl. für Suchalgorithmus

näher erläutert, die in Zusammenhang mit UAS entwickelt bzw. angewandt werden. Ein besonderer Fokus wird auf bisherige Entwicklung im Bereich von UAS-Pfadplanung in Innenräumen gelegt.

2.3 Forschungsarbeiten im Bereich der Pfadplanung in Innenräumen für UAS

Die Pfadplanung wird in den folgenden Abschnitten im Hinblick auf den Schwerpunkt Innenraum betrachtet. Es werden drei Arbeiten über die Pfadplanung mit dem Anwendungsgebiet Innenraum vorgestellt. Die Resultate der Arbeiten werden zusammen mit den Rahmenbedingungen, unter denen die Ergebnisse erzielt wurden, vorgestellt. Ziel dieses Abschnitts ist die Darlegung des aktuellen Stand der Forschung bezüglich der Pfadplanung in Innenräumen sowie der Bedeutung von diesem für die vorliegende Arbeit. Dieser Einblick wird in den Unterabschnitten 2.3.1 bis 2.3.3 gegeben. Anschließend folgt in Unterabschnitt 2.3.4 eine Einschätzung der Auswirkungen der Erkenntnisse der vorangegangen Unterabschnitte für die vorliegende Arbeit.

2.3.1 GeoSOT-basierte Pfadplanungsalgorithmus

Han, B. et al. [2] stellen einen **Geo**graphic Coordinate Subdividing Grid with **O**ne Dimension Integral Coding on 2^n **T**ree (GeoSOT)-basierten Pfadplanungsalgorithmus für Innenräume vor. GeoSOT wird für die Rasterdarstellung des Innenraums verwendet. Die Rasterdarstellung ist der *world space*, welcher Informationen über den *free space* sowie den *obstacle space* beinhaltet. Basierend auf diesen bekannten Umgebungsinformationen kommt ein verbesserter A*-Algorithmus als heuristische Methode für die Pfadplanung zum Einsatz. Das Ergebnis ist eine optimale Trajektorie zu den Zielkoordinaten. Der vorgestellte Ansatz erreicht eine schnelle Planung, die mit traditionellen Ansätzen für die Längen- und Breitengradkoordinaten nicht gelingt [2]. Zudem wird die Verwendung von alternativen Suchalgorithmen wie der **S**imulated **A**nnealing (SA)-Algorithmus und der *ant colony algorithm*[1] untersucht. Von diesen Algorithmen erzielt der verbesserte A*-Algorithmus die besten Resultate. Diese Schlussfolgerung wird basierend auf der benötigten Zeit für die Ermittlung des Pfades sowie dessen Länge gezogen.
Zu betonen ist, dass dieser Ansatz aufgrund des verwendeten GeoSOT-Raster in der Lage ist, Innenraumumgebungen mit mehreren Etagen zu navigieren. Eine Abbildung der für die Versuche genutzten Umgebung ist in Abbildung 2.2 gegeben.

[1] Ameisenkolonie-Algorithmus

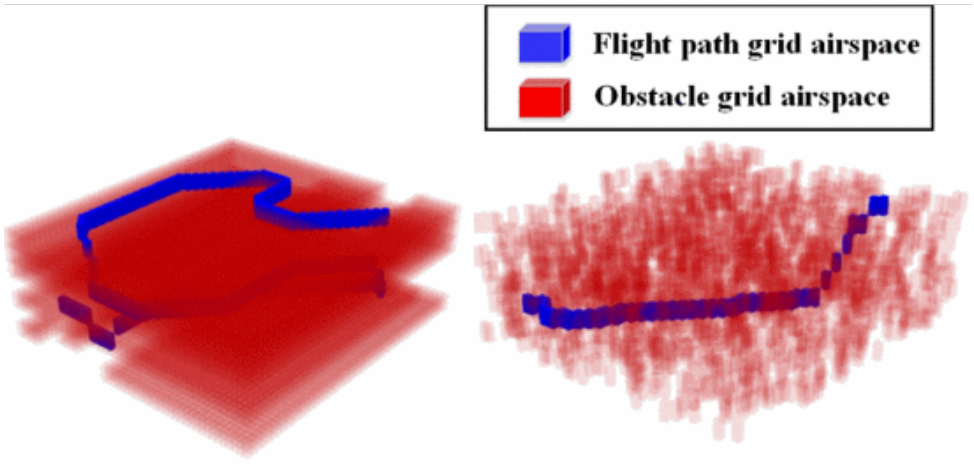


Abbildung 2.2: Veranschaulichung von einer von Han, B. et al. [2] genutzten Umgebung für die Erprobung des vorgestellten Pfadplanungsalgorithmus

Die Resultate von Han, B. et al. [2] basieren auf Simulationen in einer nicht spezifizierten Simulationssoftware [2]. Weiterhin ist anzumerken, dass die Informationen über die Umgebung nicht von der Umweltwahrnehmung des UAV aufgenommen werden. Diese werden dem Pfadplanungsalgorithmen zur Verfügung gestellt.

2.3.2 Pfadplanung mittels maschinellem Lernen

Ein weiterer Ansatz für die Pfadplanung in Innenräumen wird von Park, Jongmin / Jang, Sooyoung / Shin, Younghoon [3] vorgestellt. Bei diesem Ansatz wird ein *reinforcement learning*[1]-Modell eingesetzt. Es handelt sich demnach um einen Ansatz, der neuronale Netzwerke nutzt. Das Training findet nach der *curriculum learning*[2]-Methode statt. Bei dieser Methode wird der Lernprozess in zunehmend komplexere Schritte gegliedert. Sobald ein Lernziel erreicht ist, wird ein neues komplexeres Lernziel gesetzt und das Modell daraufhin trainiert [3] [26]. Mit diesem Vorgehen wird eine verbesserte Generalisierung sowie eine erhöhte Konvergenzgeschwindigkeit erzielt als bei dem Training eines komplexen Startziels. Mit diesem Ansatz ist es den Autoren gelungen, lokale Pfadplanungsmodelle in Gazebo-Simulationen[3] zu trainieren. Die zwei vorgestellten Modelle erzielen eine Erfolgsrate von bis zu 95.8% in Umgebungen ohne Hindernisse und bis zu 88% in Umgebungen mit Hindernissen. Die zwei Umgebungen werden in der Abbildung 2.3 visualisiert.

[1] engl. für Bestärkendes Lernen
[2] engl. für curriculares Lernen
[3] siehe Unterabschnitt 3.2.1

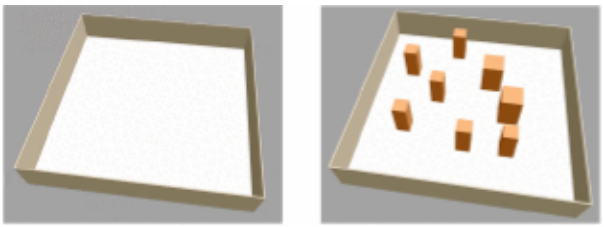


Abbildung 2.3: Ausschnitt der Veranschaulichung von den Umgebungen für die Simulationen von Park, J. et al. [3]. Auf der linken Seite ist die Umgebung ohne Hindernisse und auf der rechten Seite die Umgebung mit Hindernissen dargestellt.

Die genutzten Hindernisse sind einzelne im Raum verteilte Quader unterschiedlicher Größe. Als Erfolg wird der abgeschlossene Flug des UAV von einem zufälligen Punkt in dem simulierten Raum zu einem anderen zufälligen Punkt im gleichen Raum betrachtet. Anzumerken ist, dass die in der Arbeit von Park, J. et al. betrachteten Umgebungen lediglich eine Ebene haben und daher als zweidimensional zu betrachten sind.

2.3.3 Fusionsbasierte Ansatz für Pfadplanung

In [4] stellen Rocha, L. et al. eine Fusion von verschiedenen lokal genutzten Suchalgorithmen mit einem vorgeschlagenen globalen Planer vor. Die lokalen Suchalgorithmen umfassen A*, **Bi**directional **A star** (BiA*), APF, Bidirectional **A**rtificial **P**otential **F**ield (BiAPF), RRT und **Bi**directional **R**apidly exploring **R**andom **T**ree (BiRRT). Die mit *bidirectional*[1] benannten Suchalgorithmen unterscheiden sich von den klassischen Gegenstücken dadurch, dass die Suche nicht nur von dem Start zum Ziel, sondern parallel dazu auch von dem Ziel zum Start durchgeführt wird [4] [27] [28]. Eine Ausnahme stellt der APF-Algorithmus dar. Dieser wurde von den Autoren [4] so implementiert, dass die Suche von Start zum Ziel nur dann umgekehrt wird, wenn ein lokales Minimum erkannt wurde. Es handelt sich also um keine parallel bidirektionale Suche, sondern um eine bedarfsabhängige sequentiell bidirektionale Suche.
Der in [4] erläuterte globale Planer ermittelt ausgehend von dem Zielpunkt nach einem vorgestellten Verfahren Hilfspunkte. Die ermittelten Hilfspunkte ergeben einen groben Pfad von dem Ziel zum Start. Rocha, L. et al. [4] führen fort, die lokalen Planer werden anschließend genutzt, um Pfade zwischen diesen Hilfspunkten zu ermitteln. Das Ergebnis ist ein vollständiger Pfad, der von dem Vehikel von dem Start- bis zum Zielpunkt verfolgt werden kann. Die zusätzliche Verwendung des globalen Planers hat einige Vorteile gegenüber der ausschließlichen Verwendung der lokalen Suchalgorithmen.
Zum einen haben in dem Innenraum-Testszenario die Algorithmen RRT, APF und BiAPF keinen Pfad in einer Innenraumumgebung ermitteln können. Die Fusion dieser Algorithmen

[1] engl. für bidirektional

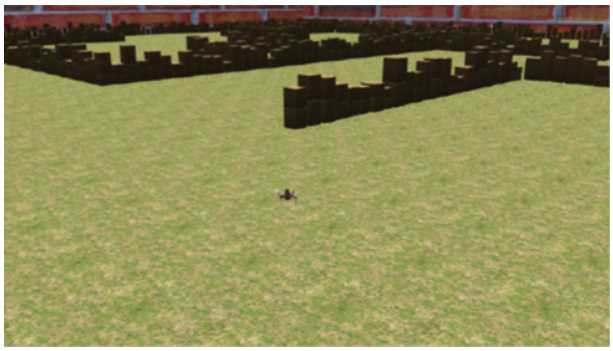


Abbildung 2.4: Ausschnitt der Veranschaulichung von der von Rocha, L. et al. [4] genutzten Innenraumumgebung. In der Mitte der Abbildung befindet sich das UAV.

mit dem globalen Planer hat in dem Testszenario dazu geführt, dass valide Pfade ermittelt werden konnten. Eine der verwendeten Testumgebungen ist in Abbildung 2.4 abgebildet. Zum anderen werden die Pfade bei der Verwendung des globalen Planers in bis zu 79% verkürzter Zeit ermittelt. Nachteilig ist, dass der Rechenaufwand durch die Verwendung des globalen Planers steigt. Zudem sind die Pfade nicht optimal. Algorithmen wie A* sind in der Lage optimale Pfade zu ermitteln. Die Kombination dieser Algorithmen mit dem nicht optimalen globalen Planer führt dazu, dass die resultierenden Pfade ebenfalls nicht optimal sind.
Der von Rocha, L. et al. [4] vorgestellte fusionierte Algorithmus wurde in Gazebo-Simulationen getestet. Der Algorithmus ist für zwei räumliche Dimensionen konzipiert und dadurch nicht in der Lage, Pfade über mehrere Etagen zu ermitteln.

2.3.4 Bedeutung für die vorliegende Arbeit

Für die Erklärung der Auswirkungen der betrachteten Arbeiten auf die vorliegende Arbeit wird das Ziel dieser wiederholt erwähnt. Das Ziel der vorliegenden Arbeit ist die Evaluation von Pfadplanungsalgorithmen in realitätsnahen Innenraumumgebungen. In Bezug auf dieses Ziel wird die Bedeutung der in den Unterabschnitten 2.3.1 bis 2.3.3 vorgestellten Arbeiten [2] [3] [4] dargelegt. Die Arbeiten stellen Ansätze für die Pfadplanung in Innenraum-ähnlichen Umgebungen vor. In diesem Unterabschnitt wird betrachtet, ob diese Umsetzungen den in 1.2 genannten Kriterien für die vorliegende Arbeit entsprechen. Konkret werden folgende Fragestellungen betrachtet:

1. Werden realitätsnahe Simulationen mit Betrachtung von physikalischen Rahmenbedingungen als Grundlage der vorgestellten Schlussfolgerungen der Arbeit genutzt?

2. Werden die Umgebungsinformationen mithilfe der Sensorik des UAS gewonnen?

3. Sind die Innenraumumgebungen mit den in Punkt 1.2 genannten Eigenschaften von Innenräumen vergleichbar?

Im Folgenden werden die vorgestellten Arbeiten anhand der Fragestellungen untersucht und es wird begründet dargelegt, wieso die Kriterien erfüllt werden oder nicht.
Der in Unterabschnitt 2.3.1 vorgestellte GeoSOT-basierte Pfadplaner wurde in einer nicht spezifizierten Simulationsumgebung getestet [2]. Entsprechend ist hierbei nicht bekannt, ob der vorgestellte Ansatz in realitätsnahen Anwendungsfällen, wie sie in dieser Arbeit gefordert werden, untersucht wurde. Damit können die Punkte 1 und 3 nicht beantwortet werden. Zudem handelt es sich um einen informierten globalen Planer. Dieser startet die Planung mit allen Informationen der Umgebung. Damit ist Anforderung 2 verletzt, da keine Sensorik existiert. Insbesondere ist nicht zu erwarten, dass in einem SAR-Szenario alle Informationen bezüglich der physischen Umgebung vor dem Einsatz des UAS bekannt sind.
Der in Unterabschnitt 2.3.2 vorgestellte **M**achine **L**earning (ML)-basierte Pfadplaner wurde ausschließlich in Umgebungen mit einer Ebene erprobt. Entsprechend sind die Ergebnisse des vorgestellten Pfadplaners anhand von 2D-Umgebungen erzielt worden. Dies verletzt Anforderung 3. Zudem sind die genutzten Hindernisse einzelne im Raum verteilte Quader unterschiedlicher Größe. Diese in dem rechten Teil der Abbildung 2.3 dargestellte Umgebung entspricht nicht der in Abschnitt 1.2 dargelegten Komplexität der Umgebungen. Das ist eine weitere Verletzung von dem Punkt 3. Weiterhin ist die Erfolgsrate des Planers durch das Hinzufügen von den in Abbildung 2.3 auf der rechten Seite dargestellten Hindernissen von $95,8\%$ auf 88% gesunken. Basierend auf diesen Informationen ist davon auszugehen, dass eine signifikante Erhöhung der Komplexität der Umgebung auch die Erfolgsrate signifikant verringern würde.
Der in Unterabschnitt 2.3.3 von Rocha, L. et al. [4] vorgestellte Fusionsalgorithmus für die Pfadplanung entspricht aus ähnlichen Gründen nicht den Anforderungen. Wie auch bei dem in Unterabschnitt 2.3.2 vorgestellten Planer wurde auch dieser Pfadplaner in weniger komplexen als den in Abschnitt 1.2 geforderten Anforderungen getestet. Abbildung 2.4 macht deutlich, dass das UAV viel passierbaren Raum in der Umgebung hat. Demnach ist auch hierbei die Anforderung 3 nicht eingehalten. Entsprechend ist auch hierbei keine Aussage darüber möglich, wie sich der Planer in signifikant komplexeren Umgebungen verhält.
In Bezug auf die vorliegende Arbeit wurden verschiedene Ansätze für die Pfadplanung in Innenräumen erprobt. Basierend auf den geschilderten Beobachtungen werden folgende Erkenntnisse festgehalten:

- Der in Unterabschnitt 2.3.1 vorgestellte Planer gehört zu der in Unterabschnitt 2.2.1 beschriebenen Kategorie von Heuristik-basierten Suchalgorithmen.
- Der in Unterabschnitt 2.3.2 vorgestellte Planer gehört zu der in Unterabschnitt 2.2.4 beschriebenen Kategorie von von der Biologie inspirierten Suchalgorithmen.

- Der in Unterabschnitt 2.3.3 vorgestellte Planer gehört zu der in 2.2.5 beschriebenen Kategorie von fusionierten Suchalgorithmen. Fusioniert werden ein globaler Planer und ein lokaler Planer. Anhand der Beschreibung des globalen Planers wird davon ausgegangen, dass es sich um einen Planer handelt, welcher zu der in Unterabschnitt 2.2.2 beschriebenen Kategorie von Stichproblen-basierten Suchalgorithmen gehört. Unter den lokalen Planern wurden neben dem von Rocha, L. et al. [4] vorgestellten A*-Planer auch APF- und RRT-Planer sowie Variationen davon untersucht. Die zwei letzteren Planer gehören zu den Kategorien von Planern, die in Unterabschnitt 2.2.3 und Unterabschnitt 2.2.2 beschriebenen werden. Entsprechend basieren die Pfadplaner zum einen auf der Potentialfeldmethode und zum anderen auf den Stichproben der Umgebung.

Damit ist sichergestellt, dass ein weites Spektrum von Ansätzen untersucht wird. Des Weiteren kann beobachtet werden, dass die Evaluation von Erfolg in den vorgestellten Arbeiten unterschiedlich ist:

- Han, B. et al. [2] haben einen möglichst kurzen Pfad bei minimalen Rechenaufwand, gemessen an der Zeit, als Ziel.
- Park, J. et al. [3] haben eine möglichst große Erfolgsrate bei Flügen zwischen zwei zufällig gewählten Koordinaten im Raum als Ziel.
- Rocha, L. et al. [4] haben eine erfolgreich ermittelte Route bei minimalen Rechenaufwand, gemessen an der Zeit, als Ziel.

Zuletzt ist zu betonen, dass die jeweils genutzten Definitionen von Innenräumen nicht mit der für diese Arbeit genutzten Definition kongruent sind. Insbesondere die Komplexität der in den vorgestellten Ansätzen erprobten Umgebungen ist signifikant geringer als die Komplexität von den Umgebungen, die in dieser Arbeit zu betrachten sind. Beispielsweise ist keine der Umgebungen der in Unterabschnitt 2.3.2 und Unterabschnitt 2.3.3 thematisierten Arbeiten nach oben hin beschränkt. Als Folge sind keine Schlussfolgerungen bezüglich der Anwendbarkeit dieser Ansätze auf Umgebungen der vorliegenden Arbeit machbar.

2.4 Evaluation von Pfadplanungsalgorithmen

In diesem Abschnitt werden verschiedene Ansätze für die Evaluation von Pfadplanungsalgorithmen betrachtet. Das Ziel ist eine Übersicht von in der Literatur bewerteten Kriterien sowie der für die Bewertung genutzten Methoden darzulegen und diese anschließend in Bezug zu der vorliegenden Arbeit zu setzen. Die Erkenntnisse der Untersuchung werden anschließend in Kapitel 4 für die Erstellung einer Metrik genutzt.

Kriterien sind Kennwerte, die für eine Bewertung genutzten werden. Es gibt eine Vielzahl von Kriterien, die bei Pfadplanern bewertet werden. Die folgende Aufzählung beinhaltet in der Literatur vorzufindene Kriterien, nach denen Pfadplaner bewertet werden:[1]

- Ermittelt der Pfadplaner in einem gegebenen Szenario einen validen Pfad? [3]
- Kosten des ermittelten validen Pfades gemessen an:
  - der Länge des Pfades [22] [29] [30].
  - dem Energieaufwand für das Verfolgen des ermittelten Pfades von dem Vehikel [31].
  - dem zeitlichen Rechenaufwand für die Ermittlung des Pfades [2] [4] [22] bzw. Speicheraufwand [4].
  - der Gefahr des ermittelten Pfades für das Vehikel bzw. die Umgebung [29] [30].
  - der Anzahl und Ausprägung von Kurven [22].
- Lokalisierbarkeit des UAV während dem Folgen des Pfades [25].

Für die Bewertung von den meisten Kriterien ist ein ermittelter Pfad eine Voraussetzung. In manchen Fällen wird jedoch lediglich bewertet, in welchen Szenarien der Planer in der Lage ist, einen Pfad zu ermitteln. Dies ist insbesondere dann sinnvoll, wenn die Umgebung komplex oder der Planer nicht erprobt ist.
Die Kostenbewertung hat entweder das Resultat des Pfadplaners oder den Pfadplanungsalgorithmus zum Ziel. Bei dem Ersteren kann beispielsweise die Länge des ermittelten Pfades bewertet werden. Im Regelfall ist ein kürzerer Pfad erstrebenswert. Bei dem Letzteren wird der Algorithmus bzw. die Implementierung von diesem bewertet. Dazu gehören beispielsweise Laufzeiten sowie benötigte Hardwareressourcen. Insgesamt ist die Kostenbetrachtung bei nahezu jeder Bewertung von Pfadplanungsalgorithmen Kern der Untersuchung.
Die Bewertung von Kriterien erfolgt mithilfe von Methoden. Methoden geben an, wie ein Kriterium erfasst wird. Die Aussagekraft von den in der vorherigen Aufzählung genannten Kriterien variiert stark abhängig von der Methodik zur Erfassung von diesen Kriterien. Zudem ist die Spezifikation von Methoden für die Reproduzierbarkeit der Ergebnisse relevant.

[1] Die Literaturverweise dienen als Beispiele für Arbeiten, in denen das entsprechende Kriterium bewertet wird, und beschränken sich nicht auf Planer für UAS.

3 Technische Grundlagen

In diesem Kapitel werden die verwendete Hardware sowie Software vorgestellt. Ziel ist die Reproduzierbarkeit der Ergebnisse der vorliegenden Arbeit. Begonnen wird mit der gebrauchten Hardware. Die einzelnen Hardwarekomponenten samt Modellbezeichnungen, die für die Bearbeitung der Aufgabenstellung relevant sind, werden dargelegt. Anschließend wird die Software vorgestellt. Die Software sowie das Nutzen von verwendeten Softwarekomponenten werden dargelegt. Anschließend folgt eine Übersicht von diesen Softwarekomponenten samt Versionsnummern.

3.1 Übersicht verwendeter Hardwarekomponenten

Die verwendete Hardware beschränkt sich im Rahmen dieser Arbeit auf den Computer, der für die Simulation der Pfadplaner genutzt wird. Eine Übersicht der genutzten Hardware ist in Tabelle 3.1 gegeben. Es ist davon auszugehen, dass andere Hardware als die genannte zu anderen Ergebnissen als den dokumentierten führt. Dies gilt insbesondere für Messungen der Laufzeiten von Algorithmen sowie die Auslastung der Hardware. Es ist anzumerken, dass anfängliche Arbeiten mit $16GB$ Arbeitsspeicher durchgeführt wurden. Die Anforderungen an den Arbeitsspeicher haben diesen Wert überschritten. Daher wurde der Speicher um weitere $32GB$ ergänzt.

3.2 UAV-Kommunikation und Steuerung innerhalb Simulationen

In diesem Abschnitt wird der Zusammenhang und die Nutzung von Softwarekomponenten erläutert. Eine Übersicht der Kommunikation der Simulationsumgebung wird in Abbildung 3.1 dargestellt.

Tabelle 3.1: Übersicht verwendeter Computerkomponenten für die Simulationen der vorliegenden Arbeit

Komponente	Marke	Modell	Anzahl
Mainboard	Gigabyte	B660 GAMING X DDR4	1
nichtflüchtiger Speicher	Samsung	860 QVO $1TB$	1
Prozessor (CPU)	Intel	Core i5-12400F	1
Grafikkarte (GPU)	Gigabyte	GeForce RTX 3070 Ti	1
Arbeitsspeicher (RAM)	Corsair	Vengeance RGB RS ($3200MHz$; $8GB$)	2
	Corsair	Vengeance RGB RS ($3200MHz$; $16GB$)	2

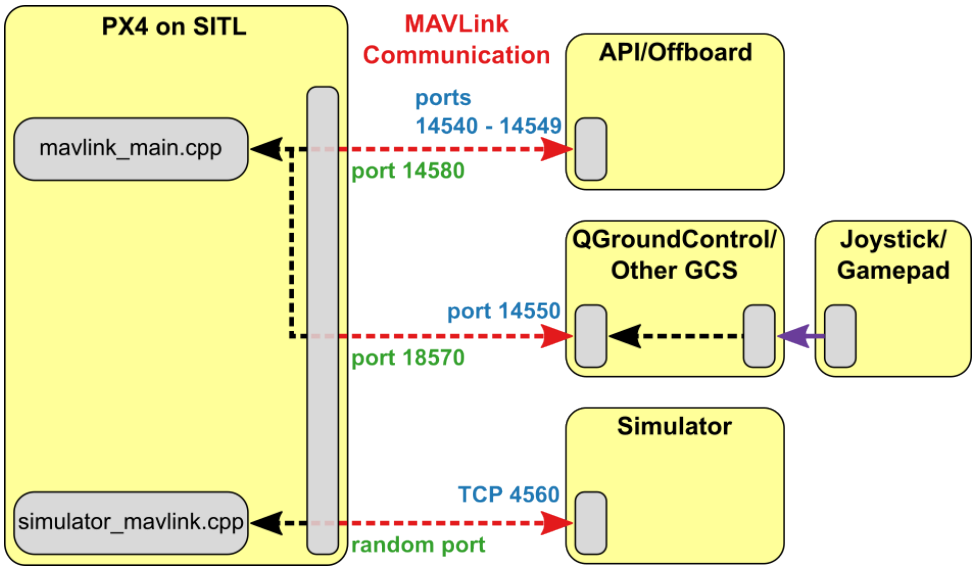


Abbildung 3.1: Übersicht der Kommunikation der SITL Simulationsumgebung [5]

Für die Simulationen dieser Arbeit werden einige Komponenten eingesetzt. Die Flugsteuerung ist ein zentraler Bestandteil des Simulationssystems. In der Abbildung 3.1 wird dies auf der linken Seite verdeutlicht. Zum Einsatz kommt die PX4-Software [32], welche im SITL-Modus genutzt wird. Der SITL-Modus erlaubt die Ausführung der Flugsteuerungssoftware auf dem simulierenden Computersystem anstatt auf dafür dedizierter Hardware. Demnach erfordert der Einsatz der SITL-Funktionalität keine spezielle Hardware. Die folgenden Unterabschnitte erläutern die Funktion der Softwarekomponenten sowie die Anwendung innerhalb der vorliegenden Arbeit. Der Fokus der Erläuterungen liegt auf den

genutzten Elementen der Softwarekomponenten.
Alternativ zu der PX4-Flugsteuerung kommen einige Umsetzungen von Pfadplanern mit eigenen Implementierungen für die Flugsteuerung. Diese erfordern keine Verbindung zu einer Bodenstation. Daher wird auch diese abhängig von der Pfadplaner-Umsetzung nicht benötigt. Unabhängig von der Umsetzung für die Flugsteuerung ist das in 3.1 grundlegende Prinzip gültig. Der Simulator kommuniziert mit der Flugsteuerung und diese ist an die ROS-API für weitere Kommunikation angebunden.

3.2.1 Gazebo-Simulator

Als Simulator wird Gazebo eingesetzt [33]. Diese Software simuliert die virtuelle Welt mit realitätsnahen physikalischen Einflüssen. Zusätzlich wird das UAV mit der zugehörigen Sensorik simuliert. Die Gazebo-Simulation kann mit oder ohne eine grafische Oberfläche gestartet werden. Ohne die **G**raphical **U**ser **I**nterface (GUI) werden weniger Ressourcen beansprucht. Die GUI kann verwendet werden, um visuelles Feedback zu erhalten. Abbildung 3.2 stellt beispielhaft einen Ausschnitt aus einer Gazebo-Simulation dar. In der Mitte des abgebildeten Fensters ist die Umgebung und das UAV dargestellt. Die untere Leiste beinhaltet Metadaten zu der Simulation wie die simulierte und reale Zeit. Auf der linken Seite befinden sich in einer Baumstruktur die simulierten Objekte. Das in der Baumstruktur ausgewählte Element wird unter der Baumstruktur in einem eigenen Fenster dargestellt und kann von dort aus modifiziert werden. Beispielsweise kann der erwünschte **R**eal **T**ime **F**actor (RTF)[1] während der Laufzeit einer Simulation angepasst werden.

