



Universität der Bundeswehr München

Institut für  
Flugsysteme

# Nutzung von Sensorleistungsmodellen zur Generierung optimaler Flugtrajektorien für unbemannte luftgestützte Aufklärungssysteme

Markus Zwick

Vollständiger Abdruck der von der Fakultät für Luft- und Raumfahrttechnik der  
Universität der Bundeswehr München zur Erlangung des akademischen Grades eines

**Doktor-Ingenieurs (Dr.-Ing.)**

angenommenen Dissertation.

Gutachter:

1. Univ.-Prof. Dr.-Ing. Peter Stütz
2. Univ.-Prof. Dr. rer. nat. Matthias Gerdts
3. Hon.-Prof. Dr.-Ing. Johann Bals

Diese Dissertation wurde am 09.04.2024 bei der Universität der Bundeswehr München  
eingereicht und durch die Fakultät für Luft- und Raumfahrttechnik am 03.08.2024  
angenommen. Die mündliche Prüfung fand am 06.08.2024 statt.



# Danksagung

Diese Arbeit entstand im Rahmen meiner Tätigkeit als wissenschaftlicher Mitarbeiter an der Professur für Luftfahrttechnik des Instituts für Flugsysteme an der Universität der Bundeswehr (UniBw) in München. Während dieser Zeit haben mich zahlreiche Menschen begleitet, inspiriert und unterstützt, ohne die diese Arbeit nicht möglich gewesen wäre. Die wichtigsten Unterstützer möchte ich nachfolgend erwähnen:

Mein größter Dank gilt meinem Doktorvater Herrn Prof. Peter Stütz für seine wertvolle Unterstützung, Expertise und das Vertrauen, das er mir all die Jahre entgegengebracht hat. Seine Anregungen und die konstruktiven Diskussionen haben maßgeblich zum Erfolg dieser Arbeit beigetragen und mich stets motiviert, mein Bestes zu geben.

Ein besonderer Dank gilt auch Herrn Prof. Matthias Gerdts und Herrn Prof. Johann Bals für das Interesse an meiner Arbeit und die wissenschaftliche Betreuung als Gutachter. Herrn Prof. Michael Schultz danke ich für die Leitung und Organisation der Promotionskommission.

Ein herzliches Dankeschön geht auch an die zahlreichen Kolleginnen und Kollegen vom Institut für Flugsysteme und insbesondere der Professur für Luftfahrttechnik. Ich bin dankbar für die vielen Gespräche, den Wissensaustausch und die Erfahrungen die meine fünf Jahre an der UniBw so bereichert haben. Besonders bedanken möchte ich mich bei meinen Mitstreitern im Heli-Sim. Wir haben gemeinsam geackert, geschwitzt, uns gegenseitig unterstützt und motiviert. Durch sie habe ich sehr viel gelernt und ich hoffe, ich konnte auch einen Teil davon zurückgeben. Ich möchte diese Erfahrung und die gemeinsamen Jahre nicht missen.

Abschließend, aber nicht weniger wichtig, möchte ich meiner Familie – meinen Eltern und meinen Geschwistern – meinen tiefsten Dank aussprechen. Neben ihrer unermüdlichen und liebevollen Unterstützung haben Sie mich geprägt und Werte vermittelt, ohne die ich diese Doktorarbeit nicht hätte realisieren können – dafür bin ich ihnen von Herzen dankbar.

Markus Zwick  
im August 2024





# Zusammenfassung

Unbemannte Luftfahrzeuge (UAVs) mit bildgebender Sensorik werden zunehmend zur luftgestützten Objektdetektion in unterschiedlichen zivilen, kommerziellen oder militärischen Anwendungen eingesetzt. Die Objektdetektion beschreibt die automatisierte Lokalisierung und Klassifizierung von Objekten (z. B. Fahrzeugen) in den sensorisch erfassten Bilddaten. Die Detektionsleistung stellt ein quantitatives Maß für die statistische Korrektheit der Objektdetektion dar. Eine hohe Detektionsleistung ist daher essenziell für eine zuverlässige Detektion und Voraussetzung für eine erfolgreiche Aufklärung.

Eine hohe Detektionsleistung stellt hohe Anforderungen an die Leistungsfähigkeit der Perzeptionskette, die sich aus der bildgebenden Sensorik, der Sensordatenverarbeitung und den Algorithmen zur Objektdetektion zusammensetzt. Verschiedene Umgebungszustände (z. B. Helligkeit, Sichtbedingungen) können sich negativ auf die Sensordatenerfassung und die nachfolgende Verarbeitungskette auswirken, was letztlich zu einer Verschlechterung der Detektionsleistung führen kann.

Um dem entgegen zu wirken, wird in dieser Arbeit ein Konzept zur sensormodellbestimmten Befliegungsplanung entwickelt und evaluiert. Dieses hat zum Ziel, lokal oder global optimale Referenz-Flugtrajektorien für das UAV zu generieren, um die daraus resultierende Detektionsleistung zu steigern. Hierfür wird ein Sensorleistungsmodell entwickelt, das die Detektionsleistung einer Perzeptionskette bezüglich spezifischer Umgebungszustände abbildet. Bei der sensormodellbestimmten Befliegungsplanung wird das Leistungsmodell genutzt, um insbesondere den Einfluss der photogrammetrischen Zustände auf die Detektionsleistung zu modellieren. Dadurch können geeignete UAV-Positionen mit hoher Detektionsleistung identifiziert werden, die wiederum Grundlage für die Generierung der Referenztrajektorien sind. Die Trajektorienoptimierung erfolgt mit Hilfe zweier mathematischer Optimierungsmethoden zur Generierung lokal oder global optimaler Referenztrajektorien unter Berücksichtigung perzeptiver, plattformspezifischer und missionsspezifischer Anforderungen und Randbedingungen.

Abschließend wird das Konzept der sensormodellbestimmten Befliegungsplanung anhand simulierter Aufklärungsszenarien zur luftgestützten Fahrzeugdetektion evaluiert. Im direkten Vergleich mit Benchmark-Trajektorien resultieren aus den lokal optimalen Referenztrajektorien der sensormodellbestimmten Befliegungsplanung Detektionsleistungen, die die Leistungswerte der Benchmark-Trajektorien übertreffen. Auf Basis global optimaler Referenztrajektorien können noch höhere Detektionsleistungen erzielt werden, die nahe am theoretischen Maximalwert liegen.



# Abstract

Unmanned aerial vehicles (UAVs) with imaging sensors are increasingly used for aerial detection of ground objects in various civil, commercial or military applications. Object detection involves the automated localization and classification of objects (e. g. vehicles) within the sensor-acquired image data. Detection performance represents a quantitative measure for the statistical correctness of object detection. A high detection performance is therefore essential for reliable detection and a prerequisite for a successful reconnaissance.

A high detection performance is demanding on the performance of the perception chain, which consists of the imaging sensor, sensor data processing and algorithms for object detection. Various environmental conditions (e. g. brightness, visibility conditions) can negatively affect sensor data acquisition and the subsequent processing chain, which can ultimately deteriorate detection performance.

To counteract this, a concept for sensor-model-based flight planning is developed and evaluated in this thesis. The goal is to generate local or global optimal reference flight trajectories for the UAV in order to increase the resulting detection performance. For this purpose, a sensor performance model is developed that maps the detection performance of a perception chain with respect to specific environmental conditions. This performance model is used in sensor-model-based flight planning, in particular to model the influence of photogrammetric conditions on the detection performance. Thereby, suitable UAV positions with high detection performance can be identified, which are the basis for the generation of the reference trajectories. Trajectory optimization is performed using two mathematical optimization methods to generate either local or global optimal reference trajectories, while taking into account perceptual, platform-specific, and mission-specific requirements.

Finally, the concept is evaluated based on simulated aerial reconnaissance scenarios for ground based vehicle detection. In direct comparison with benchmark trajectories, the local optimal reference trajectories of the sensor-model-based flight planning yield detection performances that exceed the performance values of the benchmark trajectories. Even higher detection performances can be achieved from the global optimal reference trajectories, which are close to the theoretical maximum value.



# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>xiii</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Problembeschreibung . . . . .	3
1.3 Forschungsziel . . . . .	5
1.4 Themenabgrenzung und Konkretisierung . . . . .	7
1.4.1 Positionierung von Sensorplattform und Sensorfootprint . . . . .	8
1.4.2 Einschränkung perzeptiver, plattformspezifischer und missionsspezifischer Aspekte . . . . .	9
1.5 Aufbau der Arbeit . . . . .	11
<b>2 Stand der Forschung</b>	<b>13</b>
2.1 Modellierung der Leistungsfähigkeit bildgebender Sensorik . . . . .	13
2.1.1 Einflussfaktoren auf die Detektionsleistung . . . . .	13
2.1.2 Ansätze von Sensorleistungsmodellen . . . . .	15
2.1.3 Diskussion . . . . .	17
2.2 Methoden zur Planung und Optimierung von Flugtrajektorien . . . . .	18
2.2.1 Zeit- und wertdiskrete Methoden . . . . .	19
2.2.2 Zeit- und wertkontinuierliche Methoden . . . . .	20
2.2.3 Explorative Methoden . . . . .	23
2.2.4 Diskussion . . . . .	26
2.3 Ansätze zur koordinierten Sensor- und UAV-Führung . . . . .	28
2.3.1 Ansätze zum Sensormanagement . . . . .	29
2.3.2 Sensorbestimmte UAV-Flugführung . . . . .	33
2.3.3 Diskussion . . . . .	36
2.4 Forschungsbedarf und Definition der Forschungsfragen . . . . .	37
<b>3 Konzept und Methodenauswahl</b>	<b>41</b>
3.1 Grundlage zur Steigerung der Detektionsleistung . . . . .	41

## Inhaltsverzeichnis

3.2	Anforderungen an die sensormodellbestimmte Befliegungsplanung . . . . .	43
3.3	Konzeptbeschreibung . . . . .	46
3.3.1	Ablaufbeschreibung eines Aufklärungsauftrags durch das ESPM . . . . .	47
3.3.2	Integration des ESPM in die Systemarchitektur des UAVs . . . . .	49
3.3.3	Systemarchitektur des ESPM . . . . .	50
3.3.4	Zusammenfassung . . . . .	52
3.4	Generierung von Sensorleistungsmodellen . . . . .	53
3.4.1	Grundlagen der Sensorleistungsmodelle . . . . .	53
3.4.2	Perzeptionskette . . . . .	56
3.4.3	Sensorleistungsmodelle mit Machine Learning basierten Klassifikatoren . . . . .	58
3.4.4	Sensorleistungsmodell für einen Deep Learning Objektdetektor . . . . .	61
3.4.4.1	YOLOv3-Objektdetektor . . . . .	62
3.4.4.2	Vorangegangene Arbeiten . . . . .	63
3.4.4.3	Modellierung des Sensorleistungsmodells . . . . .	66
3.4.5	Abbildung der Detektionsleistung durch Perception Maps . . . . .	73
3.4.6	Zusammenfassung . . . . .	77
3.5	Grundlagen zur Generierung der UAV-Flugtrajektorie . . . . .	79
3.5.1	Vereinfachungen und Anforderungen an die Flugtrajektorie . . . . .	79
3.5.2	UAV-Bewegungsgleichungen in der Ebene . . . . .	81
3.5.2.1	Zeitkontinuierliche Bewegungsgleichungen . . . . .	82
3.5.2.2	Zeitdiskrete Bewegungsgleichungen . . . . .	84
3.5.3	Dubins-Pfadplanung . . . . .	86
3.5.4	Zusammenfassung . . . . .	90
3.6	Mathematische Methoden zur Trajektorienoptimierung . . . . .	90
3.6.1	Optimalsteuerungsprobleme . . . . .	91
3.6.1.1	Zeitkontinuierliche Darstellung . . . . .	92
3.6.1.2	Zeitdiskrete Darstellung . . . . .	93
3.6.1.3	Lösungsverfahren für Optimalsteuerungsprobleme . . . . .	96
3.6.2	Dynamische Programmierung . . . . .	97
3.6.3	Nichtlineare modellprädiktive Regelung . . . . .	102
3.6.3.1	Diskretisierung der Systemdynamik . . . . .	102
3.6.3.2	Funktionsprinzip der modellprädiktiven Regelung . . . . .	103
3.6.3.3	Lineare und nichtlineare modellprädiktive Regelung . . . . .	104
3.6.3.4	Mathematische Grundlagen von NMPC . . . . .	106
3.6.4	Zusammenfassung . . . . .	109

<b>4</b>	<b>Umsetzung des Konzepts in ein Experimentalsystem</b>	<b>111</b>
4.1	Systembeschreibung . . . . .	111
4.2	Implementierung der Sensorfootprintpfadplanung . . . . .	113
4.2.1	Sensorpfadplanung und Bestimmung der Sensorfootprints . . . . .	115
4.2.2	Generierung footprintspezifischer Perception Maps . . . . .	117
4.3	Zeitkontinuierliche Trajektorienoptimierung . . . . .	118
4.3.1	UAV-Pfadplanung . . . . .	119
4.3.2	Kombinierte UAV-Pfadplanung und modellprädiktive Regelung . . . . .	123
4.3.3	Entwicklungsumgebung zur Implementierung der modellprädiktiven Regelung . . . . .	126
4.4	Zeit- und wertdiskrete Trajektorienoptimierung . . . . .	126
4.4.1	Dynamische Programmierung mit Dubins-Pfaden . . . . .	128
4.4.2	Realisierung der dynamischen Programmierung mit Dubins-Pfaden . . . . .	132
4.4.3	Glättung der Dubins-Pfade . . . . .	133
4.5	Implementierung der Sensorsteuerung . . . . .	133
4.6	Zusammenfassung . . . . .	135
<b>5</b>	<b>Grundlagen der Evaluierung</b>	<b>137</b>
5.1	Definition der Untersuchungsgegenstände . . . . .	137
5.2	Generierung von Benchmark-Trajektorien . . . . .	139
5.3	Einstichproben t-Test . . . . .	140
5.4	Experimentalumgebung . . . . .	142
<b>6</b>	<b>Evaluierung und Ergebnisse</b>	<b>143</b>
6.1	Evaluierung der Befliegungsplanung durch Beispielmissionen . . . . .	143
6.1.1	Versuchsplanung . . . . .	145
6.1.2	Beispielmission Routenaufklärungsszenario . . . . .	148
6.1.2.1	Befliegungsplanung mit NMPC-Trajektorienoptimierung . . . . .	149
6.1.2.2	Befliegungsplanung mit DP-Trajektorienoptimierung . . . . .	159
6.1.3	Beispielmission Flächenaufklärungsszenario . . . . .	162
6.1.3.1	Befliegungsplanung mit NMPC-Trajektorienoptimierung . . . . .	163
6.1.3.2	Befliegungsplanung mit DP-Trajektorienoptimierung . . . . .	169
6.1.4	Zusammenfassung . . . . .	172
6.2	Untersuchung von Anforderungen und Randbedingungen bei der Befliegungsplanung . . . . .	173
6.2.1	Untersuchung perzeptiver Anforderungen . . . . .	173