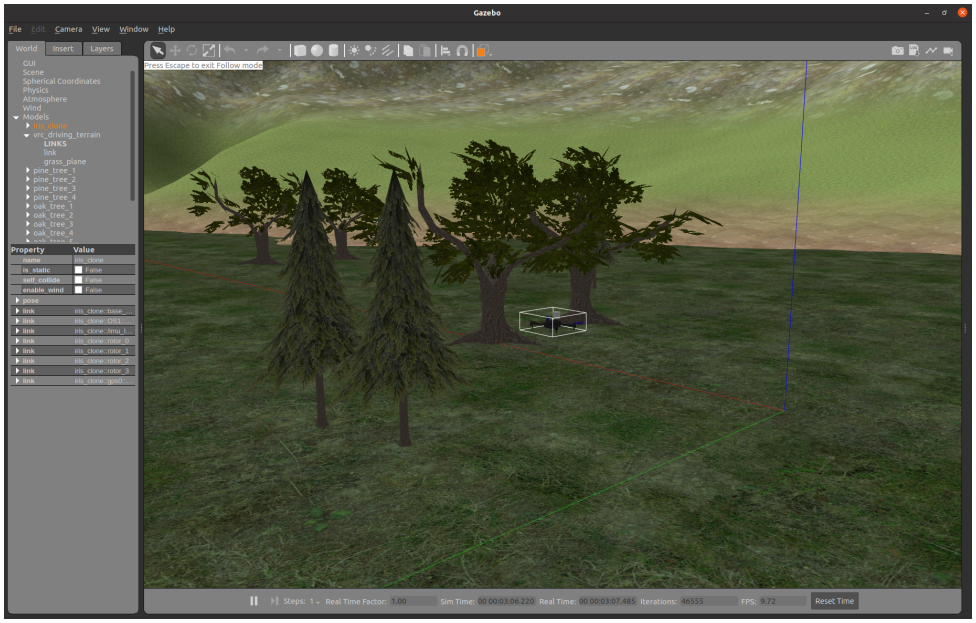


Abbildung 3.2: Gazebo-Simulation während der Verwendung eines Pfadplaners

[1] engl. für Echtzeitfaktor

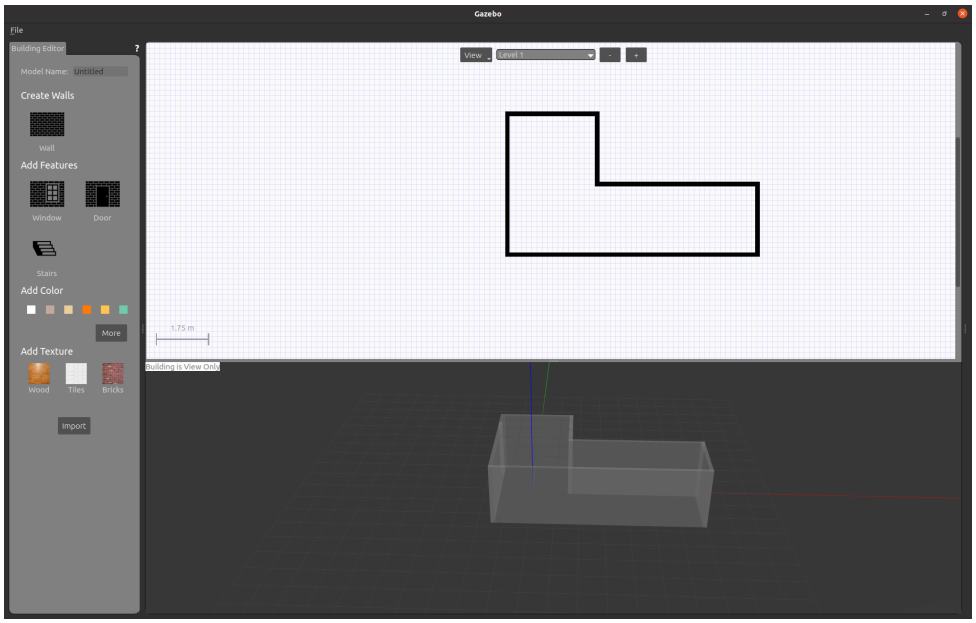


Abbildung 3.3: Gazebo Building-Editor bei der Erstellung einer Testumgebung

Eine weitere Funktionalität der Gazebo-Software, die in dieser Arbeit genutzt wird, ist der eingebaute Building-Editor. Dieser erlaubt die Erstellung von häuslichen Umgebungen. Umgebungen bestehen aus Modellen wie Wänden und Türen. Diese können in einer Gazebo-Welt simuliert werden. Diese Funktionalität wird innerhalb der Arbeit genutzt, um die Testumgebungen bzw. Testwelten für die Bewertung der Pfadplaner zu erstellen. Ein Ausschnitt des Building-Editors ist in Abbildung 3.3 dargestellt. Im linken Bereich der Abbildung befinden sich verfügbare Modelle und Texturen. Ausgewählte Modelle werden aus dem linken Bereich in dem oberen zentralen Bereich eingesetzt. Dort befindet sich die 2D-Ansicht der Umgebung. Texturen werden auf Modelle in der darunterliegenden 3D-Ansicht aufgetragen. Ein zu betonendes Merkmal des Building-Editors ist die Möglichkeit, Umgebungen mit mehreren Ebenen zu erstellen. Die erstellten Umgebungen werden als SDF-formatierte Dateien exportiert.

3.2.2 QGroundControl-Bodenstation

Als Bodenstation kommt QGroundControl [34] zum Einsatz. Diese Software wird für die Überwachung von Flugparametern genutzt. Weiterhin erfordern Planer wie die PX4-Avoidance-Pfadplaner [35] eine Verbindung zu einer Bodenstation, damit keine unerwünschte Ausfallsicherung aktiviert wird. Im Kontrast zu der Abbildung 3.1 wird kein Joystick bzw. Gamepad an die Bodenstation angebunden.
In Abbildung 3.4 ist QGroundControl während der Simulation eines Pfadplaners zu sehen. Zentral in der Abbildung ist ein Pfeil, welcher das UAV darstellt, abgebildet. Die rote Linie zeigt den geflogenen Pfad an. Im unteren Bereich befinden sich Flugdaten. Links in der

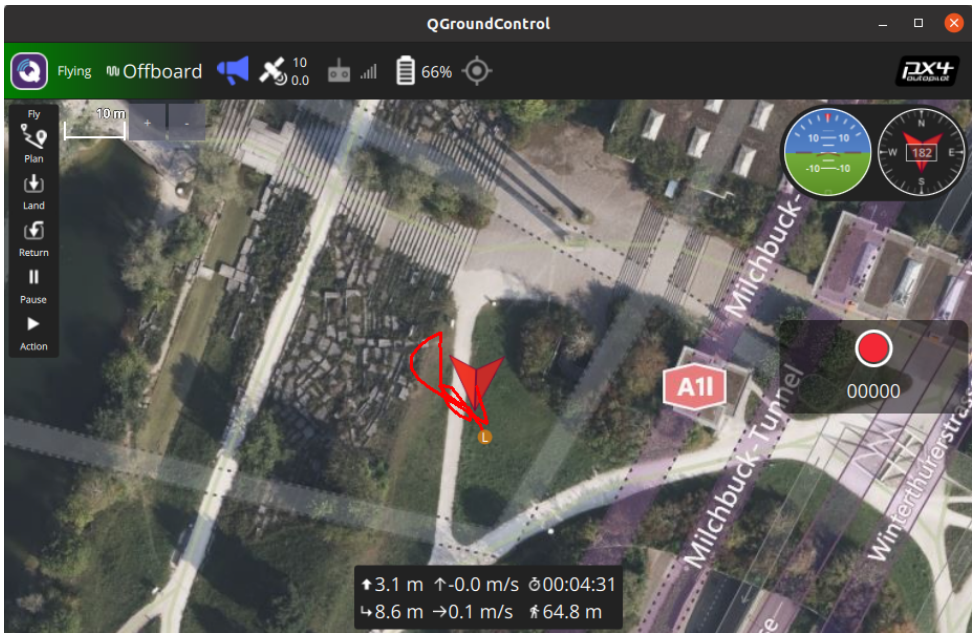


Abbildung 3.4: Übersicht QGroundControl während aktiver Simulation

GUI gibt es auswählbare Schaltflächen für die Interaktion mit dem UAV. Beispielsweise können Anweisungen für den Start oder das Landen des UAV gestartet werden.

3.2.3 ROS-API

Zuletzt wird als API das ROS-*Framework*[1] [36] angebunden. ROS wird als Schnittstelle für die Kommunikation genutzt. Diese wird verwendet, um entwickelte Software mit dem simulierten Luftfahrzeug interagieren zu lassen. Insbesondere die in dieser Arbeit untersuchten Pfadplaner nutzen diese Schnittstelle. Zudem sind alle Planer, die in dieser Arbeit anhand von Simulationen evaluierten werden, jeweils eine Sammlung von quelloffenen ROS-Paketen.

ROS arbeitet mit *Nodes* und *Nodelets*. *Nodes* bezeichnet Prozesse im ROS-*Framework* [37]. *Nodelets* sind *Nodes*, welche *zero-copy*-Operationen erlauben. Dabei existieren mehrere Ausführungsstränge innerhalb eines Prozesses, wodurch die Kommunikation von Daten zwischen den Ausführungssträngen ohne einen Kopiervorgang durchgeführt werden kann, da sich diese den Prozessspeicher teilen [38].

Die Kommunikation zwischen Nodes bzw. Nodelets kann entweder synchron oder asynchron erfolgen. Für die synchrone Kommunikation werden ROS-Services, kurz Services genannt, verwendet. Die asynchrone Kommunikation wird mittels ROS-*Topics*, kurz *Topics* genannt, durchgeführt. Dabei eignen sich *Topics* für die unidirektionale und durchgängige Kommunikation mit höheren Frequenzen. *Services* hingegen eigenen sich für ereignisbasierte

[1] engl. für Rahmenwerk

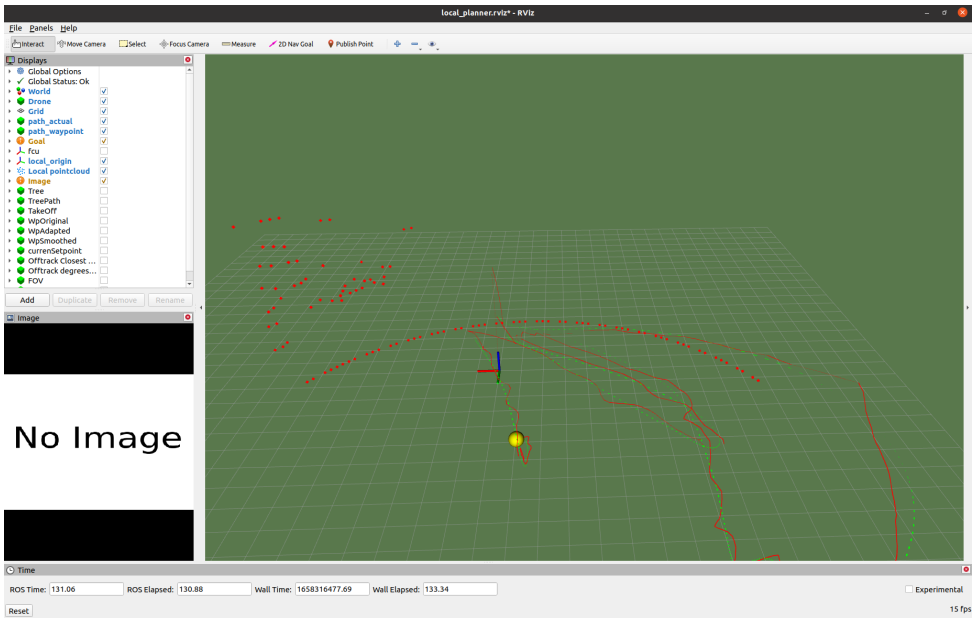


Abbildung 3.5: Übersicht von RViz während aktiver Simulation. Das Luftfahrzeug wird nicht visualisiert.

Kommunikation. Dies liegt daran, dass *Services* nach dem bidirektionalen Anfrage-Antwort-Prinzip arbeiten. Entsprechend sind *Services* im Gegensatz zu *Topics* blockierend, erlauben jedoch die Rückgabe von Daten.
Außerdem bietet ROS zusätzliche Pakete an. Eines davon ist RViz. [39] Dies ist ein ROS-Paket, welches eine modulare und anpassbare grafische Oberfläche zur Verfügung stellt. Diese wird von den untersuchten Pfadplanern genutzt, um interaktiv Flugziele zu definieren, oder Sensordaten zu visualisieren. Abbildung 3.5 ist eine Konfiguration von RViz, die von dem lokalen Planer des PX4-Avoidance Pakets genutzt wird. Auffällig ist das große mittlere Fenster. Darin werden Sensordaten, die geplante Flugroute, die tatsächliche Flugroute sowie das Ziel dargestellt. Die Sensordaten werden als einzelne rote Punkte visualisiert. Besonders im oberen linken Bereich des Fensters sind viele Messpunkte der verwendeten LiDAR-Sensorik sichtbar. Die grüne gestrichelte Linie ist der tatsächliche Pfad. Die daran meist naheliegende rote gestrichelte Linie zeigt den geplanten Pfad. Die gelbe Kugel bildet das Ziel. Dieses Ziel kann innerhalb der GUI interaktiv in zwei Dimensionen verschoben werden. Ein weiteres Fenster ist unten links sichtbar. In diesem Fenster werden Kamerabilder dargestellt, wenn welche vorhanden sind.

3.2.4 Kommunikationsprotokolle

Die Kommunikation wird über verschiedene Protokolle durchgeführt. Dies wird in der Abbildung 3.1 dargestellt. Die Kommunikation zwischen PX4 und dem Gazebo-Simulator

basiert auf dem **T**ransmission **C**ontrol **P**rotocol (TCP). Die Kommunikation zwischen PX4 und der QGroundControl-Bodenstation sowie der ROS-API basiert auf dem **U**ser **D**atagram **P**rotocol (UDP). Darauf aufbauend sind die Nachrichten ROS-API **M**icro **A**ir **V**ehicle **Link** (MAVLink)-formatiert. MAVLink ist ein Nachrichtenprotokoll für die Kommunikation mit Drohnen [40].

3.2.5 Übersicht verwendeter Softwarekomponenten

Die folgende Tabelle 3.2 beinhaltet die innerhalb dieser Arbeit maßgeblich genutzten Softwarekomponenten und deren primär genutzte Funktionalität sowie Versionsnummern.

Tabelle 3.2: Übersicht verwendeter Softwarekomponenten für die Simulationen der behandelten Pfadplaner

Funktion[1]	**Bezeichnung der Komponente**	**Version**
Flugsteuerung	PX4 [32]	v1.12.03
Simulation	Gazebo [33]	11.11.0
Visualisierung	RViz [39]	1.14.14.
Interprozesskommunikation[2]	ROS (1) [36]	Noetic
Betriebssystem	Ubuntu [41]	20.04
Bodenstation	QGroundControl [34]	4.2.3

3.3 Behandelte Pfadplaner

In diesem Abschnitt werden die vier in der vorliegende Arbeit untersuchten und evaluierten Pfadplaner-Lösungen vorgestellt. Für die unterschiedlichen Lösung werden zwecks einer erleichterten Unterscheidung identifizierende Bezeichnungen genutzt. Die folgende Aufzählung nennt die untersuchten Pfadplaner sowie die Bezeichnung innerhalb der Arbeit:

[1] Hierbei wird die primäre Funktion angegeben.
[2] Weiterhin wird ROS als strukturelles Rahmenwerk für wiederverwendbare Softwarekomponenten genutzt.

- Der lokale Planer aus dem PX4-Avoidance Repository [35] wird behandelt und als lokaler PX4-Planer bezeichnet. Dessen Funktionsweise wird in der Arbeit [19] vorgestellt.
- Der globale Planer aus dem PX4-Avoidance Repository [35] wird behandelt und als globaler PX4-Planer bezeichnet. Dessen Funktionsweise wird in der Arbeit [42] vorgestellt.
- Der Planer aus dem gbplanner_ros Repository [6] wird behandelt und als CERBERUS-Planer bezeichnet. Dessen Funktionsweise wird in den Arbeiten [43] und [44] vorgestellt.
- Der Planer aus dem mbplanner_ws Repository [45] wird behandelt und als MB-Planer bezeichnet. Dessen Funktionsweise wird in der Arbeit [46] vorgestellt.

Diese Pfadplaner-Lösungen werden in gleicher Reihenfolge in den folgenden Unterabschnitten behandelt. Es werden die vier Pfadplaner mit der zugehörigen Visualisierung vorgestellt. Dabei ist anzumerken, dass die Visualisierungen sich voneinander stark unterscheiden. Die relevanten Merkmale der Visualisierungen werden erläutert.

3.3.1 Grundlagen der PX4-Avoidance-Pfadplaner

Das PX4-Avoidance Repository besteht aus drei Pfadplanern. Es gibt

1. den *local_planner*,
2. den *global_planner* und
3. den *safe_landing_planner*,

von denen die ersten zwei behandelt werden. Der dritte ist auf den Landungsprozess spezialisiert und daher nicht für die Untersuchungen in der vorliegenden Arbeit geeignet. Der *local_planner* wird im Folgendem als lokaler Planer bezeichnet. Dieser ist ein **V**ector **F**ield **H**istogram (VFH)+* basierter Planer [35], dessen Funktionsweise in der bereits in Abschnitt 2.2 behandelten Arbeit von Baumann, T. [19] dargelegt wird. Dieser VFH+*-Algorithmus basiert auf der Potentialfeldmethode. Da es sich um einen lokalen Planer handelt, wird keine Karte der Umgebung erstellt. Jedoch werden bei dieser Implementierung nicht lediglich die aktuellen Sensorinformationen, sondern auch kürzlich vergangene Sensorinformationen berücksichtigt. In Abbildung 3.7 wird der aktive lokale Planer dargestellt. Die benötigten Sensorinformationen sind Distanzpunkte. In der Umsetzung aus dem

PX4-Avoidance Paket besteht die Sensorik aus Stereokameras. Das verwendete Luftfahrzeug wird als Iris-3Cam bezeichnet. Dieses wird in Abbildung 3.6 visualisiert. An diesem Luftfahrzeug befinden sich drei Stereokameras.

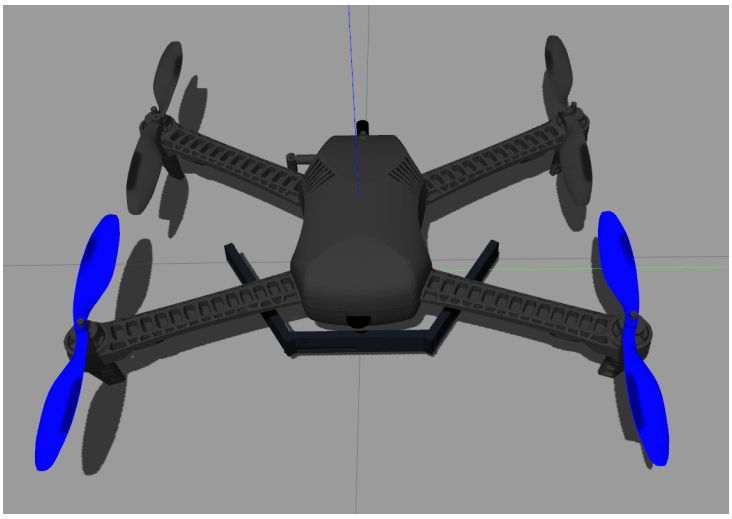


Abbildung 3.6: Darstellung des Iris-3Cam-UAS in Gazebo. Das Luftfahrzeug ist von der vorderen Seite zu sehen. Es sind drei Stereokameras unter den vorderen Armen des Luftfahrzeugs angebracht.

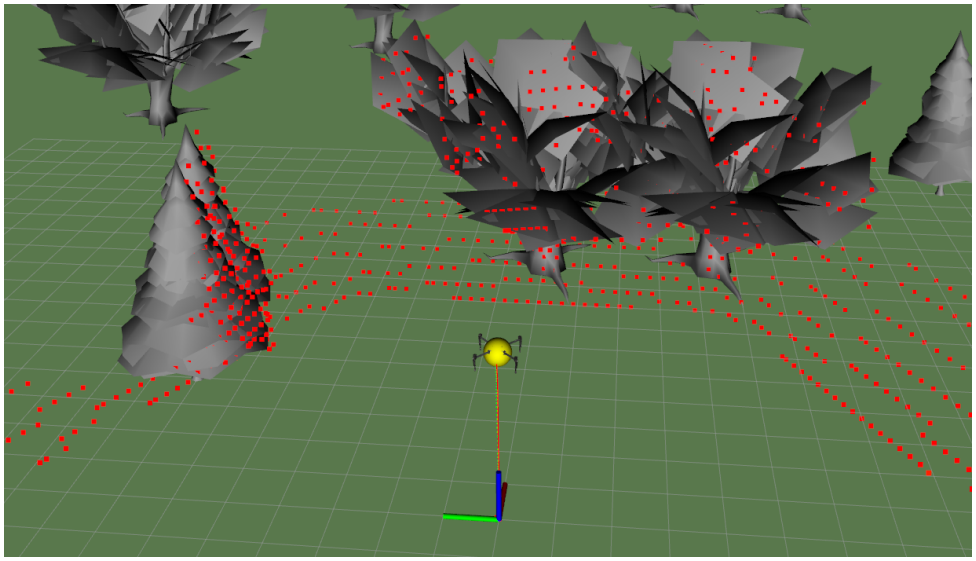
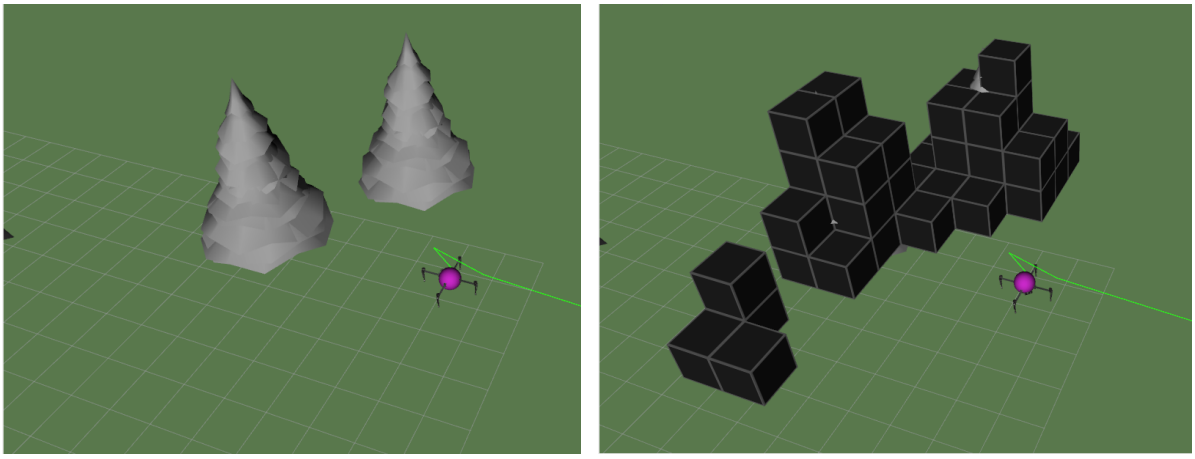


Abbildung 3.7: Ausschnitt der Visualisierung mittels RViz des lokalen Planers aus dem PX4-Avoidance Paket. Die Umgebung und Sensorwahrnehmung des Iris-3Cam-UAS werden dargestellt. Die Drohne schwebt in der Nähe von Bäumen. Die gelbe Sphäre stellt die Zielposition dar.

Der *global_planner* wird im Folgenden als globaler Planer bezeichnet. Die Funktionsweise von diesem Planer wird in [42] vorgestellt [35]. Dieser globale Planer erstellt die Umgebung mithilfe von OctoMap. OctoMap ist ein Framework für die Erstellung von 3D-Karten [47]. Eine Gegenüberstellung der visualisierten OctoMap in einer Simulation ist in Abbildung 3.8

gegeben. Aufbauend auf der 3D-Karte wird der A*-Algorithmus für die Ermittlung eines Pfades eingesetzt. Bei dem globalen Planer kommt die gleiche Drohne und Sensorik zum Einsatz wie bei dem lokalen Planer.



(a) Visualisierung der Umgebung mit OctoMap-Visualisierung ausgeschaltet.

(b) Visualisierung der Umgebung mit OctoMap-Visualisierung eingeschaltet.

Abbildung 3.8: Ausschnitt der Visualisierung mittels RViz des globalen Planers aus dem PX4-Avoidance Paket. Die Drohne schwebt vor zwei Bäumen. Die pinke Sphäre stellt die Zielposition dar. Die Iris-Drohne befindet sich auf der Zielposition. Der grüne Pfad stellt den geflogenen Pfad bis zu der Ist-Position dar. Die schwarzen Blöcke stellen die OctoMap dar.

3.3.2 Grundlagen des CERBERUS-Pfadplaners

Der CERBERUS-Pfadplaner ist im Rahmen der DARPA-*Subterranean Challange* entstanden [44]. In der vorliegenden Arbeit wird der Pfadplaner nach dem entwickeltem Team als CERBERUS-Pfadplaner bezeichnet. Das Ziel der *Challenge* war die Entwicklung von innovativen Technologien für den Einsatz in komplexen unterirdischen Tunnelsystemen [48]. Eine der Kernaufgaben war die Umsetzung von autonomer Navigation, Kartierung und Suche in den komplexen Umgebungen. Der von dem Team CERBERUS entwickelte Pfadplaner hat in dieser *Challenge* den ersten Platz erreicht. Aufgrund der Konzeptionierung für komplexe Tunnelsysteme wird vermutet, dass dieser Pfadplaner auch in häuslichen Umgebungen funktionieren kann. Eine in der DARPA-*Subterranean Challenge* genutzte Gazebo-Welt für das Testen von Algorithmen ist in Abbildung 3.9 gezeigt. Der aktive CERBERUS-Pfadplaner ist in Abbildung 3.10 visualisiert. Es handelt sich um das gleiche Tunnelsystem wie in der Abbildung 3.9. Neben der Visualisierung mittels Gazebo gibt es die Visualiserung von Sensordaten mithilfe von RViz. Ein Ausschnitt von der gleichen Situation wie in Abbildung 3.10 ist in Abbildung 3.11 zu sehen. Die Kamera befindet sich weiter von dem UAS entfernt als in der vorherigen Grafik.

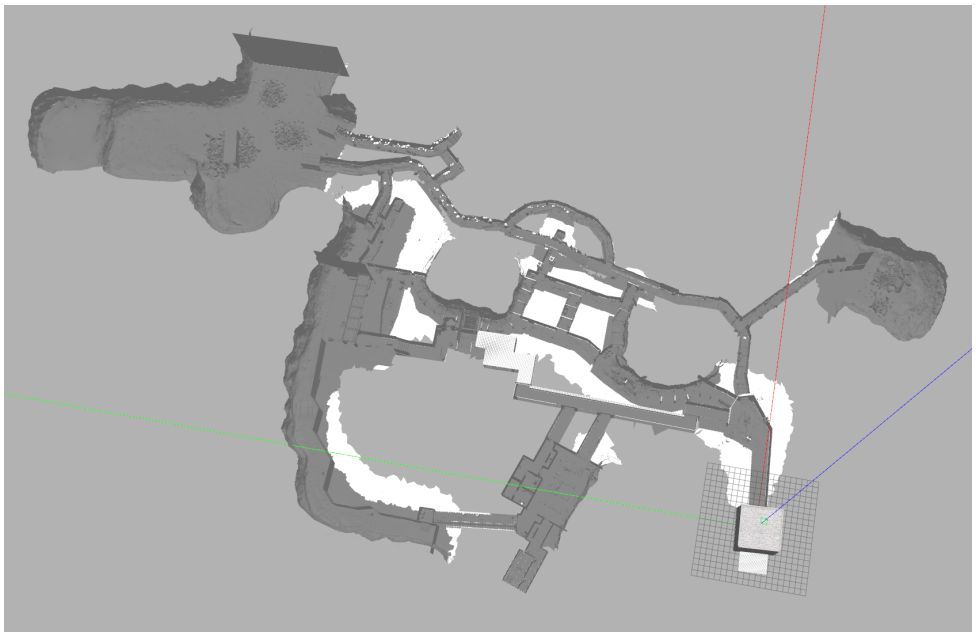


Abbildung 3.9: Ausschnitt einer in Gazebo geöffneten Welt. Diese Umgebung wurde bei der DARPA-*Subterranean Challenge* für das Testen von Algorithmen verwendet [6]. Zu sehen ist ein Überblick eines Tunnelsystems.

Das verwendete UAS unterscheidet sich von dem des PX4-Pfadplaner aus dem Unterabschnitt 3.3.1. Das UAS-Modell hat die Bezeichnung *rmf_obelix* und ist mit einem LiDAR-Sensor ausgestattet.
Der Pfadplanungsalgorithmus ist ein fusionsbasierter Ansatz. Das Ziel der Umsetzung ist die effiziente Exploration der Umgebung [44]. Es wird ein lokaler und ein globaler Pfadplaner kombiniert. Der lokale Pfadplaner ist solange aktiv, bis dieser keine ausreichend hohe Exploration erreicht. Der für den lokalen Pfadplaner verwendete Suchalgorithmus ist der Dijkstra-Algorithmus. Der Dijkstra-Algorithmus wird auf einen Graphen angewandt, welcher nach dem **R**apidly Exploring **R**andom **G**raphs (RRG)-Algorithmus dicht besetzt aufgebaut wird. Der RRG-Algorithmus ist ein stichprobenbasierter Ansatz für die inkrementelle Erweiterung der Stichproben in Richtung des Zieles [49]. Zudem beachtet der lokale Planer die Limitierungen des Luftfahrzeugs [44]. Wenn die Betriebszeit des Luftfahrzeugs ausgeschöpft ist, wird der globale Pfadplaner aktiv, welcher das Luftfahrzeug zu der Start-Position zurückführt.
Der globale Planer arbeitet mit einem dünn besetzten Graphen von Knoten [44]. Die Knotenpunkte werden nach einer Filterung von dem lokalen Planer übernommen und langfristig abgelegt [43]. Der Suchalgorithmus für den globalen Planer ist ebenfalls der Dijkstra-Algorithmus.

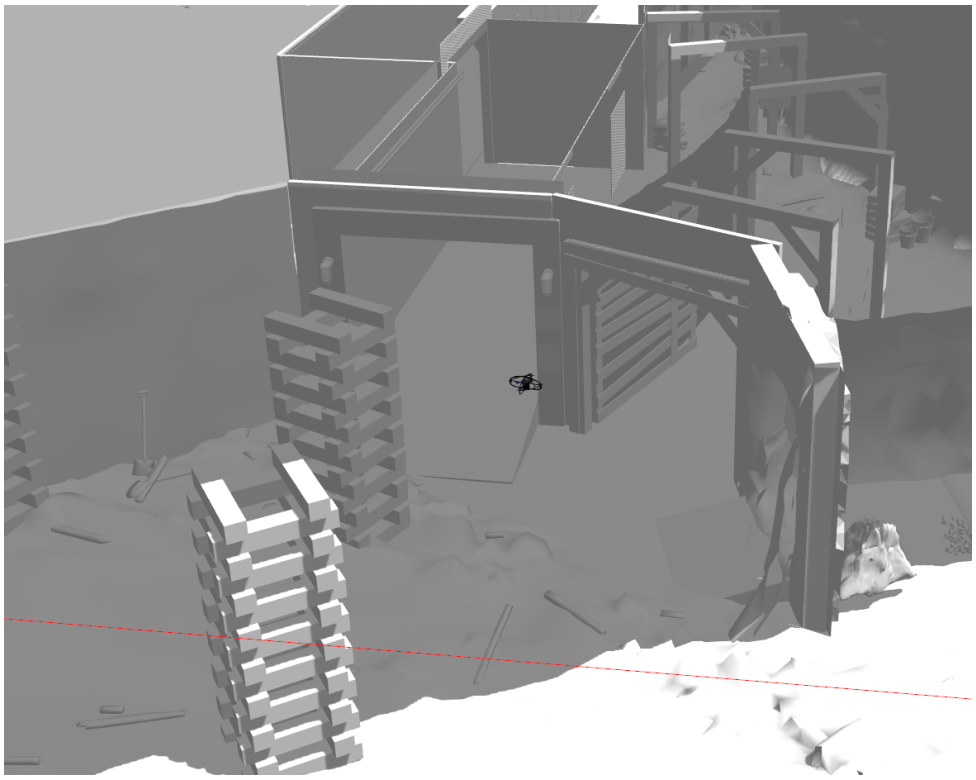


Abbildung 3.10: Ausschnitt aus Gazebo bei aktivem CERBERUS-Pfadplaner. In der Mitte der Abbildung ist das UAS sichtbar. Die Umgebung beinhaltet verschiedene Hindernisse.

3.3.3 Grundlagen des MB-Pfadplaners

Der aus dem mbplanner_ws Repository [45] genutzte Pfadplaner ist ebenfalls aus der DARPA-*Subterranean Challenge*. Die theoretischen Grundlagen des Planers werden in [46] vorgestellt. Es handelt sich ebenfalls um einen explorierenden Pfadplaner. Dieser ist für die gleichen Umgebungen konzipiert wie der in Unterabschnitt 3.3.2 beschriebene CERBERUS-Planer. Entsprechend wird auch bei diesem Pfadplaner vermutet, dass dieser für Innenräume geeignet ist. Der Pfadplaner hat einen stichprobenbasierten Ansatz [46]. Das verwendete UAV wird als *m100* bezeichnet. Es ist mit einem LiDAR-Sensor ausgestattet. Das Luftfahrzeug ist in Abbildung 3.12 während einer Simulation dargestellt. Die zugehörigen Sensordaten des UAS werden in Abbildung 3.13 visualisiert. Der Sensor hat eine Auflösung von $440 \cdot 16$ Datenpunkten.
Die verfügbaren Kommunikationsschnittstellen sind mit denen des CERBERUS-Planers identisch.

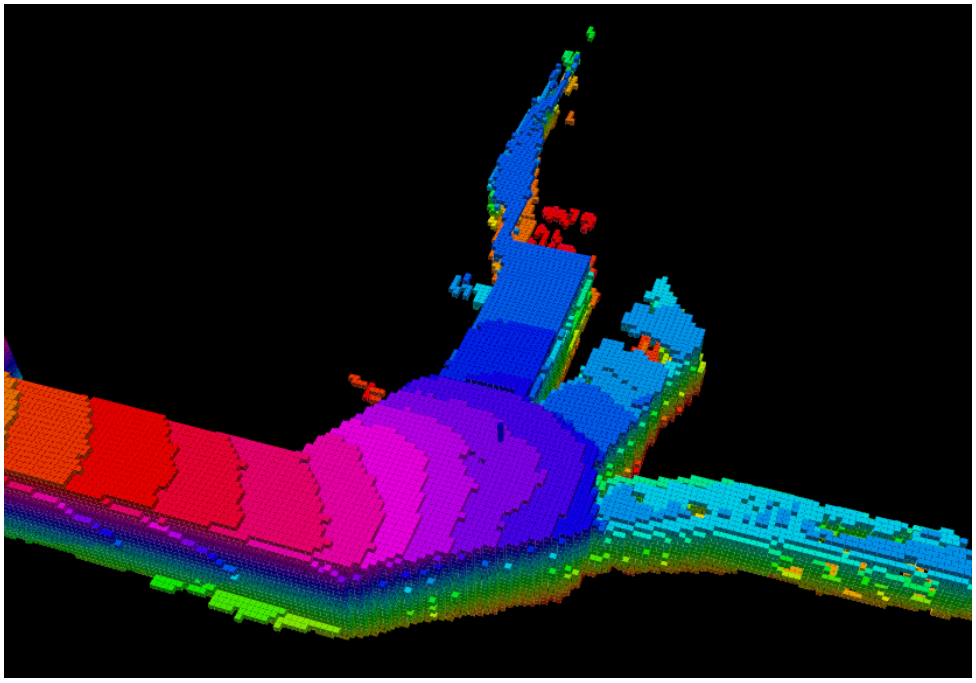


Abbildung 3.11: Ausschnitt aus RViz bei aktivem CERBERUS-Pfadplaner. Sichtbar sind erkannte Hindernisse, welche als farbige Quadrate in RViz dargestellt werden. Die einzelnen Tunnel in des Tunnelsystems sind sichtbar.

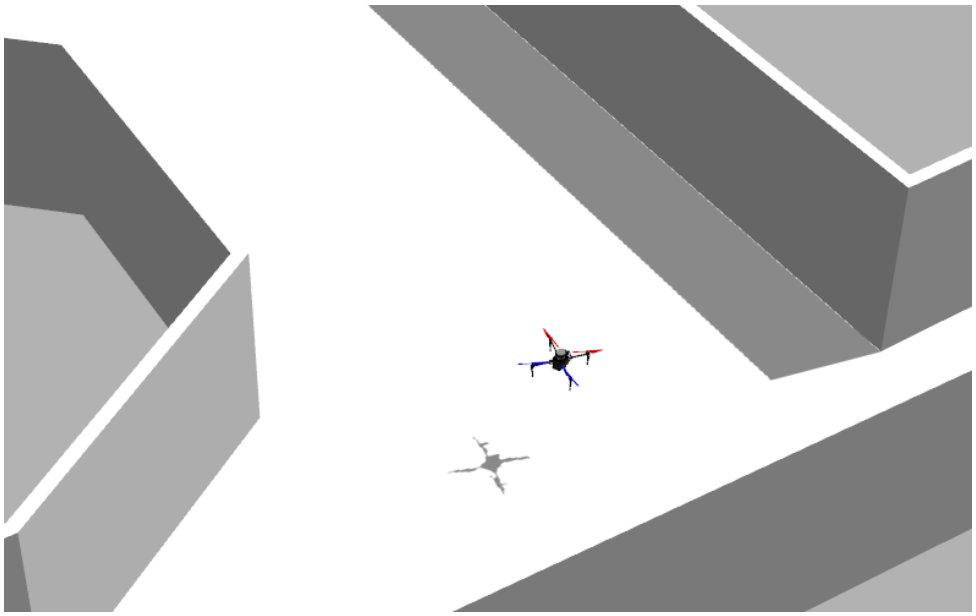


Abbildung 3.12: Ausschnitt aus Gazebo bei aktivem MB-Pfadplaner. In der Mitte der Abbildung ist das UAS sichtbar. Die Umgebung besteht aus einem Boden und mehreren Wänden.

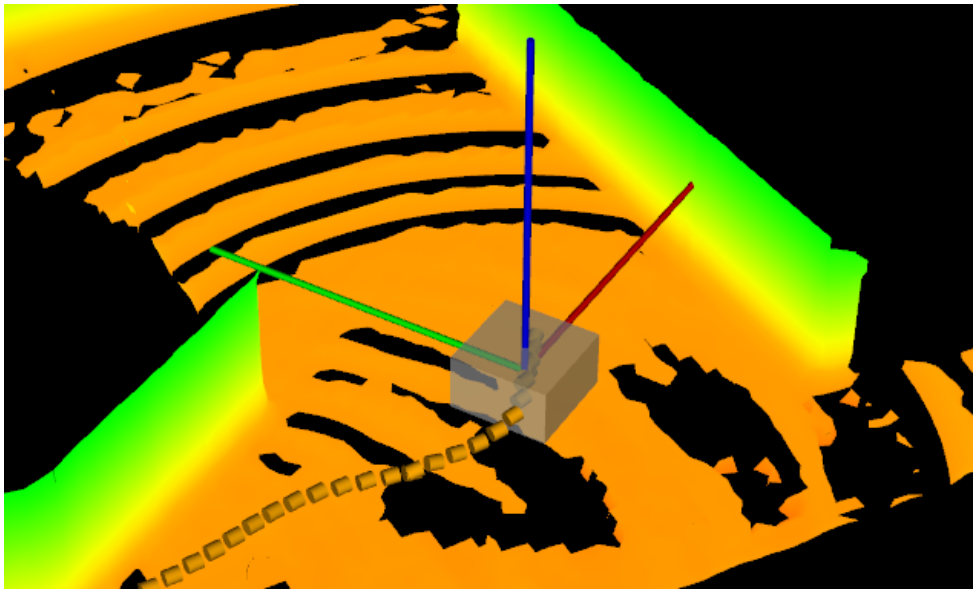


Abbildung 3.13: Ausschnitt aus RViz bei aktivem MB-Pfadplaner. Sichtbar sind erkannte Hindernisse, welche farbig dargestellt werden. Das UAV wird mit drei Koordinatenachsen sowie durch ein Quader dargestellt. Der geflogene Pfad wird mithilfe von orange-farbigen Zylindern visualisiert.

4 Umsetzung

In diesem Kapitel wird die Evaluation durchgeführt. Die Umsetzung ist wie folgt gegliedert:

- Abschnitt 4.1 beschäftigt sich mit der Inbetriebnahme der quelloffenen Implementierungen von Pfadplanern.
- Abschnitt 4.2 beschäftigt sich mit dem Entwurf der Kriterien für die Bewertung von Pfadplanungsalgorithmen.
- Abschnitt 4.3 beschäftigt sich mit der Konzeptionierung der Methoden für die Erfassung der im vorherigen Abschnitt entworfenen Kriterien.
- Abschnitt 4.4 beschäftigt sich mit der Evaluation der Pfadplanungsalgorithmen mittels der erstellten Kriterien und den zugehörigen Methoden der Erfassung.
- Abschnitt 4.5 beschäftigt sich mit einer gewichteten anwendungsspezifischen Auswertung der Evaluation der Pfadplanungsalgorithmen.

4.1 Auswahl, Analyse und Modifikation quelloffener Implementierungen von Pfadplanern

In diesem Abschnitt wird die Umsetzung der Evaluation begonnen. Ziel dieses Abschnitts sind vier quelloffene Pfadplaner, welche in weiteren Verlauf der vorliegenden Arbeit evaluiert werden. Das Vorgehen gliedert sich wie folgt:

- Zunächst werden in Unterabschnitt 4.1.1 vier Pfadplaner ausgewählt. Dies geschieht nach der Darlegung der kritischen Erkenntnisse aus der initialen Recherche.
- Die Unterabschnitte 4.1.2 bis 4.1.5 analysieren und modifizieren die einzelnen ausgewählten Pfadplaner. Zunächst werden jeweils die Funktionalitäten betrachtet und anschließend die kommunikativen Schnittstellen identifiziert. Herausstechende Merkmale der Pfadplaner werden erklärt und diskutiert. Zudem werden Veränderungen, die an den Pfadplanern bzw. an den genutzten Luftfahrzeugen durchgeführt werden,

dokumentiert. Die Veränderungen die durchgeführt werden, haben das Ziel, die Anwendbarkeit der Pfadplaner in Innenräumen zu verbessern.

- Im dem Unterabschnitt 4.1.6 werden zunächst die kritischen Funktionalitäten aller Pfadplaner tabellarisch gegenübergestellt. Die Auswirkungen der Unterschiede werden hinsichtlich der Evaluation diskutiert. Anschließend werden die kommunikativen Schnittstellen der Pfadplaner dargelegt und Unterschiede zwischen diesen dargelegt. Auch die Auswirkungen der unterschiedlichen Kommunikationsschnittstellen werden diskutiert.

4.1.1 Auswahl quelloffener Implementierungen von Pfadplanern

Der von Baumann, T. [19] vorgestellte lokale Planer basiert auf der Potentialfeldmethode. Dieser erweist sich als funktional. Hindernisse werden gemieden und ein kollisionsfreier Pfad zum Ziel kann ermittelt und verfolgt werden. Diese Schlussfolgerung wird basierend auf simulierten sowie physischen experimentellen Untersuchungen im Außenbereich gezogen. Die von Rocha, L. et al. [4] durchgeführten Untersuchungen beinhalten den APF-Algorithmus. Dieser arbeitet wie der von Baumann, T. [19] vorgestellte Planer nach der Potentialfeldmethode. In den Untersuchungen von Rocha, L. et al. [4] hat der APF-Algorithmus keinen Pfad in dem komplexeren Innenbereich ermitteln können. Dies führt zu der Annahme, dass in der Außenumgebung erfolgreich getestete Planer in Innenraumumgebungen nicht erfolgreich sein werden. Dies ist auf die Komplexität der Umgebungen und als Konsequenz davon der Komplexität des benötigten Pfades zurückzuführen.
Basierend auf den geschilderten Recherchen sowie des vorangegangenen Vergleichs werden einige Vermutungen bezüglich des Verhaltens von verschiedenen Arten von Planern in einer Innenraumumgebung gemacht. Es werden im Folgenden lokale Planer und informierte bzw. uninformierte globale Planer betrachtet.
Für die vorliegende Arbeit macht der Vergleich deutlich, dass vor allem lokale Pfadplaner für die in Kapitel 1 geschilderten Anwendungsfälle nicht praktikabel sind. Lokale Planer arbeiten ohne die Erstellung einer Karte von der Umgebung.
Um diese Aussage mit einem theoretischen Beispiel zu untermauern, wird angenommen, dass ein auf dem UAS ausgeführter Planer einen Pfad in einer Etage von einem Wohnblock mit mehreren Wohnungen ermittelt. Der Startpunkt ist in der Wohnung 1 und der Zielpunkt ist in der Nachbarwohnung 2. Die Wahrscheinlichkeit ist hoch, dass zunächst das UAV in der Lage ist, sich dem Zielknoten zu nähern. Dies ist möglich, bis die Wand angetroffen wird, welche die beiden Wohnungen trennt. An dieser Stelle ist der lokale Planer nicht in der Lage näher an das Ziel zu kommen. Die einzige Möglichkeit an den Zielknoten in der Wohnung 2 zu gelangen, ist ein Umweg, welcher es erfordert, den Abstand zum Zielknoten temporär zu erhöhen. Dies wird von dem lokalen Planer jedoch nicht

erlaubt. Im Fall eines lokalen Planers, der auf der Potentialfeldmethode basiert, würden sich die anziehenden Kräfte des Zielknoten mit den abstoßenden Kräfte der Wand zu Null summieren. Das bedeutet, dass der Planer an der Stelle in ein lokales Minimum fallen würde.
In dem gleichen Szenario wäre ein informierter globaler Planer in der Lage, einen Pfad zu ermitteln. Bei einem derartigen Planer würde erkannt werden, dass der direkt auf das Ziel zusteuernder Pfad nicht zum Ziel führt. Jedoch kann in realen Anwendungsfällen nicht davon ausgegangen werden, dass die Umgebungsinformationen von Anfang an bekannt sind. Daher wäre ein informierter globaler Planer nicht immer anwendbar.
Bei einem uninformierten globalen Planer wird vermutet, dass dieser ein ähnliches Verhalten wie der lokale Planer aufweist. Alternativ könnte der Planer anfangen, die Umgebung zu erkunden. Erkunden bedeutet, den Pfad zunächst über unbekannte Gebiete zu planen, bis dort über die Sensorik erkannt wird, dass dieser Pfad ebenfalls nicht passierbar ist. Anschließend wird dies wiederholt. Wenn dieser Planer nicht auf maximalen Informationsgewinn optimiert ist, wird dieser Prozess sehr ineffizient. Dieser Vorgang könnte auch bedeuten, dass das Luftfahrzeug über bereits bekannte Stellen hin und her fliegt.
Das angenommene Szenario macht deutlich, dass Pfadplaner mit hoher Wahrscheinlichkeit in komplexen Innenräumen entweder suboptimale oder keine Lösungen ermitteln. Die Ausnahme bildet der informierte globale Planer. Jedoch ist in der Realität die Wahrscheinlichkeit gering, dass die Umgebungsinformationen in allen Anwendungsfällen vorliegen. Entsprechend ist auch dieser nicht immer anwendbar. Neben den behandelten Pfadplanern gibt es Pfadplaner, die auf für Exploration von Umgebungen optimiert sind. Bei derartigen Umsetzungen von Pfadplanern wird die Umgebung mit dem Ziel von maximalen Informationsgewinn erkundet [43]. Diese explorierenden Pfadplaner können eine Lösung für die Problematik von uninformierten globalen Pfadplanern sein, die die Umgebung nicht systematisch erkunden.
Basierend auf den vorangegangenen Betrachtungen werden die folgenden Pfadplanern für die Evaluation ausgewählt:

- der lokale PX4-Planer, welcher in Unterabschnitt 3.3.1,
- der globale PX4-Planer, welcher ebenfalls in Unterabschnitt 3.3.1 vorgestellt wird,
- der explorierende globale CERBERUS-Planer, welcher in Unterabschnitt 3.3.2 vorgestellt wird und
- der explorierende globale MB-Planer, welcher in Unterabschnitt 3.3.3 vorgestellt wird.

Von dem lokalen Planer wird erwartet, dass dieser in Innenräumen als Gesamtlösung nicht funktionieren wird. Dieser Vermutung basiert auf der Tatsache, dass es sich um einen

APF-basierten Planer handelt und diese für lokale Minima anfällig sind. Jedoch könnte dieser Planer bestimmte Kriterien besser erfüllen als die anderen Planer. Entsprechend würde der lokale Planer einen realistischen Referenzwert für das betrachtete Kriterium liefern.
Der globale Planer ist kein explorierender Planer. Entsprechend wird vermutet, dass dieser bei wenigen Hindernissen zum Ziel gelingen kann. Weiterhin wird angenommen, dass die Problematik der lokalen Minima bei diesem Pfadplaner nicht vorzufinden ist. Es kann jedoch nicht erahnt werden, ob ein valider Pfad in komplexen Innenraum-Umgebungen ermittelt werden kann. Die Leistungsanforderungen an die Hardware sollte aufgrund der zu erstellenden Abbildung der Umgebung des Planers höher sein als die des lokalen Planers.
Neben diesen zwei Planern werden die zwei explorierenden Pfadplaner untersucht. Beide Planer stammen aus der DARPA-*Subterranean Challange* und wurden für komplexe Umgebungen erstellt. Bei diesen beiden Planern wird erwartet, dass diese in den komplexen Innenräumen operieren können. Da diese für den autonomen Betrieb konzipiert sind, wird weiterhin vermutet, dass die Ausführung mehr Computerressourcen benötigt als die zwei anderen Pfadplaner.
Mit den vier ausgewählten Pfadplaner wird ein großes Spektrum von Ansätzen von Pfadplanern abgedeckt. Es werden lokale, globale und explorierende globale Pfadplanerumsetzungen untersucht. Die Auswahl von den verschiedenen Umsetzungen stellt die Anforderung an die Metrik, dass diese allgemeine Kriterien enthält, die die Vergleichbarkeit von derartig unterschiedlichen Ansätzen erlaubt.
Die aufgestellten Vermutungen werden in Abschnitt 4.4 im Rahmen der Evaluation der Pfadplaner aufgegriffen und beantwortet.

4.1.2 Funktionalitäten und Schnittstellen des lokalen PX4-Pfadplaners

In diesem Unterabschnitt wird der lokale PX4-Pfadplaner untersucht und modifiziert. Das Ziel ist diesen Pfadplaner in einen Zustand zu bringen, der die Anwendung im Innenraum erlaubt. Zudem werden die Schnittstellen identifiziert, die für die Interaktion mit Skripten genutzt werden können. Diese Skripte werden für die Evaluation benötigt. Beispielsweise müssen die Positionsdaten des UAS über eine Schnittstelle ermittelbar sein, um berechnen zu können, ob eine Zielposition erreicht ist. In den folgenden Absätzen wird zunächst die Anwendung bzw. die Eigenheiten der Pfadplaner untersucht und es werden gegebenenfalls Modifizierungen vorgenommen. Anschließend werden die Kommunikationsschnittstellen identifiziert.
Der lokale PX4-Planer nutzt PX4 für die Flugsteuerung. Die Flugsteuerung hat einige Modi, in denen das UAS betrieben wird. Für die Vorgabe von Zielkoordinaten muss sich das

UAS im OFFBOARD-Modus befinden. Anschließend können Zielkoordinaten gesetzt werden und der Pfadplaner steuert unmittelbar das UAS zu der Zielposition. Diese Zielposition kann frei gewählt werden. Jedoch kommt es bei Zielpositionen nahe dem Boden zu einem unerwünschten Verhalten des Pfadplaners. Wie in Abbildung 4.1 dargestellt, hat der Pfadplaner eine kreisende Bewegung über dem Ziel durchgeführt.

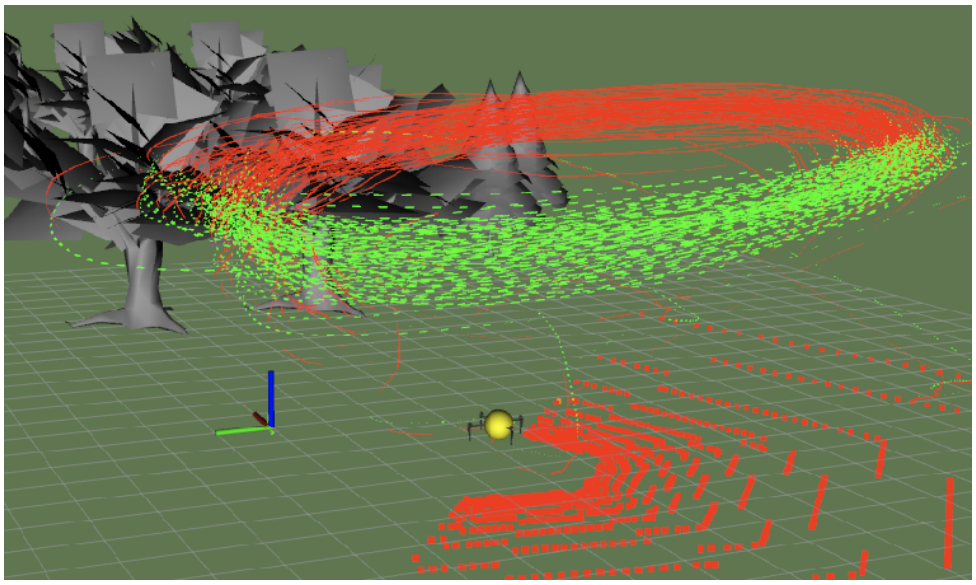


Abbildung 4.1: Ausschnitt des lokalen PX4-Planers in RViz. Das UAS führte eine kreisende Bewegung über der gelb markierten Zielposition aus. Die verfolgte Pfad ist grün dargestellt und der geplante Pfad ist rot dargestellt.

Die kreisende Bewegung wurde einige Minuten lang durchgeführt. Schließlich wurde die Bewegung unterbrochen und das UAS zum Ziel bewegt. Dabei war kein menschliches Handeln nötig. Dieses Phänomen konnte wiederholt in verschiedenen Szenarien beobachtet werden. Es konnte auch bei höherem Abstand zum Boden beobachtet werden.
Es ist nicht gelungen, die Ursache für dieses Verhalten zu ermitteln. Dennoch ist gelungen, dieses Verhalten zu vermeiden, indem eine andere Umweltwahrnehmung verwendet wird. In Abbildung 4.2 ist das gleiche Luftfahrzeug mit einem LiDAR ausgestattet. Jedoch befindet sich der Sensor nicht unter dem Luftfahrzeug, sondern über diesem. Der LiDAR-Sensor hat einen horizontalen **F**ield **o**f **V**iew (FoV)[1] von 360°. Der lokale PX4-Pfadplaner funktioniert mit diesem Sichtfeld nicht. Daher muss dieses reduziert werden. Mit einem horizontalen FoV von ca. 60° ist der Pfadplaner in der Lage, die Zielposition zu erreichen und dabei keine Kreisbewegungen über diesem zu machen. Dieses Luftfahrzeug wird als Iris-LiDAR bezeichnet.

[1] engl. für Sichtfeld

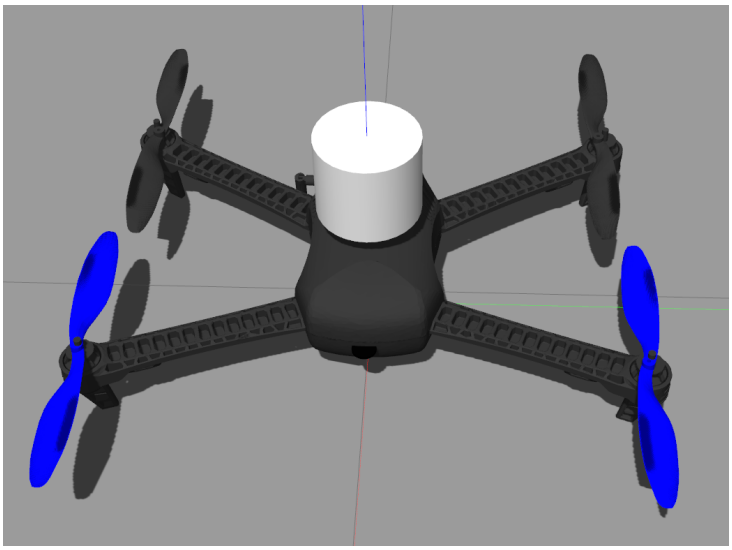


Abbildung 4.2: Darstellung des Iris-LiDAR Luftfahrzeuges. Der Zylinder stellt den LiDAR-Sensor dar.

Der lokale PX4-Pfadplaner erlaubt das setzen von Zielkoordinaten. Diese werden zweigeteilt übergeben. Mittels der RViz-Oberfläche kann interaktiv ein Marker gesetzt werden, der die x- und y-Koordinaten vorgibt. Die z-Koordinate wird über eine RViz-externe grafische Oberfläche, welche auf dem ROS-Paket `dynamic_reconfigure` [50] basiert, übergeben.
Die Kommunikation der x- und y-Koordinate mittels RViz wird mithilfe von ROS-Topics durchgeführt. Diese Kommunikation über Topics muss nicht zwangsläufig über die RViz-Oberfläche durchgeführt werden. Alternativ kann dies gleichermaßen mittels der C++-Schnittstelle von ROS durchgeführt werden. Das bedeutet, dass die Positionsvorgabe der x- und y-Koordinaten über eine Softwareschnittstelle möglich ist. Dies erlaubt das Erstellen von Skripten für diesen Zweck.
Die Kommunikation der z-Koordinate ist nicht mittels Topics möglich. Das für diese Koordinate genutzte ROS-Paket hat keine C++ Schnittstelle [51]. Jedoch kann mit C++ ein System-Funktionsaufruf über die `system`-Funktion von C++ durchgeführt werden, welcher das setzten der z-Koordinate erlaubt. Dadurch ist sichergestellt, dass alle drei Koordinaten in einem C++-Programm gesetzt werden können.
Weiterhin werden von der genutzten PX4-Software die Flugdaten über Topics kommuniziert. Dazu wird die PX4-Software mit dem zusätzlichen MAVROS-Paket [52] verwendet. Dieses Paket betreibt eine Schnittstelle zwischen der PX4-Software und ROS. Von den Kommunizierten Daten sind folgende interessant, um die Erstellung von Skripten für die Evaluation zu ermöglichen:

- der aktuellen Betriebsmodus des UAS,
- die exakte Position von dem Luftfahrzeug sowie

- die Trägheitsmomente der verbauten **I**nertia **M**easurement **U**nit (IMU) des Luftfahrzeugs.

Der Betriebsmodus gibt Auskunft darüber, ob das Luftfahrzeug Zielpositionen verfolgen kann. Die exakte Position des Luftfahrzeugs kann genutzt werden, um zu ermitteln, ob eine Zielposition erreicht ist. Die Daten der IMU beinhalten die Geschwindigkeit des Luftfahrzeugs. Mit den Geschwindigkeitsinformationen kann ermittelt werden, ob das Luftfahrzeug in Bewegung ist.

4.1.3 Funktionalitäten und Schnittstellen des globalen PX4-Pfadplaners

In diesem Unterabschnitt wird der lokale PX4-Pfadplaner untersucht und modifiziert. Wie auch bei der Untersuchung des lokalen PX4-Pfadplaners in Unterabschnitt 4.1.2 ist das Ziel die Anwendbarkeit des Planers und die Identifikation von Schnittstellen für die spätere Erstellung von Skripten für die Evaluation.
Bei diesem Pfadplaner wird ebenfalls das in Abbildung 4.2 dargestellte Iris-Modell mit dem LiDAR-Sensor verwendet. Damit ist gewährleistet, dass alle Pfadplaner mit einer LiDAR-Sensorik ausgestattet sind. Dies verbessert die Vergleichbarkeit der Resultate der verschiedenen Pfadplaner, da alle Pfadplaner eine LiDAR-Sensorik für die Umweltwahrnehmung verwenden.
Die Verwendung des globalen PX4-Pfadplaners ist mit der des lokalen PX4-Pfadplaners sehr ähnlich. Der benötigte Flugmodus ist ebenfalls `OFFBOARD`. Das Prozess für das Setzen der Zielposition ist identisch.
Das Verhalten des globalen Pfadplaners zeigt jedoch auch Unterschiede. Beispielsweise wird keine Zielposition mit einer Höhe von unter $2m$ angeflogen. Für den Betrieb in Innenräumen ist dies jedoch zwingend erforderlich. Für die Lösung dieses Problems wird der Quellcode analysiert und anschließend angepasst. Die Funktion für die Ermittlung des Risikos für Postionen hat eine hartkodierte Mindesthöhe für valide Positionen. Der relevante Ausschnitt dieser Funktion ist in Listing 4.1 gegeben. Die Abfrage setzt den Rückgabewert der Funktion statisch auf 1.0, wenn `cell.zIndex()` kleiner als 1 ist. Es konnte experimentell bestätigt werden, dass der Rückgabewert der Funktion `cell.zIndex()` mit der erlaubten Flughöhe korreliert. Der Vergleich dieses Rückgabewerts mit dem statischen Wert 1 hindert den Pfadplaner daran, eine Zielkoordinate unter diesem Grenzwert zu akzeptieren. Die Verringerung der Mindesthöhe in dem Quellcode löst das Problem. Die modifizierte Abfrage ist in Listing 4.2 gegeben. Der konstante Wert 1 ist zu 0 geändert. Nach dieser Modifizierung kann die Zielposition ab $1m$ Höhe frei gewählt werden. Dies ist ausreichend, um Ziele in Innenräume vorzugeben.

```
133 if (cell.zIndex() < 1 || !octree_) {
134   return 1.0; // Octomap does not keep track of the ground
135 }
```

Listing 4.1: Ausschnitt aus der getSingleCellRisk-Funktion der GlobalPlanner-Klasse.

```
133 if (cell.zIndex() < 0 || !octree_) {
134   return 1.0; // Octomap does not keep track of the ground
135 }
```

Listing 4.2: Ausschnitt aus der modifizierten getSingleCellRisk-Funktion der GlobalPlanner-Klasse.