## *Inhaltsverzeichnis*

6.2.2	Untersuchung sensorplattformspezifischer Randbedingungen . . . . .	174
6.2.3	Zusammenfassung . . . . .	176
6.3	Evaluierung der Aufklärungs- und Berechnungsdauer . . . . .	177
6.3.1	Evaluierung der Aufklärungsdauer . . . . .	177
6.3.2	Untersuchung der Berechnungsdauer . . . . .	180
6.3.3	Zusammenfassung . . . . .	182
<b>7</b>	<b>Diskussion und Ausblick</b>	<b>183</b>
7.1	Zusammenfassung der Zwischenziele und des Forschungsziels . . . . .	183
7.2	Diskussion . . . . .	185
7.2.1	Defizite bei der Modellierung perceptiver Einflussfaktoren . . . . .	185
7.2.2	Einschränkungen bei der Modellierung sensorplattformspezifischer Randbedingungen . . . . .	186
7.3	Technisch-wissenschaftliche Beiträge . . . . .	186
7.4	Ausblick und Anknüpfungspunkte für die weitere Forschung . . . . .	188
	<b>Literatur</b>	<b>189</b>



# Abkürzungsverzeichnis

AP	Average precision
APF	Artificial potential field
BLOB	Binary large object
CC	Classification cascade
CC-SLM	Sensorleistungsmodell mit CC-Klassifikator
CNN	Convolutional neural network
DL	Deep learning
DOTA	Dataset for object detection in aerial images
DP	Dynamische Programmierung
DPM	Deformable part model
EO	Elektrooptisch
FN	False negative
FP	False positive
GNC	Guidance, navigation and control
GSD	Ground sample distance
GUI	Graphical user interface
IR	Infrarot
KNN	Künstliches neuronales Netz
MALE	Medium altitude long endurance
MAP	Multiarmed bandit
MDP	Markow decision process
MPC	Model predictive control
NMPC	Nonlinear model predictive control
PM	Perception Map
POMDP	Partially observable Markow decision process
PRM	Probabilistic roadmap
RCNN	Region-based convolutional neural network

## *Abkürzungsverzeichnis*

RRT	Rapidly-exploring random tree
SLAM	Simultaneous localization and mapping
SQP	Sequentielle quadratische Programmierung
SSD	Single-shot multibox detection
TM	Template matching
TN	True negative
TP	True positive
UAS	Unmanned aerial system
UAV	Unmanned aerial vehicle
YOLO	You only look once
YOLO-SLM	Sensorleistungsmodell mit YOLO-Objektdetektor

# 1 Einleitung

Unbemannte Luftfahrzeuge (engl.: Unmanned Aerial Vehicles, UAVs<sup>1</sup>) werden häufig eingesetzt, um mit bildgebenden Sensoren Aufklärungsaufgaben in verschiedenen Anwendungsdomänen durchzuführen. Das Ziel in dieser Arbeit ist es, durch eine engere Koordination der UAV-Flugführung mit der Sensorsteuerung die Detektionsleistung für die luftgestützte Aufklärung zu steigern. Hierzu wird die Detektionsleistung der bildgebenden Sensorik durch Sensorleistungsmodelle modelliert. Auf Grundlage der Leistungsmodelle werden optimale Sensor- und UAV-Positionen ermittelt. Diese sind Ausgangspunkt für die Generierung entsprechend verbesserter UAV-Flugtrajektorien, aus deren Anwendung eine gesteigerte Detektionsleistung resultiert.

Das erste Kapitel soll einen kurzen Einblick in die verschiedenen Anwendungsbereiche von UAVs zum Zweck der luftgestützten Aufklärung geben. Anschließend wird auf die Herausforderungen und Probleme für die automatisierte Planung und Durchführung von Aufklärungsaufträgen eingegangen, woraus das Forschungsziel dieser Arbeit abgeleitet wird. Abschließend folgen eine Themenabgrenzung und ein Überblick über den Aufbau dieser Arbeit.

## 1.1 Motivation

Die Nutzung von UAVs hat in den letzten Jahren eine starke Zunahme erlebt [2]. Ursächlich hierfür sind u. a. eine kontinuierliche Steigerung der Leistungsfähigkeit (z. B. Nutzlast, Flugdauer, Sensorintegration) und eine Verbesserung der Flugführung (u. a. Navigation, Autopilot), weshalb sich UAVs in spezifischen Anwendungsbereichen zu einfach bedienbaren und kostengünstigen Alternativen gegenüber bemannten fliegenden Systemen (z. B. Hubschrauber) entwickelt haben. UAVs werden für unterschiedliche Aufgaben in zivilen, kommerziellen und militärischen Anwendungen eingesetzt. Beispiele sind u. a.

---

<sup>1</sup>Als UAV wird ein unbemanntes Luftfahrzeug bezeichnet, wobei die Definition unterschiedlich und teils domänenspezifisch ist. Im Kontext dieser Arbeit wird folgende Definition von [1] verwendet: Ein UAV ist ein Luftfahrzeug, das ohne menschlichen Piloten oder Personal an Bord betrieben wird. Es ist wiederverwendbar und in der Lage, mit aerodynamischen Mitteln dauerhaft zu fliegen. Das UAV wird ferngesteuert oder fliegt automatisch ein voreingestelltes Flugprofil ab.

## 1 Einleitung

Überwachungs- und Aufklärungsmissionen [3–6], Such- und Rettungsmissionen [7–11], die Umweltüberwachung [12–14], Einsätze in der Land- und Forstwirtschaft [15–19] oder die photogrammetrische Kartierung und Inspizierung aus der Luft [20, 21]. In Abb. 1.1 sind Beispiele aus den genannten Einsatzbereichen abgebildet.

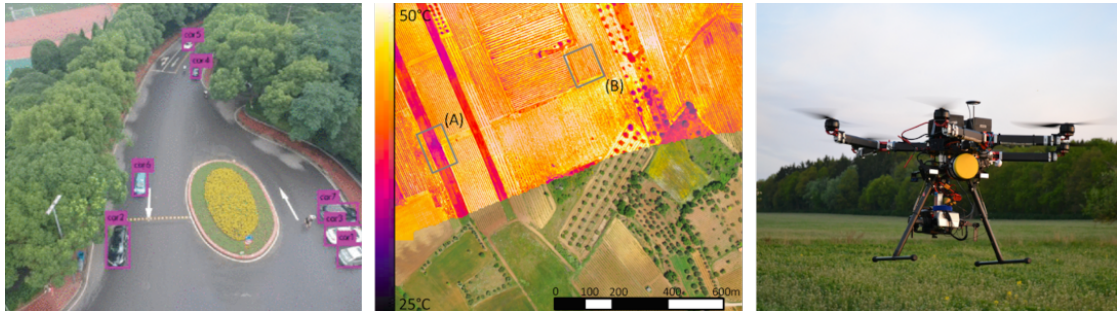


Abbildung 1.1: Beispiele für UAV-Anwendungen: luftgestützte Verkehrsüberwachung [4] (links), Analyse von Rebstößen [13] (Mitte), UAV zur Detektion von Rehkitzen in Feldern [16] (rechts)

UAVs mit bildgebender Sensorik, beispielsweise im visuellen oder infraroten Spektrum, werden zunehmend zur Detektion von Objekten am Boden eingesetzt. Die *Objektdetektion* umfasst hierbei die Lokalisierung und Klassifizierung spezifischer Objekte (z. B. Fahrzeuge) in den Bilddaten.

Die Bedienung großer UAVs, wie z. B. der Kategorie „Medium Altitude Long Endurance“ (MALE), erfolgt in der Regel durch einen UAV-Operator, der das UAV führt, und einen Sensoroperator, der die Parametrisierung, Sensorführung und Sensordatenauswertung übernimmt [22]. Bei Bedarf wird der Sensoroperator durch einen oder mehrere Luftbildauswerter unterstützt. Bei kleineren UAVs übernimmt der UAV-Operator häufig auch die Aufgaben der Sensorführung und Auswertung.

Mit der Entwicklung und Verfügbarkeit immer leistungsfähigerer Computer und Algorithmen zur automatisierten Objektdetektion (z. B. YOLOv3 [23], YOLOv7 [24], RetinaNet [25]) zeichnet sich ein Trend ab, dass die UAV- und/oder Sensorführung zunehmend automatisiert durchgeführt werden [26–29]. Ein Beispiel ist im linken Bild von Abb. 1.1 zu sehen. Durch einen maschinellen Detektionsalgorithmus wurde die Position der Fahrzeuge im Bild lokalisiert, durch Begrenzungsrahmen markiert und als „car“ (Auto) klassifiziert.

Bei der automatisierten Objektdetektion soll eine *hohe Detektionsleistung* angestrebt werden, um die Missionaufgabe, z. B. eine Fahrzeugzählung, korrekt, effizient und zuverlässig durchführen zu können. In dieser Arbeit ist die Detektionsleistung definiert als ein quantitatives Maß für die statistische Korrektheit einer Objektdetektion. Eine hohe

Detektionsleistung stellt hohe Anforderungen an die Fähigkeit und Zuverlässigkeit der Sensordatenverarbeitung und Auswertung, insbesondere wenn automatisch Handlungsentscheidungen auf Grundlage der Detektionsergebnisse getroffen werden sollen. Dies soll nachfolgend an zwei Beispielen erläutert werden:

In der Arbeit von Lygouras et al. [11] wird ein Ansatz vorgestellt, bei dem ein Multikopter-UAV mit bildgebender Sensorik hochautomatisiert Personen im Wasser detektieren kann. Darüber hinaus ist das System in der Lage, Person im Wasser zu identifizieren, die sich in einer Notlage befinden. Wird eine solche Person erkannt, sendet das UAV selbstständig ein Notrufsignal und wirft einen mitgeführten Rettungsring nahe der Person ab.

Im zweiten Beispiel wird von Cukor et al. [16] ein Verfahren untersucht, bei dem mit Hilfe luftgestützter optischer Sensoren Tiere in Feldern detektiert werden können (siehe Abb. 1.1 rechts). Dieser Vorgang wird durchgeführt, bevor ein Feld gemäht werden soll und Rehkitze darin vermutet werden. Durch die luftgestützte Detektion können die Tiere entdeckt und rechtzeitig in Sicherheit gebracht werden. Wird ein Rehkitz nicht korrekt detektiert und damit auch nicht erkannt, kann es durch die Mäharbeiten verletzt oder getötet werden.

Diese beiden Beispiele stehen exemplarisch für zahlreiche weitere Anwendungsfälle für die unbemannte luftgestützte Aufklärung und automatisierte Detektion mit bildgebender Sensorik. Des Weiteren soll durch die beiden Beispiele verdeutlicht werden, dass eine zuverlässige und korrekte Objektdetektion Grundvoraussetzung für eine erfolgreiche Missionsdurchführung ist. Demgegenüber kann eine fehlerhafte oder unzuverlässige Objektdetektion, und eine damit korrelierende niedrige Detektionsleistung, in den beschriebenen Beispielen zu Verletzungen oder Tod von Menschen oder Tieren führen.

Aus diesem Grund soll mit der vorliegenden Arbeit ein Beitrag geleistet werden, um die Detektionsleistung für unbemannte luftgestützte Aufklärungssysteme zu steigern.

## 1.2 Problembeschreibung

Die im vorangegangenen Absatz beschriebenen Anwendungen erfordern eine hohe Leistungsfähigkeit der automatisierten Detektion, um die gewünschte Aufklärungs- bzw. Detektionsaufgabe erfolgreich ausführen zu können. Darüber hinaus gibt es Einflussfaktoren, die sich (negativ) auf die Detektionsleistung auswirken können. Beispielsweise verringert Nebel die Sichtbarkeit, den Kontrast und die Farbsättigung, wohingegen Wolken die Helligkeit und den Schattenwurf verringern und Niederschlag das Sensorrauschen erhöht und den Bildkontrast reduziert [30].

## 1 Einleitung

Zudem können spezifische Missionsanforderungen (z. B. Missionszeit, Taktik), Plattformeneigenschaften und Sensorrestriktionen (z. B. UAV- und Gimbal-Dynamik, Sichtfeld, Sensorauflösung) die Sensordatenerfassung beeinflussen und sich damit auf die Informationsqualität der Sensordaten auswirken [31].

Das Ergebnis der nachgeschalteten Bildverarbeitungsalgorithmen (z. B. Kantendetektion, Mustererkennung, Objektklassifizierung) ist daher stark von der Qualität der Sensordaten abhängig. Letztlich führt eine geringe Sensordatenqualität zu einer eher fehlerhaften Objektdetektion, wodurch die Leistungsfähigkeit der automatisierten Detektion reduziert wird.

In der Arbeit von Zwick, Koch & Stütz [31] wurden verschiedene Umweltbedingungen, Missionsanforderungen und Systemeigenschaften, die eine Auswirkung auf die Detektionsleistung haben, zusammengestellt und kategorisiert. Diese Zusammenstellung ist im Folgenden aufgelistet:

- Atmosphärische Zustände (z. B. Sichtbarkeit, Lichtstreuung, Tageszeit)
- Topographische Zustände (z. B. Gebäude, Vegetation, Wasser, Straßen)
- Photogrammetrische Zustände (z. B. Sensor-Elevationswinkel, Bodenpixelauflösung)
- Sensoreigenschaften, Sensordatenverarbeitung und Auswertung (z. B. Sensortyp, Sichtfeld, Sensorauflösung, Bildverarbeitungsalgorithmen)
- Eigenschaften der Sensorplattform (z. B. Gimbal-Kinematik, UAV-Flugdynamik)
- Missionsauftrag und Missionsrandbedingungen (z. B. räumliche und zeitliche Beschränkungen, Anzahl und Verteilung aufzuklärender Objekte)

In einem weiteren Schritt wurden die Variablen identifiziert, die (bis zu einem gewissen Grad) gezielt beeinflusst werden können. Unter der Annahme, dass die missionsspezifischen Aspekte unveränderlich sind, führt dies zu den folgenden beeinflussbaren Variablen:

- Photogrammetrische Zustände
- Sensorspezifische Eigenschaften
- Eigenschaften der Sensorplattform

Es besteht somit die Möglichkeit, durch eine gezielte Beeinflussung dieser Variablen die Detektionsleistung zu steigern bzw. einer Reduktion (in Grenzen) entgegen zu wirken.

Um die Auswirkung unterschiedlicher Umgebungszustände auf die Detektionsleistung quantifizieren zu können, wurden unter anderem von Hellert & Stütz [32] *Sensorleistungsmodelle* entwickelt, die den Einfluss topographischer, atmosphärischer und photogrammetrischer Umgebungszustände auf die Detektionsleistung für spezifische *Perzeptionsketten* abbilden. Die Perzeptionskette beschreibt hierbei eine spezifische Kombination aus der

bildgebenden Sensorik, der Sensordatenverarbeitung und Aufbereitung, sowie dedizierten Bildverarbeitungsalgorithmen zur Objektdetektion.

Zusammenfassend kann die *Problemstellung* wie folgt beschrieben werden: die zuverlässige Durchführung von Aufklärungs- bzw. Detektionsaufgaben durch unbemannte luftgestützte Aufklärungssysteme setzt eine hohe Detektionsleistung voraus. Allerdings können unterschiedliche Umweltbedingungen, Missionsanforderungen und Systemeigenschaften die Detektionsleistung negativ beeinflussen. Durch die gezielte Beeinflussung spezifischer Zustände und Eigenschaften kann die Detektionsleistung gesteigert bzw. einer Reduktion entgegengewirkt werden.

Auf der Grundlage von Sensorleistungsmodellen wird in dieser Arbeit ein Konzept zur Steigerung der Detektionsleistung unbemannter luftgestützter Aufklärungssysteme vorgestellt. Hierfür werden automatisiert optimale UAV-Flugtrajektorien generiert, die die photogrammetrischen Zustände gezielt derart beeinflussen, dass die resultierende Detektionsleistung gesteigert werden kann.