Im Gegensatz zum lokalen PX4-Planer erlaubt der globale PX4-Planer das Setzen der minimal und maximal erlaubten Flughöhe. In Tests hat sich jedoch gezeigt, dass diese nicht immer eingehalten werden. Da dies jedoch nicht kritisch ist, wird dieses Phänomen nicht weiter analysiert.
Die nächste Auffälligkeit des Planers ist, dass die Auflösung der OctoMap gering ist. Die Auflösung muss erhöht werden, damit enge Durchgänge in Innenräumen als passierbarer Raum für den Pfadplaner nutzbar sind. In einer der Launch-Dateien des Projekts wird der Parameter für die Auflösung der OctoMap festgelegt. Die Datei hat den Namen `global_planner_octomap.launch`. In dieser Datei wird der Parameter mit dem Namen `resolution` von 1.0 auf 0.5 angepasst. Diese Veränderung verdoppelt die Auflösung. Kleinere Werte als 0.5 erweisen sich als problematisch. Beispielsweise konnte mit einem Parameterwert von 0.25 beobachtet werden, dass das UAS über der Start-Position schwebt, jedoch nicht zum Ziel fliegt.
Zudem kann beobachtet werden, dass Werte unter 0.5 die Häufigkeit, mit der das UAS in Hindernisse fliegt, erhöhen. Es wird vermutet, dass die Auflösung des Sensors zu gering ist, um alle Blöcke der OctoMap in einem Bereich als Hindernis zu identifizieren. Das resultiert darin, dass beispielsweise eine Wand nicht vollständig erfasst wird und der Pfadplaner durch eine nonexistente Lücke in der Wand plant. Dies führt zu einer Kollision des Luftfahrzeugs mit der Wand. Der Wert 0.5 wurde experimentell als bester Wert für die Auflösung ermittelt. Die Auflösung ist erhöht und die Häufigkeit, mit der Blöcke der OctoMap fälschlich als frei identifiziert werden, ist gering.
Die Kommunikationsschnittstellen des globalen PX4-Pfadplaner sind mit den des lokalen PX4-Pfadplaners, dessen Schnittstellen in Unterabschnitt 4.1.2 dargelegt werden, identisch. Das Setzen der Zielposition funktioniert gleich. Jedoch unterscheidet sich der Name der z-Koordinate. Dies muss bei der Erstellung eines Skriptes, welches mit dem Pfadplaner interagieren soll, beachtet werden. Es werden keine weiteren relevanten Unterschiede festgestellt.

4.1.4 Funktionalitäten und Schnittstellen des explorierenden CERBERUS-Pfadplaners

In diesem Unterabschnitt wird der explorierende CERBERUS-Pfadplaner untersucht und modifiziert. Wie auch in Unterabschnitt 4.1.2 und Unterabschnitt 4.1.3 ist das Ziel die Anwendbarkeit des Planers zu untersuchen und Schnittstellen für die Erstellung von Skripten, die für die Evaluation benötigt werden, zu identifizieren.
Bei der Untersuchung der Anwendbarkeit des CERBERUS-Pfadplaners fallen einige Merkmale, die für die Methodik der Evaluation bedeutend sind, auf:

- Die Zielposition wird über die RViz-Oberfläche gesetzt. Dies erlaubt das Setzen von Zielkoordinaten mittels der ROS-Schnittstelle. Es wird die x-, y- und z-Koordinaten mittels RViz gesetzt.
- Die Zielpositionen werden nur sehr grob verfolgt. Dies lässt sich damit erklären, dass es sich um einen stichprobenbasierten Pfadplaner handelt. Es wird vermutet, dass bei dem Setzten der Zielposition der nächste verfügbare Knoten der Stichprobe als tatsächliche Zielposition verwendet wird.
- Der Pfadplaner fliegt nicht über Hindernisse, die so groß sind, wie eine Wand. Stattdessen werden die Hindernisse umflogen, wenn dies möglich ist. Dies deutet auf eine maximal zulässige Höhe hin.
- Der Pfadplaner ist in der Lage, Umgebungen autonom zu explorieren.
- Der Pfadplaner hat eine zeitliche begrenzte Betriebszeit. Nach dem Ablauf dieser Zeit führt der Pfadplaner das UAS zurück zu der Anfangsposition.
- Der Pfadplaner hat eine Initialisierungssequenz. Diese lässt das Luftfahrzeug auf eine bestimmte Position fliegen und anschließend dort schweben.
- Es können ausschließlich Zielpositionen aus der direkten Umgebung bzw. aus der bekannten Umgebung verfolgt werden. Das bedeutet, der Pfad zu einer Zielposition außerhalb des bereits erkundeten Gebietes kann nicht ermittelt werden.

In den folgenden Absätzen werden einige der aufgelisteten Punkte und deren Bedeutung für die Evaluation erläutert.
Von diesen Eigenschaften des Pfadplaners ist besonders interessant, dass nur Zielposition aus dem bereits bekannten Gebiet valide sind. Das bedeutet für die Evaluation, dass keine beliebigen Zielkoordinaten gegeben werden können. Bei der Vorgabe von Zielkoordinaten muss sichergestellt werden, dass sich diese im bekannten Gebiet des Planers befinden. Weiterhin ist anzumerken, dass die Planung zu einer Zielposition erst dann möglich ist,

wenn die Erkundung zuvor aktiviert wurde. Es ist nicht möglich, die Initialisierung durchzuführen und anschließend die Planung zu einer Zielposition zu starten. Es muss zwangsläufig eine Exploration angefangen werden, bevor die Planung zu einer Zielposition gestartet werden kann. Die Exploration kann umgehend nach dem Start von dieser beendet werden. Dadurch dass der Pfadplaner in der Lage ist, die Umgebungen autonom zu erkunden, können führende Umgebungen gebaut werden. In Kombination mit der Tatsache, dass Hindernisse nicht überflogen werden, können häusliche Umgebungen erstellt werden, die nur in eine Richtung passierbar sind. So könnte beispielsweise überprüft werden, ob der Pfadplaner durch eine Verengung planen kann.
Weiterhin ist in dieser Arbeit die Beschränkung der Betriebszeit nicht erwünscht, da die Tests beliebig lange dauern dürfen. In der Konfigurationsdatei `gbplanner_config.yaml` für das `rmf_obelix`-Luftfahrzeug wird der Parameter `time_budget_limit` von 480 auf 3600 erhöht. Damit ist die erlaubte Simulationszeit von 480 Sekunden auf 3600 Sekunden bzw. eine Stunde erhöht.
Als Nächstes werden die Kommunikationsschnittstellen des Pfadplaner dargelegt. Der Pfadplaner verwendet nicht die PX4-Flugsteuerung. Das hat zur Konsequenz, dass keine MAVROS-Topics mit Informationen zum Flugmodus zur Verfügung stehen. Jedoch gibt es andere ROS-Topics, welche unter anderen Topic-Namen teilweise ähnliche Daten veröffentlichen. Es ist gelungen, Topics zu identifizieren, welche Positionsdaten sowie IMU-Daten des UAS veröffentlichen. Hingegen konnte keine Topic identifiziert werden, welche den Flugmodus des UAS darlegt. Letzteres hat zur Konsequenz, dass keine Schnittstelle gegeben ist, die Auskunft darüber gibt, ob der Pfadplaner aktuell erkundet oder eine gesetztes Ziel verfolgt. Zusätzlich werden Zielpositionen aufgrund der stichprobenbasierten Art des Planers nur ungenau verfolgt. Diese zwei Eigenschaften erschweren die effiziente Überprüfung, ob der Planer eine vorgegeben Zielkoordinate bereits als erreicht betrachtet. Die Auswirkung davon auf die Evaluation ist, dass Messungen erschwert sind. Beispielsweise könnte eine Zeitmessung nicht durch absolute Positionen ausgelöst werden. Alternativ zu absoluten Positionen können die Geschwindigkeitsdaten der IMU genutzt werden, um festzustellen, ob sich das Luftfahrzeug bewegt. Dadurch könnte beispielsweise eine Zeitmessung beendet werden, sobald eine Geschwindigkeit unterschritten ist.
Zuletzt ist zu erwähnen die Funktionalitäten des Pfadplaners über ROS-Services geschaltet werden. Das bedeutet, dass der Pfadplaner mithilfe der ROS-Schnittstelle gesteuert werden kann. Entsprechend können Skripte für diesen Zweck erstellt werden.

4.1.5 Funktionalitäten und Schnittstellen des explorierenden MB-Pfadplaners

In diesem Unterabschnitt wird der explorierende MB-Pfadplaner untersucht. Wie auch bei den vorangegangen Unterabschnitten ist das Ziel die Anwendbarkeit des Planers zu untersuchen und Schnittstellen für die Erstellung von Skripten, die für die Evaluation benötigt werden, zu identifizieren.
Die Beobachtungen bei der Untersuchungen des MB-Pfadplaner sind mit den Beobachtungen des CERBERUS-Pfadplaners nahezu identisch. Die relevanten Gemeinsamkeiten zwischen MB-Pfadplaner und CERBERUS-Pfadplaner sind:

- Die Zielposition wird über die RViz-Oberfläche gesetzt.
- Die Zielpositionen werden nur sehr grob verfolgt. Auch der MB-Planer ist ein stichprobenbasierter Planer.
- Der Pfadplaner umfliegt Hindernisse auf konstanter Höhe und fliegt nicht über diese.
- Der Pfadplaner ist in der Lage, Umgebungen autonom zu explorieren.
- Der Pfadplaner hat eine identischen Initialisierungssequenz.

Die Unterschiede des MB-Pfadplaners im Vergleich mit dem CERBERUS-Pfadplaners sind:

- Es kann keine Zielposition vorgegeben werden, auch nicht im bereits erkundeten Gebiet.
- Der Pfadplaner hat keine zeitliche begrenzte Betriebszeit.

Die geschilderten Gemeinsamkeiten haben für den MB-Pfadplaner die gleichen Konsequenzen, die für den CERBERUS-Pfadplaner bereits in Unterabschnitt 4.1.4 geschildert sind. Um Wiederholungen zu vermeiden, werden ausschließlich die Konsequenzen der relevanten Unterschiede für die Erstellung des Skripte, die für die Evaluation benötigt werden, diskutiert.
Der MB-Pfadplaner ist der einzige untersuchte Pfadplaner, der nicht die Funktionalität hat, zu einer wählbaren Zielposition ein Pfad zu planen. Es handelt sich um einen ausschließlich explorierenden Pfadplaner. Dies reduziert die Anzahl der Gemeinsamkeiten aller untersuchten Pfadplaner. Entsprechend müssen für die Evaluationen Methoden entwickelt werden, die die Fähigkeit des Pfadplaners so untersuchen, dass dieser mit den anderen vergleichbar ist.
Zuletzt ist zu erwähnen, dass die Funktionalitäten bei dem MB-Pfadplaner wie auch bei dem CERBERUS-Pfadplaner mittels ROS-Services geschaltet werden. Damit ist die

Tabelle 4.1: Übersicht von funktionalen Merkmalen von den untersuchten Pfadplanern. Die Namen der Pfadplaner sind aus räumlichen Gründen verkürzt: lokaler PX4-Pfadplaner (lok. PX4), globaler PX4-Pfadplaner (glob. PX4), CERBERUS-Pfadplaner (CERBERUS) und MB-Pfadplaner (MB). Vorhandene Merkmale werden mit dem Symbol ✓ gekennzeichnet, wenn der entsprechende Pfadplaner dieses Merkmal erfüllt oder mit ✗ gekennzeichnet, wenn der entsprechende Pfadplaner das Merkmal nicht erfüllt.

Merkmal	lok. PX4	glob. PX4	CER-BERUS	MB
Vorgabe von Zielpositionen ist möglich.	✓	✓	✓	✗
Die Verfolgung der Zielposition ist akkurat.	✓	✓	✗	✗
Die Verfolgung der Zielposition ist in unbekannten Gebiet ist möglich.	✓	✓	✗	✗
Die maximale Flughöhe ist limitiert.	✗	✓	✓	✓
Die Funktionalität für eine autonome Exploration der Umgebung ist gegeben.	✗	✗	✓	✓
Der aktuelle Flugmodus des UAS kann abgefragt werden.	✓	✓	✗	✗

Möglichkeit der Steuerung des Pfadplaner mittels einer Software-Schnittstelle basierend auf ROS gegeben.

4.1.6 Zusammenfassung der ermittelten Gemeinsamkeiten und Unterschiede der untersuchten Pfadplaner

In diesem Unterabschnitt wird basierend auf den Untersuchungen aus den Unterabschnitten 4.1.2 bis 4.1.5 die für die Evaluation relevanten Merkmale zusammenfassend und gegenüberstellend dargelegt. Die Struktur der Zusammenfassung orientiert sich an der Struktur der vorangegangenen Unterabschnitte. Zunächst werden funktionale Merkmale hervorgehoben sowie diskutiert und anschließend werden die kommunikativen Schnittstellen der Pfadplaner dargelegt.
Beginnend mit den funktionalen Merkmalen der Pfadplaner ist in der Tabelle 4.1 eine Übersicht gegeben. Die in der Tabelle 4.1 aufgezählten Merkmale zeigen die fundamentalen Gemeinsamkeiten sowie Unterschiede zwischen den untersuchten Pfadplanern. Für die

Evaluation bedeuten die Unterschiede, dass für die Bewertung eines Kriteriums nicht bei allen Planern die gleiche Methodik angewandt werden kann. Beispielsweise unterscheiden sich die explorierenden Pfadplaner CERBERUS und MB von den beiden PX4-Pfadplanern darin, dass die PX4-Planer keine Funktionalität für die Erkundung der Umgebung haben. Dadurch könnte beispielsweise bei den explorierenden Pfadplanern die Erkundungsfunktion genutzt werden, um festzustellen, ob eine Verengung in einem Raum von dem Planer passierbar ist. Um das gleiche bei den PX4-Planern zu ermitteln, muss eine andere Methodik genutzt werden. Beispielsweise können manuell mithilfe der RViz-Oberfläche die Zielpositionen so gesetzt werden, dass der Pfadplaner gezwungen ist, einen Pfad, der durch die Verengung führt, zu ermitteln.
Ähnlich verhält es sich bei der akkuraten Verfolgung von Zielkoordinaten. Dies ist bei den PX4-Planern möglich, jedoch bei dem CERBERUS-Planer nicht. Daher könnte bei den PX4-Planern überprüft werden, ob die Zielposition erreicht ist, indem die Ist-Position mit der Ziel-Position abgeglichen wird. Da der CERBERUS-Planer die Zielposition aufgrund der technischen Funktionsweise mit hoher Wahrscheinlichkeit nicht erreichen kann, muss für den CERBERUS-Planer eine andere Methodik angewandt werden. Beispielsweise kann die Geschwindigkeit des Luftfahrzeugs bei dem CERBERUS-Planer genutzt werden, um zu ermitteln, ob sich das Luftfahrzeug bewegt oder bereits aufgehört hat dies zu tun.
In beiden geschilderten Fällen bedarf es unterschiedlicher Methodiken für das Erfassen eines Kriteriums. Es ist davon auszugehen, dass ein methodischer Fehler entsteht. Dies gilt es bei der Auswertung zu berücksichtigen.
Als nächstes wird die Kommunikation der Pfadplaner verglichen. Zunächst werden Gemeinsamkeiten und schließend Unterschiede genannt. Die Positionsdaten des UAS werden bei allen untersuchten Pfadplaner mittels ROS-Topics veröffentlicht. Die Vorgabe der x- und y- Koordinate wird bei den PX4-Planern sowie bei dem CERBERUS-Planer ebenfalls über Topics durchgeführt. Zusätzlich muss bei den PX4-Planern die z-Koordinate mittels einer Systemfunktion angepasst werden.
Als Unterschied ist festzuhalten, dass die PX4-Planer keine ROS-Services verwenden. Dahingegen verwendet der CERBEREUS- und der MB-Planer jeweils Services für das Schalten der Erkundungsfunktionalität sowie für das Planen zu einer zuvor gesetzten Zielposition.
Für die Interaktion mit den Pfadplanern mittels Skripten bedeuten die geschilderten Unterschiede bei der Kommunikation, dass mehrere Skripte erstellt werden müssen. Durch die Unterschiede ist weiterhin davon auszugehen, dass das Setzen von Zielpositionen unterschiedlich viel Zeit in Anspruch nimmt. Hierbei gilt es festzustellen, ob die zeitlichen Unterschiede ausreichen gering sind, um bei der Auswertung vernachlässigt werden zu können.

4.2 Kriterien der Metrik für die Bewertung von Pfadplanungsalgorithmen

In diesem Abschnitt wird die Umsetzung der Metrik durchgeführt. Das Ziel ist die Auswahl von Kriterien, die Auskunft darüber geben, ob sich ein Pfadplaner für den Einsatz in Innenräumen eignet. Bei diesem Prozess wird darauf geachtet, dass die Ermittlung der Kriterien technisch umsetzbar ist. Dies wird in den folgenden Schritten durchgeführt:

1. In Unterabschnitt 4.2.1 werden Überlegungen bezüglich der Erstellung der Metrik diskutiert. Es werden drei Überlegungen basieren auf den theoretischen und technischen Grundlagen unter Betrachtung der Anforderungen an die vorliegende Arbeit angestellt.

2. In Unterabschnitt 4.2.2 wird die Metrik bestehend aus acht Kriterien erstellt. Davon sind vier Kriterien diskret und vier haben eine stetige Messgröße. Eine Zusammenfassung der Kriterien und deren Wertebereich ist in Tabelle 4.2 gegeben.

4.2.1 Überlegungen bezüglich der Erstellung einer Metrik

Im Folgenden wird diskutiert, welche Überlegungen bei der Erstellung einer Metrik für die Evaluation von Pfadplanern anzustellen sind, damit die erstrebte Aussagekraft erreicht wird. Die Überlegungen geschehen mit dem Ziel, quelloffene Pfadplaner in Innenräumen zu evaluieren. Die Metrik soll eine Bewertung erlauben, die aussagt, ob sich ein Pfadplaner für diesen Anwendungsfall eignet. Insbesondere werden die Überlegungen bezüglich

- der Bewertung der Pfadlänge,
- der Realitätsnähe der Simulationen und
- machbaren Vereinfachungen am Beispiel von der Flugzeit innerhalb einer Simulation

angestellt. Da die konkreten Kriterien der Metrik abhängig von der Art der Innenräume unterschiedlich gewichtet werden können, ist die Gewichtung der Kriterien nicht Teil der Überlegungen.
Die erste Überlegung betrifft die Bewertung der ermittelten Pfadlänge eines Pfadplaners in einem gegebenen Szenario. Hierbei kann weiter zwischen der theoretisch geplanten Länge und der praktisch zurückgelegten Distanz, die bei der Verfolgung des Pfades von einem Luftfahrzeug zurückgelegt wird, unterschieden werden. In der Praxis kann die exakte Verfolgung von einem geplanten Pfad wenig sinnvoll sein. Zum Beispiel kann kein rechtwinkliger Pfad von einem UAV verfolgt werden, ohne das Luftfahrzeug vollständig abzubremsen. Das vollständige Abbremsen ist im Regelfall weder sinnvoll noch gewünscht.

Dadurch sind theoretisch optimale Pfade unter Umständen nicht umsetzbar und im Affekt nicht besser als suboptimale Pfade, welche von dem Luftfahrzeug besser verfolgt werden können. Die Verfolgung der Pfade hängt maßgeblich von der genutzten Flugsteuerung ab. Neben der Flugsteuerung können auch Pfadplaner große Auswirkung auf die Verfolgbarkeit der Pfade haben. Wie in dem Unterabschnitt 2.2.2 dargelegt wird, gibt es Pfadplaner, welche die Beschränkungen der Luftfahrzeuge betrachten. Daher sollte die tatsächlich zurückgelegte Distanz und nicht die theoretisch geplanten Länge des Pfades betrachtet werden. Die Überprüfung, ob ein Pfad, welcher nach der zugrundeliegenden Kostenfunktion günstig ist, tatsächlich praktikabel ist, kann mithilfe von Simulationen durchgeführt werden. Festzuhalten ist, die Länge des Pfades sollte nicht an dem geplanten Pfad ermittelt werden, da die exakte Verfolgung von diesem nicht garantiert ist. Stattdessen sollte die Länge des Pfades an der Distanz, die das Luftfahrzeug bei der Verfolgung von diesem zurücklegt, ermittelt werden.
Die nächste Überlegung ist, dass realitätsnahe Resultate mit einer Simulation, welche die physischen Bedingungen betrachtet, aufschlussreicher sind als rein mathematische Betrachtungen. Ein besonders markanter Unterschied zwischen mathematischen Betrachtungen von Pfadplanungsalgorithmen und realitätsnahen Simulationen ist der Informationsgewinn. In der Realität ist ein UAS auf die verbaute Umweltwahrnehmung angewiesen. Diese Umweltwahrnehmung ist nicht fehlerfrei. Dem gegenübergestellt sind rein mathematische Simulationen mit fehlerfreien und vollständigen Daten. Letzteres ist in der Realität nicht umsetzbar. Der Pfadplanungsalgorithmus muss in der Lage sein, die Fehler der Sensorik zu kompensieren. Eine Konsequenz von realitätsnahen Simulationen ist, dass neben dem Pfadplanungsalgorithmus weitere Algorithmen benötigt werden, damit dieser Pfadplaner angewandt werden kann. Beispielweise müssen Sensordaten bei einem PRM-Planer verarbeitet werden, um in das benötigte Format gebracht zu werden. Hierbei kann ein Verarbeitungsschritt der Sensordaten die Erstellung einer OctoMap-Karte sein [42]. Diese Konvertierung wird nicht von dem Pfadplanungsalgorithmus durchgeführt, jedoch ist sie für den Pfadplaner eine Voraussetzung. Dadurch steigt die Anzahl an erforderlichen Operationen von einer in der Realität nutzbaren Pfadplaner-Implementierung. Trotz des Anstiegs bleibt der erzeugt Pfad unverändert. Diese Überlegung zeigt, dass die Aussagekraft von realitätsnahen Simulationen diejenige von rein mathematischen Simulationen übersteigt. Festzuhalten ist, dass die Ermittlung von benötigten Rechenressourcen eines Pfadplaners nicht nur anhand der benötigten Rechenressourcen für den Pfadplanungsalgorithmus selbst zu ermitteln ist. Stattdessen sollten die benötigten Ressourcen an der Gesamtimplementierung mit allen Modulen, die von dem Pfadplaner genutzt werden, ermittelt werden.
Weiterhin wird unter Berücksichtigung der in Abschnitt 1.2 dargelegten Vorgabe dieser Arbeit die dritte Überlegung angestellt. Die Rahmenbedingungen setzen voraus, dass ein Multikopter als Luftfahrzeug eingesetzt wird. In Verbindung mit dem Anwendungsgebiet in Innenräumen kann angenommen werden, dass die Fluggeschwindigkeit über die gesamte

Betriebsdauer gering ist. Demnach kann es sinnvoll sein, statt der Vermessung der zurückgelegten Pfadlänge die Flugzeit für diesen Pfad zu ermitteln. Dies liegt daran, dass in der Realität das gleiche UAV eine annähernd gleiche akkubedingte Flugzeit in Innenräumen haben wird. Damit ist gemeint, dass sich die maximale akkubedingten Flugzeit des gleichen Luftfahrzeugs in Innenräumen bei wiederholten Messungen dieser Flugzeit wenig verändern wird. Das UAV wird nicht auf eine hohe Geschwindigkeit gebracht, da die hindernisreiche Umgebung dies nicht zulässt. Das UAV wird entsprechend die meiste Zeit in einem Schwebeflug bzw. bei niedrigen Geschwindigkeiten operieren. Dieser Betrieb resultiert in einem annähend konstanten elektrischen Leistungsaufwand. Entsprechend ist auch die akkubedingte maximale Betriebszeit eines UAV unter diesen Umständen annähernd konstant. Dies hat zwei Voraussetzungen:

1. Der Pfad enthält wenige Höhenunterschiede.
2. Die benötigte elektrische Leistung für den Flug ist signifikant höher als die elektrische Leistung für die Ausführung der Algorithmen.

Der erste Punkt ist visuell ermittelbar. Jeder der untersuchten Pfadplaner verwendet Gazebo als Simulationssoftware. Dadurch ist die visuelle Erfassung von Höhenunterschieden möglich. Der zweite Punkt ist in der in dieser Arbeit durchgeführten Simulation nicht direkt ermittelbar. Dies liegt daran, dass in der vorliegenden Arbeit die Algorithmen auf dem simulierenden Computersystem ausgeführt werden. Die Hardware dieses Computersystems unterscheidet sich maßgeblich von mobil genutzter Hardware. Die geschilderte Überlegung macht deutlich, dass ein Kriterium der Metrik mit unterschiedlichen Methoden ermittelt werden kann. Diese sind bei der Umsetzung mit unterschiedlich hohem Aufwand verbunden. Entsprechend gilt es, dies bei der Erstellung der Metrik zu beachten.

4.2.2 Entwurf der Kriterien

In diesem Unterabschnitt werden die Kriterien für die Metrik entworfen. Ziel ist eine Liste von Kriterien, die für die Bewertung von Pfadplanern in Innenräumen genutzt werden kann. Die Auswahl der Kriterien basiert auf den theoretischen Grundlagen aus Kapitel 2 und den ermittelten Eigenschaften der untersuchten Pfadplaner aus Abschnitt 4.1. Es werden dabei die Anforderungen und Rahmenbedingungen aus Abschnitt 1.2 berücksichtigt. Die ausgewählten Kriterien werden zunächst nacheinander genannt und erläutert. Eine Zusammenfassung der Ergebnisse dieses Abschnitts ist in Tabelle 4.2 gegeben.
Es werden zwei Arten von Kriterien unterschieden. Zum einen gibt es diskrete Kriterien und zum anderen Kriterien mit einem stetigen Wertebereich. Diese werden in der genannten Reihenfolge im Folgenden dargelegt.
Ist ein diskretes Kriterium in zwei Kategorien unterteilt, handelt es sich um ein binäres

diskretes Kriterium. Binäre diskrete Kriterien entsprechen dem Wert 1, wenn sie erfüllt sind. Ist ein binäres Kriterium nicht erfüllt, entspricht dies dem Wert 0. Bei diskreten Kriterien mit mehr als zwei Kategorien wird bei der Erklärung des Kriteriums die möglichen Zustände und die zugehörigen Werte genannt. Kriterien mit einem stetigen Wertebereich werden als stetige Kriterien bezeichnet. Alle Kriterien werden auf den Intervall $[0, 1]$ abgebildet, um die folgende Auswertung zu erleichtern. Für alle Kriterien werden folgende Informationen gegeben:

- Art des Kriteriums,
- Relevanz für den Anwendungsfall und
- wie der Wert für das Kriterium ermittelt wird.

Zum Schluss werden alle Kriterien in einer Aufzählung zusammengefasst. Anzumerken ist, dass die Reihenfolge, in der die Kriterien aufgeführt sind, nicht die Gewichtung widerspiegelt.

Diskrete Kriterien

Folgende diskrete Kriterien werden festgelegt:

- „Höhenlimitierung“,
- „3D-Pfadplaner“,
- „Kontinuierlicher Pfadplaner“ und
- „Zielpositionsvorgabe“.

Das erste Kriterium wird als „Höhenlimitierung“ bezeichnet. Dieses diskrete Kriterium gibt an, ob der Pfadplaner eine Limitierung der maximalen Höhe erlaubt. Es handelt sich um ein binäres Kriterium. Die Limitierung der maximalen Höhe kann in dem Anwendungsfall SAR genutzt werden, um das UAS nicht durch gefährliche Gebiete zu führen. In anderen Situationen kann die gleiche Funktionalität genutzt werden, um die Pfadplanung durch ein bestimmtes Gebiet zu forcieren. Dies ist dann sinnvoll, wenn Randinformationen die Beschränkung des zu untersuchenden Gebietes erlaubt. Dies bedeutet, weniger Ressourcen werden für ungewollte Informationen aufgewandt bzw. riskiert, wenn die Limitierung der Flughöhe möglich ist. Die Limitierung der maximalen Höhe kann entweder zu einem absoluten und uveränderlichen oder einem relativen Referenzpunkt sein. Ist eine der zwei Möglichkeit einer Limitierung der maximalen Höhe gegeben, ist das Kriterium erfüllt.
Das zweite diskrete Kriterium wird als „3D-Pfadplaner“ bezeichnet. Dieses Kriterium gibt an, ob die Funktionalität des Pfadplaners, drei räumliche Dimensionen für die Pfadplanung

zu nutzen, gegeben ist. Es handelt sich um ein binäres Kriterium. Dieses Kriterium äußert sich darin, dass der Pfadplaner Pfade über mehrere Ebenen ermitteln kann. Gebäude haben häufig mehrere Etagen. Daher ist das Anwendungsspektrum von 3D-Pfadplanern signifikant größer als das von 2D-Pfadplanern. Insbesondere in dem Anwendungsfall SAR ist es wünschenswert, die Fähigkeit von UAS, sich vertikal zum Boden fortzubewegen, auszunutzen. Ist der Pfadplaner in der Lage, einen Pfad über mindestens zwei Etagen eines Gebäudes zu planen, ist das Kriterium erfüllt.
Das dritte Kriterium wird als „Kontinuierlicher Pfadplaner“ bezeichnet. Dieses diskrete Kriterium gibt an, ob die Pfadplanung kontinuierlich oder schrittweise erfolgt. Es handelt sich um ein binäres Kriterium. Eine schrittweise Planung äußert sich dadurch, dass das UAS vor der Verfolgung einer neuen Zielposition abgebremst wird. Bei einer kontinuierlichen Pfadplanung kann ein neues Ziel während der Verfolgung des vorherigen Ziels vorgegeben werden, ohne das Luftfahrzeug dadurch abzubremsen. Diese Funktionalität ist wünschenswert, um in Innenräumen die verfügbare Flugzeit optimal zu nutzen. Es ist unerheblich, ob die Vorgabe der Zielpositionen von einem Algorithmus oder durch menschliches Interagieren mit der GUI erfolgt. Wird das Luftfahrzeug nicht sichtbar abgebremst, wenn ein neue Zielvorgabe gesetzt wird, dann ist das Kriterium erfüllt.
Das vierte Kriterium wird als „Zielpositionsvorgabe“ bezeichnet. Dieses diskrete Kriterium gibt an, in welcher Form die Vorgabe von Zielpositionen möglich ist. Es wird in drei Abstufungen unterschieden:

1. Die Vorgabe von Zielpositionen ist nicht möglich. Dies ist dann der Fall, wenn die Funktionalität von dem Pfadplaner nicht gegeben ist. Das entsricht dem Wert 0 für das Kriterium.

2. Die Vorgabe von Zielpositionen ist beschränkt möglich. Dies ist dann der Fall, wenn nur Zielpositionen in bekannten Gebieten valide sind. Das entspricht dem Wert 0.5 für das Kriterium.

3. Die Vorgabe von Zielpositionen ist vollständig möglich. Dies ist dann der Fall, wenn Zielpositionen in bekannten und unbekannten Gebieten gesetzt werden können. Eine Ausnahme ist, wenn die Zielposition mit einem bekannten Hindernis kollidiert. Dann darf die Zielposition von dem Pfadplaner verworfen werden. Das entspricht dem Wert 1 für das Kriterium.

Die Abstufung basiert auf den in Abschnitt 4.1 durchgeführten Untersuchungen. Die Vorgabe von Zielkoordinaten kann für eine Anwendung eine Notwendigkeit darstellen. Im Anwendungsfall SAR könnte die Vorgabe von Zielpositionen genutzt werden, um das UAS schnell dort zu platzieren, wo es benötigt wird. Weniger Einschränkungen bei der Zielpositionsvorgabe sind wünschenswert, da dies die Nutzung der Funktionalität erleichtert. Daher wird die vorgestellte Abstufung des Kriteriums vorgenommen.

Stetige Kriterien

Folgende stetige Kriterien werden festgelegt:

- „Zeitkosten“,
- „Minimale Freiraum“,
- „Durchschnittliche Prozessorauslastung“ und
- „Maximaler Speicheraufwand“.

Das erste Kriterium wird als „Zeitkosten“ bezeichnet. Dieses stetige Kriterium ist die benötigte Zeit für das Zurücklegen der Strecke zwischen zwei Zielpositionen. Einer der kritischen Faktoren von UAS ist die durch den Akkumulator begrenzte Betriebszeit. Dadurch dass Multikopter das Eigengewicht durch den erzeugten Schub ausgleichen müssen, um in der Luft zu verweilen, wird die Betriebszeit wenig von der zurückgelegten Strecke beeinflusst. Zu dem Zurücklegen der Strecke gehört die benötigte Zeit für die Planung dazu. Die Zeit wird in Sekunden gemessen. Der Wert für das Kriterium „Zeitkosten“ wird aus den ermittelten Zeiten aller Pfadplaner wie folgt ermittelt. Zunächst werden alle Zeiten auf den Intervall $[0, 1]$ normalisiert. Daraus ergibt sich für jeden Pfadplaner eine Variable $t_{GesamtNormalisiert}$. Nach dieser Operation entspricht die längste und somit schlechteste benötigte Zeit $t_{GesamtNormalisiert}$ dem Wert 1. Damit der beste Zeit dem Wert 1 entspricht, wird der jeweilige normalisierte Wert $t_{GesamtNormalisiert}$ von 1 subtrahiert:

$$K_{Zeitkosten} = 1 - t_{GesamtNormalisiert} \tag{4.1}$$

Die nach dieser Vorschrift ermittelte Variable $K_{Zeitkosten}$ entspricht dem Kriterium „Zeitkosten“. Falls Das Ziel nicht erreicht wird, entspricht das pauschal dem Wert 0. Dieser Wert wird nicht in die Normalisierung einbezogen. Das Kriterium bewertet nicht nur den Pfadplanungsalgorithmus, sondern die Anwendung von diesem Algorithmus in einem realitätsnahen Gesamtsystem. Es ist anzunehmen, dass die Pfadplaner mit unterschiedlichen Geschwindigkeiten die Strecken verfolgen. Eine längere Strecke, die schneller von dem Luftfahrzeug verfolgt wird, könnte schneller zum Ziel führen, als eine kürzere Strecke, die langsamer von dem Luftfahrzeug verfolgt wird. Die Geschwindigkeit, mit der das Luftfahrzeug fortbewegt wird, hängt maßgeblich von dem Pfadplaner ab. Daher spiegelt sich an diesem Kriterium die Gesamtleistung des Pfadplaners wider. Demnach ist dieser Kennwert nicht nur aussagekräftiger als die Ermittlung der Länge des Pfades, sondern ist der Ermittlungsprozess zusätzlich simpler. Die Entscheidung für dieses Kriterium basiert auf den in Unterabschnitt 4.2.1 dargelegten und für dieses Kriterium fortgeführten Überlegungen bezüglich der Pfadlänge.
Das zweite Kriterium wird als „Minimaler Freiraum“ bezeichnet. Dieses stetige Kriterium

gibt an, wieviel breiter ein Durchgang in Relation zu dem Luftfahrzeug mindestens sein muss, damit der Pfadplaner das UAS kollisionsfrei durch den Durchgang führen kann. In Innenräumen ist dies ein kritischer Faktor. Es ist zu erwarten, dass nicht alle Verengungen durch die das Luftfahrzeug physisch passt, auch von dem Pfadplaner genutzt werden. Weiterhin ist zu beachten, dass die Größe der verwendeten Luftfahrzeuge unterschiedlich ist. Um die Vergleichbarkeit zu gewährleisten, wird bei der Berechnung des Wertes für das Kriterium die Größe des Luftfahrzeugs berücksichtigt. Zudem wird davon ausgegangen, dass das Luftfahrzeug quadratisch ist. Diese Annahme dient der Vereinfachung der Erfassung des Kriteriums. Das Kriterium bedarf einiger Rechenoperationen. Im Folgenden wird die Herleitung des Kriteriums erläutert:

$$S_{VerengungFrei} = S_{passierbar} - S_{BreiteVehikel} \tag{4.2}$$

Die Variable $S_{passierbar}$ entspricht der passierbaren Strecke, die parallel zum Boden verläuft. Die Variable $S_{BreiteVehikel}$ ist die Breite des Luftfahrzeugs, welche orthogonal zu der vorderen Seite des Luftfahrzeugs und parallel zum Boden gemessen wird. Die Variable $S_{VerengungFrei}$ ergibt sich aus der Differenz zwischen $S_{passierbar}$ und $S_{BreiteVehikel}$. Die Abbildung 4.3 visualisiert die Variablen beispielhaft an dem Luftfahrzeug Iris-LiDAR. Die Einheit für die Strecke ist Zentimeter. Um die Vergleichbarkeit zwischen den verschiedenen Pfadplanern zu erleichtern, werden die erfassten Werte für das Kriterium auf den Intervall $[0, 1]$ normalisiert. Der normalisierte Wert wird als $S_{VerengungNormalisiert}$ bezeichnet. Um den besten Wert mit dem Wert 1 abzubilden, wird dieser nach der folgenden Formel invertiert:

$$K_{MinFreierRaum} = 1 - S_{VerengungNormalisiert} \tag{4.3}$$

Die Variable $K_{MinFreierRaum}$ gibt den normalisierten und invertierten minimalen Freiraum an, den der Pfadplaner benötigt, um das Luftfahrzeug durch einen Durchgang kollisionsfrei führen zu können. Diese Variable entspricht dem Kriterium „Minimaler Freiraum“ . Ein höherer Wert entspricht einem besseren Resultat. Falls die Zielposition nicht erreicht wird, wird das Kriterium mit dem Wert 0 gewertet.

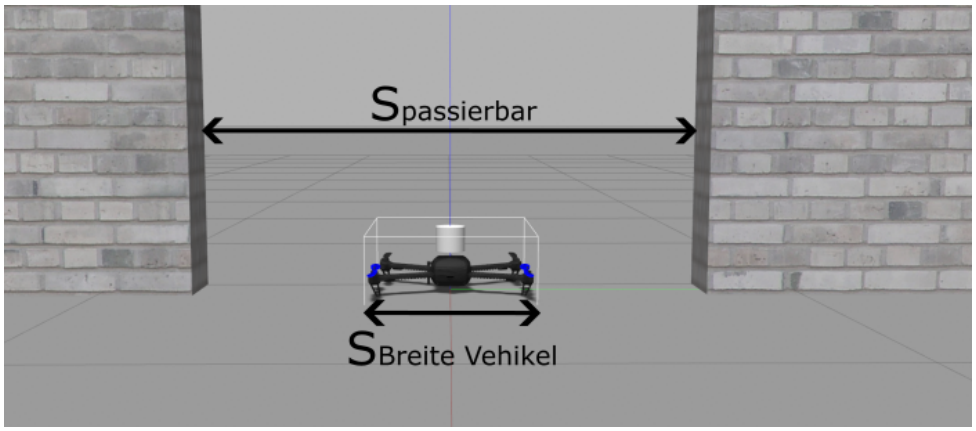


Abbildung 4.3: Darstellung der relevanten Strecken des Kriteriums „Minimaler Freiraum“ am Beispiel des Luftfahrzeugs Iris-LiDAR in einer Gazebo-Simulation. Der Durchgang zwischen zwei Wänden wird mit der Variable $S_{passierbar}$ angegeben. Die Breite des Luftfahrzeugs wird als $S_{BreiteVehikel}$ angegeben und wird durch die Kollisionsbox des Luftfahrzeugs verdeutlicht.

Das dritte Kriterium wird als „Durchschnittliche Prozessorauslastung“ bezeichnet. Dieses stetige Kriterium ist die Prozessorauslastung, die durch alle Module der Pfadplaner-Umsetzung verursacht wird. Dazu gehört nicht die Simulationsumgebung und die Prozessorauslastung, die durch diese verursacht wird. Es ist davon auszugehen, dass die verursachte Prozessorauslastung volatil ist. Beispielsweise kann vermutet werden, dass die Prozessauslastung besonders hoch ist, wenn der Pfadplanungsalgorithmus ausgeführt wird. Um die Prozessorauslastung des Pfadplaners über den gesamten Zeitraum zu betrachten, wird sich dazu entschieden, die durchschnittliche Auslastung für das Kriterium zu verwenden. Es ist davon auszugehen, dass die in der vorliegenden Arbeit genutzte Computerhardware leistungsstärker als mobile und akkubetriebene Computersysteme ist. Dennoch ist die gemessene Prozessorauslastung ein Indiz für die benötigte CPU, um die Pfadplaner ausführen zu können. Dies ist in Innenräumen besonders relevant, da die benötigte Hardware auf den Formfaktor des UAS Auswirkungen hat. Innenräume haben wenig verfügbaren Platz. Daher ist eine geringe Prozessauslastung ein relevantes Kriterium. Da die Auslastung der CPU einen stetigen Wertebereich hat, wird eine Normierung der gemessenen CPU-Auslastungen aller Pfadplaner auf das Intervall $[0, 1]$ durchgeführt. Das bedeutet, die höchste verursachte Prozessorauslastung hat nach der Normierung den Wert 1 und die geringste verursachte Prozessorauslastung hat den Wert 0. Um mit dem restlichen Format konsistent zu sein, wird auch dieser normierte Wert invertiert, sodass die geringste Prozessauslastung den Wert 1 hat und die höchste Prozessorauslastung den Wert 0. Der auf diese Weise ermittelte Wert entspricht dem Kriterium „Durchschnittliche Prozessorauslastung“.
Das vierte stetige Kriterium wird als „Maximaler Speicheraufwand“ bezeichnet. Dieses Kriterium gibt die maximale Menge des gebrauchten Arbeitsspeichers, die für die Aus-

führung aller Module der Pfadplaner-Umsetzung benötigt wird, an. Im Gegensatz zu der Prozessorauslastung ist davon auszugehen, dass der benötigte Arbeitsspeicher mit der Zeit konstant zunimmt. Das liegt daran, dass mit zunehmender Zeit das UAS stetig mehr von der Umgebung erfasst. Bei den globalen bzw. explorierenden Pfadplanern wird die Karte der bekannten Umgebung nicht verworfen, sondern stetig erweitert. Dadurch werden stetig weitere Informationen im Arbeitsspeicher abgelegt. Lokale Pfadplaner hingegen verwerfen die Umgebungsdaten zyklisch. Bei den lokalen Pfadplanern ist daher kein konstanter Anstieg zu erwarten. Dennoch ist die Menge des benötigten Arbeitsspeichers für alle Pfadplaner relevant, da dies wie bei der CPU-Auslastung Rahmenbedingungen für die benötigte Hardware setzt. Je kleiner der Wert ist, desto besser ist es. Auch bei diesem Kriterium wird eine Normalisierung auf den Intervall $[0, 1]$ durchgeführt. Nach dieser Normalisierung entspricht der Wert von dem Pfadplaner mit dem höchsten Speicheraufwand dem Wert 1. Der normalisierte Wert für den geringsten Speicheraufwand entspricht dem Wert 0. Um auch hierbei mit dem restlichen Format konsistent zu sein, wird auch dieser normierte Wert invertiert. Nach dieser Invertierung hat der Pfadplaner mit dem geringsten Speicheraufwand den Wert 1 für das Kriterium. Der Pfadplaner mit dem höchsten Speicheraufwand hat den Wert 0 für das Kriterium. Der nach dieser Vorgehensweise ermittelte Wert entspricht dem Wert für das Kriterium „Maximaler Speicheraufwand“ .
Ein weiteres erwogenes Kriterium ist die Auslastung der GPU. Jedoch hat sich bei Untersuchungen gezeigt, dass die GPU nicht von den Pfadplanern beansprucht wird. Die GPU wird hingegen von der verwendeten Simulationsumgebung Gazebo beansprucht. Insbesondere die Simulation von der LiDAR-Sensorik profitiert von der GPU. Da die Simulation von dieser Sensorik bei der realen Anwendung nicht nötig ist, wird die Auslastung der GPU nicht als Kriterium verwendet.
Damit sind alle Kriterien für die Bewertung von Pfadplanern in Innenräumen vorgestellt. Die Tabelle 4.2 gibt eine Zusammenfassung der gewählten Kriterien.

Tabelle 4.2: Zusammenfassung von Kriterien für die Bewertung von Pfadplanern. Bei allen Kriterien entspricht der Wert 1 der besten Bewertung.

#	Kriterium	Beschreibung	Werte-bereich
Endliche Wertebereiche			
1.	„Höhenlimitierung“	Die Maximale Höhe ist limitierbar.	{0,1}
2.	„3D-Pfadplaner“	Es handelt sich um einen 3D-Pfadplaner.	{0,1}
3.	„Kontinuierlicher Pfadplaner“	Es gibt keine Bremsung bei einer Zieländerung.	{0,1}
4.	„Zielpositionsvorgabe“	Die Zielposition kann gesetzt werden.	{0,0.5,1}
Stetige Wertebereiche			
1.	„Zeitkosten“	Die Dauer für die Planung und den Flug zwischen den Start- und Zielpositionen.	[0,1]
2.	„Minimaler Freiraum“	Der Minimale Freiraum zwischen dem Luftfahrzeug und dem Hindernis, durch welchen der Planer das Luftfahrzeug führen kann.	[0,1]
3.	„Durchschnittliche Prozessorauslastung“	Durchschnittliche Prozessorauslastung, die durch alle Planer-Module verursacht wird.	[0,1]
4.	„Maximaler Speicheraufwand“	Maximaler Arbeitsspeicher, der durch die alle Planer-Module beansprucht wird.	[0,1]

4.3 Konzeptionierung der Methoden für die Erfassung von Kriterien

In diesem Unterabschnitt werden die Methoden für die Erfassung der Kriterien, welche in Unterabschnitt 4.2.2 vorgestellt werden, konzeptioniert. Das Ziel ist der Entwurf der Methoden für die Bewertung der Kriterien für alle Pfadplaner. Allerdings weisen die

untersuchten Pfadplaner fundamentale technische Unterschiede auf, welche in Unterabschnitt 4.1.1 identifiziert werden. Aus diesem Grund ist es nicht möglich, ein Kriterium bei allen Pfadplanern mit der gleichen Methodik zu untersuchen. Dies wird bei dem Entwurf der Methoden für die Erfassung der Kriterien berücksichtigt.
In den Unterabschnitten 4.3.1 bis 4.3.8 werden die einzelnen Kriterien behandelt. Es wird detailliert dargelegt, wie ein Kriterium erfasst wird. Falls ein Kriterium nicht bei allen Pfadplanern mit der gleichen Methodik erfassbar ist, werden verschiedene Methoden entworfen. Die genutzte Methodik wird abhängig davon gewählt, ob entweder die Vorgabe einer Zielkoordinate möglich ist, oder die Funktionalität der autonomen Exploration bei einem Pfadplaner gegeben ist. Die Reihenfolge, in der die Methodik für die Kriterien entworfen wird, orientiert sich an der Reihenfolge vorgestellten Kriterien aus Unterabschnitt 4.2.2.
Unabhängig von der Art eines Kriteriums kann es entweder manuell oder programmatisch erfasst werden. Programmatisch bedeutet, eine Software wird entwickelt, um das untersuchte Kriterium zu erfassen. Manuell bedeutet, dass die Erfassung mittels menschlicher Interaktion mit der Pfadplaner-Software bzw. durch Analyse der Software, Dokumentation oder Konfigurationsdateien geschieht.

4.3.1 Erfassung des diskreten Kriteriums „Höhenlimitierung“

Das Kriterium „Höhenlimitierung“ wird anhand der Funktion für die Limitierung der Höhe manuell erfasst. Die Erfassung erfolgt, indem die Dokumentation, die GUI und die Konfigurationsdateien des betrachteten Pfadplaners nach der Funktionalität untersucht werden. Wenn diese Funktionalität gefunden wird, wird diese in einem Test bestätigt. Der Test besteht darin, diese Funktionalität mittels einer Simulation in einer Gazebo-Umgebung zu verifizieren.

4.3.2 Erfassung des diskreten Kriteriums „3D-Pfadplaner“

Das Kriterium „3D-Pfadplaner“ wird manuell erfasst, indem überprüft wird, ob der Pfadplaner drei räumliche Dimensionen nutzt. Die Überprüfung wird je nach Funktionalität der Pfadplaner unterschiedlich durchgeführt. Es wird zwischen Pfadplanern mit der Funktionalität, zu explorieren, und Pfadplanern mit der Funktionalität, eine Zielposition vorzugeben, unterschieden. Wenn der untersuchte Pfadplaner die Funktionalität besitzt, die Umgebung autonom zu explorieren, dann wird diese genutzt. Wenn keine explorierende Funktionalität gegeben ist, dann werden manuell Ziele, die entlang der Treppe führen, vorgegeben. Für die Erfassung des Kriteriums wird eine simulierbare Umgebung mit einer Steigung in Form einer Treppe genutzt. Die für dieses Kriterium genutzte Gazebo-Umwelt ist in Abbildung 4.4 dargestellt. Die Treppe ist großflächig gestaltet, um den Pfadplanern nicht durch

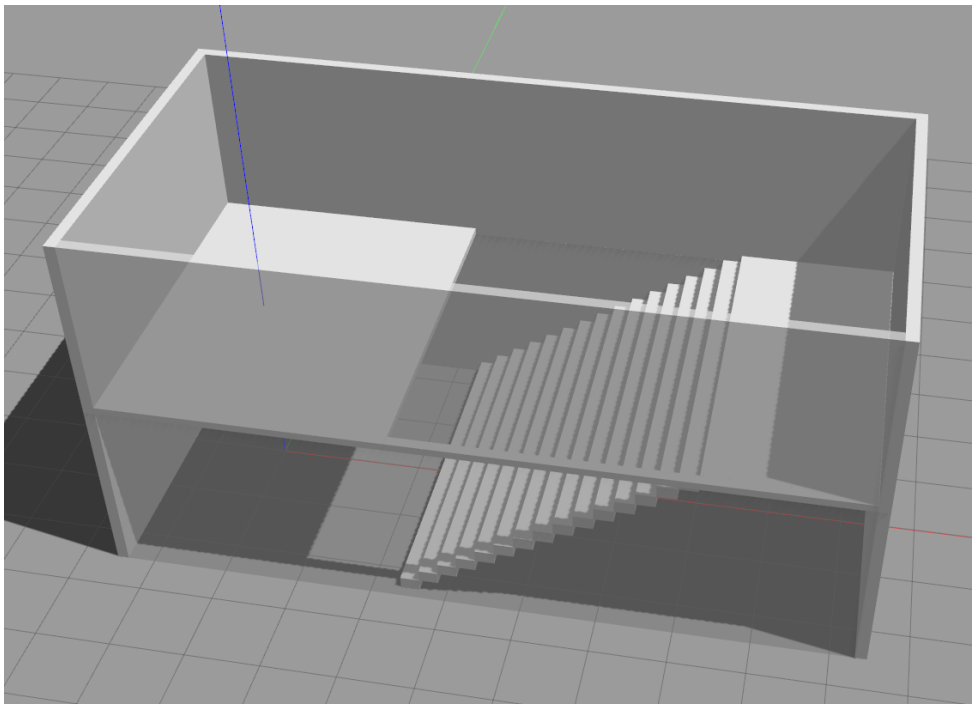


Abbildung 4.4: Genutzte Gazebo-Umgebung für die Erfassung des Kriteriums „3D-Pfadplaner“. Die Fläche des Grundrisses ist $10m \cdot 5m$. Die vorderen Wände sind transparent, um die Visualisierung des Innenraumes zu ermöglichen. Dies hat keine Auswirkungen auf die Simulation.

starke räumliche Beschränkung zu hindern. Unabhängig von der gegebenen Funktion des Pfadplaners wird die gleiche Umgebung genutzt. Das Luftfahrzeug startet im Erdgeschoss vor der Treppe. Kann die Treppe zu der höheren Etage verfolgt werden, ist das Kriterium erfüllt.

4.3.3 Erfassung des diskreten Kriteriums „Kontinuierlicher Pfadplaner“

Das Kriterium „Kontinuierlicher Pfadplaner“ wird manuell überprüft. Die Erfassung ist bei Pfadplanern mit der Funktionalität, zu explorieren, und Pfadplanern mit der Funktionalität, eine Zielposition vorzugeben, unterschiedlich. Wenn der untersuchte Pfadplaner die Funktionalität hat, die Umgebung autonom zu explorieren, dann wird diese genutzt. Bei explorierenden Pfadplanern wird das Kriterium daran bewertet, ob das Luftfahrzeug zwischen zwei Erkundungsetappen verlangsamt wird. Bei nicht explorierenden Pfadplanern wird ein neues Ziel vorgegeben, bevor das vorherige Ziel erreicht ist. In beiden Fällen werden ausschließlich Richtungungsänderungen, die unter etwa 45° betragen, betrachtet. Bei größeren Winkeln ist eine Geschwindigkeitsreduktion durch die abrupte Richtungsänderung zu erwarten.

4.3.4 Erfassung des diskreten Kriteriums „Zielpositionsvorgabe“

Das Kriterium „Zielpositionsvorgabe“ wird manuell ermittelt. Dazu wird geprüft, ob die Funktionalität der Vorgabe einer Zielposition bei dem untersuchten Pfadplaner gegeben ist. Nur in diesem Fall wird der Pfadplaner hinsichtlich dieses Kriteriums untersucht. Die Vorgabe der Zielposition erfolgt innerhalb einer Simulation. Durch die Visualisierung der Sensordaten mittels RViz wird festgestellt, welcher Teil der Umgebung dem Pfadplaner bekannt ist. Zunächst wird eine Zielposition, die sich außerhalb des bekannten Gebietes befindet, gesetzt. Falls sich das UAS in Richtung des Zieles bewegt, wird das Kriterium mit dem Wert 1 bewertet. Wenn es nicht zu einer derartigen Reaktion des UAS kommt, wird eine Zielposition innerhalb des bekannten Gebietes gesetzt. Falls sich das UAS in Richtung des Zieles bewegt, wird das Kriterium mit dem Wert 0.5 bewertet.

4.3.5 Erfassung des stetigen Kriteriums „Zeitkosten“

Das Kriterium „Zeitkosten“ wird mittels eines Simulationsszenarios programmatisch erfasst. Das Simulationsszenario besteht aus einer Umgebung, welche sich in mehrere Räume aufteilt, sowie den festgelegten Start- und Zielpositionen. Zwischen den Start- und Zielpositionen gibt es eine Menge von Positionen, die als Zwischen-Zielpositionen vorgegeben werden. Das Simulationsszenario wird in Abbildung 4.5 visualisiert. Für die Messung der Zeit wird die Gesamtdauer für den Flug zwischen Start und Ziel gemessen. Die Vorgabe der Zielpositionen besteht zum größten Teil aus Zielposition, welche in dem direkten Sichtbereich des UAS liegen. Darunter befinden sich Zielkoordinaten, die das Erhöhen der Flughöhe erfordern. Weiterhin gibt es eine Zielposition, die sich nicht in dem direkten Sichtbereich des UAS befindet. Diese befindet sich in dem bereits bekannten Gebiet. Es wird angenommen, dass diese Zielposition von Pfadplanern, die eine Karte der Umgebung erstellen, erreicht werden kann. Die Kombination von den verschiedenen Arten von Zielpositionen resultiert in einer verbesserten Aussagekraft der Bewertung. Bei diesem Kriterium werden ausschließlich Pfadplaner mit der Funktionalität einer setzbaren Zielvorgabe untersucht. Es wird sichergestellt, dass alle Pfadplaner exakt die gleichen Informationen erhalten. Dies wird umgesetzt, indem ein ROS-Node erstellt wird, der die Vorgabe der Zielkoordinaten anhand einer definierten Liste von Zielpositionen durchführt. Es wird empirisch sichergestellt, dass alle Pfadplaner die Vorgabe von allen Zielpositionen akzeptieren. Die Messung der Zeit, die von dem untersuchten Pfadplaner benötigt wird, um das Luftfahrzeug von der Start- zu der Zielposition zu führen, wird ebenfalls mittels des ROS-Nodes programmatisch gemessen.

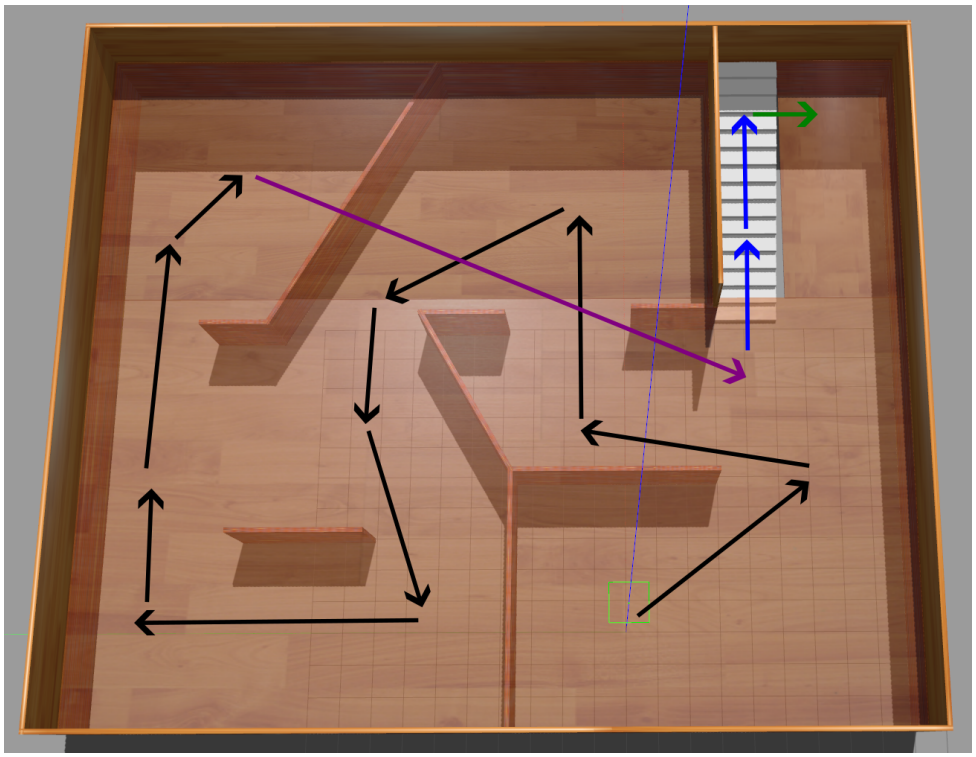


Abbildung 4.5: Genutzte Gazebo-Umgebung für die Erfassung des Kriteriums „Zeitkosten“. Die schwarzen Pfeile zeigen den Pfad des Luftfahrzeugs durch das Erdgeschoss. Diese befinden sich immer im Sichtbereich des UAS. Der lila Pfeil zeigt den direkten Pfad durch die bereits bekannte Umgebung. Die blauen Pfeile zeigen den aufsteigenden Pfad entlang der Treppe an. Der grüne Pfeil zeigt den Pfad im ersten Obergeschoss an. Die Decke ist aus Gründen der Visualisierung transparent.

4.3.6 Erfassung des stetigen Kriteriums „Minimaler Freiraum“

Das Kriterium „Minimaler Freiraum“ wird mittels einer Simulation manuell überprüft. Es werden Gazebo-Umgebungen erstellt, welche jeweils eine zu passierende verengte Passage haben. Ein Beispiel ist in Abbildung 4.6 gegeben. Das Kriterium wird ermittelt, indem überprüft wird, ob der Pfadplaner das UAS durch die Verengung führen kann. Der Weg durch die Verengung wird in der Abbildung mit einem Pfeil kenntlich gemacht. Es werden mehrere Umgebungen mit unterschiedlich ausgeprägten Verengungen entworfen. Die Untersuchung beginnt mit der geringsten Verengung. Wenn der Pfadplaner das Luftfahrzeug erfolgreich durch die Verengung führt, wird die Umgebung mit der nächstkleineren Verengung untersucht. Dies wird wiederholt, bis der Pfadplaner das Luftfahrzeug nicht durch eine neue Verengung führt. Die Breite der letzten erfolgreich passierten Verengung wird zur Bewertung notiert und nach nach dem Verfahren, welches in Unterabschnitt 4.2.2 erläutert wird, verarbeitet. Die untersuchten Verengungen haben eine passierbare Breite von $200cm$, $150cm$, $125cm$, $100cm$, $90cm$, $80cm$, $70cm$, $60cm$ und $50cm$. Es wird angenommen, dass eine Verengung von mehr als $200cm$ nicht relevant für Pfadplaner mit dem Einsatzgebiet

in Innenräumen sind. Weiterhin wird angenommen, dass eine Verengung von unter $50cm$ nicht realistisch passierbar ist.