## 1.3 Forschungsziel

Das *übergeordnete Forschungsziel* ist wie folgt definiert:

**Forschungsziel:** Entwicklung und Validierung eines funktionalen Konzepts zur Steigerung der Detektionsleistung unbemannter luftgestützter Aufklärungssysteme durch die Generierung optimaler, sensormodellbestimmter UAV-Flugtrajektorien. Für die Trajektorienoptimierung sind ausgewählte perzeptive, sensorplattformspezifische und missionsspezifische Anforderungen zu berücksichtigen.

Das zu entwickelnde Konzept wird im weiteren Verlauf der Arbeit unter dem Begriff *sensormodellbestimmte Befliegungsplanung* zusammengefasst. Im Folgenden wird das Konzept kurz umrissen, um dem Leser einen ersten Überblick zu verschaffen.

Der Grundgedanke der sensormodellbestimmten Befliegungsplanung ist es, durch Nutzung von Sensorleistungsmodellen, automatisiert optimale Flugtrajektorien (sogenannte Referenztrajektorien) für das UAV zu generieren. Diese Trajektorien sind Sollwertvorgaben für die UAV-Flugbewegung. Die Generierung der optimalen Referenztrajektorien erfolgt dabei unter Berücksichtigung unterschiedlicher perzeptiver (z. B. atmosphärische, topographische und photogrammetrische Zustände, Sensoreigenschaften), sensorplattformspezifischer (z. B. UAV-Dynamik) und missionsspezifischer (z. B. Position und Ausdehnung des aufzuklärenden Gebiets) Anforderungen und Randbedingungen.

## 1 Einleitung

Die Sensorleistungsmodelle bilden dabei die statistisch zu erwartende Detektionsleistung in Abhängigkeit spezifischer Umgebungsbedingungen ab. Diese Umgebungsbedingungen umfassen u. a. atmosphärische Einflüsse (z. B. Helligkeit und Sichtbedingungen), topographische Zustände innerhalb des vom Sensor erfassten Bereichs (z. B. Straßen, urbanes Gebiet, Flächen mit Bewuchs) und photogrammetrische Zustände (z. B. Bodenpixelauflösung, Elevationswinkel). Insbesondere die photogrammetrischen Zustände werden maßgeblich durch die Sensor- bzw. UAV-Position relativ zum Sensorfootprint am Boden beeinflusst.

In Abb. 1.2 sind beispielhaft photogrammetrische, topographische und atmosphärische Umgebungszustände, sowie die bildgebende Sensorik und die Sensorplattform dargestellt. Mit Hilfe der sensormodellbestimmten Befliegungsplanung werden optimale Referenztrajektorien für das UAV generiert.

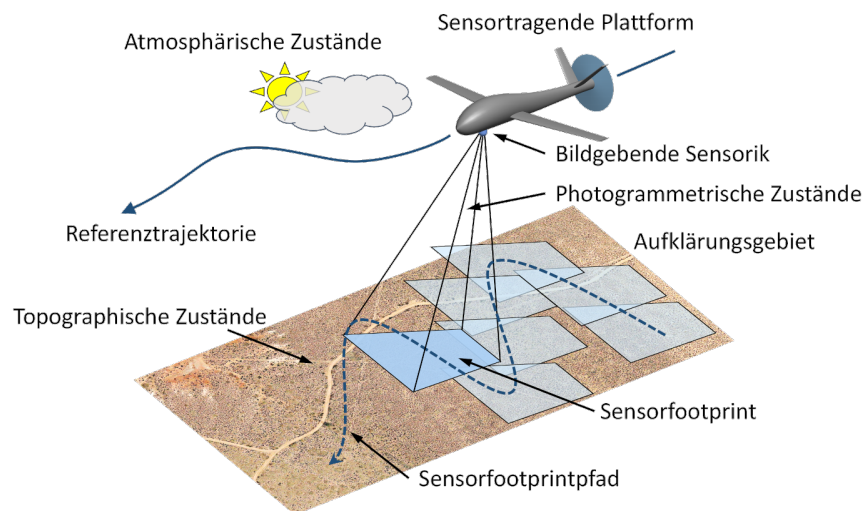


Abbildung 1.2: Darstellung photogrammetrischer, topographischer und atmosphärischer Umgebungszustände, sowie die Sensorik und die Sensorplattform. Aus der sensormodellbestimmten Befliegungsplanung resultiert eine optimale Referenztrajektorie für das UAV (übernommen und angepasst aus [33]).

Das Ziel ist es, unter Verwendung eines Sensorleistungsmodells den Sensor während des Fluges so zu positionieren und auszurichten, dass ein möglichst optimales Gesamtergebnis erreicht wird. Darüber hinaus kann der UAV- / Sensoroperator den quantitativen Wert der so ermittelten Detektionsleistung für die abschließende Bewertung der Detektionsergebnisse verwenden. Ein hoher Wert korreliert mit einer hohen statistischen Korrektheit des Messergebnisses, wohingegen bei einem niedrigen Wert eine höhere Unsicherheit mit dem Messergebnis verbunden ist. Für die Evaluierung der sensormodellbestimmten Be-



## 1.4 Themenabgrenzung und Konkretisierung

fliegungsplanung wird das Konzept in ein Experimentalsystem überführt und am Beispiel der simulierten luftgestützten Detektion am Boden befindlicher Fahrzeuge bewertet.

Um das übergeordnete Forschungsziel zu erreichen, werden nachfolgend zusätzliche *Zwischenziele* definiert. Mit Hilfe der Zwischenziele soll das Forschungsziel weiter konkretisiert, eingegrenzt und dessen Komplexität reduziert werden. In Abschnitt 2.4 werden die Zwischenziele zusätzlich durch spezifische Forschungsfragen ergänzt.

**Zwischenziel 1:** Entwicklung eines funktionalen Konzepts zur sensormodellbestimmten Befliegungsplanung. Die Befliegungsplanung soll ausgewählte perzeptive, sensorplattformspezifische und missionsspezifische Anforderungen und Randbedingungen berücksichtigen.

**Zwischenziel 2:** Entwicklung und Untersuchung von Sensorleistungsmodellen zur Modellierung der Detektionsleistung in Abhängigkeit ausgewählter perzeptiver Einflüsse und Randbedingungen.

**Zwischenziel 3:** Auswahl und Implementierung geeigneter mathematischer Optimierungsmethoden zur Generierung optimaler Flugtrajektorien in Abhängigkeit der Detektionsleistung sowie sensorplattformspezifischer und missionsspezifischer Anforderungen.

**Zwischenziel 4:** Überführung des Konzepts der sensormodellbestimmten Befliegungsplanung in ein Experimentalsystem.

**Zwischenziel 5:** Validierung der sensormodellbestimmten Befliegungsplanung im Experimentalsystem anhand repräsentativer Aufklärungsmissionen unter Berücksichtigung perzeptiver und sensorplattformspezifischer Anforderungen.

## 1.4 Themenabgrenzung und Konkretisierung

Die Entwicklung und Untersuchung des Konzepts zur sensormodellbestimmten Befliegungsplanung berührt unterschiedliche Forschungsfelder, wie z. B. das Feld der Trajektorienoptimierung für UAVs oder mathematische Methoden zur Lösung multikriterieller Optimierungsproblemen. Die umfassende und tiefgehende Betrachtung der einzelnen Forschungsfelder würde den Rahmen dieser Arbeit sprengen. Aus diesem Grund wird im Folgenden eine Themenabgrenzung und Einschränkung auf die relevanten Aspekte des Untersuchungsschwerpunkts durchgeführt.

### 1.4.1 Positionierung von Sensorplattform und Sensorfootprint

Das unbemannte luftgestützte Aufklärungssystem kann in zwei wesentliche Teilsysteme aufgeteilt werden: das UAV als *Sensorplattform* und den *Sensor* zur Bilddatenerfassung. Aus der Variation der Bewegungsfreiheitsgrade dieser beiden Systeme können unterschiedliche Anwendungsfälle abgeleitet werden. Tabelle 1.1 zeigt eine Übersicht ausgewählter Anwendungsfälle mit Bezug auf die Bewegungsfreiheit der Sensorplattform und des Sensorfootprints<sup>2</sup> des bildgebenden Sensors am Boden.

Tabelle 1.1: Anwendungfallmatrix bezüglich der Freiheitsgrade zur Positionierung von Plattform und Sensorfootprint für unbemannte luftgestützte Aufklärungs- und Überwachungssysteme

Positionierung der Sensorplattform	Positionierung des Sensorfootprints	Anwendungsfall	Optimierungsgröße	Ref.
vorgegeben / beschränkt veränderlich	fest / vorgegeben	Hochfliegendes Überwachungssystem	Energiebedarf, Flugdauer	[34, 35]
vorgegeben / beschränkt veränderlich	frei	Tracking mehrerer Objekte	Zustandsschätzung	[36, 37]
frei	fest / vorgegeben	Überwachung, Tracking	Sichtverbindung zum Objekt, Überwachungsdauer	[38, 39]
frei	frei	Geländeaufklärung, Objektdetektion	Detektionsleistung, Aufklärungsdauer	[40, 41]

Die Positionierung der Sensorplattform kann einerseits vorgegeben bzw. nur beschränkt veränderlich sein. Ein Beispiel ist eine solarbetriebene Sensorplattform mit hohen Anforderungen an eine effiziente und energiesparende Flugführung. Andererseits gibt es Anwendungsfälle, bei denen die Positionierung des UAVs innerhalb der flugdynamischen Begrenzungen und regulatorischen Beschränkungen weitestgehend frei erfolgen kann, wie z. B. bei der luftgestützten Kartografie durch ein Multikopter-UAV.

Eine ähnliche Unterscheidung in eine freie oder vorgegebene Positionierung kann für den Sensorfootprint am Boden gemacht werden. Ein Beispiel für die freie Positionierung ist die Aufklärung eines Gebiets zur Detektion von Fahrzeugen. Hierbei kann der Sensorfootprint (in physikalischen Grenzen relativ zur Sensorplattform) räumlich und zeitlich

<sup>2</sup>Als Sensorfootprint wird die Fläche am Boden bezeichnet, die durch den bildgebenden Sensor des UAVs erfasst wird. In dieser Arbeit wird der englische Begriff *Sensorfootprint* oder kurz *Footprint* verwendet, da es keinen adäquaten deutschen Begriff gibt.

frei positioniert werden. Demgegenüber ist die Positionierung des Footprints fest bzw. vorgegeben, wenn beispielsweise ein Objekt ununterbrochen überwacht werden soll.

Die Optimierungsgröße in Tabelle 1.1 steht exemplarisch für den Fall, dass für die Anwendung ein Optimierungsproblem gelöst werden soll. In der letzten Tabellenspalte sind Referenzen als Beispiel für den jeweiligen Anwendungsfall angegeben.

Der Anwendungsfall, der in dieser Arbeit betrachtet wird, ist der letzten Zeile in Tabelle 1.1 mit freier räumlicher und zeitlicher Positionierung der Sensorplattform und des Sensorfootprints zugeordnet. Hierbei wird die räumlich-zeitliche Positionierung des UAVs durch die Referenz-Flugtrajektorie bestimmt, die aus der sensormodellbestimmten Befliegungsplanung (siehe Kapitel 4) resultiert. Um die Komplexität des Optimierungsproblems zu reduzieren wird die räumlich-zeitliche Positionierung des Sensorfootprints intrinsisch durch eine Heuristik festgelegt, die in Abschnitt 4.2 beschrieben wird. Demgegenüber unterscheidet sich der Anwendungsfall „Überwachung, Tracking“ dadurch, dass die Positionierung des Footprints extrinsisch durch das zu überwachende Objekt (z. B. ein Fahrzeug, das sich entlang einer Straße bewegt) vorgegeben wird und dadurch nicht beeinflusst werden kann.

### 1.4.2 Einschränkung perzeptiver, plattformspezifischer und missionsspezifischer Aspekte

Im Folgenden werden Einschränkungen aufgelistet, die hinsichtlich der *perzeptiven Aspekte* auf die sensormodellbestimmte Befliegungsplanung getroffen werden:

- Die Entwicklung und Nutzung von Sensorleistungsmodellen wird auf bildgebende Sensorik im elektrooptischen und infraroten Spektrum (EO/IR) beschränkt. Anderweitige bildgebende Sensorik, wie z. B. Multi- und Hyperspektralkameras oder Synthetic Aperture Radar (SAR), wird nicht untersucht.
- Es wird davon ausgegangen, dass die Sensordatengenerierung, Datenaufbereitung und Auswertung (Objektdetektion) automatisiert an Bord des UAVs abgearbeitet wird. Hierdurch soll ein UAV- / Sensoroperator von der manuellen Sensordatenauswertung entbunden werden.
- Relevante atmosphärische und topographische Umgebungszustände mit Einfluss auf die Detektionsleistung werden als a priori bekannt angenommen. Zudem werden die atmosphärischen Bedingungen als konstant im gesamten Operationsgebiet festgelegt. Diese Annahme wird als valide betrachtet, da die flächige Ausdehnung der Befliegungsplanung in der Regel auf wenige Kilometer begrenzt ist.

## 1 Einleitung

Des Weiteren werden die folgenden Einschränkungen und Vereinfachungen für die *sensortragende Plattform* festgelegt:

- Die bildgebende Sensorik ist in einem Gimbal-System verbaut und kardanisch mit dem UAV als sensortragende Plattform verbunden. Es wird angenommen, dass die Gimbal-Dynamik zur Ausrichtung der Sensorik, wie das Schwenken und Neigen der optischen Achse, deutlich schneller erfolgt, als die Fluglageänderung des Luftfahrzeugs. Aus diesem Grund wird die Gimbal-Dynamik, sowie die mögliche Blockade der kardanischen Aufhängung (Gimbal Lock<sup>3</sup>), nicht weiter berücksichtigt.
- Die Trajektorienoptimierung für die sensormodellbestimmte Befliegungsplanung erfolgt für ein einzelnes UAV. Der gleichzeitige Einsatz mehrerer (kooperierender) UAVs bzw. der Einsatz von UAV-Schwärmen zur Durchführung eines Aufklärungsauftrags wird in dieser Arbeit nicht betrachtet. Hier wird auf die Arbeiten [43, 44] verwiesen.
- Das flugdynamische System des UAVs wird als Starrflügel-Flugzeug modelliert. Es wird festgelegt, dass sich das UAV in konstanter Flughöhe und unter Vorgabe eines konstanten Sollwerts für die Fluggeschwindigkeit bewegt.

Die *missionsspezifischen Aspekte* werden wie folgt festgelegt:

- Das Konzept der sensormodellbestimmten Befliegungsplanung wird durch die Simulation von Aufklärungsaufgaben zur luftgestützten Detektion von Fahrzeugen in einem vordefinierten Gebiet evaluiert. Die in dieser Arbeit zur Anwendung kommenden Sensorleistungsmodelle (siehe Abschnitt 3.4) bilden daher die Detektionsleistung für die luftgestützte Fahrzeugdetektion ab.
- Für die Aufklärungsaufgabe ist vorab nicht bekannt, ob bzw. an welcher Position sich Fahrzeuge im Gebiet befinden. Aus diesem Grund wird festgelegt, dass die aufzuklärende Fläche vollständig und deterministisch mit dem Sensorfootprint überstrichen und erfasst werden muss. Ein stochastischer oder wahrscheinlichkeit-basierter Ansatz zur Bewegungsplanung des Sensorfootprints, wie er beispielsweise in [45–47] vorgestellt wird, wird nicht untersucht.
- Luftraumbegrenzungen im oder um den Operationsraum des UAVs werden vor der Befliegungsplanung festgelegt und sind bekannt. Zudem ist der Operationsraum, in dem sich das UAV bewegt, frei von Hindernissen.