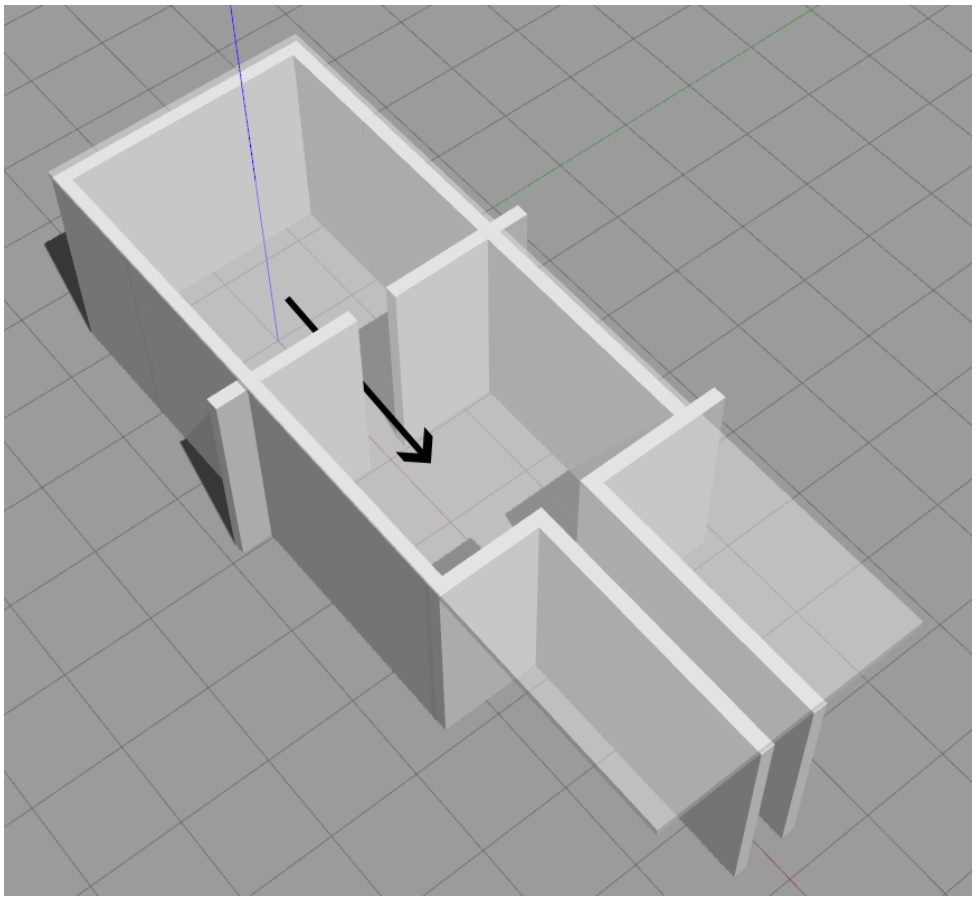


Abbildung 4.6: Genutzte Gazebo-Umgebung für die Erfassung des Kriteriums „Minimaler Freiraum“. Der schwarze Pfeil zeigt den Pfad, der durch das Luftfahrzeug passiert werden muss, um das Kriterium zu erfüllen. Die Verengung hat eine Breite von $60cm$. Die Decke ist aus Gründen der Visualisierung transparent.

4.3.7 Erfassung des stetigen Kriteriums „Durchschnittliche Prozessorauslastung“

Das Kriterium „Durchschnittliche Prozessorauslastung“ wird programmatisch mittels einer Simulation und eines erstellten Skriptes für die Erfassung der CPU-Auslastung ermittelt. Anzumerken ist, dass die in diesem Unterabschnitt thematisierte Software sowie betrachteten Prozesse für das Kriterium „Maximaler Speicheraufwand“ identisch sind. In diesem Unterabschnitt werden folgende Aspekte der Methodik thematisiert:

- die genutzte Simulationsumgebung mit dem antizipierten Pfad des Luftfahrzeugs,
- die Unterscheidung der Durchführung anhand von verschiedenen Funktionalitäten der Pfadplaner,

- die genutzte Software für die Ermittlung der Prozessorauslastung und
- die berücksichtigten sowie nicht berücksichtigten Prozesse für die Ermittlung der Prozessorauslastung.

Die Umgebung der genutzten Simulation besteht aus mehreren Wänden. Diese Umgebung wird in Abbildung 4.7 dargestellt. Die Pfeile der Abbildung geben den ungefähren antizipierten Pfad an. Durch das Voranschreiten des Luftfahrzeugs wird zunehmend mehr von der Umgebung erkundet. Die CPU-Auslastung wird während dieses Prozesses für jeden Pfadplaner einzeln festgehalten.
Die Durchführung der Simulation wird je nach Funktionalität der Pfadplaner unterschiedlich durchgeführt. Es wird zwischen Pfadplanern mit der Funktionalität, zu explorieren, und Pfadplanern mit der Funktionalität, eine Zielposition vorzugeben, unterschieden. Bei explorierenden Pfadplanern wird die Exploration solange durchgeführt, bis die Umgebung vollständig erkundet ist. Für die Ermittlung der durchschnittlichen Prozessorauslastung wird die Initialisierungssequenz dieser Pfadplaner nicht betrachtet, um die Verfälschung der Daten zu vermeiden. Bei Pfadplanern ohne diese Funktion werden Zielkoordinaten manuell vorgegeben. Auch hierbei wird sichergestellt, dass die gesamte Umgebung von dem UAS beflogen ist. Während dieser Simulation wird die Prozessorauslastung erfasst.
Für die Erfassung der CPU-Auslastung wird das Programm `psrecord` [53] genutzt. Neben der CPU-Auslastung wird das gleiche Tool für die Erfassung des verwendeten Arbeitsspeichers genutzt. Um dieses Programm wird ein Skript erstellt, welches die CPU-Auslastung aller Module des untersuchten Pfadplaners sekündlich in einer log-Datei festhält. Als Nächstes wird von den geloggten Werten für die Prozessorauslastung der Durchschnitt ermittelt. Der Durchschnitt wird anschließend nach dem Verfahren, welches in Unterabschnitt 4.2.2 erläutert wird, verarbeitet.
Bei der Erfassung der Prozessorauslastungen der Pfadplaner werden alle Prozesse berücksichtigt, die für die Ausführung des Pfadplaners vonnöten sind. Insbesondere gehören dazu folgende Prozesse:

- Alle grundlegenden ROS-Prozesse, die für das Funktionieren des ROS-Framework erforderlich sind: Ein grundlegender ROS-Prozesse ist der `rosmaster`-Prozess. Dieser ist für jede Kommunikation innerhalb des ROS-Frameworks vonnöten.
- Alle Pfadplaner-spezifische ROS-Prozesse, welche bei jedem Pfadplaner unterschiedlich sind: Diese Prozesse führen die Algorithmen für die Pfadplanung sowie die Kartierung der Umgebung aus. Letzteres geschieht nur bei globalen Pfadplanern.
- Alle Prozesse, die für die Flugsteuerung erforderlich sind: Diese sind erforderlich, um das Luftfahrzeug steuern zu können.

Weiterhin gibt es Prozesse, die während der Simulation genutzt werden, jedoch in der Form bei einer realen Anwendung nicht erforderlich wären. Dazu gehören folgende Prozesse:

- Der RViz-Prozess wird für visuelles Feedback sowie als Schnittstelle genutzt. Jedoch sind diese Funktionalitäten in einem realen Anwendungsfall in der Form nicht zwangsläufig erforderlich und die Pfadplanungsalgorithmen funktionieren auch ohne diese. Daher ist der RViz-Prozess nicht Gegenstand der Messung.
- Alle Gazebo-Prozsse werden ebenfalls nicht betrachtet. Gazebo wird für die Simulation genutzt. Diese wäre in einem realen Anwendungsfall nicht erforderlich. Daher werden diese Prozesse nicht betrachtet.

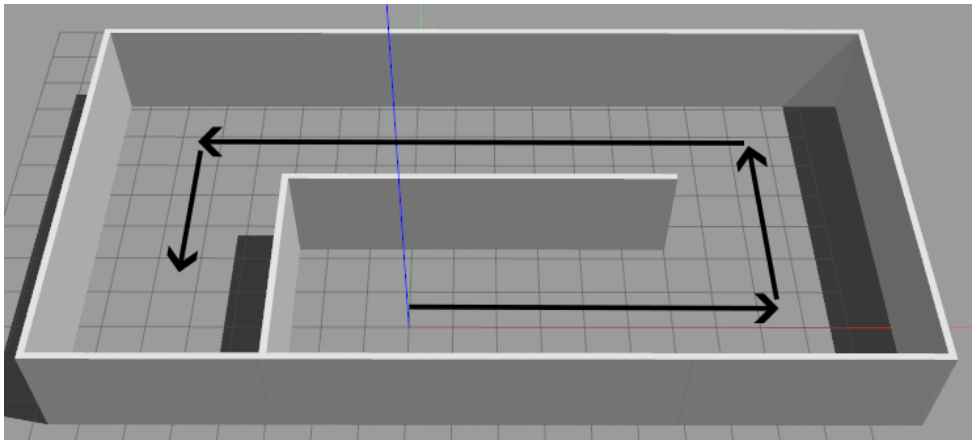


Abbildung 4.7: Genutzte Gazebo-Umgebung für die Evaluation des Kriteriums „Durchschnittliche Prozessorauslastung“. Die Pfeile geben den ungefähren antizipierten Pfad, der von dem jeweiligen Luftfahrzeug verfolgt werden soll, an.

4.3.8 Erfassung des stetigen Kriteriums „Maximaler Speicheraufwand“

Das Kriterium „Maximaler Speicheraufwand“ wird nach dem gleichen Prinzip wie das Kriterium „Durchschnittliche Prozessorauslastung“, dessen Erfassung in Unterabschnitt 4.3.7 erläutert wird, ermittelt. Die Erfassung des Arbeitsspeichers wird parallel zu der Erfassung der Prozessorauslastung durchgeführt. Es wird in der gleichen Simulation von dem gleichen Skript erfasst. Daher erfolgt auch hier die Erfassung des Kriteriums auf Basis von `psrecord`. Zunächst wird die Erfassung des genutzten Arbeitsspeichers über den Zeitraum der Simulation durchgeführt. Anschließend wird der maximal genutzte Arbeitsspeicher ermittelt und nach dem Verfahren, welches in Unterabschnitt 4.2.2 erläutert wird, verarbeitet und evaluiert.

Tabelle 4.3: Zusammenfassung der Ergebnisse der Evaluation von allen untersuchten Pfadplaner. Die Namen der Pfadplaner sind aus räumlichen Gründen verkürzt: lokaler PX4-Pfadplaner (lok. PX4), globaler PX4-Pfadplaner (glob. PX4), CERBERUS-Pfadplaner (CERBERUS) und MB-Pfadplaner (MB).

Kriterium	lok. PX4	glob. PX4	CER-BERUS	MB
„Höhenlimitierung“	0.00	1.00	1.00	0.00
„3D-Pfadplaner“	1.00	1.00	1.00	1.00
„Kontinuierlicher Pfadplaner“	1.00	1.00	0.00	0.00
„Zielpositionsvorgabe“	1.00	1.00	0.50	0.00
„Zeitkosten“	0.00	0.00	1.00	k.A.
„Minimaler Freiraum“	0.00	0.00	1.00	0.30
„Durchschnittliche Prozessorauslastung“	1.00	1.00	0.51	0.00
„Maximaler Speicheraufwand“	1.00	0.90	0.39	0.00

4.4 Evaluation der Pfadplaner

In diesem Abschnitt wird die Evaluation der Pfadplaner vorgestellt. Für jedes Kriterium wird jeder Pfadplaner einzeln betrachtet. Die bei der Evaluation genutzten Methoden werden in Abschnitt 4.3 erläutert.
Die Struktur der Evaluation ist wie folgt:

- In den Unterabschnitten 4.4.1 bis 4.4.8 werden die acht Kriterien evaluiert. Dabei werden zunächst die vier diskreten und anschließend die vier stetigen Kriterien betrachtet. Bei der Untersuchung werden eventuelle Auffälligkeiten genannt. Jeder Wert wird bei der Evaluation begründet.

- Schließlich werden in dem Unterabschnitt 4.4.9 die Vermutungen, welche in Unterabschnitt 4.1.1 aufgestellt werden, aufgegriffen. Zu jeder Vermutung wird klargestellt, ob diese richtig oder falsch ist.

Eine Zusammenfassung der Ergebnisse der gesamten Evaluation ist in Tabelle 4.3 gegeben. Messungen, die aufgrund von technischen Gegebenheiten nicht erfolgen konnten, werden mit den Worten „keine Angabe“ bzw. der Abkürzung „k.A.“ kenntlich gemacht.

4.4.1 Evaluation des Kriteriums „Höhenlimitierung“

In diesem Unterabschnitt wird das Kriterium „Höhenlimitierung“ bei allen Pfadplanern evaluiert. Eine Zusammenfassung der Ergebnisse ist in Tabelle 4.3 in der entsprechenden Zeile gegeben.
Bei der Evaluation der Pfadplaner wurden folgende Beobachtungen gemacht:

- Bei dem lokalen PX4-Pfadplaner konnte keine Möglichkeit für die Limitierung der Höhe festgestellt werden. Es wurden die Dokumentation des PX4-Avoidance-Paktes [35], die GUI und die Konfigurationsdateien des Pfadplaners untersucht. Aus diesem Grund wird das Kriterium mit dem Wert 0.00 bewertet.

- Bei dem globalen PX4-Pfadplaner kann die Höhe mittels der GUI limitiert werden. Die `rqt_reconfigure`-GUI enthält dem Parameter `max_altitude_`, welcher die maximale Höhe limitiert. Bei Untersuchungen mittels Simulation hat sich gezeigt, dass Zielpositionen über der erlaubten Höhe verworfen werden. Dies entspricht dem erwarteten Verhalten. Daher wird das Kriterium mit dem Wert 1.00 bewertet.

- Bei dem CERBERUS-Pfadplaner kann die Höhe mittels Parameter in einer Konfigurationsdatei festgelegt werden. Die Höhe wird relativ zum Boden festgelegt. In der Datei `gbplanner_config.yaml` bestimmt der Parameter `max_ground_height` die maximal erlaubte Höhe zu dem Boden. In Untersuchungen hat sich gezeigt, dass die Höhe nicht überschritten wird. Daher wird das Kriterium mit dem Wert 1.00 bewertet.

- Bei dem MB-Pfadplaner ist die maximale Höhe ebenfalls in Relation zu dem Boden beschränkt. Diese Schlussfolgerung wurde auf Basis von zahlreichen Beobachtungen in Simulationen gezogen. Jedoch konnte nicht der entsprechende Parameter identifiziert werden. Weder in der Dokumentation noch in den Konfigurationsdateien konnte der Parameter festgestellt werden. Die Konsequenz davon ist, dass eine gezielte Veränderung dieses Parameters nicht möglich ist. Ohne die Möglichkeit diesen Parameter zu verändern, ist die Nutzung der Höhenlimitierung, wie es in Unterabschnitt 4.2.2 dargelegt wird, nicht möglich. Daher wird das Kriterium mit dem Wert 0.00 bewertet.

4.4.2 Evaluation des Kriteriums „3D-Pfadplaner“

In diesem Unterabschnitt wird das Kriterium „3D-Pfadplaner“ bei allen Pfadplanern evaluiert. Die Treppe der in Unterabschnitt 4.3.2 vorgestellten Gazebo-Umgebung konnte von allen Pfadplanern verfolgt werden. Daher erfüllen alle Pfadplaner das Kriterium. Eine

Zusammenfassung der Ergebnisse ist in Tabelle 4.3 in der entsprechenden Zeile gegeben. Bei der Evaluation der Pfadplaner wurden folgende Beobachtungen gemacht:

- Bei dem lokalen PX4-Pfadplaner konnten die Zielpositionen iterativ entlang der Treppe der in Abbildung 4.4 dargestellten Umgebung gesetzt werden. Mit diesem Vorgehen ist es gelungen, das UAS in die Zielposition zu führen. Aus diesem Grund wird das Kriterium mit dem Wert 1.00 bewertet.
- Bei dem globalen PX4-Pfadplaner ähnelt das Vorgehen dem des lokalen PX4-Pfadplaners. Zusätzlich musste die maximal erlaubte Flughöhe iterativ angepasst werden. Ist diese nicht korrekt eingestellt, kollidiert der Pfadplaner mit der Decke. Das liegt daran, dass bei den erkannten Hindernissen versucht wird, über diese zu fliegen. Jedoch wird dadurch, dass die Sensorik nicht nach oben ausgerichtet ist, die Decke unmittelbar über den Luftfahrzeug nicht wahrgenommen. Dadurch wird diese nicht als Hindernis erkannt. Dennoch ist das Ziel kollisionsfrei mit gegebenen Funktionalitäten erreichbar. Daher wird das Kriterium mit dem Wert 1.00 bewertet.
- Bei dem CERBERUS-Pfadplaner kann die Zielposition kollisionsfrei mit der explorierenden Funktion erreicht werden. Daher wird das Kriterium mit dem Wert 1.00 bewertet.
- Bei dem MB-Pfadplaner kann die Zielposition ebenfalls kollisionsfrei mit der explorierenden Funktion erreicht werden. Daher wird das Kriterium mit dem Wert 1.00 bewertet.

4.4.3 Evaluation des Kriteriums „Kontinuierlicher Pfadplaner“

In diesem Unterabschnitt wird das Kriterium „Kontinuierlicher Pfadplaner“ bei allen Pfadplanern evaluiert. Eine Zusammenfassung der Ergebnisse ist in Tabelle 4.3 in der entsprechenden Zeile gegeben.
Bei der Evaluation der Pfadplaner wurden folgende Beobachtungen gemacht:

- Bei dem lokalen PX4-Pfadplaner kann eine Zielpositionen während der Verfolgung einer vorherigen Zielposition vorgegeben werden, ohne das Luftfahrzeug abzubremsen. Daher wird das Kriterium mit dem Wert 1.00 bewertet.
- Bei dem globalen PX4-Pfadplaner ist das Verhalten des UAS identisch zu dem lokalen PX4-Pfadplaner. Daher wird das Kriterium bei diesem Pfadplaner ebenfalls mit dem Wert 1.00 bewertet.

- Bei dem CERBERUS-Pfadplaner wird das Luftfahrzeug zwischen den Erkundungsetappen abgebremst. Dies geschieht auch bei Richtungsänderungen mit einem geringen Winkel. Daher wird das Kriterium mit dem Wert 0.00 bewertet.
- Bei dem MB-Pfadplaner wird das Luftfahrzeug zwischen den Erkundungsetappen teilweise vollständig abgebremst. Daher wird das Kriterium mit dem Wert 0.00 bewertet.

4.4.4 Evaluation des Kriteriums „Zielpositionsvorgabe“

In diesem Unterabschnitt wird das Kriterium „Zielpositionsvorgabe“ bei allen Pfadplanern evaluiert. Eine Zusammenfassung der Ergebnisse ist in Tabelle 4.3 in der entsprechenden Zeile gegeben.
Bei der Evaluation der Pfadplaner wurden folgende Beobachtungen gemacht:

- Bei dem lokalen PX4-Pfadplaner kann eine beliebige Zielposition gesetzt werden. Die Zielposition wird auch dann akzeptiert und verfolgt, wenn diese sich in einem unbekannten Gebiet befinden. Daher wird das Kriterium mit dem Wert 1.00 bewertet.
- Bei dem globalen PX4-Pfadplaner ist dies ebenfalls möglich. Anzumerken ist, dass Zielpositionen, die sich in bekannten Hindernissen befinden, verworfen werden. Das ist für das Kriterium zulässig. Daher wird das Kriterium mit dem Wert 1.00 bewertet.
- Bei dem CERBERUS-Pfadplaner ist die Vorgabe von Zielpositionen beschränkt möglich. Zielpositionen außerhalb von bekannten Gebieten können nicht gesetzt werden. Derartige Zielpositionen werden entweder auf die nächste valide Zielposition in dem bekannten Gebiet umgeändert, oder sie werden gänzlich verworfen. Daher wird das Kriterium mit dem Wert 0.50 bewertet.
- Bei dem MB-Pfadplaner besteht keine Möglichkeit eine Zielposition zu setzen. Daher wird das Kriterium mit dem Wert 0.00 bewertet.

4.4.5 Evaluation des Kriteriums „Zeitkosten“

In diesem Unterabschnitt wird das stetige Kriterium „Zeitkosten“ bei allen Pfadplanern evaluiert. Eine Zusammenfassung der Ergebnisse ist in Tabelle 4.3 in der entsprechenden Zeile gegeben.
Bei der Evaluation der Pfadplaner wurden folgende Beobachtungen gemacht:

- Bei dem lokalen PX4-Pfadplaner konnte kein erfolgreicher Versuch durchgeführt werden. Trotz validen Zielpositionen ist es bei allen Versuchen zu Kollisionen mit der Decke oder einer Wand oder zu einem Stillstehen des Luftfahrzeugs gekommen. Die Kollisionen mit der Decke können nicht verhindert werden, da die Funktion für die Beschränkung der Flughöhe nicht gegeben ist. Die Kollision einer Wand passiert trotz sensorischer Wahrnehmung von dieser. Das bedeutet, die Sensorik erfasst die Wand und dennoch führt der Pfadplaner das Luftfahrzeug in die Wand. Die korrekte sensorische Wahrnehmung der Wand konnte mit der Visualisierung mittels RViz verifiziert werden. Das Stillstehen des Luftfahrzeugs wird mit lokalen Minima erklärt. Der zugrundeliegende APF-Algorithmus ist besonders anfällig für dieses Problem. Aufgrund von diesen Problemen ist es nicht gelungen, einen vollständigen Versuch, wie es in Unterabschnitt 4.3.5 geschildert wird, durchzuführen. Daher wird das Kriterium mit dem Wert 0.00 bewertet.

- Bei dem globalen PX4-Pfadplaner konnte ebenfalls kein erfolgreicher Versuch durchgeführt werden. Bei den Versuchen kam es zu Kollisionen mit Wänden bzw. zu einem Stillstehen des Luftfahrzeugs. Die Kollisionen sind vorgekommen, obwohl die Sensorik die Hindernisse korrekt erkannt hat. Dies konnte mit der Visualisierung mittels RViz verifiziert werden. Das Stillstehen des Luftfahrzeugs wird damit erklärt, dass der Algorithmus nicht für die Hindernisdichte konzipiert ist. Aufgrund der wiederholt fehlgeschlagenen Versuche wird das Kriterium mit dem Wert 0.00 bewertet.

- Bei dem CERBERUS-Pfadplaner konnte der Versuch wiederholt erfolgreich durchgeführt werden. Die dafür benötigte Zeit liegt zwischen $179.281s$ und $195.223s$. Es kam bei keinem der Versuche zu Kollisionen. Jeder der gestarteten Versuche wurde erfolgreich durchgeführt. Der CERBERUS-Pfadplaner ist der einzige Pfadplaner, bei dem dieser Versuch erfolgreich durchgeführt werden konnte. Daher wird dieses Kriterium für den CERBERUS-Pfadplaner mit dem Wert 1.00 bewertet.

- Bei dem MB-Pfadplaner hat keine Funktionalität für die Vorgabe von Zielpositionen. Daher wird das Kriterium für den Pfadplaner nicht bewertet.

4.4.6 Evaluation des Kriteriums „Minimaler Freiraum“

In diesem Unterabschnitt wird das stetige Kriterium „Minimaler Freiraum“ bei allen Pfadplanern evaluiert. Eine Zusammenfassung der Ergebnisse ist in Tabelle 4.3 in der entsprechenden Zeile gegeben. Für die Breiten der Luftfahrzeuge werden die jeweiligen Breiten der Kollisionsboxen der Luftfahrzeuge verwendet.
Bei der Evaluation der Pfadplaner wurden folgende Beobachtungen gemacht:

- Bei dem lokalen PX4-Pfadplaner hat das Luftfahrzeug eine Breite von $47cm$. Die geringste Verengung, durch die der Pfadplaner das Luftfahrzeug führt, beträgt $200cm$. Dies entspricht nach der in Unterabschnitt 4.2.2 vorgestellten Vorschrift einem Wert von 0.0 für das Kriterium.
- Bei dem globalen PX4-Pfadplaner wird das gleiche Luftfahrzeug verwendet. Die geringste passierbare Verengung beträgt ebenfalls $200cm$. Dies entspricht ebenfalls einem Wert von 0.00 für das Kriterium.
- Bei dem CERBERUS-Pfadplaner hat das Luftfahrzeug eine Breite von $25cm$. Die geringste passierbare Verengung beträgt $60cm$. Dies entspricht einem Wert von 1.00 für das Kriterium.
- Bei dem MB-Pfadplaner hat das Luftfahrzeug eine Breite von $82cm$. Die geringste passierbare Verengung beträgt $200cm$. Dies entspricht einem Wert von 0.30 für das Kriterium.

4.4.7 Evaluation des Kriteriums „Durchschnittliche Prozessorauslastung“

In diesem Unterabschnitt erfolgt die Auswertung des Kriteriums „Durchschnittliche Prozessorauslastung“. Hierbei wird für jeden Pfadplaner visuell dargelegt, wie sich die CPU-Auslastung im Verlauf der Simulation verändert.
Eine Zusammenfassung der Ergebnisse ist in Tabelle 4.3 in der entsprechenden Zeile gegeben. Für die Ermittlung der vorgestellten Werte werden alle Prozesse berücksichtigt, die für die Nutzung der Pfadplaner benötigt werden. Die berücksichtigten Prozesse werden in Unterabschnitt 4.3.7 dargelegt. Weiterhin ist betonend anzumerken, dass die Skalen der Abbildungen in diesen Unterabschnitten unterschiedlich sind. Es wird stets die Zeit auf der X-Achse und die CPU-Auslastung auf der Y-Achse dargestellt. Die Zeit auf den Skalen der Abbildungen ist unterschiedlich, da die Dauer der Ausführung der jeweiligen Pfadplaner unterschiedlich ist. Die Skalen der CPU-Auslastung ist unterschiedlich, da die verursachten Auslastungen zwischen den Pfadplaner variieren. Bei jeder Abbildung wird der durchschnittliche Wert der Prozessauslastung mit einer gestrichelten Linie kenntlich gemacht. Die ersten gemessenen Datenpunkte für die CPU-Auslastung liegen stets um den Wert 0, da die Messung vor der eigentlichen Ausführung des Pfadplaners gestartet wird. Das liegt daran, dass die Prozesse registriert sind, bevor die Ausführung von diesen Prozessen beginnt.
Bei der Evaluation der Pfadplaner wurden folgende Beobachtungen gemacht:

- Bei dem lokalen PX4-Pfadplaner wurde eine durchschnittliche Prozessorauslastung von 23.5% ermittelt. Dieser Wert liegt um 0.2% über dem des globalen PX4-Pfadplaners. Damit hat sich die in Unterabschnitt 4.1.1 geäußerten Vermutung, dass der lokale PX4-Pfadplaner mangels der Kartierung der Umgebung eine geringere Auslastung der CPU verursacht, widerlegt. Der Verlauf der CPU-Auslastung ist in Abbildung 4.8 dargestellt. Zu Beginn der Ausführung steigt die Auslastung auf etwa den durchschnittlichen Wert und oszilliert über die gesamte Ausführungszeit um diesen Wert. Die durchschnittliche Auslastung des Planers entspricht dem Wert 1.00 für das Kriterium.

- Bei dem globalen PX4-Pfadplaner wurde eine durchschnittliche Prozessorauslastung von 23.2% ermittelt. Dies ist die geringste Auslastung, die bei allen Pfadplanern ermittelt wurde. Der Verlauf der CPU-Auslastung ist in Abbildung 4.9 dargestellt. Dieser hat einen ähnlichen Verlauf wie bei dem lokalen PX4-Pfadplaner. Es fällt jedoch auf, dass die Streuung der Messpunkte stärker ausgeprägt ist. Die durchschnittliche Auslastung des Planers entspricht dem Wert 1.00 für das Kriterium.

- Bei dem CERBERUS-Pfadplaner wurde eine durchschnittliche Prozessorauslastung von 77.83% ermittelt. Der Verlauf der CPU-Auslastung ist in Abbildung 4.10 dargestellt. Der Initialisierungsprozess ist nach etwa Sekunde 30 abgeschlossen. Danach ist die verursachte CPU-Auslastung konstant und beinhaltet hervorstehende Maxima. Es wird angenommen, dass dies die Zeitpunkte sind, an denen die Planung der nächsten Zielkoordinate durchgeführt wurde. Die durchschnittliche Auslastung des Planers entspricht dem Wert 0.51 für das Kriterium.

- Bei dem MB-Pfadplaner wurde eine durchschnittliche Prozessorauslastung von 134.29% ermittelt. Dies ist damit die höchste Auslastung von allen Pfadplanern. Der Verlauf der CPU-Auslastung ist in Abbildung 4.11 dargestellt. Der Initialisierungsprozess ist bei diesem Pfadplaner nach etwa 20 Sekunden abgeschlossen. Danach liegt die CPU-Auslastung etwa zwischen 120 und 160 Prozent. Es sind mehr Maxima erkennbar, als Planungsetappen bei der Durchführung beobachtet werden konnten. Die Ursache davon ist nicht bekannt. Weiterhin ist bei der Durchführung des Versuches nicht der vollständige antizipierte Pfad, welcher in Abbildung 4.7 dargestellt wird, verfolgt worden. Der Pfadplaner hat nicht die Funktionalität, Zielpositionen vorzugeben. Daher ist es dem Pfadplaner nicht gelungen, die vollständige Umgebung zu erkunden. Der von dem Pfadplaner erfasste Teil der Umgebung entspricht etwa 95% der Gesamtumgebung. Die durchschnittliche Auslastung des Planers entspricht dem Wert 0.00 für das Kriterium.

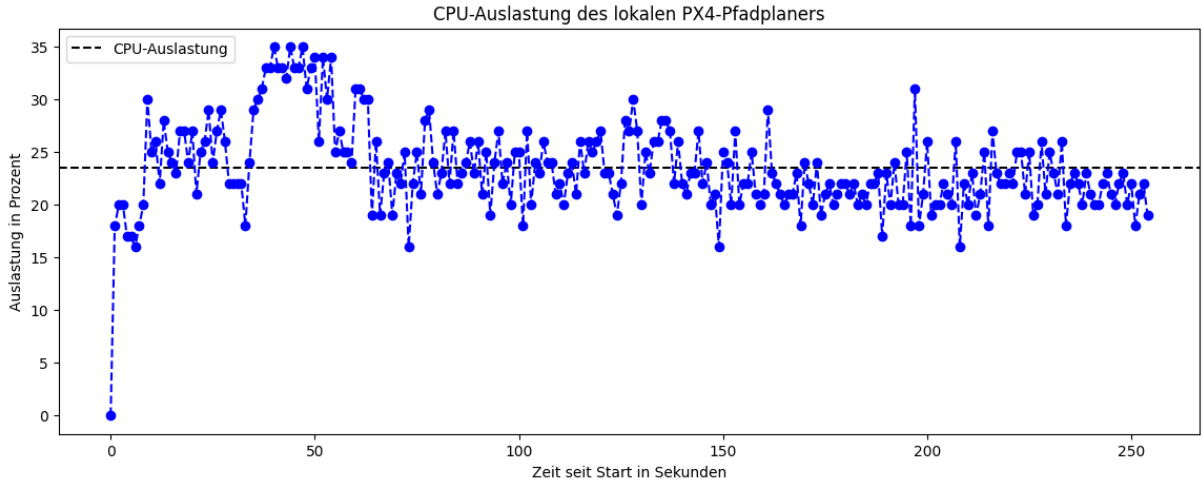


Abbildung 4.8: Die Prozessorauslastung bei der Ausführung des lokalen PX4-Pfadplaners in der in Abbildung 4.7 dargestellten Gazebo-Umgebung. Die gestrichelte Linie visualisiert die durchschnittliche Prozessorauslastung.

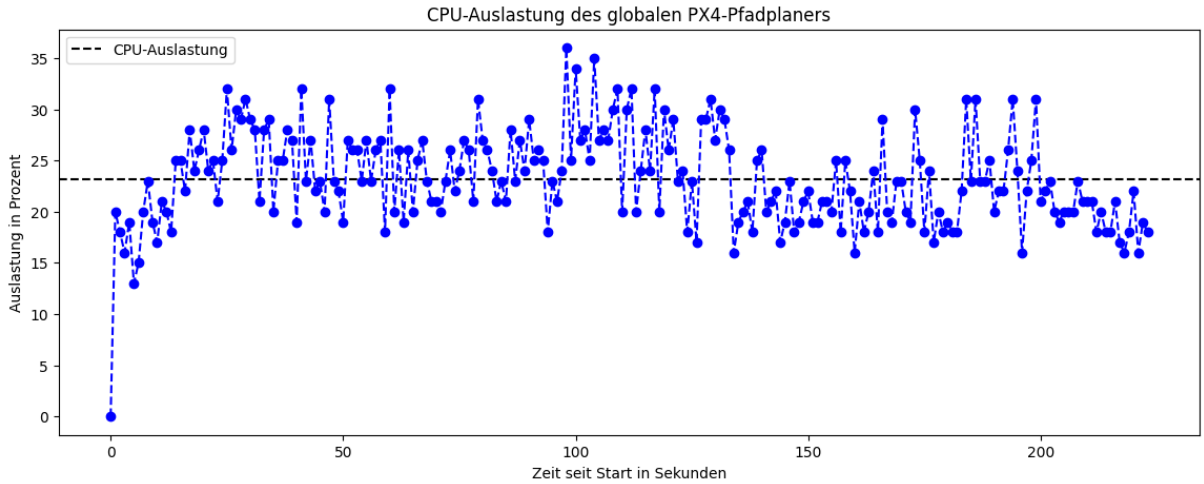


Abbildung 4.9: Die Prozessorauslastung bei der Ausführung des globalen PX4-Pfadplaners in der in Abbildung 4.7 dargestellten Gazebo-Umgebung. Die gestrichelte Linie visualisiert die durchschnittliche Prozessorauslastung.

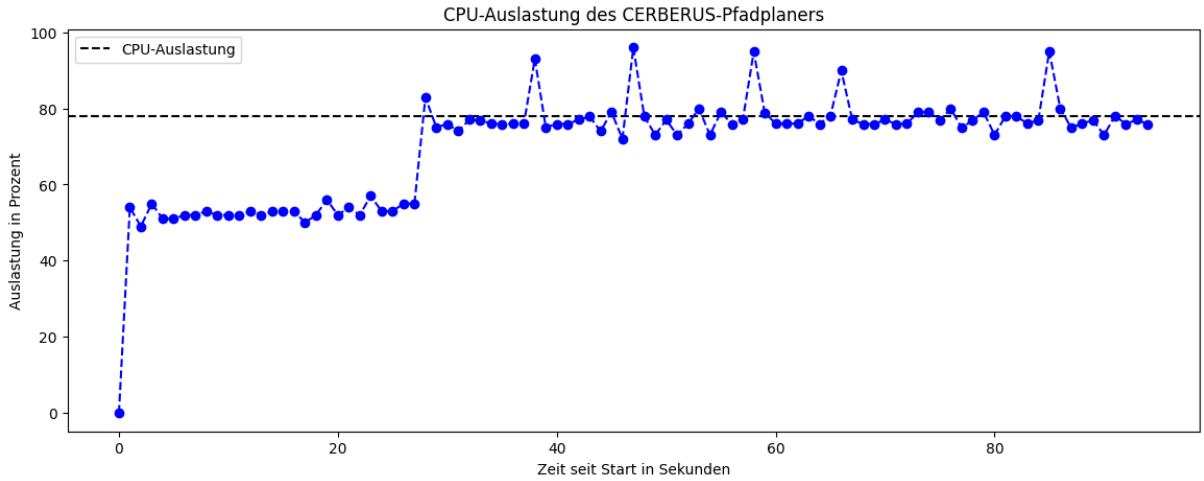


Abbildung 4.10: Die Prozessorauslastung bei der Ausführung des globalen CERBERUS-Pfadplaners in der in Abbildung 4.7 dargestellten Gazebo-Umgebung. Die gestrichelte Linie visualisiert die durchschnittliche Prozessorauslastung.

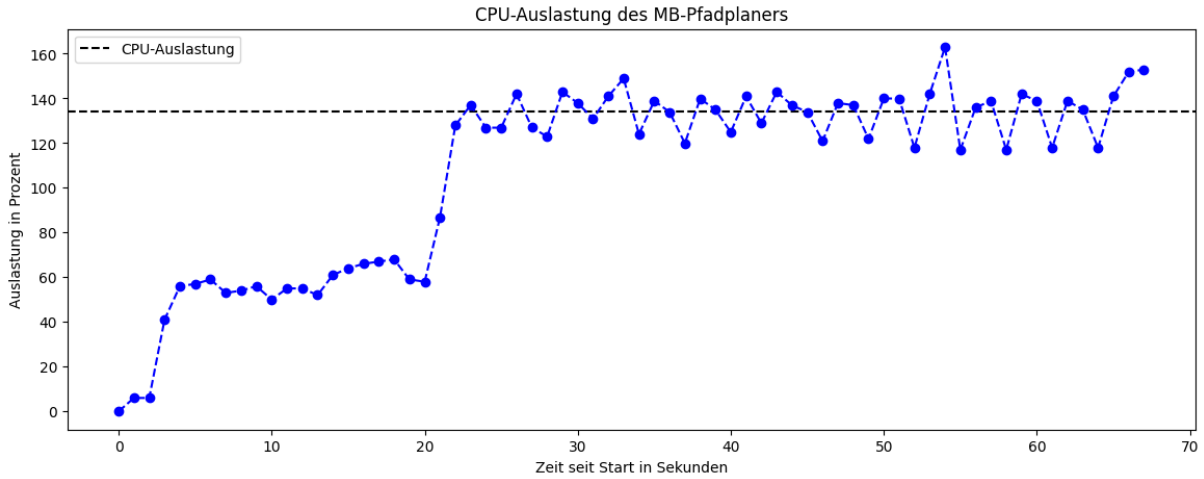


Abbildung 4.11: Die Prozessorauslastung bei der Ausführung des globalen MB-Pfadplaners in der in Abbildung 4.7 dargestellten Gazebo-Umgebung. Die gestrichelte Linie visualisiert die durchschnittliche Prozessorauslastung.

4.4.8 Evaluation des Kriteriums „Maximaler Speicheraufwand“

In diesem Unterabschnitt erfolgt die Auswertung des Kriteriums „Maximaler Speicheraufwand“ bei allen Pfadplanern. Eine Zusammenfassung der Ergebnisse ist in Tabelle 4.3 in der entsprechenden Zeile gegeben.
Im Folgenden wird für jeden Pfadplaner visuell dargelegt, wie sich der verwendete Arbeitsspeicher im Verlauf der Simulation verändert. Bei den folgenden vorgestellten Ergebnissen gelten die in Unterabschnitt 4.4.7 dargelegten Rahmenbedingungen. Auch hierbei ist zu beachten, dass die Skalen in den Abbildungen unterschiedlich sind. Die gestrichelten Linien in den Abbildungen visualisieren den jeweils höchsten gemessenen Wert für den genutzten Arbeitsspeicher. Die Vorschrift, nach der aus dem maximalen genutzten Arbeitsspeichers der Wert für das Kriterium ermittelt wird, wird in Unterabschnitt 4.2.2 vorgestellt.
Bei der Evaluation der Pfadplaner wurden folgende Beobachtungen gemacht:

- Bei dem lokalen PX4-Pfadplaner wurde ein maximal genutzter Arbeitsspeicher von $249.13MB$ ermittelt. Dies ist der geringste Wert für den maximal genutzten Arbeitsspeicher von allen Pfadplanern. Der Verlauf von dem genutzten Arbeitsspeicher ist in Abbildung 4.12 dargestellt. Zu Beginn der Ausführung steigt der genutzte Arbeitsspeicher nahezu auf das Maximum. Der Verlauf ist ab etwa Sekunde 30 jedoch konstant leicht steigend. Der Pfadplaner erstellt keine Karte der Umgebung, die zunehmend Speicher benötigt. Daher wird davon ausgegangen, dass die Logs, welche bei der Ausführung erstellt werden, den konstanten Anstieg des genutzten Speichers verursachen. Zusätzlich ist ein herausstehendes Maximum circa bei der Sekunde 160 zu erkennen. Dieser maximal benötigte Arbeitsspeicher des Planers entspricht dem Wert 1.00 für das Kriterium.

- Bei dem globalen PX4-Pfadplaner wurde ein maximal genutzter Arbeitsspeicher von $249.13MB$ ermittelt. Der Verlauf von dem genutzten Arbeitsspeicher ist in Abbildung 4.13 dargestellt. Dieser Verlauf zeigt, dass der Arbeitsspeicher zunächst auf circa $270MB$ steigt und bei diesem Wert bis etwa Sekunde 100 bleibt. Anschließend beginnt der benötigte Arbeitsspeicher bis zu der Terminierung der Prozesse zu steigen. Der globale PX4-Pfadplaner verwendet für die Erstellung der Karte der Umgebung eine OctoMap. Der konstante Teil des Verlaufs wird damit begründet, dass die Initiale Kartierung der Umgebung ein größeres Gebiet umfasst, als die Sensorik des UAS erreicht. Dadurch wird mehr Speicher allokiert, als zunächst für die Kartierung benötigt wird. Durch die Bewegung des UAS wird zunehmend mehr von der Umgebung wahrgenommen. Sobald die Karte der Umgebung erweitert werden muss, fängt der Speicher an zu steigen. Daher wird der steigende Teil der Kurve mit

der Erkundung der Umgebung begründet. Der maximal benötigte Arbeitsspeicher dieses Pfadplaners entspricht dem Wert 0.90 für das Kriterium.

- Bei dem CERBERUS-Pfadplaner wurde ein maximal genutzter Arbeitsspeicher von $414.47MB$ ermittelt. Der Verlauf von dem genutzten Arbeitsspeicher ist in Abbildung 4.14 dargestellt. Der Verlauf zeigt einen stetigen Anstieg des genutzten Arbeitsspeichers. Für die Kartierung wird ein stichprobenbasierter Ansatz genutzt. Mit der Erkundung der Umgebung wird die Karte stetig um neue Stichproben erweitert. Damit wird der konstante Anstieg erklärt. Der maximal benötigte Arbeitsspeicher dieses Pfadplaners entspricht dem Wert 0.39 für das Kriterium.

- Bei dem MB-Pfadplaner wurde ein maximal genutzter Arbeitsspeicher von $557.91MB$ ermittelt. Hierbei ist zu betonen, dass etwa 95% der Umgebung erfasst wurde, da die restliche Umgebung nicht erkundet wurde. Der Verlauf von dem genutzten Arbeitsspeicher ist in Abbildung 4.15 dargestellt. Dieser zeigt einen stetigen Anstieg des genutzten Arbeitsspeichers. Auch bei diesem Pfadplaner wird ein stichprobenbasierter Ansatz genutzt. Daher wird auch hier angenommen, dass die zunehmende Anzahl an Stichproben der Umgebung den Anstieg des Speichers verursacht. Um die fehlenden fünf Prozent der Umgebung zu korrigieren, wird ein linearer Anstieg des Speichers über die gesamte Simulation angenommen. Zu Beginn der Simulation beträgt der benötigte Arbeitsspeicher etwa $479MB$. Dies wird mit der Variable $a_{SpeicherBeginn}$ festgehalten. Zum Ende der Simulation liegt der maximal gemessene Wert bei $557.91MB$. Dies wird mit der Variable $a_{SpeicherEnde}$ festgehalten. Die Korrektur des fehlenden Teils der Umgebung wird nach folgender Formel durchgeführt:

$$a_{SpeicherEndeKorrigiert} = a_{SpeicherBeginn} + \frac{a_{SpeicherEnde} - a_{SpeicherBeginn}}{0.95} \tag{4.4}$$

  Die Variable $a_{SpeicherEndeKorrigiert}$, die den korrigierten Wert darstellt, entspricht 562.06. Nach dieser Rechnung wird angenommen, dass die vollständige Kartierung der Umgebung mit einem Speicheraufwand von $562,06MB$ verbunden wäre. Dieser Wert wird für die weitere Berechnung des Kriteriums „Maximaler Speicheraufwand“ verwendet. Dies ist der höchste Wert für den maximal genutzten Arbeitsspeicher von allen Pfadplanern. Das entspricht dem Wert 0.00 für das Kriterium „Maximaler Speicheraufwand“.

Weiterhin kann die in Unterabschnitt 4.2.2 geäußerte Vermutung, dass sich der genutzte Arbeitsspeicher mit die Kartierung der Umgebung erhöht, bestätigt werden. Alle globalen Pfadplaner zeigen einen steigenden Speicheraufwand.

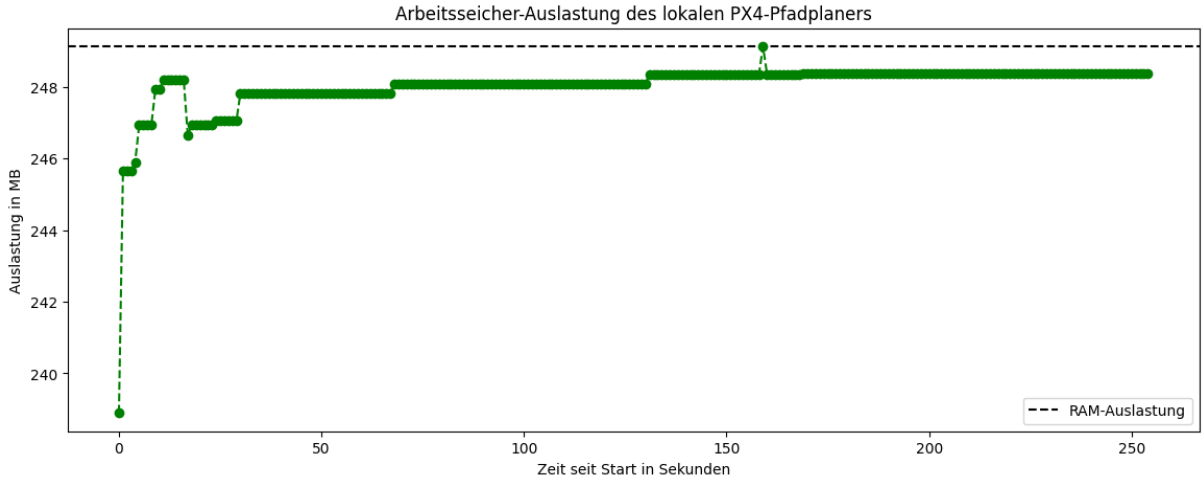


Abbildung 4.12: Der genutzte Arbeitsspeicher bei der Ausführung des lokalen PX4-Pfadplaners auf der in Abbildung 4.7 dargestellten Gazebo-Umgebung. Die gestrichelte Linie visualisiert den maximal genutzten Arbeitsspeicher.

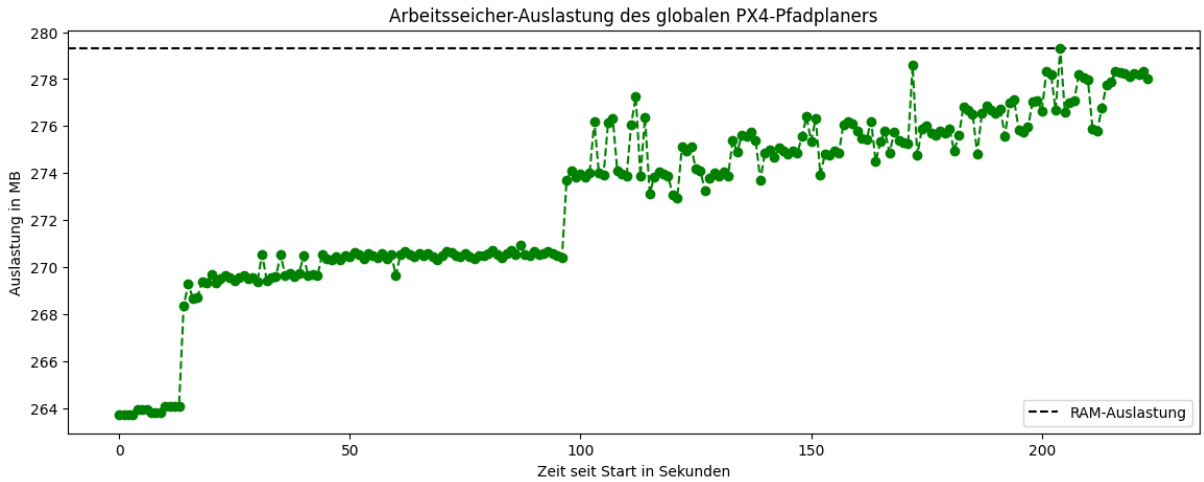


Abbildung 4.13: Der genutzte Arbeitsspeicher bei der Ausführung des globalen PX4-Pfadplaners auf der in Abbildung 4.7 dargestellten Gazebo-Umgebung. Die gestrichelte Linie visualisiert den maximal genutzten Arbeitsspeicher.

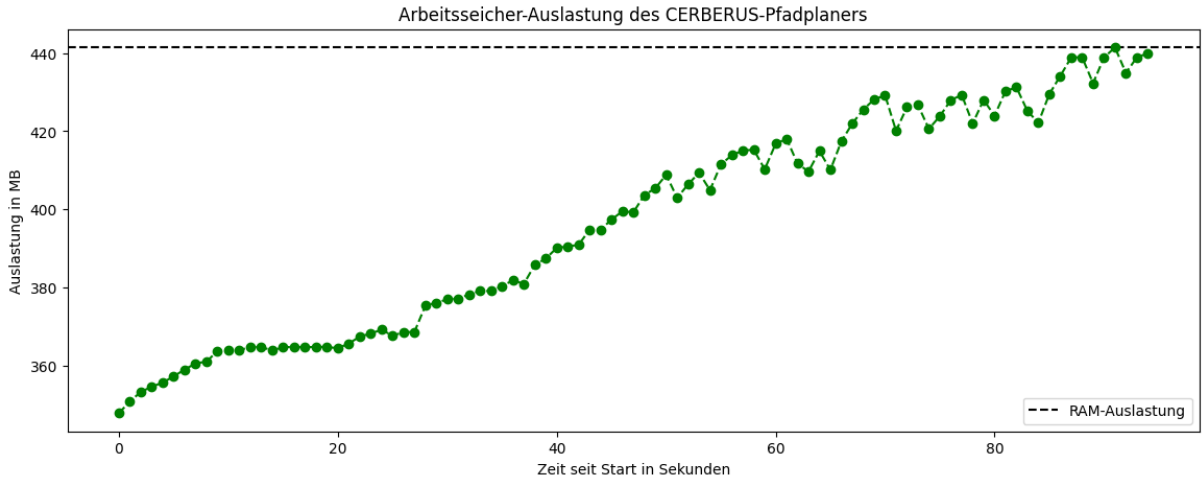


Abbildung 4.14: Der genutzte Arbeitsspeicher bei der Ausführung des globalen CERBERUS-Pfadplaners auf der in Abbildung 4.7 dargestellten Gazebo-Umgebung. Die gestrichelte Linie visualisiert den maximal genutzten Arbeitsspeicher.

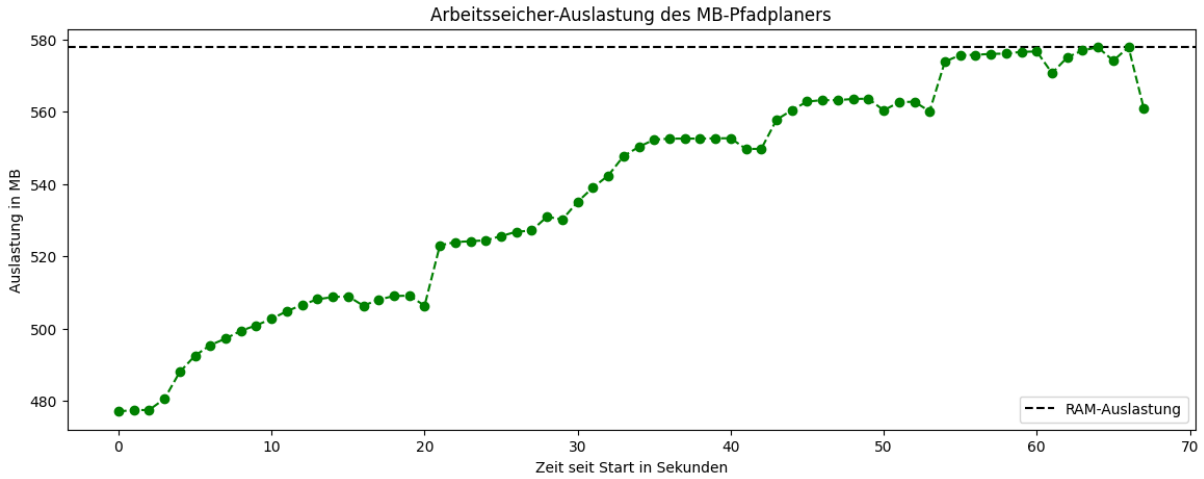


Abbildung 4.15: Der genutzte Arbeitsspeicher bei der Ausführung des globalen MB-Pfadplaners auf der in Abbildung 4.7 dargestellten Gazebo-Umgebung. Die gestrichelte Linie visualisiert den maximal genutzten Arbeitsspeicher.

4.4.9 Beobachtungen bei der Evaluation

In diesem Unterabschnitt werden die in Unterabschnitt 4.1.1 aufgestellten Vermutungen bezüglich der Pfadplaner aufgegriffen und beantwortet. Alle folgenden Beobachtungen basieren auf den experimentellen Versuchen, welche in den vorangegangenen Unterabschnitten 4.4.1 bis 4.4.8 vorgestellt werden.
Die Vermutung, dass der lokale PX4-Pfadplaner in Innenräumen häufig von lokalen Minima betroffen sein würde, konnte in experimentellen Versuchen bestätigt werden. Es konnte auch bestätigt werden, dass der lokale Pfadplaner in einem Kriterium als Referenz dienen kann. Wie in der Tabelle 4.3 dargelegt wird, hat der Planer die geringste maximale Speichernutzung von allen Pfadplanern und dient somit für das Kriterium „Maximaler Speicheraufwand“ als Benchmark.
Bei dem globalen PX4-Pfadplaner konnte bestätigt werden, dass lokale Minima kein Problem darstellen. Jedoch konnte nicht bestätigt werden, dass die Leistungsanforderung des globalen PX4-Planers höher ist als diejenigen des lokalen PX4-Planers. Diese Vermutung ist widerlegt.
Bei den beiden explorierenden Pfadplanern wurde zum einen vermutet, dass die Ressourcenanforderungen höher als bei den PX4-Planern wären. Dies kann anhand der Kriterien „Durchschnittliche Prozessorauslastung“ und „Maximaler Speicheraufwand“ bestätigt werden. Zum anderen wurde vermutet, dass explorierende Pfadplaner besser für hindernisreiche Umgebungen geeignet sind. Dies kann teilweise bestätigt werden. Das Kriterium „Minimaler Freiraum“ zeigt beispielsweise, dass der CERBERUS-Pfadplaner wenig freien Raum benötigt. Dies gilt jedoch nicht für den MB-Planer.

4.5 Anwendungsspezifische Auswertung der Evaluation

In diesem Abschnitt werden die Pfadplaner anhand einer beispielhaften Gewichtung der Kriterien ausgewertet. Die Bewertungen der Pfadplaner werden aus dem Abschnitt 4.4 übernommen. Die Gewichtungen sind im Rahmen der Arbeit vorgegeben. Eine Zusammenfassung der Ergebnisse ist in Tabelle 4.5 gegeben.
Der größte Teil der Gewichtung wird auf die vier Kriterien „3D-Pfadplaner“, „Zielpositionsvorgabe“, „Zeitkosten“ und „Minimaler Freiraum“ gesetzt. Diese Kriterien haben jeweils eine Gewichtung von 15% oder 20%. Insgesamt machen diese vier Kriterien 70% der Gesamtgewichtung aus.
Die restlichen 30% der Gewichtung sind auf die Kriterien „Höhenlimitierung“, „Kontinuierlicher Pfadplaner“, „Durchschnittliche Prozessorauslastung“ und „Maximaler Speicheraufwand“ aufgeteilt.

Es ist anzumerken, dass das Kriterium „3D-Pfadplaner“ von allen Pfadplanern gleichermaßen erfüllt ist. Daher differenzieren sich die Pfadplaner in diesem Kriterium nicht. Es wird dennoch betrachtet, da die im Rahmen der Arbeit vorgegebene Gewichtung dies verlangt. Folgende Gesamtergebnisse werden erreicht

- Der CERBERUS-Pfadplaner hat die höchste und somit beste Gesamtbewertung von 79%.
- Die zweithöchste Gesamtbewertung hat der globale PX4-Pfadplaner mit einem Wert von 60%.
- Danach folgt der lokale PX4-Pfadplaner mit einer Gesamtbewertung von 50%.
- Die geringste erreichte Gesamtbewertung hat der MB-Pfadplaner mit 21%.

Bei dem letzten Pfadplaner ist anzumerken, dass das Kriterium „Zeitkosten“ mit 0.00 Punkten gewertet wurde. Der Pfadplaner hat keine Funktionalität, die die Überprüfung dieses Kriteriums erlaubt. Dennoch wird es von der vorgegebenen Gewichtung verlangt und daher bewertet.
Diese Ergebnisse der Evaluation decken sich mit den Beobachtungen, die bei der Arbeit mit den vorgestellten Pfadplanern in den Abschnitten 4.1 und 4.4 gemacht wurden. Die in diesen Abschnitten erkannten Probleme der Pfadplaner werden in Tabelle 4.4 zusammengefasst. Neben den Problemen, die bereits in den genannten Abschnitten behandelten werden, werden allgemeine Beobachtungen genannt, welche den Einsatz in Innenräumen erschweren.

Tabelle 4.4: Übersicht der beobachteten Probleme der Pfadplaner für den Einsatz in Innenräumen.

Pfadplaner	**Probleme in Innenräumen**
lokaler PX4-Pfadplaner	• Der Pfadplaner verfällt häufig in lokale Minima. Dies resultiert in einem Stillstehen des Luftfahrzeugs. • Es kommt häufig zu Kollisionen mit Wänden und Decken. Die Kollision mit Decken kann nicht verhindert werden. • Die Limitierung der maximalen Höhe ist nicht möglich. • Es können keine engen Pfade passiert werden.
globaler PX4-Pfadplaner	• Der Pfadplaner kann häufig keinen Pfad zum Ziel ermitteln, obwohl ausreichend Platz verfügbar ist. • Es kommt zu Kollisionen mit Wänden. • Es können keine engen Pfade passiert werden.
CERBERUS-Pfadplaner	• Der Pfadplaner eignet sich für den Einsatz in Innenräumen.
MB-Pfadplaner	• Es gibt keine Funktionalität, Zielpositionen vorzugeben. • Zwischen den Planungsetappen wird das Luftfahrzeug stark abgebremst, wodurch es schwankt. Dies hat in seltenen Fällen zu einem Absturz des Luftfahrzeugs geführt. • Die Exploration erfasst nicht immer die vollständige Umgebung. • Es können keine engen Pfade passiert werden. • Die Anpassung der Flughöhenlimitierung ist nicht möglich.

Tabelle 4.5: Zusammenfassung der gewichteten Auswertung von allen untersuchten Pfadplaner. Die Namen der Pfadplaner sind aus räumlichen Gründen verkürzt: lokaler PX4-Pfadplaner (lok. PX4), globaler PX4-Pfadplaner (glob. PX4), CERBERUS-Pfadplaner (CERBERUS) und MB-Pfadplaner (MB).

Kriterium	lok. PX4	glob. PX4	CER-BERUS	MB	Gewich-tung
„Höhenlimitierung“	0.00	1.00	1.00	0.00	0.10
„3D-Pfadplaner“	1.00	1.00	1.00	1.00	0.15
„Kontinuierlicher Pfadplaner“	1.00	1.00	0.00	0.00	0.05
„Zielpositionsvorgabe“	1.00	1.00	0.50	0.00	0.15
„Zeitkosten“	0.00	0.00	1.00	k.A.	0.20
„Minimaler Freiraum“	0.00	0.00	1.00	0.30	0.20
„Durchschnittliche Prozessorauslastung“	0.96	1.00	0.45	0.00	0.10
„Maximaler Speicher-aufwand“	1.00	0.90	0.39	0.00	0.05
Gesamtbewertung	0.50	0.60	0.79	0.21	1.00

5 Ergebnisse und Diskussion

Die vorliegende Arbeit untersucht vier Pfadplaner für unbemannte Luftfahrzeuge hinsichtlich ihrer Anwendbarkeit in Innenräumen. Dazu wurde eine Metrik bestehend aus acht Kriterien erstellt und eine Evaluation der Pfadplaner mit dieser durchgeführt. Die Kriterien wurden anhand einer oder mehrerer Simulationen bewertet. Die Evaluation hatte methodisch und quantifiziert zu erfolgen, um eine Aussage darüber treffen zu können, welcher der Pfadplaner für den Einsatz in Innenräumen am besten geeignet ist.
Das Ergebnis der Evaluation ist in Tabelle 5.1 dargestellt. Der durchgeführten Evaluation zufolge eignet sich der CERBERUS-Pfadplaner von den untersuchten Pfadplanern für den Einsatz in Innenräumen am besten. Die Gewichtung der Kriterien war vorgegeben. Eine Übersicht der Gewichtung sowie Evaluation aller Pfadplaner ist in Tabelle 4.5 gegeben.