---

<sup>3</sup>Als „*Gimbal Lock*“ wird der Zustand einer kardanischen Aufhängung bezeichnet, wenn zwei Drehachsen parallel zu einander ausgerichtet sind und dadurch dem System ein Freiheitsgrad entzogen wird. Eine tiefergehende Beschreibung zum *Gimbal Lock* findet sich u. a. in [42].

- Die Lage und die Ausdehnung des aufzuklärenden Gebiets am Boden werden vorgegeben und die Topographie innerhalb des Gebiets wird als bekannt vorausgesetzt.

Für die eigentliche *Befliegungsplanung* soll gelten:

- Die sensormodellbestimmte Befliegungsplanung wird entwickelt, zum Zweck der luftgestützten Aufklärung eines vordefinierten Gebiets. Die Aufklärung besteht im Wesentlichen aus der einmaligen sensorischen Erfassung und Analyse des aufzuklärenden Gebiets. Damit grenzt sich die Aufklärung von der Überwachung bzw. dem Tracking ab, bei dem ein Gebiet oder Objekt gezielt mehrfach oder dauerhaft durch den Aufklärungssensor erfasst wird. Diesbezüglich wird der Leser auf die Arbeiten [36, 48, 49] verwiesen.
- Die Befliegungsplanung wird für unterschiedliche Sensorleistungsmodelle durchgeführt und evaluiert. Da in dieser Arbeit der Fokus auf der Optimierung der UAV-Flugtrajektorie liegt, findet keine vergleichende Bewertung der Leistungsmodelle untereinander statt. Diesbezüglich wird auf die Arbeit von Hellert [50] verwiesen.
- Die Befliegungsplanung verfolgt den Ansatz der Offline-Planung. Die damit generierten optimalen Referenztrajektorien fungieren als Sollwertvorgabe für die koordinierte UAV- und Sensorsteuerung an Bord des Luftfahrzeugs. Bei der Offline-Planung wird die gesamte Flugtrajektorie berechnet, bevor der eigentliche Vorgang des Abfliegens der Trajektorie beginnt [51]. Aus diesem Grund ist es erforderlich, dass die Eigenschaften des UAVs, des Sensorsystems und die relevanten Umgebungsbedingungen a priori bekannt sind, wie beispielsweise die atmosphärischen Bedingungen oder Hindernisse im Operationsgebiet.

Im Verlauf der Arbeit werden weitere spezifische Annahmen und Einschränkungen getroffen, auf die an entsprechender Stelle hingewiesen wird.

## 1.5 Aufbau der Arbeit

Nachdem die Problemstellung und das Forschungsziel dieser Arbeit beschrieben wurden, wird in Kapitel 2 der Stand der Forschung zu den relevanten Bereichen dieser Arbeit untersucht, um die Forschungslücken aufzeigen zu können. Hierfür wird in Abschnitt 2.1 auf die Modellierung der Leistungsfähigkeit bildgebender Sensorik eingegangen. Daneben werden in Abschnitt 2.2 Methoden zur Planung und Optimierung von Flugtrajektorien

## 1 Einleitung

betrachtet. Die Verknüpfung von Sensorleistungsmodellen mit der Optimierung von UAV-Flugtrajektorien führt in Abschnitt 2.3 zu Ansätzen zur koordinierten Sensor- und UAV-Führung. Abschließend werden in Abschnitt 2.4 der Forschungsbedarf aufgezeigt und die Forschungsfragen definiert.

In Kapitel 3 wird das Konzept der sensormodellbestimmten Befliegungsplanung vorgestellt und geeignete Methoden untersucht. Zu diesem Zweck werden in einem ersten Schritt die Grundlagen zur Steigerung der Detektionsleistung in Abschnitt 3.1 vorgestellt. Anschließend werden in Abschnitt 3.2 Anforderungen an die Befliegungsplanung festgelegt, die aus den Forschungszielen abgeleitet werden. Darauf aufbauend wird in Abschnitt 3.3 das Konzept der sensormodellbestimmten Befliegungsplanung detailliert erläutert und die dafür notwendige Anpassung des Sensor- und Perzeptionsmanagement beschrieben. Darüber hinaus wird in Abschnitt 3.4 auf die Generierung eines Sensorleistungsmodells eingegangen. Um die Befliegungsplanung durchführen zu können werden die Grundlagen zur Generierung der UAV-Flugtrajektorien in Abschnitt 3.5 sowie unterschiedliche mathematische Methoden für die Trajektorienoptimierung in Abschnitt 3.6 betrachtet.

Die sensormodellbestimmte Befliegungsplanung wird in Kapitel 4 in ein Experimentalsystem umgesetzt. Um dies zu erreichen, wird in Abschnitt 4.1 das Experimentalsystem beschrieben und in Abschnitt 4.2 auf die Implementierung der Sensorpfadplanung eingegangen. Darauf aufbauend folgt die Umsetzung der Befliegungsplanung für Systeme mit zeitkontinuierlicher Trajektorienoptimierung in Abschnitt 4.3 und für die zeit- und wertdiskrete Trajektorienoptimierung in Abschnitt 4.4. In einem letzten Schritt wird die Implementierung der Sensorsteuerung in Abschnitt 4.5 beschrieben.

In Kapitel 5 werden die Grundlagen für die Evaluierung der Befliegungsplanung gelegt. Zu diesem Zweck werden in Abschnitt 5.1 die Untersuchungsgegenstände definiert und die Generierung von Benchmark-Trajektorien in Abschnitt 5.2 vorgestellt.

Die eigentliche Evaluierung der sensormodellbestimmten Befliegungsplanung wird in Kapitel 6 durchgeführt. Hierbei wird die Befliegungsplanung anhand von Beispielmissionen in Abschnitt 6.1 evaluiert. Zudem wird in Abschnitt 6.2 überprüft, ob spezifische Anforderungen und Randbedingungen eingehalten werden. Auf die Aufklärungs- und Berechnungsdauer wird in Abschnitt 6.3 eingegangen.

Am Ende der Arbeit werden in Kapitel 7 die Ergebnisse diskutiert und ein Ausblick gegeben. Dabei werden in Abschnitt 7.1 die erreichten Zwischenziele und das Forschungsziel zusammengefasst, gefolgt von einer Diskussion der Kritikpunkte in Abschnitt 7.2. Das Kapitel endet mit einer Übersicht der technisch-wissenschaftlichen Beiträge in Abschnitt 7.3 und einem Ausblick und Anknüpfungspunkte für die weitere Forschung in Abschnitt 7.4.

## 2 Stand der Forschung

In diesem Kapitel werden relevante Themen mit Bezug auf die sensormodellbestimmte Befliegungsplanung aus dem Stand der Forschung untersucht. Dazu wird zu Beginn auf die Modellierung der Leistungsfähigkeit bildgebender Sensorik eingegangen. Anschließend werden Methoden zur Planung und Optimierung von Flugtrajektorien aus dem Stand der Forschung vorgestellt. Dem folgt eine Übersicht von Ansätzen zur koordinierten Sensor- und UAV-Führung. Abschließend wird der Forschungsbedarf dargestellt und die Forschungsfragen definiert.

### 2.1 Modellierung der Leistungsfähigkeit bildgebender Sensorik

Die Entwicklung eines Sensorleistungsmodells ist eng mit der Modellierung spezifischer Einflussfaktoren auf die Detektionsleistung verbunden. Aus diesem Grund wird im folgenden Abschnitt 2.1.1 zuerst auf die Einflussfaktoren eingegangen, bevor in Abschnitt 2.1.2 unterschiedliche Ansätze von Sensorleistungsmodellen vorgestellt werden.

#### 2.1.1 Einflussfaktoren auf die Detektionsleistung

In diesem Abschnitt werden Arbeiten aus dem Stand der Forschung betrachtet, in denen Einflussfaktoren auf die Detektionsleistung untersucht wurden. Die Auswahl der Arbeiten ist auf die luftgestützte Detektion bodengebundener Objekte (z. B. Fahrzeuge, Personen) mit Hilfe bildgebender Sensorik (z. B. Sensorik im elektrooptischen und infraroten Spektrum) fokussiert. Die Detektionsleistung beschreibt dabei ein quantitatives Maß für die Korrektheit der Lokalisierung und Klassifizierung ausgewählter Objekte in den Bilddaten.

Von Wu et al. [52] wird beschrieben, dass bei Algorithmen zur Objektdetektion, die auf bodengebundenen Bilddaten (z. B. für die Anwendung beim hochautomatisierten Fahren [53, 54]) trainiert wurden, ein erheblicher Leistungsabfall beobachtet wird, wenn diese direkt auf Luftbildaufnahmen angewendet werden. Als Ursache wird der Einfluss spezifischer Umgebungszustände genannt, die Auswirkung auf die Bilddatenerfassung und letztendlich auf die Detektionsleistung haben. In der Arbeit von Wu et al. wurden insbesondere die folgenden Umgebungszustände untersucht:

## 2 Stand der Forschung

- Die UAV-Flughöhe beeinflusst den Maßstab und den Grad der Detaillierung der im Bild befindlichen Objekte.
- Der Sensor-Elevationswinkel hat einen großen Einfluss auf die Erscheinungsform der zu erfassenden Objekte (z. B. Personen), weshalb der Detektionsalgorithmus unterschiedliche visuelle Erscheinungsformen desselben Objekts erkennen muss.
- Die Sichtbedingungen und die Helligkeit ergeben sich aus den Beleuchtungsverhältnissen (Tag und Nacht) und Wetterbedingungen (z. B. sonnig, bewölkt, neblig, regnerisch). Sie beeinflussen damit insbesondere die Sichtbarkeit und das Aussehen von Objekten.

Um die teilweise negativen Effekte dieser Umgebungszustände auf die Detektionsleistung zu kompensieren, wurden diese Umgebungszustände als Metainformationen, zusätzlich zu den eigentlichen Bilddaten, für das Training eines Faster-RCNN [55, 56] Objekterkennungsalgorithmus verwendet. Wu et al. konnten zeigen, dass durch diesen Ansatz die Leistung der bildbasierten Objektdetektion gesteigert werden konnte.

Einen ähnlichen Ansatz verfolgen auch Jin, Jia & Chu in [57]. Auch in ihrer Arbeit wird beschrieben, dass die UAV-Flughöhe, der Sensor-Elevations- und Azimutwinkel sowie die Helligkeit teils negative Auswirkungen auf die automatisierte bildbasierte Objektdetektion haben. Um dem zu begegnen, wurde die Struktur eines einstufigen YOLOv5-Objektdetektors [58] um eine Komponente erweitert, mit der spezifische Merkmale im Sensorbild identifiziert und als Metainformationen zur Verbesserung der Detektionsleistung genutzt werden können. Für diese Metainformationen wurden die folgenden Umgebungszustände herangezogen:

- Die UAV-Flughöhe mit den Merkmalsausprägungen: niedrig, mittel, hoch
- Die Sensorausrichtung mit den Ausprägungen: Front-, Seiten- und Vogelperspektive
- Die Helligkeit mit der Unterteilung in: Tageslicht, Nacht

In einer Evaluierung mit dem modifizierten YOLOv5-Objektdetektor konnten die Autoren zeigen, dass durch die Nutzung der Metainformationen die Detektionsleistung des Detektors im direkten Vergleich mit ähnlichen Objektdetektoren aus den Arbeiten von [52, 59] gesteigert werden konnte.

Auch von Acatay et al. [60] wurde die Detektionsleistung unterschiedlicher Objektdetektoren im Kontext der luftgestützten Detektion bodengebundener Objekte untersucht. Hierfür wurden die Objektdetektoren Faster-RCNN [55, 56], SSD [61] und YOLOv2 [62] adaptiert und trainiert. Anschließend wurde die Detektionsleistung u. a. in Bezug auf die Objektkategorie und die Bodenpixelauflösung (engl.: Ground Sample Distance, GSD) bewertet, wobei die Bilddaten aus dem DOTA-Bilddatensatz [63] genutzt wurden.



## 2.1 Modellierung der Leistungsfähigkeit bildgebender Sensorik

Um die Auswirkungen der GSD auf die Detektionsleistung zu analysieren, wurden alle Bilder aus dem Datensatz auf zwei einheitliche GSD-Werte skaliert und anschließend mit den Detektionsalgorithmen ausgewertet. Die Untersuchung ergab, dass der adaptierte Faster-RCNN Detektor mit den skalierten Bilddaten eine deutlich höhere Detektionsleistung erzielte, als bei Bildern mit unterschiedlicher GSD. Demgegenüber war die Steigerung der Detektionsleistung bei einer einheitlichen GSD mit den adaptierten SSD- und YOLOv2-Detektoren weniger stark ausgeprägt.

### 2.1.2 Ansätze von Sensorleistungsmodellen

Nachdem im vorangegangenen Abschnitt Einflussfaktoren auf die Detektionsleistung betrachtet wurden, werden im Folgenden Ansätze zur Modellierung von Sensorleistungsmodellen für den luftgestützten Einsatz bildgebender Sensorik im Stand der Forschung untersucht. Das Sensorleistungsmodell bildet dabei die zu erwartende Detektionsleistung für eine spezifische Perzeptionskette (siehe Abschnitt 3.4.2) in Abhängigkeit relevanter Einflussfaktoren (vgl. Abschnitt 2.1.1) ab. Die Auswahl der Leistungsmodelle wird dabei auf den Anwendungsbereich der luftgestützten Detektion bodengebundener Objekte (z. B. Fahrzeuge, Personen) durch maschinelle Objekterkennung beschränkt, da dieser Aspekt für die sensormodellbestimmte Befliegungsplanung die höchste Relevanz besitzt. Obwohl in mehreren Arbeiten der Einfluss von Umgebungszuständen auf die Detektionsleistung evaluiert wurde, wurden Ansätze zur expliziten Modellierung der Einflussfaktoren auf die Detektionsleistung in der Forschung bislang nur wenig untersucht. Auf zwei dieser Ansätze wird im Folgenden eingegangen:

Ein Ansatz für ein Sensorleistungsmodell stammt von Lee et al. in [64]. In ihrer Arbeit beschreiben die Autoren, dass Deep Learning (DL) basierte Algorithmen zur Objektdetektion häufig anhand von Daten trainiert und validiert werden, die aus anderen Domänen (z. B. dem hochautomatisierten Fahren) stammen. Werden diese Detektoren für die Auswertung von Luftbildaufnahmen verwendet, kann dies zu einer großen Fehlerquote und damit zu einer reduzierten Detektionsleistung führen. Die Fehler resultieren auf Grund unterschiedlicher Umgebungszustände zwischen den UAV-Bildern und den trainierten Bilddaten. Um das Problem zu lösen stellen Lee et al. einen Ansatz vor, mit dem die Grenzbereiche der DL-basierten Objektdetektoren identifiziert werden können, jenseits derer die Detektoren eine starke Reduktion der Detektionsleistung aufweisen. Zu diesem Zweck haben die Autoren synthetische Bilddaten von Personen in unterschiedlichen Landschaften mit Hilfe einer Grafik-Engine generiert. Durch die synthetischen Daten können gezielt verschiedene Umgebungszustände, wie z. B. die Sensorposition, der Bildwinkel,

## 2 Stand der Forschung

die Beleuchtungsbedingungen oder die Objektposition variiert sowie die Ground Truth ermittelt werden. Anhand dieser Bilddaten wurde anschließend die Abhängigkeit der Detektionsleistung von den verschiedenen Umgebungszuständen untersucht, wobei die folgenden Anwendungsfälle betrachtet wurden:

- Die Abnahme der Detektionsleistung, wenn die Sensorposition senkrecht über die Objektposition verschoben wird (Elevationswinkel =  $90^\circ$ ).
- Die Abhängigkeit der Detektionsleistung von der Variation der Objektorientierung.
- Die Änderung der Detektionsleistung bei unterschiedlichen Beleuchtungsbedingungen.

Die Evaluierung wurde anhand der luftgestützten Personendetektion mit Hilfe der DL-basierten Objektdetektoren Tiny-YOLOv3 [65], YOLOv3 [23] und RetinaNet [25] durchgeführt und die Abhängigkeit der Detektionsleistung von den oben genannten Umgebungszuständen quantifiziert.

Ein ähnlicher Ansatz wird auch in der Arbeit von Hellert [50] untersucht. In der Arbeit wird die Entwicklung von Sensorleistungsmodellen zur luftgestützten Fahrzeugdetektion beschrieben, deren Perzeptionsketten aus unterschiedlichen Objektklassifikatoren bestehen. Auf diese Leistungsmodelle wird in Abschnitt 3.4.3 noch näher eingegangen. Für die Modellierung dieser Leistungsmodelle wurden in einer Simulationsumgebung synthetische Datensätze mit visuellen und Infrarotbildern von Fahrzeugen in verschiedenen Umgebungen erzeugt. Diese Datensätze enthalten auch den zugehörigen Umgebungszustandsvektor, der die Ground Truth der simulierten atmosphärischen (u. a. Bewölkung, Nebel, Niederschlag, Helligkeit), photogrammetrischen (Bodenpixelauflösung und Sensor-Elevationswinkel) und topographischen Zustände (u. a. Landbedeckung und Oberflächenunebenheit) zum Zeitpunkt der Bilddatenerfassung beschreibt. Jeder dieser Zustände wirkt sich auf die Sensordatenerfassung aus und beeinflusst letztendlich die Detektionsleistung. Für die Generierung der Leistungsmodelle hat Hellert vier verschiedene Objektklassifikatoren untersucht: *Classification Cascade* (CC) [66], *Deformable Part Model* (DPM) [67], *Template Matching* (TM) [68] und *Binary Large Object* (BLOB) [69]. Die beiden Klassifikatoren CC und DPM sind Verfahren aus dem Machine Learning, wohingegen TM dem Template Matching und BLOB den wissensbasierten Methoden zugeordnet werden kann. Für jeden dieser Klassifikatoren wurde ein separates Leistungsmodell erstellt, das die Detektionsleistung unter dem Einfluss der Umgebungszustände abbildet.

### 2.1.3 Diskussion

In Tabelle 2.1 sind die untersuchten Einflussfaktoren auf die Detektionsleistung sowie die verwendeten Objektdetektoren bzw. Klassifikatoren aus Abschnitt 2.1.1 zusammengetragen. Es zeigt sich, dass in allen genannten Arbeiten Geometriebeziehungen (u. a. Bodenpixelauflösung, Elevationswinkel, Flughöhe) zwischen dem UAV und dem Objekt am Boden evaluiert wurden. In einem Teil der Arbeiten wurden darüber hinaus auch die vorherrschenden Sichtbedingungen als Einflussfaktoren betrachtet.

Tabelle 2.1: Untersuchte Einflussfaktoren auf die Detektionsleistung

Referenz	Untersuchte Einflussfaktoren auf die Detektionsleistung	Verwendeter Objektdetektor
Wu et al. [52]	UAV-Flughöhe, Elevationswinkel, Sichtbedingungen, Helligkeit	Faster-RCNN
Acatay et al. [60]	Bodenpixelauflösung	Faster-RCNN, SSD, YOLOv2
Jin, Jia & Chu [57]	UAV-Flughöhe, Elevations- und Azimutwinkel, Helligkeit	YOLOv5

In Tabelle 2.2 sind die Ansätze zur Modellierung der Detektionsleistung für Perzeptionsketten mit bildgebender Sensorik aus Abschnitt 2.1.2 zusammengefasst.

Tabelle 2.2: Sensorleistungsmodelle zur Modellierung der Detektionsleistung von Perzeptionsketten

Referenz	Modellierte Umgebungszustände	Merkmale der Perzeptionskette
Lee et al. [64]	Elevationswinkel, Objektorientierung, Beleuchtungsbedingungen	unterschiedliche Objektdetektoren: Tiny-YOLOv3, YOLOv3, RetinaNet
Hellert [50]	topographische, atmosphärische, photogrammetrische Zustände	unterschiedliche Klassifikatoren: TM, DPM, CC, BLOB

Bei den Sensorleistungsmodellen von Hellert [50] werden Objektklassifikatoren zur bilddatenbasierten Detektion genutzt, die auf Template Matching, Machine Learning und wissensbasierten Verfahren beruhen. Die Entwicklung leistungsfähiger Deep Learning basierter Objektdetektoren, wie z. B. YOLOv3 [23], hat dazu geführt, dass die Bedeutung von Verfahren, die nicht auf tiefen neuronalen Netzen basieren, in vielen Bereichen der automatisierten Objekterkennung abgenommen hat [70]. Ungeachtet dessen wird die

Weiterverwendung der Leistungsmodelle von Hellert für die sensormodellbestimmte Befliegungsplanung als sinnvoll erachtet. Ein wesentlicher Grund hierfür ist, dass die Modelle in der Simulation und im Realflug verifiziert und validiert wurden [50]. Zudem bilden diese Leistungsmodelle neben den topographischen und atmosphärischen Umgebungszuständen auch explizit den Einfluss der photogrammetrischen Zustände (Elevationswinkels und Bodenpixelauflösung) auf die Detektionsleistung ab. Dieser Aspekt wird bei der sensormodellbestimmten Befliegungsplanung gezielt ausgenutzt, um letztendlich die Detektionsleistung der Perzeptionskette zu steigern.

### 2.2 Methoden zur Planung und Optimierung von Flugtrajektorien

In diesem Abschnitt werden Methoden zur Planung bzw. Optimierung von UAV-Flugtrajektorien aus dem Stand der Forschung vorgestellt, die für die sensormodellbestimmte Befliegungsplanung von Relevanz sind. Auf Grund der Vielzahl an Methoden wurde bereits durch das *Zwischenziel 3* in Abschnitt 1.3 eine Spezifizierung durchgeführt. Des Weiteren liegt der Fokus der Analyse auf den Methoden, die eine Trajektorienoptimierung unter Berücksichtigung zeitlicher und flugdynamischer Beschränkungen (z. B. Limitierung von Rollwinkel und Fluggeschwindigkeit) ermöglichen. Dies grenzt die Trajektorienoptimierung von der reinen Pfadplanung ab.

Im Folgenden werden Arbeiten aus dem Stand der Forschung bezüglich zeit- und wertdiskreter, zeit- und wertkontinuierlicher sowie explorativer Verfahren zur Planung bzw. Optimierung von UAV-Flugtrajektorien betrachtet. Optimierungsverfahren, die biologischen Prozessen nachempfunden sind, wie beispielsweise *evolutionäre Algorithmen* [71, 72], *künstliche neuronale Netze* [73, 74], *Partikelschwarmoptimierung* [75, 76] oder *Ant Colony Optimization* [77, 78] werden nicht weiter untersucht. Dies wird damit begründet, dass nicht oder nur teilweise bekannt ist, wie die Eingangsgrößen (unabhängige Variablen) die Ausgangsgröße (abhängige Variable) beeinflussen. Es handelt sich daher um Black-Box-Methoden. Das erschwert die Interpretation der Ergebnisse, insbesondere dann, wenn die Kriterien, die der Algorithmus für die Lösungsfindung anwendet, nicht oder nur ansatzweise bekannt sind [79–81].

Darüber hinaus wird in dieser Arbeit festgelegt, dass relevante System- und Umgebungszustände (z. B. atmosphärische Zustände, Flugzustand des UAVs) für die sensorbestimmte Befliegungsplanung (siehe Kapitel 3) entweder a priori bekannt sind, durch Expertenwissen vorgegeben werden oder (implizit) in den Sensorleistungsmodellen abgebildet sind

## 2.2 Methoden zur Planung und Optimierung von Flugtrajektorien

und die Optimierung daher deterministisch erfolgt. Vor diesem Hintergrund werden Optimierungsmethoden, deren Entscheidungen wahrscheinlichkeitsbasiert getroffen werden, nicht betrachtet. Hierzu zählen Methoden aus dem Bereich des stochastischen Schließens, wie beispielsweise *Markow-Entscheidungsprozesse* (engl.: Markov Decision Process, MDP) [82] und *partially observable Markow Decision Process* (POMDP) [83].

Bevor in diesem Abschnitt Methoden zur Trajektorienoptimierung gesichtet werden, sollen vorab die Begriffe *Pfadplanung* und *Trajektorienplanung* erläutert und die Unterschiede herausgestellt werden. Dies wird als notwendig erachtet, da einerseits beide Begriffe in der Literatur häufig gleichbedeutend verwendet werden und andererseits keine allgemeingültige Definition existiert. In [84] wird die *Pfadplanung* beschrieben als eine Methode zur Erzeugung einer kollisionsfreien geometrischen Bahn zwischen einem Anfangs- und einem Endpunkt ohne zeitliche Betrachtung. Demgegenüber werden bei der *Trajektorienplanung* dynamische Randbedingungen berücksichtigt und der (Flug-)Bahn ein Geschwindigkeits- und Beschleunigungsverlauf zugewiesen. Häufig geht der Trajektorienplanung eine Pfadplanung voraus oder die Trajektorienoptimierung wird direkt durchgeführt.

Das Ziel der Trajektorienoptimierung ist es, Sollwerte für die UAV-Flugsteuerung zu generieren, damit eine gewünschte Flugbewegung ausgeführt werden kann. Die Berechnung der Flugtrajektorie ist somit inhärent verbunden mit dem flugdynamischen System des Luftfahrzeugs. Weitere Anforderungen an die Flugtrajektorie werden in Abschnitt 3.5.1 festgelegt.

### 2.2.1 Zeit- und wertdiskrete Methoden

Zeit- und wertdiskrete Optimierungsmethoden können den gitter- bzw. graphenbasierten Verfahren zugeordnet werden, für die in der Literatur zahlreiche Methoden beschrieben werden. Zu den bekanntesten Verfahren zählen u. a. der *Dijkstra-Algorithmus* [85] und der *A\*-Algorithmus* [86], der eine Erweiterung des Dijkstra-Algorithmus darstellt.

Obwohl im Stand der Forschung viele Anwendungsbeispiele des Dijkstra- bzw. A\*-Algorithmus für die UAV-Pfadplanung und Optimierung zu finden sind, konnten keine Anwendungen gefunden werden, bei denen diese Algorithmen zur UAV-Trajektorienoptimierung unter Berücksichtigung flugdynamischer Limitierungen (z. B. Rollwinkelbegrenzung) genutzt werden. Aus diesem Grund werden diese beiden Verfahren nicht weiter betrachtet. Insgesamt konnten nur wenige Methoden identifiziert werden, die für die zeit- und wertdiskrete Trajektorienoptimierung unter Nebenbedingungen geeignet sind. Eines der am häufigsten verwendeten Verfahren wird im Folgenden vorgestellt.

### Dynamische Programmierung

Bei der *dynamischen Programmierung* wird ein diskretes Optimalsteuerungsproblem in Teilprobleme zerlegt und diese nach dem *Prinzip der Optimalität* [87] systematisch gelöst (siehe auch Abschnitt 3.6.2). Das Ergebnis der dynamischen Programmierung ist eine global optimale Lösung des Optimalsteuerungsproblems (siehe Abschnitt 3.6.1). Durch die Diskretisierung des Zustandsraums kann dieser als (multidimensionales) Gitter betrachtet werden, wodurch die dynamische Programmierung den zeit- und wertdiskreten Optimierungsverfahren zugeordnet werden kann.

In [88] wurde ein Verfahren zur UAV-Flugbahnoptimierung auf der Grundlage der dynamischen Programmierung vorgestellt. Hierbei soll die Flugbahn von UAVs mit Mobilfunkverbindung unter Berücksichtigung der Dauer der Verbindungsunterbrechung optimiert werden. Das Ziel ist es, die Länge der Flugtrajektorie zu minimieren, wobei die Verbindungsunterbrechung eine spezifische Zeitdauer nicht überschreiten darf. Zu diesem Zweck wurden dedizierte Stützpunkte auf Basis eines Overlay-Gitters im Fluggebiet des UAVs definiert. Darauf aufbauend wurde mit Hilfe der dynamischen Programmierung eine optimale Flugtrajektorie, unter Berücksichtigung von Randbedingungen, generiert. Bei diesem Ansatz wurde allerdings kein explizites flugdynamisches Modell zur Modellierung der Zustandsübergänge verwendet, lediglich die Fluggeschwindigkeit und die Flughöhe des UAVs wurden berücksichtigt.

In der Arbeit von Quintero et al. [89] wurden optimale Flugtrajektorien für zwei mit Kameras ausgestattete UAVs generiert, die kooperativ ein Fahrzeug am Boden verfolgen. Die UAVs wurden als Starrflügel-UAVs modelliert, die sich mit konstanter Fluggeschwindigkeit in einer festen Flughöhe bewegen, wobei das flugdynamische System mit Hilfe von Dubins-Pfaden (siehe auch Abschnitt 3.5.3) nachgebildet wurde. Das Ziel in dieser Arbeit war es, mit Hilfe der dynamischen Programmierung die optimale koordinierte Steuerstrategie für die beiden UAVs zu berechnen, um damit die fusionierte Kovarianz für den Geolokalisierungsfehler des Fahrzeugs zu minimieren. Um den Berechnungsaufwand bei der dynamischen Programmierung reduzieren zu können, wurde der Systemzustandsraum der UAVs auf drei Zustände ( $x$ -,  $y$ -Position und Gierwinkel) begrenzt.

#### 2.2.2 Zeit- und wertkontinuierliche Methoden

In diesem Abschnitt werden zwei verbreitete Methoden aus dem Bereich der zeit- und wertkontinuierlichen Trajektorienoptimierung vorgestellt, deren Eigenschaften erläutert und Anwendungsbeispiele aus dem Stand der Forschung aufgezeigt.

## 2.2 Methoden zur Planung und Optimierung von Flugtrajektorien

Die Optimierung der nachfolgenden Methoden basiert auf einem kontinuierlichen Zeit- und Wertebereich (Zustandsraum). Da diese Methoden in der Regel numerisch und nur selten analytisch gelöst werden können, findet während des Optimierungsprozesses eine Diskretisierung (Quantisierung des Zustandsraums) statt. Dennoch werden die nachfolgenden Methoden als zeit- und wertkontinuierlich betrachtet, da die Quantisierungsstufen in der Regel so klein gewählt werden, dass der Fehler zwischen dem diskretisierten und dem kontinuierlichen System vernachlässigt werden kann oder einen vorab definierten Grenzwert nicht überschreitet. Auf die Eigenschaften der Diskretisierung kontinuierlicher dynamischer Systeme wird in Abschnitt 3.5.2.2 nochmals eingegangen.

### Modellprädiktive Regelung

Bei der *modellprädiktiven Regelung* (engl.: Model Predictive Control, MPC) wird ein Modell des dynamischen Systems genutzt, um die zukünftige Entwicklung des Systemzustands unter Berücksichtigung von Systemzustands- und/oder Steuerbeschränkungen zu prognostizieren. Das Ziel ist es, auf Grundlage einer Kostenfunktion eine optimale Steuerstrategie zu berechnen, mit der das dynamische System einer Sollwertvorgabe folgt, unter Einhaltung unterschiedlicher Anforderungen und Randbedingungen [90]. Bei der modellprädiktiven Regelung wird ein Optimalsteuerungsproblem (siehe Abschnitt 3.6.1) zyklisch für einen endlichen Zeithorizont gelöst. Daraus resultiert eine optimale Steuerstrategie, deren erste Steuereingabe auf den folgenden Optimierungszeitschritt angewendet wird. Nach der Aktualisierung des (gemessenen) Systemzustands wird die Optimierung von neuem durchgeführt. Eine ausführlichere Betrachtung der (nichtlinearen) modellprädiktiven Regelung folgt in Abschnitt 3.6.3.