Tabelle 5.1: Gesamtergebnis der Bewertung der Pfadplaner für den Einsatz in Innenräumen. Der beste erreichbare Wert ist 1 und der schlechteste erreichbare Wert ist 0. Die genutzten Gewichtungen werden in Abschnitt 4.5 vorgestellt.

Pfadplaner	**Gesamtbewertung**
lokaler PX4-Pfadplaner	0.50
globaler PX4-Pfadplaner	0.60
CERBERUS-Pfadplaner	0.79
MB-Pfadplaner	0.21

Die Evaluation wurde unter Einhaltung der Anforderungen, die an die vorliegende Arbeit gestellt wurden, durchgeführt. Eine Zusammenfassung der Anforderungen ist in Tabelle 5.2 gegeben.

Tabelle 5.2: Übersicht der Anforderungen an die vorliegende Arbeit für die Evaluation der Pfadplaner. Mit dem ✓-Symbol wird gekennzeichnet, dass die Anforderung eingehalten wurde.

#	**Anforderung**	**Eingehalten**
1.	Für die Erfassung der Kriterien werden realitätsnahe[1] Simulationen verwendet.	✓
2.	Die erstellte Metrik besteht aus objektiv messbaren Kriterien.	✓
3.	Die Evaluation ist anhand von für Innenräume repräsentativen Umgebungen durchgeführt.	✓
4.	Die Leistungsanforderungen der Pfadplanungsalgorithmen werden bei der Evaluation berücksichtigt	✓

Die folgenden Abschnitte erläutern die in den Tabellen 5.1 und 5.2 dargestellten Ergebnisse:

- In Abschnitt 5.1 werden die tabellarisch aufgeführten Ergebnisse näher erläutert. Es wird dargelegt, wie die Anforderungen an die vorliegende Arbeit bei der Evaluation berücksichtigt wurden.
- In Abschnitt 5.2 werden die vorgestellten Ergebnisse diskutiert. Es werden die Vergleichbarkeit, Erweiterbarkeit und Aussagekraft der Kriterien thematisiert. Anschließend wird die Übertragbarkeit der Ergebnisse der vorliegenden Arbeit auf die Realität betrachtet.

5.1 Zusammenfassung anhand der Anforderungstabelle

In diesem Abschnitt werden die Anforderungen in der Reihenfolge, wie sie in Tabelle 5.2 aufgezählt sind, betrachtet:

1. Für die realitätsnahen Simulationen wird Gazebo eingesetzt. Alle Kriterien bestehen aus einem Teil der Untersuchung, in dem das UAS einen Flug durchführt. Damit ist stets sichergestellt, dass die Ergebnisse soweit wie möglich auf die Realität übertragbar sind.

[1] im Sinne der physikalischen Gesetze

2. Jedes der Kriterien der verwendeten Metrik ist objektiv quantifizierbar. Alle Kriterien werden auf das Intervall [0,1] abgebildet, sodass eine Interpretation der Ergebnisse leicht möglich ist. Zusätzlich sind die Kriterien so spezifiziert, dass die Bewertung dieser bei Pfadplanern mit unterschiedlichen Funktionen möglich ist. Die Kriterien werden in Unterabschnitt 4.2.2 beschrieben und in Tabelle 4.2 tabellarisch zusammengefasst.

3. Für die Erfassung der Kriterien werden für Innenräume repräsentative Simulationsumgebungen verwendet. Beispiele sind in den Abbildungen 4.5 und 4.6 dargestellt. Die verwendeten Umgebungen beinhalten grundsätzlich wenig Platz für das Passieren des Luftfahrzeugs. Es befinden sich stets Hindernisse in nahezu jeder Richtung nahe dem Luftfahrzeug. Ein Teil der genutzten Umgebungen erfordert die Pfadplanung über mehrere Ebenen hinweg. Dies sind Eigenschaften, die in realen Gebäuden ebenfalls anzutreffen sind.

4. Die Leistungsanforderungen der Pfadplaner werden mit zwei Kriterien evaluiert. Ein Kriterium evaluiert die durchschnittliche CPU-Auslastung und ein weiteres Kriterium erfasst den maximal beanspruchten Arbeitsspeicher des untersuchten Pfadplaners. Weiterhin wurde im Rahmen der Arbeit festgestellt, dass die Pfadplaner keine GPU-Ressourcen beanspruchen. Die GPU wird ausschließlich für die Simulation benötigt.

Weiterführend ist zu betonen, dass die untersuchten Pfadplaner unterschiedliche Funktionalitäten und Interaktionsmöglichkeiten haben. Beispielsweise besitzen nicht alle Pfadplaner die Funktion, Zielpositionen beliebig zu setzen. Daher sind im Rahmen dieser Arbeit für einige der Kriterien unterschiedliche Methoden für die Erfassung von diesen Kriterien entstanden. Dadurch dass ein Kriterium mittels unterschiedlicher Methoden erfassbar ist, ist ein weites Spektrum von Pfadplanern mit den in der vorliegenden Arbeit vorgestellten Kriterien und Methoden evaluierbar.

5.2 Diskussion der Ergebnisse

In diesem Abschnitt werden die Ergebnisse hinsichtlich der methodischen Durchführung, Aussagekraft und Realitätsnähe diskutiert. Das Vorgehen gliedert sich wie folgt:

- In dem Unterabschnitt 5.2.1 wird die Vergleichbarkeit und Vollständigkeit der genutzten Kriterien diskutiert. Für eine aussagekräftige Evaluation muss sichergestellt werden, dass die untersuchten Kriterien alle relevanten Aspekte der Pfadplaner betrachten.

- In dem Unterabschnitt 5.2.2 wird die Erweiterbarkeit der Evaluationen hinsichtlich neuer Datensätze bei Hinzuziehung weiterer Pfadplaner untersucht. Insbesondere wird diskutiert, welche Auswirkungen neue Datensätze auf die Evaluation bisheriger Datensätze haben.
- In dem Unterabschnitt 5.2.3 wird der genutzte Wertebereich für die Kriterien diskutiert. Insbesondere werden die Implikationen der Werte von Kriterien, die in Relation zueinander gesetzt werden, untersucht.
- In dem Unterabschnitt 5.2.4 wird die Übertragbarkeit der Ergebnisse auf die Realität diskutiert. Insbesondere wird die genaue Positionsschätzung der UAS im Raum untersucht.

5.2.1 Vergleichbarkeit und Vollständigkeit der Kriterien

In der vorliegenden Arbeiten wurden Pfadplaner mit unterschiedlichen Funktionalitäten bewertet. Aufgrund dieser unterschiedlichen Funktionalitäten ist es nicht immer möglich gewesen, die gleiche Funktionalität bei allen Pfadplanern für die Evaluation eines Kriteriums zu verwenden. Jedoch konnte das Kriterium bei allen Pfadplanern mit dem Einsatz verschiedener Funktionalitäten bewertet werden.
Beispielsweise wurde für das Kriterium „Durchschnittliche Prozessorauslastung“ bei zwei von vier Pfadplanern die explorierende Funktionalität eingesetzt. Bei den zwei anderen Pfadplanern wurden die Zielpositionen manuell interaktiv vorgegeben. Nicht alle Pfadplaner haben die Funktionalität, zu explorieren. Auch Zielpositionen können nicht bei allen Pfadplanern gesetzt werden. Durch eine speziell konstruierte Umgebung ist es dennoch gelungen, die Exploration eines prädestinierten Pfades zu forcieren. Dadurch wurde trotz unterschiedlicher Funktionalitäten von allen Pfadplanern der annähernd gleiche Pfad verfolgt.
Bei der Auswertung der Messungen konnte klar beobachtet werden, dass die verursachte Prozessorauslastung der explorierenden Pfadplaner signifikant höher war als diejenige der anderen Pfadplaner. Daher sind die nicht explorierenden Pfadplaner bei diesem Kriterium signifikant besser bewertet.
Auch wenn mit der Vorgehensweise die tatsächliche Prozessauslastung der jeweiligen Pfadplaner erfasst wird, wird dabei ein Aspekt vernachlässigt: Die explorierenden Pfadplaner haben einen signifikant höheren Grad der Automatisierung. Dies wird durch zusätzliche Algorithmen erreicht, welche den höheren Bedarf an Rechenressourcen zur Konsequenz haben. Für die vorliegende Arbeit war die explorierende Funktion der Pfadplaner nicht Kern der Untersuchungen. Daher ist die Vergleichbarkeit der unterschiedlichen Methoden für die Erfassung des gleichen Kriteriums legitim. Jedoch kann bei anderen Anwendungsfällen

die explorierende Funktion ein kritischer Aspekt sein, welcher durch ein eigenes Kriterium repräsentiert werden sollte.
Daher gilt es bei zukünftigen Arbeit sicherzustellen, dass die genutzten Kriterien bei allen Pfadplanern durch eine für den Kontext ausreichend vergleichbare Methodik erfasst werden. Zusätzlich gilt es zu überprüfen, ob die genutzten Kriterien vollständig die relevanten Merkmale der Untersuchung abdecken.

5.2.2 Erweiterbarkeit der Untersuchungen mit neuen Daten

In der vorliegenden Arbeit werden alle Kriterien auf das Intervall [0,1] abgebildet. Dabei entspricht der Wert 1 stets dem besten Resultat. Bei den stetigen Kriterien werden gemessene Werte, die weit größer als der Wert 1 sein können, normalisiert. Ein Beispiel für ein derartiges Kriterium ist „Maximaler Speicheraufwand“. Der Speicheraufwand wird in Byte gemessen und anschließend verarbeitet. Einer der Verarbeitungsschritte ist die Normalisierung auf das besagte Intervall. Für die Normalisierung werden jeweils alle Werte des Kriteriums der vier Pfadplaner verwendet. Darausfolgend entspricht der Pfadplaner mit dem schlechtesten Resultat dem Wert 0 und der beste Planer dem Wert 1. Durch die Verarbeitung der stetigen Werte sind die Ergebnisse bei allen Kriterien im gleichen Format. Dies erleichtert die Interpretation der Ergebnisse.
Dennoch geht damit die Problematik der Erweiterbarkeit einher: Sobald ein neuer Pfadplaner untersucht wird und den bisherigen Wertebereich eines Kriteriums unter- oder überschreitet, muss die Normalisierung für alle Pfadplaner erneut durchgeführt werden.
Damit eine Neuberechnung möglich ist, muss sichergestellt werden, dass die Messgrößen der bisherigen Evaluationen verfügbar bleiben. Dies wiederum erfordert logistischen Aufwand und kann zu Fehlern führen.

5.2.3 Aussagekraft der Kriterien unter Berücksichtigung des Wertebereiches

Wie bereits erläutert, werden alle Kriterien auf dem Intervall [0,1] abgebildet. Der Wert 0 suggeriert, dass ein Kriterium nicht erfüllt ist oder im Vergleich zu den anderen Pfadplanern am schlechtesten erfüllt ist. Weiterhin werden die Kriterien entweder anhand einer absoluten oder einer relativen Skala bewertet. Kriterien mit einer relativen Skala sind Kriterien mit zugehörigen Messgrößen, die einen stetigen Wertebereich haben.
Die Abbildung von stetigen Kriterien auf das besagte Intervall hat folgende Implikation: Wenn in der Menge der untersuchten Pfadplaner ein stetiges Kriterium von allen Pfadplanern sehr gut erfüllt wäre, würde dennoch mindestens ein Planer den Wert 0 für das

Kriterium erhalten. Dies liegt daran, dass die Verarbeitung der Messgröße so durchgeführt wird, dass der beste Pfadplaner immer mit dem Wert 1 und der schlechteste Pfadplaner immer mit dem Wert 0 bewertet wird. Dies ist auch dann der Fall, wenn die Varianz zwischen den Messgrößen des Kriteriums bei allen Pfadplanern klein ist.
In dem vorgestellten Szenario würde die Evaluation des Kriteriums fälschlicherweise suggerieren, dass mindestens ein Pfadplaner signifikant besser oder schlechter als ein anderer ist. Daher muss bei Kriterien, die relativ zu den Ergebnissen der anderen Pfadplaner betrachtet werden, der gesamte Kontext berücksichtigt werden. Nur so kann eine Fehlinterpretation der Ergebnisse ausgeschlossen und die Aussagekraft beibehalten werden.

5.2.4 Übertragbarkeit auf die Realität

Bei den Simulationen der vorliegenden Arbeit werden die Multikopter-Luftfahrzeuge unter Betrachtung der realitätsnahen physikalischen Rahmenbedingungen simuliert. Hierbei werden Effekte wie wirkende Trägheitsmomente berücksichtigt. Dennoch gibt es Teile der Simulation, welche in der Realität nicht umsetzbar sind.
Beispielsweise ist in den Simulationen die exakte Position des Luftfahrzeugs im Raum akkurat ermittelbar. Der Pfadplaner kann nur dann funktionieren, wenn die Position des Luftfahrzeugs bekannt ist.
Die technische Umsetzung einer annähernd so genauen Positionierung eines Multikopters ist in der Realität nur mit hohem Aufwand möglich. Für eine möglichst genaue Positionsschätzung werden **S**imultaneous **L**ocalization **a**nd **M**apping (SLAM)-Algorithmen eingesetzt [54]. Mit der Ausführung eines derartigen Algorithmus werden zusätzlich Systemressourcen beansprucht. Ein derartiger Algorithmus ist jedoch eine Notwendigkeit bei der Nutzung von Pfadplanern, da diese nur mit einer genauen Positionseinschätzung funktionieren.
Da die akkurate Positionseinschätzung des Luftfahrzeugs nicht Teil der Arbeit war, ist dies nicht berücksichtigt worden. Dennoch ist zu bedenken, dass für den Einsatz der Pfadplaner auf realen UAS eine genaue Positionseinschätzung eine Voraussetzung darstellt und Systemressourcen beansprucht. Dies bedeutet, dass der in der Simulation ermittelte Bedarf an Systemressourcen niedriger als der reale Bedarf an Systemressourcen ist. Dies ist bei der Auswahl der Hardwarekomponenten bei der Übertragung der Pfadplaner von der Simulation auf die Realität zu berücksichtigen.

6 Ausblick

Die Untersuchungen der vorliegenden Arbeit zeigen, dass der Einsatz von UAS in Innenräumen möglich ist. Diese Erkenntnis wurde anhand realitätsnahen Simulationen gewonnen. Aufbauende Forschungsarbeiten sollten diese Erkenntnis systematisch bestätigen und erweitern.
Zunächst gilt es die Allgemeingültigkeit der Ergebnisse dieser Arbeit zu überprüfen. Insbesondere sollte überprüft werden, ob stichprobenbasierte Pfadplaner, welche aus einer Fusion von lokalen und globalen Planern bestehen, die beste Lösung für Innenräume darstellen. Diese Konklusion wurde basierend auf der Untersuchung von vier Pfadplanern getroffen. Da dies eine zu geringe Datengrundlage für eine allgemeine Aussage ist, gilt es hierbei die Datengrundlage zu erweitern. Dafür müssen weitere Pfadplaner-Umsetzungen in gleichen bzw. ähnlichen Versuchen evaluiert werden.
Weiterhin können aufbauende Arbeiten das Wissen in folgenden Aspekten bzw. Bereichen ausbauen:

- Die Optimierung der Leistung der Pfadplaner in Innenräumen.
- Die Verbesserung der Aussagekraft der Evaluation derartiger Pfadplaner.
- Die Erschließung von weiteren Anwendungsbereichen für den Einsatz von UAS in komplexen Umgebungen.

Diese Bereiche werden in gleicher Reihenfolge in den folgenden Abschnitten betrachtet.
Die Leistung der Pfadplaner in Innenräumen kann zum einen an der Software und zum anderen an der Hardware optimiert werden.
Bei der Software können die Algorithmen für die Pfadplanung oder weitere benötigte Softwares verbessert werden. Beispielsweise werden die Ergebnisse stark von der Flugsteuerung beeinflusst. Eine Verbesserung der Flugsteuerung kann sich vorteilhaft auf die Effizienz, mit der Pfade verfolgt werden, auswirken.
Die Hardware kann ebenfalls auf den Einsatzbereich in Innenräumen optimiert werden. Dazu können die Rahmenbedingen, die durch die Innenräume auferlegt werden, bei der Konzeptionierung des eingesetzten Luftfahrzeuges berücksichtigt werden. Beispielsweise kann die Flugdynamik, die Robustheit sowie Größe des Luftfahrzeuges auf hindernisreiche Umgebungen optimiert werden. Die erwarteten Geschwindigkeiten sind gering und die

Wahrscheinlichkeit für Kollisionen ist durch die hindernisreiche Umgebung erhöht. Zusätzlich ermöglicht ein kleiner Formfaktor des Luftfahrzeuges die effizientere Nutzung des begrenzten Freiraums.
Der zweite Aspekt, welcher in folgenden Forschungsarbeiten untersucht werden kann, ist die Verbesserung der Aussagekraft der Evaluation. Insbesondere sind die stetigen Kriterien, deren Wert relativ ist unter Umständen anfällig für Fehlinterpretation. Daher könnte anstatt einer relativen Skala für die Kriterien eine absolute Skala entworfen werden. Damit würde die Vergleichbarkeit und Interpretierbarkeit von den Evaluation sichergestellt werden. Die absoluten Kriterien würden die Anfälligkeit der Evaluation für Fehlinterpretation verringern.
Der dritte Aspekt, der in Folgearbeiten untersucht werden kann, ist die Erschließung von weiteren Anwendungsbereichen für den Einsatz von UAS in komplexen Umgebungen. In der vorliegenden Arbeit wurden komplexe Umgebungen in Form von Innenräumen behandelt. Jedoch können die Ergebnisse auch auf andere komplexe Umgebungen übertragen werden. Da es sich um eine moderne Technologie handelt, ist davon auszugehen, dass noch weitere Anwendungsbereiche zu erschließen sind. Insbesondere verschieben sich durch technologischen Fortschritt stetig die Grenzen dieser Technologie. Es ist davon auszugehen, dass zukünftig die Reichweite von derartigen Systemen zunimmt, die Rechenleistung sowie Robustheit steigt, die Sensorik eine verbesserte Umweltwahrnehmung ermöglicht und die Kosten gesenkt werden. Aufbauende Arbeiten können sich mit den erwarteten Verbesserungen der Technologie auseinandersetzten und neue Anwendungsbereiche erschließen.

Danksagung

An dieser Stelle möchte ich mich bei allen bedanken, die mich bei der Durchführung dieser Arbeit unterstützt haben.
Allem voran möchte ich meinem betrieblichen Betreuer Nikolaus Alexander Ammann danken. Bedanken möchte ich mich für die fachliche und technische Unterstützung sowie die aktive Hilfe beim Umgang mit Schwierigkeiten. Ohne die geleistete Hilfe wäre die Umsetzung der vorliegenden Arbeit in der Form nicht möglich gewesen.
Ein besonderer Dank gebührt Professor Holger Gerhards, der mich bei dem wissenschaftlichen Schreiben unterstützt hat und stets hilfreiche Anregungen und Ideen für mich hatte.
Abschließend möchte ich mich bei meiner Kommilitonin Maikhanh Dang bedanken, die mir mit viel Geduld und Hilfsbereitschaft zur Seite stand.

Literatur

[1] C. Goerzen, Z. Kong und B. Mettler. „A Survey of Motion Planning Algorithms from the Perspective of Autonomous UAV Guidance“. In: *Journal of Intelligent and Robotic Systems* (2009).

[2] Bing Han, Qingmei Li und Chengqi Cheng. „Research on UAV Indoor Path Planning Algorithm Based on Global Subdivision Grids“. In: *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*. 2021, S. 8503–8506. DOI: 10.1109/IGARSS47720.2021.9554242.

[3] Jongmin Park, Sooyoung Jang und Younghoon Shin. „Indoor Path Planning for an Unmanned Aerial Vehicle via Curriculum Learning“. In: *2021 21st International Conference on Control, Automation and Systems (ICCAS)*. 2021, S. 529–533. DOI: 10.23919/ICCAS52745.2021.9649794.

[4] Lidia Rocha u. a. „A UAV Global Planner to Improve Path Planning in Unstructured Environments“. In: *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2021, S. 688–697. DOI: 10.1109/ICUAS51884.2021.9476692.

[5] o. V. *Simulation*. 2022. URL: https://docs.px4.io/main/en/simulation/ (besucht am 26. 08. 2022).

[6] Dharmadhikari, Mihir. *Graph-based Exploration Planner 2.0*. 2021. URL: https://github.com/ntnu-arl/gbplanner_ros (besucht am 26. 08. 2022).

[7] Keller, Sarah. *Gesamtinvestitionen in die weltweite Drohnen-Industrie in den Jahren 2008 bis 2020*. 2022. URL: https://de.statista.com/statistik/daten/studie/981239/umfrage/weltweite-gesamtinvestitionen-in-die-drohnen-industrie/ (besucht am 26. 08. 2022).

[8] Market Prospects. *UAV Market and Technology Development Trend*. 2022. URL: https://www.market-prospects.com/articles/uav-market-trend (besucht am 26. 08. 2022).

[9] Amazon Staff. *Amazon Prime Air prepares for drone deliveries*. 2022. URL: https://www.aboutamazon.com/news/transportation/amazon-prime-air-prepares-for-drone-deliveries (besucht am 26. 08. 2022).

[10] XGuard Robotics. *integrated autonomous security systems*. o. J. URL: https://www.xguardrobotics.com/#commercial (besucht am 26. 08. 2022).

[11] Azur Drones. *SKEYETECH*. 2022. URL: https://www.azurdrones.com/product/skeyetech/ (besucht am 26. 08. 2022).

[12] Alex Netto. *X2 Color, 3D Scan & Cloud: An Integrated Solution for Autonomous Commercial Drone Inspection*. 2021. URL: https://www.skydio.com/blog/autonomous-commercial-drone-inspection/ (besucht am 26. 08. 2022).

[13] Airbone Drones. *Inspection drone: Upgraded with Autonomy and AI*. 2021. URL: https://www.airbornedrones.co/inspection-drone-upgraded-with-autonomy-and-ai/ (besucht am 26. 08. 2022).

[14] Zendai Kashino, Goldie Nejat und Beno Benhabib. „Multi-UAV based Autonomous Wilderness Search and Rescue using Target Iso-Probability Curves“. In: *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2019, S. 636–643. DOI: 10.1109/ICUAS.2019.8798354.

[15] Rabah Louali u. a. „Implementation of an UAV Guidance, Navigation and Control System based on the CAN data bus: Validation using a Hardware In the Loop Simulation“. In: *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. 2017, S. 1418–1423. DOI: 10.1109/AIM.2017.8014217.

[16] Guan-Ting Tu und Jih-Gau Juang. „Path Planning and Obstacle Avoidance Based on Reinforcement Learning for UAV Application“. In: *2021 International Conference on System Science and Engineering (ICSSE)*. 2021, S. 352–355. DOI: 10.1109/ICSSE52999.2021.9537945.

[17] Mohammadreza Radmanesh u. a. „Overview of Path Planning and Obstacle Avoidance Algorithms for UAVs: A Comparative Study“. In: *Unmanned Systems* 6 (Apr. 2018), S. 1–24. DOI: 10.1142/S2301385018400022.

[18] Y. K. Hwang und N. Ahuja. „Gross motion planning—a survey“. In: *ACM Computing Surveys* (1992).

[19] Tanja Baumann. „Obstacle Avoidance for Drones Using a 3DVFH* Algorithm“. In: *Spring Term 2018* 67 (2018).

[20] Zheng Yu u. a. „Overview of Research on 3D Path Planning Methods for Rotor UAV“. In: *2021 International Conference on Electronics, Circuits and Information Engineering (ECIE)*. 2021, S. 368–371. DOI: 10.1109/ECIE52353.2021.00081.

[21] S. Kuraman und E. Franzzoli. „Sampling-based Algorithms for Optimal Motion Planning“. In: *The International Journal of Robotics Research* (2011).

[22] Mehmet Korkmaz und Akif Durdu. „Comparison of optimal path planning algorithms“. In: *2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET)*. 2018, S. 255–258. DOI: 10.1109/TCSET.2018.8336197.

[23] L.E. Kavraki u. a. „Probabilistic roadmaps for path planning in high-dimensional configuration spaces“. In: *IEEE Transactions on Robotics and Automation* 12.4 (1996), S. 566–580. DOI: 10.1109/70.508439.

[24] Chunyu Ju, Qinghua Luo und Xiaozhen Yan. „Path Planning Using Artificial Potential Field Method And A-star Fusion Algorithm“. In: *2020 Global Reliability and Prognostics and Health Management (PHM-Shanghai)*. 2020, S. 1–7. DOI: 10.1109/PHM-Shanghai49105.2020.9280929.

[25] Yang Gao u. a. „A New Path Evaluation Method for Path Planning With Localizability“. In: *IEEE Access* 7 (2019), S. 162583–162597. DOI: 10.1109/ACCESS.2019.2950725.

[26] Yoshua Bengio u. a. „Curriculum learning“. In: *Proceedings of the 26th annual international conference on machine learning*. 2009, S. 41–48.

[27] A. Sohn. „Parallel bidirectional A* search on a symmetry multiprocessor“. In: *Proceedings of 1993 5th IEEE Symposium on Parallel and Distributed Processing*. 1993, S. 19–22. DOI: 10.1109/SPDP.1993.395556.

[28] Pratyaksa Ocsa Nugraha Saian, Suyoto und Pranowo. „Optimized A-Star algorithm in hexagon-based environment using parallel bidirectional search“. In: *2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE)*. 2016, S. 1–5. DOI: 10.1109/ICITEED.2016.7863246.

[29] Yang Gao u. a. „An online path planning approach of mobile robot based on particle filter“. In: *Industrial Robot: An International Journal* 40.4 (2013), S. 305–319. ISSN: 0143-991X. DOI: 10.1108/01439911311320813.

[30] Bakir Lacevic, Paolo Rocco und Morten Strandberg. „Safe motion planning for articulated robots using RRTs“. In: *2011 XXIII International Symposium on Information, Communication and Automation Technologies*. 2011, S. 1–7. DOI: 10.1109/ICAT.2011.6102089.

[31] Nuwan Ganganath, Chi-Tsun Cheng und Chi K. Tse. „A Constraint-Aware Heuristic Path Planner for Finding Energy-Efficient Paths on Uneven Terrains“. In: *IEEE Transactions on Industrial Informatics* 11.3 (2015), S. 601–611. DOI: 10.1109/TII.2015.2413355.

[32] o.V. *Software Overview*. o. J. URL: https://px4.io/software/software-overview/ (besucht am 26. 08. 2022).

[33] Open Robotics. *Simulate before you build*. o. J. URL: https://gazebosim.org/home (besucht am 26. 08. 2022).

[34] Dronecode. *QGroundControl*. o. J. URL: http://qgroundcontrol.com/ (besucht am 26. 08. 2022).

[35] Jaeyoung-Lim. *Obstacle Detection and Avoidance*. 2020. URL: https://github.com/PX4/PX4-Avoidance (besucht am 26. 08. 2022).

[36] ShaneLoretz. *ROS Noetic Ninjemys*. 2020. URL: http://wiki.ros.org/noetic (besucht am 26. 08. 2022).

[37] HabibOladepo. *nodelet*. 2018. URL: http://wiki.ros.org/Nodes (besucht am 26. 08. 2022).

[38] Saito, Isaac. *nodelet*. 2017. URL: http://wiki.ros.org/nodelet (besucht am 26. 08. 2022).

[39] Woodall, William. *rviz*. 2018. URL: http://wiki.ros.org/rviz (besucht am 26. 08. 2022).

[40] o. V. *MAVLink Developer Guide*. 2021. URL: https://mavlink.io/en/ (besucht am 26. 08. 2022).

[41] Canonical Ltd. *Alternative downloads*. 2022. URL: https://ubuntu.com/download/alternative-downloads (besucht am 26. 08. 2022).

[42] Vilhjalmur Vilhjalmsson. „Risk-based Pathfinding for Drones“. In: (2016).

[43] Tung Dang u. a. „Graph-based subterranean exploration path planning using aerial and legged robots“. In: *Journal of Field Robotics* 37.8 (2020). Wiley Online Library, S. 1363–1388.

[44] Mihir Kulkarni u. a. „Autonomous Teamed Exploration of Subterranean Environments using Legged and Aerial Robots“. In: *2022 International Conference on Robotics and Automation (ICRA)*. 2022, S. 3306–3313. DOI: 10.1109/ICRA46639.2022.9812401.

[45] Dharmadhikari, Mihir. *mbplanner_ws*. 2020. URL: https://github.com/ntnu-arl/mbplanner_ros (besucht am 26. 08. 2022).

[46] Mihir Dharmadhikari u. a. „Motion Primitives-based Path Planning for Fast and Agile Exploration using Aerial Robots“. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020, S. 179–185. DOI: 10.1109/ICRA40945.2020.9196964.

[47] Armin Hornung u. a. „OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees“. In: *Autonomous Robots* (2013). DOI: 10.1007/s10514-012-9321-0. URL: https://octomap.github.io.

[48] Defense Advanced Research Projects Agency. *DARPA Subterranean (SubT) Challenge (Archived)*. o. J. URL: https://www.darpa.mil/program/darpa-subterranean-challenge (besucht am 26. 08. 2022).

[49] Rahul Kala. „Rapidly exploring random graphs: motion planning of multiple mobile robots“. In: *Advanced Robotics* 27.14 (2013), S. 1113–1122. DOI: 10.1080/01691864.2013.805472.

[50] MichaelFerguson. *dynamic_reconfigure*. 2015. URL: http://wiki.ros.org/dynamic_reconfigure (besucht am 26. 08. 2022).

[51] Saito, Isaac. *UsingDynparamToChangeHokuyoLaserParameters*. 2020. URL: http://wiki.ros.org/hokuyo_node/Tutorials/UsingDynparamToChangeHokuyoLaserParameters#PythonAPI (besucht am 26. 08. 2022).

[52] Dronecode. *UsingDynparamToChangeHokuyoLaserParameters*. o. J. URL: https://dev.px4.io/v1.10_noredirect/en/ros/mavros_installation.html (besucht am 26. 08. 2022).

[53] Robitaille, Thomas. *astrofrog/psrecord*. 2020. URL: https://github.com/astrofrog/psrecord (besucht am 26. 08. 2022).

[54] Hriday Bavle u. a. „VPS-SLAM: Visual Planar Semantic SLAM for Aerial Robotic Systems“. In: *IEEE Access* 8 (2020), S. 60704–60718. DOI: 10.1109/ACCESS.2020.2983121.