Für die Berechnung der optimalen Steuerstrategie des Optimalsteuerungsproblems existieren unterschiedliche effiziente Verfahren und Algorithmen. Zu diesen zählt beispielsweise die sequentielle quadratische Programmierung (SQP) [91]. Da die Systemdynamik des Modells häufig durch Differentialgleichungen modelliert wird, kann MPC den zeit- und wertkontinuierlichen Optimierungsmethoden zugeordnet werden. Darüber hinaus erfolgt die Optimierung bei MPC in einer geschlossenen Regelschleife, weshalb diese Methode zu den optimierungsbasierten Reglern zählt. Das Ergebnis der modellprädiktiven Regelung ist eine optimale Steuerstrategie bezüglich einer spezifischen Startkonfiguration.

In [92] wurde ein Ansatz zur Kollisionsvermeidung bei der Trajektorienplanung mit MPC vorgestellt. Zu diesem Zweck verwendeten die Autoren einen modifizierten MPC-Algorithmus, bei dem ein Unsicherheitsfaktor als Randbedingung explizit mit berück-

sichtigt wurde. Hierfür wurde die Struktur des Algorithmus weiterentwickelt, um die Wahrscheinlichkeitsbedingungen in die Zielfunktion des MPC einbinden zu können.

In [93] beschreiben Stastny, Dash & Siegwart eine nichtlineare modellprädiktive Regelung für ein Starrflügel-UAV, um einer vorgegebenen Flugtrajektorie unter dem Einfluss von Wind zu folgen. Hierfür wurde eine Modellidentifikation des Rollverhaltens durchgeführt und als Systemmodell in den MPC-Algorithmus integriert. Abschließend wurde das Ergebnis der modellprädiktiven Regelung unter Variation des Zeithorizonts und unter dem simulierten Einfluss von Wind überprüft.

Von Lee et al. wurde in [94] ein Ansatz zur Optimierung des UAV-Flugbahnwinkels für ein Kipprotor-UAV mit Hilfe der modellprädiktiven Regelung untersucht. Dieser Ansatz hatte zum Ziel, den Flugbahnwinkel für den Tiefflug und unter dem Einfluss von Wind zu optimieren. Eine Herausforderung bei der Optimierung bestand darin, dass kleine Kipprotor-Drohnen empfindlich auf Wind und Turbulenzen reagieren, wie sie insbesondere in der Nähe von Bergen auftreten. Aus diesem Grund wurde der MPC-Algorithmus um eine nichtlineare Backstepping-Methode erweitert, um den optimalen Flugbahnwinkel unter Berücksichtigung von Randbedingungen (u. a. Neigung und Flughöhe) zu ermitteln.

### **Künstliches Potentialfeld**

Ein weiterer Vertreter aus dem Feld der zeit- und wertkontinuierlichen Trajektorienoptimierung ist das *künstliche Potentialfeld* (engl.: Artificial Potential Field, APF). Potentialfeldmethoden beruhen auf der Idee, dem Konfigurationsraum eine Potentialfunktion zuzuordnen. Der Konfigurationsraum beschreibt die Summe aller zulässigen und hindernisfreien Positionen, die das UAV einnehmen kann. Ein Objekt (z. B. ein UAV), das durch dieses Potentialfeld bewegt werden soll, wird hierzu als Teilchen simuliert, das auf die Kräfte des Potentialfelds reagiert. Dem Zielpunkt wird das niedrigste Potential zugewiesen, wodurch das Objekt angezogen wird, während Hindernisse das Objekt abstoßen [95]. Ein grundlegendes Problem bei der Pfad- oder Trajektorienoptimierung mit APFs stellen lokale Minima dar. Dabei kann der Pfad / die Trajektorie durch ein lokales Minima blockiert werden, bevor der Zielpunkt oder der Zielzustand erreicht wird [96].

In [97] stellten Batinovic et al. ein Verfahren zur Optimierung von UAV-Flugtrajektorien auf Basis eines modifizierten APF vor. Das Ziel war es, dass das UAV einer vorgeplanten Trajektorie folgt und dabei Hindernissen ausweicht. Die Hindernisse wurden hierbei als Potentialfelder modelliert. Die Modifikation des APF-Algorithmus wurde so umgesetzt, dass der translatorischen Kraftwirkung, die aus den abstoßenden Potentialfeldern resul-



## 2.2 Methoden zur Planung und Optimierung von Flugtrajektorien

tiert, eine rotatorische Komponente überlagert wurde. Dadurch sollte die Blockierung der Flugtrajektorie, auf Grund von lokalen Minima, verhindert werden.

In Luo et al. [98] wird ein weiterer Ansatz beschrieben, bei dem das künstliche Potentialfeld durch zusätzliche Steuerkräfte modifiziert wurde, um das Problem der lokalen Minima zu vermeiden. Diese Steuerkräfte wurden auf der Basis einer modellprädiktiven Regelung berechnet und dem APF überlagert. Das ursprüngliche APF-Optimierungsproblem wurde in ein multikriterielles Optimierungsproblem umgewandelt, wobei eine zusätzliche Optimierungsvariable (hier: die Überlebensfähigkeit) eingeführt wurde. Durch diese zusätzliche Optimierungsvariable konnte verhindert werden, dass der Trajektorienverlauf durch lokale Minima blockiert wurde.

Aus den oben genannten Beispielen zur Trajektorienoptimierung mit APF können zwei Nachteile identifiziert werden. Ein Nachteil ist, dass geeignete Maßnahmen getroffen werden müssen, um die Gefahr einer Blockierung der Trajektorie, auf Grund lokaler Minima, zu verhindern. Obwohl durch die Verwendung konzentrischer Potentialfelder glatte (Flug-)Trajektorien (siehe hierzu Erläuterung in Abschnitt 3.5.1) realisiert werden können, muss eingeschränkt werden, dass flugdynamische Beschränkungen, insbesondere die Limitierung des Rollwinkels und der damit verbundenen Krümmungsbeschränkung der Flugtrajektorie, nur näherungsweise umgesetzt werden können.

Sowohl bei der modellprädiktiven Regelung, als auch mit künstlichen Potentialfeldern, können Flugtrajektorien von einem spezifischen Startzustand zu einem festgelegten Zielzustand geplant werden. Ein Vorteil besteht darin, dass der Verlauf der Trajektorie zwischen dem Start- und Zielzustand in Grenzen gezielt beeinflusst werden kann.

### 2.2.3 Explorative Methoden

Abschließend wird in diesem Abschnitt auf explorative Methoden zur Planung bzw. Optimierung von UAV-Flugtrajektorien aus dem Stand der Forschung eingegangen und deren Eigenschaften untersucht. Bei explorativen Methoden wird der Zustandsraum iterativ stochastisch abgetastet und die daraus resultierende Systemkonfiguration auf Validität (z. B. Kollisionsfreiheit) überprüft.

#### Rapidly-exploring Random Tree

Beim Verfahren *Rapidly-Exploring Random Tree* (RRT) [99, 100] wird ein (Flug-)Pfad für das UAV durch den Konfigurationsraum ermittelt, indem durch iteratives und stochastisches Abtasten des Konfigurationsraums Stichproben genommen werden, aus denen eine Baumstruktur generiert wird. Nach jedem Iterationsschritt wird in der Baumstruktur

## 2 Stand der Forschung

geprüft, ob eine räumliche Verbindung zwischen einem definierten Start- und Endpunkt erreicht wurde. Mit RRT kann asymptotisch die global kürzeste (optimale) Verbindung zwischen einem Start- und Endpunkt im Konfigurationsraum identifiziert werden, falls eine Lösung existiert.

Auf Grund der Beschränkung von RRT auf dynamische Systeme mit holonomer Dynamik, wurde *kinodynamic RRT* [100, 101] entwickelt. Bei der *kinodynamischen Planung* [102] geht es darum, ein dynamisches System (z. B. ein UAV) von einem Start- oder Ausgangszustand zu einem Zielzustand zu überführen und dabei kinematische und dynamische (kurz: kinodynamische) Beschränkungen zu berücksichtigen. Beispiele finden sich u. a. in [103, 104].

In [105] wird ein Verfahren vorgestellt, bei dem ein dynamischer RRT-Algorithmus für die Bahnplanung von Starrflügel-UAVs in einer dynamischen dreidimensionalen Umgebung eingesetzt wird. Hierfür wurde die Baumstruktur, die mit Hilfe des RRT-Algorithmus ermittelt wird, durch die Einführung von Zwangsgleichungen erweitert, um die flugdynamischen Beschränkungen des UAVs zu erfüllen. Unter Verwendung von B-Splines wurde der Flugpfad, der aus der Baumstruktur resultiert, interpoliert und geglättet. Als Ergebnis konnte der Algorithmus einen glatten Pfad generieren, der die dynamischen Beschränkungen des UAVs in einer dreidimensionalen Umgebung mit statischen und/oder dynamischen Hindernissen erfüllt. Aus der Bahnplanung kann durch die Vorgabe eines Geschwindigkeits- und Beschleunigungsverlaufs eine Flugtrajektorie generiert werden.

In der Arbeit von Penicka & Scaramuzza [106] wird auf Grundlage von RRT ein Verfahren zur Planung einer zeitminimalen Trajektorie für einen Quadrotor entlang vorgegebener Wegpunkte durch eine Hindernisskulisse vorgestellt. Für das flugdynamische System wurde die volle Quadrotordynamik berücksichtigt. Hierfür wurde zunächst eine initiale Trajektorienplanung für ein kinematisch beschränktes Punkt-Masse-Modell unternommen. Anschließend wurde das Verfahren *kinodynamic RRT*, unter Verwendung eines vollständigen Quadrotor-Modells auf die Lösung des Punkt-Masse-Modells angewendet, woraus eine fliegbare UAV-Trajektorie resultierte.

### **Probabilistic Roadmap**

*Probabilistic Roadmap* (PRM) [107] ist ein weiteres stichprobenbasiertes Verfahren zur Bestimmung eines Pfads zwischen einem UAV-Startzustand und einem Zielzustand, unter Berücksichtigung von Hindernissen. Das Funktionsprinzip von PRM besteht darin, Stichproben von UAV-Konfigurationen (Position und Gierwinkel) aus dem Konfiguri-

## 2.2 Methoden zur Planung und Optimierung von Flugtrajektorien

onsraum zu nehmen und zu überprüfen, ob diese hindernissfrei sind. Anschließend wird mit Hilfe eines Planungsalgorithmus versucht, naheliegende Konfigurationen miteinander zu verbinden. Unter Verwendung eines Graphensuchalgorithmus wird geprüft, ob es einen Pfad zwischen der Start- und Zielkonfigurationen im Graphen gibt.

In Abhängigkeit der verwendeten Heuristik zur Generierung der Stichproben (z. B. durch ein Voronoi-Diagramm) und des verwendeten Abbruchkriteriums kann mit PRM die global kürzeste (optimale) Verbindung zwischen dem Start- und dem Endpunkt im Konfigurationsraum asymptotisch identifiziert werden. In der Literatur finden sich nur wenige Arbeiten, die PRM für die Planung bzw. Optimierung von UAV-Flugtrajektorien verwenden. Zwei dieser Arbeiten werden im Folgenden beschrieben:

In [108] stellen die Autoren ein Verfahren für die Planung einer Flugtrajektorie für ein UAV vor, das ein urbanes Gebiet aufklären soll. Der Fokus liegt auf der Maximierung der untersuchten Fläche unter Berücksichtigung flugdynamischer Beschränkungen des UAVs. Zudem ist die Umgebung in der sich das UAV bewegt in Teilen veränderlich. Für die Planung der Flugtrajektorien wurde zunächst das PRM-Verfahren genutzt, um einen kinematisch zulässigen Pfad durch mehrere Wegpunkte zu planen. Anschließend wurde die Dubins-Pfadplanung (siehe Abschnitt 3.5.3) angewendet, um diesen Pfad zu glätten und eine fliegbare Trajektorie zu erzeugen.

In [109] wurde ein modifizierter PRM-Algorithmus entwickelt, mit dem Flugtrajektorien generiert werden können, die den Energieverbrauch des UAVs, auf Grund von Richtungsänderungen, reduziert. Hierfür wurde der Abtastbereich im Konfigurationsraum optimiert, um die räumliche Auflösung zu verbessern und gleichzeitig die Berechnungsdauer zu reduzieren. Die durch den PRM-Algorithmus berechneten Pfade wurden mit Hilfe von B-Splines geglättet, um daraus abschließend die Flugtrajektorie zu generieren.

Ein wesentlicher Nachteil von PRM besteht darin, dass das flugdynamische System des UAVs nur näherungsweise durch Heuristiken bei der Pfad- bzw. Trajektorienplanung berücksichtigt werden kann. Aus diesem Grund wird in den oben genannten Beispielen zusätzlich die Dubins-Pfadplanung oder ein Spline-Glättung verwendet, um den Pfad zu glätten, aus dem abschließend die eigentliche Flugtrajektorie generiert wird.

Sowohl bei RRT, als auch bei PRM wird ein Pfad / eine Flugtrajektorie von einem spezifischen Startzustand zu einem dedizierten Endzustand geplant. Für den Verlauf des Pfads bzw. der Trajektorie zwischen den Zuständen wird nur geprüft, ob dieser frei von Hindernissen ist. Eine Beeinflussung des Pfadverlaufs (z. B. durch Potentialfelder) ist nicht vorgesehen.

### 2.2.4 Diskussion

In Tabelle 2.3 sind die verschiedenen Methoden zur Planung bzw. Optimierung von Flugtrajektorien aus den vorangegangenen Abschnitten in einer Übersicht zusammengefasst. Über die Ausprägung *Optimalität* können die Optimierungsmethoden u. a. den folgenden Kategorien zugeordnet werden:

- Methoden zur Generierung *lokal optimaler* Flugtrajektorien auf Grundlage einer vorgegebenen Startkonfiguration.
- Methoden zur Generierung *global optimaler* Flugtrajektorien.
- Methoden zur Generierung *asymptotisch global optimaler* Flugtrajektorien.

Ausgehend von *Anforderung 3* aus Abschnitt 1.3 zur Implementierung geeigneter Optimierungsmethoden zur Generierung *optimaler* Flugtrajektorien werden in dieser Arbeit zwei Optimierungsmethoden für die sensormodellbestimmte Befliegungsplanung näher untersucht. Zu diesem Zweck wird jeweils ein Verfahren aus der Kategorie der *lokal optimalen* sowie der *global optimalen* Methoden ausgewählt. Die beiden ausgewählten Methoden sind in Tabelle 2.3 grün hinterlegt.

Für die Generierung global optimaler Flugtrajektorien wird die Methode *dynamische Programmierung* (DP) als geeignet betrachtet. Bei diesem Verfahren wird der gesamte diskrete Zustandsraum für die Generierung einer Flugtrajektorie berücksichtigt. Aus der Optimierung resultiert eine, bezüglich des diskreten Zustandsraums, global optimale UAV-Flugtrajektorie sowie ein optimaler UAV-Startzustand. Für die Berechnung der Zustandsübergänge wird in Abschnitt 3.5.3 ein Ansatz vorgestellt, um ausgewählte flugdynamische Beschränkungen (z. B. eine Krümmungsbeschränkung der Flugtrajektorie) bei der Trajektorienoptimierung berücksichtigen zu können.

Aus den Optimierungsmethoden zur Generierung lokal optimaler Flugtrajektorien wird die *modellprädiktive Regelung* (MPC) als eine geeignete Methode für die sensormodellbestimmte Befliegungsplanung identifiziert. Ein wesentlicher Vorteil von MPC ist, dass mit dieser Methode multikriterielle Optimierungsprobleme mit vorgegebener Systemdynamik gelöst werden können. Dies ermöglicht u. a. die Generierung optimaler UAV-Flugtrajektorien durch die explizite Berücksichtigung der, im Allgemeinen nichtlinearen, UAV-Flugdynamik. Mit Hilfe von MPC können lokal optimale Flugtrajektorien für eine vorgegebene Startkonfiguration berechnet werden.

Mit der Auswahl und Untersuchung der beiden Optimierungsmethoden zur Generierung zeit- und wertdiskreter global optimaler Trajektorien bzw. zeit- und wertkontinuierlicher lokal optimaler Trajektorien wird ein breites Anwendungsgebiet abgedeckt. Zudem werden unterschiedliche, zum Teil komplementäre, Eigenschaften und Anforderungen

Tabelle 2.3: Vergleich ausgewählter Methoden zur Trajektorienoptimierung. Grün sind diejenigen Verfahren markiert, die für die Untersuchung der Befliegungsplanung in Frage kommen.

Methode	Zustandsraum	Optimalität der Trajektorie	Einbeziehung dynamischer Beschränkungen	Bemerkung	Referenzen
Dynamische Programmierung	diskret	global optimal bzgl. der Diskretisierung	ja, durch diskretisierte Bewegungsgleichung	Für jeden Startpunkt wird ein optimaler Endpunkt ermittelt	[88, 89]
Modellprädiktive Regelung	kontinuierlich	lokal optimal	ja, durch Systemmodell	Trajektorienoptimierung erfolgt in einer Regelschleife	[92–94]
Künstliches Potentialfeld	kontinuierlich	lokal optimal	durch Heuristiken	Anfällig für lokale Minima	[97, 98]
Rapidly-exploring Random Tree	diskret	asymptotisch global optimal	ja, durch <i>kinodynamic RRT</i>	konvergiert nur langsam	[100, 101, 105, 106]
Probabilistic Roadmap	diskret	asymptotisch global optimal	durch Heuristiken	konvergiert nur langsam	[108, 109]

betrachtet. Die Entscheidung, ob im Rahmen der Befliegungsplanung eine lokal oder global optimale Trajektorie generiert werden soll, ist u. a. missionsabhängig und obliegt dem UAV-Operator. Darüber hinaus weisen beide Optimierungsverfahren spezifische Vorteile als auch Einschränkungen auf. Ein universeller Vergleich zur Bestimmung des „besten“ Verfahrens ist aus diesen Gründen nicht zielführend und wird daher auch nicht durchgeführt.

Im Folgenden wird kurz auf die Nachteile bzw. Einschränkungen der anderen Methoden in Tabelle 2.3 eingegangen. Daraus leitet sich die Begründung ab, warum diese Methoden nicht weiter für die sensormodellbestimmte Befliegungsplanung untersucht werden: Ein wesentlicher Nachteil der Verfahren APF und PRM besteht darin, dass flugdynamische Beschränkungen des UAVs (z. B. krümmungsbeschränkte Flugtrajektorie) nur angenähert durch Heuristiken umgesetzt und im Optimierungsprozess berücksichtigt werden können. Darüber hinaus ist die Trajektorienoptimierung durch künstliche Potentialfelder anfällig für lokale Minima. Ein weiterer Nachteil der Trajektorienoptimierung auf Basis der stichprobenbasierten Verfahren PRM und RRT ist, dass die optimale (kürzeste) Flugtrajektorie nur asymptotisch erreicht werden kann, wodurch das Verfahren prinzipbedingt und abhängig vom gewählten Abbruchkriterium gegebenenfalls nur langsam konvergiert.

### 2.3 Ansätze zur koordinierten Sensor- und UAV-Führung

Nach der Betrachtung des aktuellen Stands der Forschung bezüglich der Leistungsmodelle bildgebender Sensorik (Abschnitt 2.1) und den Methoden zur Trajektorienoptimierung (Abschnitt 2.2) werden in diesem Abschnitt Ansätze zur koordinierten Sensor- und UAV-Führung untersucht, die beide Themenbereiche miteinander verbindet. Um einen Aufklärungsauftrag automatisiert mit einem UAV ausführen zu können, bedarf es spezifischer Fähigkeiten zur Planung, Koordination und Durchführung des Sensoreinsatzes an Bord des UAVs. Zu diesem Zweck werden im Folgenden zuerst Konzepte zum Sensormanagement vorgestellt und anschließend Ansätze zur sensorbestimmten UAV-Flugführung<sup>4</sup> betrachtet. Die Auswahl der Konzepte berücksichtigt die Themenabgrenzung und Konkretisierung aus Abschnitt 1.4. Aus diesem Grund werden Ansätze zum Sensormanagement und/oder zur UAV-Flugführung, die primär für Multi-UAV-Anwendungen oder für reine Tracking-Anwendungen entwickelt wurden, nicht weiter betrachtet.

---

<sup>4</sup>Die Bezeichnung *sensorbestimmte UAV-Flugführung* wurde an dieser Stelle gewählt, um eine begriffliche Trennung zum Konzept der *sensormodellbestimmten Befliegungsplanung* zu erreichen, die im Rahmen dieser Arbeit entwickelt wird.

### 2.3.1 Ansätze zum Sensormanagement

Das *Sensormanagement* zielt auf die Verwaltung, Koordinierung und Integration der Sensornutzung ab, um spezifische und dynamische Missionsziele zu erreichen [110]. Das Sensormanagement kann auch als ein Konzept zur optimierten Nutzung begrenzter Sensor- und Systemressourcen beschrieben werden, mit dem Ziel, komplexe Aufgaben der sensorischen Datenerfassung und Auswertung (wie z. B. die luftgestützte Aufklärung bodengebundener Objekte) hochautomatisiert durchführen zu können [111].

In [110] werden von den Autoren Ng & Ng unterschiedliche Gründe angegeben, die die Zweckmäßigkeit eines Sensormanagements unterstreichen:

- Verringerung des Arbeitsaufwands für den Sensorbediener durch Automatisierung der Sensorzuweisung, Ausrichtung und Überwachung.
- Priorisierung, Planung und Zuweisung von Aufklärungsanfragen.
- Effiziente Planung und Nutzung der Sensorik zur Durchführung eines Aufklärungsauftrags.
- Effiziente Nutzung der begrenzten Systemressourcen.

#### Generisches Sensormanagement System

Von Ng & Ng wird in [110] ein *generisches Sensormanagement System* (GSMS) vorgestellt. In Abb. 2.1 ist die Systemarchitektur des GSMS dargestellt, wobei der Datenfusionsprozess ausgelagert und als eigenständiger Prozess betrachtet wird.

Das GSMS umfasst die folgenden vier Hauptmodule:

- Die *Eingabeverarbeitung* (Input Processing Unit) empfängt Daten vom Bediener sowie aus der Sensordatenfusion und führt eine erste Verarbeitung der Daten durch, wie z. B. die Überwachung des Sensorstatus.
- Durch die *Entscheidungsfindung* (Decision-making Unit) wird festgelegt, welche Aufgaben zuerst und in welcher Reihenfolge abgearbeitet werden. Hierfür wird eine Priorisierung der Ziele und Aufgaben sowie die Identifizierung zeitkritischer Aufgaben durchgeführt.
- Dem schließt sich die *Optimierung und Steuerung* (Optimising Unit / Control Unit) der Sensorik an. Hierzu zählen u. a. die Auswahl und Ansteuerung (Ausrichtung) der Sensoren, sowie die sensorische Datenerfassung.
- Die *Planung und Aufgabenzuweisung* (Scheduling Unit / Task assignment Unit) soll sicherstellen, dass die Aufklärungsaufträge entsprechend der Priorisierung und Dringlichkeit korrekt geplant und den jeweiligen Sensoren zugewiesen werden.

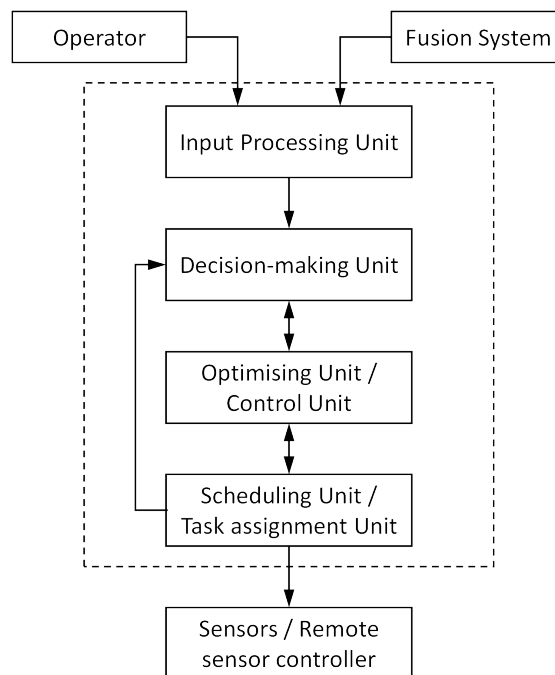


Abbildung 2.1: Systemarchitektur des generischen Sensormanagement Systems (übernommen und angepasst aus [110]).

### Sensormanagement

Ein weiterer Ansatz für ein *Sensormanagement* (SM) wurde von Hero & Cochran in [112] vorgestellt. Die Autoren beschreiben das Sensormanagement als die bestimmende Funktion für die dynamische Auswahl eines spezifischen Sensors aus einer Reihe verfügbarer Sensoren. Die Sensorauswahl wird für den Messvorgang (z. B. die sensorische Datenerfassung) derart angepasst, dass eine bestimmte Leistungsmetrik optimiert wird.

In Abb. 2.2 sind die grundlegenden Elemente und die Funktionsweise des Sensormanagement Systems mit geschlossenem Regelkreis dargestellt. Die Funktionsweise des Sensormanagements ist wie folgt [112]: Sobald ein Sensor (S1, S2, S3) durch den *Sensor Selector* ausgewählt und eine Messung durchgeführt wurde, werden aus den Rohdaten relevante Informationen extrahiert. Dies erfordert im Allgemeinen die Sensordatenfusion (Blöcke *Sensor Fusion* und *Signal Processing*), um die Sensorinformation mit weiteren Informationen aus früheren Messungen oder aus anderen Quellen, die nicht zum Sensorsystem gehören, zu fusionieren. Im Kontext des Sensormanagements müssen die fusionierten Daten einen Informationsstand liefern, auf dessen Grundlage die Vorzüge jeder möglichen Sensorauswahl für den nächsten Zeitschritt quantifiziert werden können



(Block *Predict System Performance*). Eine solche Quantifizierung erfolgt im Modul *Optimization* und kann durch unterschiedliche statistische oder heuristische Ansätzen erfolgen. Hierzu zählen u. a. *Markov Decision Processes* (MDP) [82], *partially observable Markov Decision Process* (POMDP) [83] und *multiarmed bandit* (MAP) [113]. Auf Grundlage der Optimierung wählt der *Sensor Selector* den entsprechenden Sensor aus, wodurch der Regelkreis geschlossen wird.

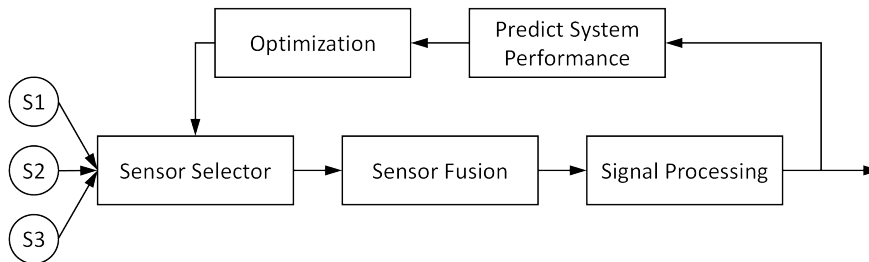


Abbildung 2.2: Systemarchitektur des Sensormanagement Systems mit geschlossenem Regelkreis von Hero & Cochran (in Anlehnung an [112]).

### Sensor- und Perzeptionsmanagement System

Während in den vorangegangenen Konzepten primär die Sensorauswahl und Steuerung betrachtet wurde, wird von Russ & Stütz in [114, 115] das Paradigma des *Sensor- und Perzeptionsmanagement Systems* (SPMS) für den Einsatz an Bord luftgestützter Plattformen vorgestellt. Dieser Ansatz beschreibt sowohl das Management der Sensorik als auch die automatisierte Sensordatenerfassung, Aufbereitung und Auswertung. Das SPMS ist dafür konzipiert, perzeptive Aufgaben (z.B. die Suche nach Fahrzeugen in einem bestimmten Gebiet) abzuarbeiten und damit komplexe Überwachungs- und Aufklärungsmissionen hochgradig automatisiert durchführen zu können. Zu diesem Zweck ist das SPMS in der Lage, durch die an Bord des UAVs verfügbaren Hardware- (z. B. bildgebende Sensoren) und Softwareressourcen (z. B. Filter, Bildverarbeitungsalgorithmen) die perzeptiven Fähigkeiten und Limitierungen zu bestimmen, die für die Durchführung des Aufklärungs- oder Überwachungsauftrags erforderlich sind [116]. Darüber hinaus verfügt das SPMS über ein Active Perception [117] Softwaremodul, das dynamisch mit der Signalverarbeitungskette und deren Algorithmen interagiert, um die Perzeptionsleistung zu verbessern.

In Abb. 2.3 ist die Architektur eines unbemannten Flugsystems (engl.: Unmanned Aerial System, UAS) dargestellt, das aus einer Bodenkontrollstation und dem unbemannten Luftfahrzeug (UAV) besteht, die über einen Datenlink verbunden sind. Das System UAV

## 2 Stand der Forschung

ist wiederum in die Komponenten SPMS, Missionsmanagement, Flugführung, Navigation und Steuerung (GNC), Flug- und Missionssensorik sowie das Luftfahrzeug unterteilt.

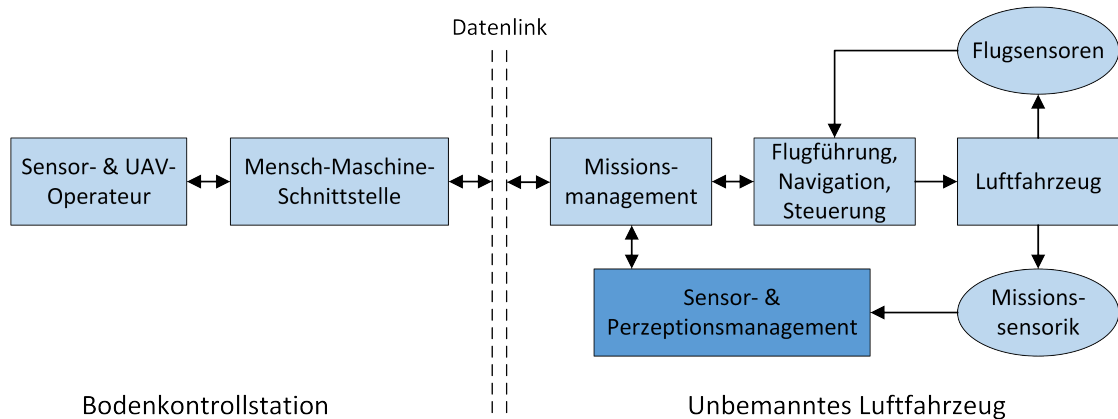


Abbildung 2.3: Systemschaubild zur UAV-Flugführung mit integriertem Sensor- und Perzeptionsmanagement System (SPMS) an Bord des UAVs (In Anlehnung an [114])

Das SPMS interagiert direkt mit dem Missionsmanagement und der Missionssensorik. Unter Anwendung des Prinzips der aufgabenbasierten Führung [118] ist das UAV in der Lage, seine eigene Vorgehensweise entsprechend der vom Bediener zugewiesenen hochrangigen Missionsaufgaben abzuleiten, wobei relevante Beschränkungen (u. a. Luftraumgrenzen) und Hintergrundwissen (z. B. aktuelles Lagebild) berücksichtigt werden. Die entsprechende Funktionalität ist in das Missionsmanagement integriert [114]. Die Architektur des SPMS (siehe Abb. 2.4) umfasst zwei miteinander verbundene Ebenen: Das *Perzeptionsmanagement* und das *Sensormanagement*. Ersteres verarbeitet den vom Missionsmanagement gestellten Perzeptionsauftrag, indem es die am besten geeignete Perzeptionskette (siehe Abschnitt 3.4.2) für die Aufgabe auswählt. Anschließend werden die Anforderungen aus dem Missionsmanagement an das Sensormanagement übergeben, das daraus die Aufgaben für die Sensorsteuerung (z. B. Anpassung des Sensorbildwinkels, Einstellung der Bildrate, kardanische Ausrichtung des Sensors) ableitet und ausführt.

## 2.3 Ansätze zur koordinierten Sensor- und UAV-Führung

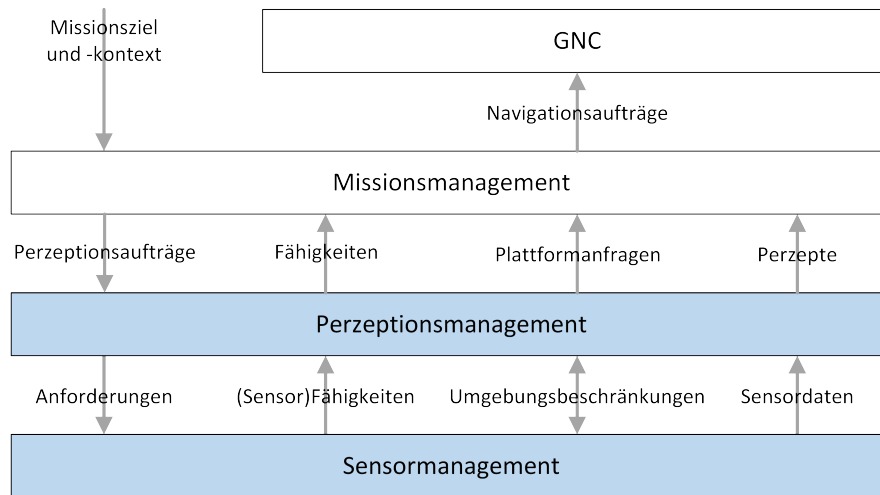


Abbildung 2.4: Systemarchitektur des Sensor- und Perzeptionsmanagement System. Das System beinhaltet die beiden interagierenden Module für das Perzeptionsmanagement und das Sensormanagement [50].

### 2.3.2 Sensorbestimmte UAV-Flugführung

In diesem Abschnitt werden Ansätze zur *sensorbestimmten UAV-Flugführung* aus dem Stand der Forschung untersucht. Wie zu Beginn von Abschnitt 2.3 erwähnt, wurde für die Untersuchung eine Themenabgrenzung und Konkretisierung in Abschnitt 1.4 festgelegt. Da der Flugzustand des UAVs und das aufzuklärende Gebiet als bekannt vorausgesetzt werden, werden Ansätze zur simultanen Lokalisierung und Kartierung (engl.: Simultaneous Localization and Mapping, SLAM) [119–121] nicht weiter berücksichtigt. Im Gegensatz zu SLAM oder Ansätzen zum UAV-basierten Multi-Target-Tracking [36, 122–124] wurde die sensorbestimmte UAV-Flugführung zur Detektion bodengebundener Objekte bislang nur wenig untersucht. Nachfolgend wird eine Auswahl relevanter Ansätze vorgestellt:

Von Yifeng et al. wird in [125] ein Verfahren zur luftgestützten Aufklärung bodengebundener Objekte mit Hilfe bildgebender Sensorik vorgestellt. Hierfür wird aus der Länge der Sensorsichtlinie zwischen dem UAV und dem Sensorfootprint am Boden sowie dem Sensorbildwinkel die Bodenpixelauflösung (GSD) berechnet und daraus die Größe des Objekts bestimmt. Aus der Objektgröße wird mit Hilfe der empirisch ermittelten Johnson Kriterien [126] die Wahrscheinlichkeit berechnet, mit der das Objekt detektiert werden kann. Zusätzlich wurde ein Umweltmodell erstellt, das die Wahrscheinlichkeit der Detektion in Abhängigkeit des Kontrasts zwischen Objekt und Hintergrund sowie der atmosphärischen Dämpfung berücksichtigt. Durch ein Bayessches Wahrscheinlichkeitsmodell

## 2 Stand der Forschung

wird die Wahrscheinlichkeit ermittelt, dass sich das Objekt im Sensorfootprint befindet. Für die Planung der UAV-Flugtrajektorie wurden die Flughöhe, der Sensorbildwinkel und der Sensor-Elevationswinkel fest vorgegeben. Mit diesen Annahmen ergibt sich für die Trajektorienplanung ein Flugpfad, der kreisförmig um das Objekt am Boden verläuft. Durch die Festlegung des Elevationswinkels wird ein Freiheitsgrad der Optimierung entzogen, wodurch der Optimierungsprozess deutlich vereinfacht wird. Demgegenüber kann bei der Planung der Flugtrajektorie mit einem vorgegebenen (flachen) Elevationswinkel nicht bzw. nur beschränkt auf Hindernisse in der Sensorsichtlinie reagiert werden.

In der Arbeit von Skoglar et al. [37] wird ein Verfahren zur sensorbestimmten UAV-Flugführung vorgestellt, bei dem eine Reihe straßengebundener Fahrzeuge von einem UAV mit einem kardanisch aufgehängten bildgebenden Sensor detektiert und überwacht werden sollen. Das Problem besteht darin, alle entdeckten Ziele zu überwachen und gleichzeitig nach neuen Zielen zu suchen, indem die Ausrichtung des Sensors und die Bewegung des UAV koordiniert werden. Die Bewegung der Fahrzeuge wird durch ein Straßennetz beschränkt, das als vorab bekannt angenommen wird. Von den Autoren wird ein Planer auf der Grundlage einer Zustandsmaschine mit drei verschiedenen Modi vorgeschlagen:

- Zielverfolgung (Tracking)
- Wiederauffinden bekannter Ziele
- Suche nach neuen Zielen (Detektion)

Nachfolgend wird ausschließlich der letzte Modus betrachtet. Für die Suche nach neuen Zielen (Fahrzeugen) wird ein gitterbasierter Filter (hier: ein Punkt-Masse-Filter [127]) verwendet mit dem modelliert wird, welche Straßen bereits untersucht wurden. Mit der Nutzung des gitterbasierten Filters ist der Nachteil verbunden, dass ein vereinfachtes Bewegungsmodell für die Fahrzeuge verwendet werden muss, um die Dimensionalität des Zustandsraums zu verringern. Als Zielfunktion für das Optimierungsproblem wird die kumulative Detektionswahrscheinlichkeit genutzt. Das Modell der Detektionswahrscheinlichkeit berücksichtigt die Ausrichtung und den Bildwinkel des Sensors, die Beeinträchtigung der Sensorsichtlinie durch Gebäude, sowie ein Modell der Detektionsleistung, das die Entfernung des Objekts und atmosphärische Störungen berücksichtigt. Das UAV bewegt sich in einer konstanten Höhe mit konstanter Fluggeschwindigkeit. Als Steuereingabe dient die Rollrate, die durch den minimalen Kurvenradius beschränkt wird. Zu jedem Zeitschritt wird die Steuereingabe bestimmt, sodass das UAV einen Referenzpunkt ansteuert, der sich in einer bestimmten Entfernung vor dem Sensorfootprint am Boden befindet. Bei dem Ansatz von Skoglar et al. wird der Abstand zwischen dem UAV und dem Objekt / dem

### 2.3 Ansätze zur koordinierten Sensor- und UAV-Führung

Sensorfootprint am Boden optimiert. Das Sensorleistungsmodell wird stark vereinfacht, weshalb beispielsweise keine explizite Berücksichtigung des Elevationswinkels oder der Bodenpixelauflösung erfolgt.

Einen Ansatz zur Trajektorienplanung für die automatisierte visuelle Detektion und Inspektion dreidimensionaler Objekte mit Hilfe von Multikopter-UAVs stellen Ivić et al. in [128] vor. Die Autoren verwenden eine Methode auf Grundlage eines Potentialfelds, womit die UAV-Flugtrajektorien generiert werden. Darüber hinaus werden mit der Methode Abstandsflächen erzeugt, die zur Vermeidung von Kollisionen und zur Festlegung der Kameraausrichtung am UAV dienen. Das Sichtfeld des Sensors wird als Kegel modelliert, dessen Spitze sich im Zentrum des UAVs befindet. Die Ausrichtung des Kegels entspricht damit der Ausrichtung des Sensors. Die Höhe des Kegels legt die maximal zulässige Entfernung des UAVs von der Oberfläche fest, die inspiziert wird. Die UAV-Trajektorienoptimierung basiert auf einer stichprobenbasierten Pfadplanung in Kombination mit modellprädiktiver Regelung (MPC), die in [129] detailliert vorgestellt wurde. Bei dem Ansatz von Ivić et al. werden weder der Sensorbildwinkel adaptiert noch die Bodenpixelauflösung oder atmosphärische Einflussfaktoren berücksichtigt. Es wird lediglich ein vereinfachtes Modell des Sensorfootprints für die Bewertung der vom Footprint überstrichenen Fläche verwendet.

Zhang & Huang untersuchen in [3] ein Verfahren zur luftgestützten Aufklärung und Überwachung in komplexen geometrischen Umgebungen (z. B. in bergigen und städtischen Gebieten). Hierfür haben die Autoren eine Methode entwickelt, um die minimale Anzahl von Wegpunkten zu ermitteln, die überflogen werden müssen, um ein Gebiet vollständig aufzuklären oder überwachen zu können. Die Trajektorienplanung durch die Wegpunkte erfolgt unter Berücksichtigung kinematischer Beschränkungen des UAVs und wird durch Bézier-Kurven [130] modelliert. Das UAV wird als Starrflügel-Flugzeug modelliert, das sich in konstanter Flughöhe und mit konstanter Fluggeschwindigkeit bewegt, wobei der minimale Kurvenradius beschränkt ist. Das UAV ist mit einem nach unten gerichteten bildgebenden Sensor ausgestattet, dessen Sichtfeld, ähnlich zum Ansatz von [128], als Kegel modelliert wird. Objekte (z. B. Personen) werden detektiert, sobald sie sich im Sichtfeld befinden und die Sensorsichtlinie frei von Hindernissen (z. B. Gebäude, Berge) ist. Wie bereits beim Ansatz von Ivić et al. festgestellt wurde, verwenden auch die Autoren Zhang & Huang ein stark vereinfachtes Sensorleistungsmodell. Die UAV-Flugtrajektorien werden durch Bézier-Kurven modelliert, weshalb flugdynamische Beschränkungen nur näherungsweise umgesetzt werden können.

### 2.3.3 Diskussion

In Abschnitt 2.3.1 wurden verschiedenen Ansätze für ein Sensormanagement an Bord des UAVs vorgestellt, die in Tabelle 2.4 zusammengetragen und gegenübergestellt sind. Für die Gegenüberstellung wurde untersucht, ob Funktionalitäten für eine *Sensorsteuerung*, eine *Sensordatenauswertung*, sowie eine *sensorbestimmte UAV-Flugführung* unterstützt und umgesetzt werden.

Tabelle 2.4: Gegenüberstellung unterschiedlicher Ansätze für ein Sensormanagement

Referenz	Sensorsteuerung	Sensordatenauswertung	Sensorbestimmte UAV-Flugführung
Ng & Ng [110]	✓	—	—
Hero & Cochran [112]	✓	—	—
Russ & Stütz [114, 115]	✓	✓	—

Die Ansätze für ein Sensormanagement von Ng & Ng [110] und Hero & Cochran [112] fokussieren sich im Wesentlichen auf die dynamische Auswahl geeigneter Sensorsysteme und die Anpassung der Parametrisierung, um die Aufklärungsleistung zu verbessern. Für die Sensordatenauswertung werden Schnittstellen bereitgestellt, eine automatisierte Auswertung durch die Systeme findet aber nicht statt. Von den aufgelisteten Systemen stellt nur das Sensor- und Perceptionsmanagement System von Russ & Stütz [114, 115] inhärente Funktionen für die automatisierte Sensorsteuerung und Sensordatenauswertung bereit. Zudem wird in [50] beschrieben, dass das SPMS, als Bestandteil eines UAS, bereits in der Simulation und im Realflug untersucht und validiert wurde. Obwohl das SPMS über keine dedizierten Fähigkeiten zur sensorbestimmten UAV-Flugführung verfügt, wird das System als geeignet für die Integration der *sensormodellbestimmten Befliegungsplanung* angesehen, welches in dieser Arbeit entwickelt wird.

Aus den Ansätzen zur sensorbestimmten UAV-Flugführung aus Abschnitt 2.3.2 wurden die Merkmale zur Modellierung der Sensorleistung und der Trajektorienoptimierung identifiziert und in Tabelle 2.5 zusammengefasst. Aus der Tabelle geht hervor, dass in den vorgestellten Arbeiten vielfach nur ein vereinfachtes Sensorleistungsmodell verwendet wurde, um die Detektionsleistung des Sensorsystems zu modellieren. Eine UAV-Trajektorienoptimierung auf Grundlage von Sensorleistungsmodellen, die die relevanten Umgebungsbedingungen detailliert und realitätsnah abbilden, wurde nicht durchgeführt.

## 2.4 Forschungsbedarf und Definition der Forschungsfragen

Tabelle 2.5: Merkmale der sensorbestimmten UAV-Flugführung aus den vorgestellten Ansätzen

Referenz	Modellierung der Sensorleistung	Merkmale der Trajektorienoptimierung
Yifeng et al. [125]	verwendet Johnson Kriterien, berücksichtigt atmosphärische Dämpfung und Kontrast zum Objekt	festgelegter Elevationswinkel, Variation des Rollwinkels
Skoglar et al. [37]	berücksichtigt Länge der Sichtlinie und atmosphärische Störungen	Variation der Rollrate
Ivić et al. [128]	fester Bildwinkel, keine Modellierung der Detektionsleistung	Trajektorienoptimierung durch MPC
Zhang & Huang [3]	fester Bildwinkel, Unterbrechnung der Sensorsichtlinie wird berücksichtigt	Verwendung von Bézier-Kurven

## 2.4 Forschungsbedarf und Definition der Forschungsfragen

Ausgehend von der Problemstellung in Abschnitt 1.2 und der Untersuchung zum Stand der Forschung wird der folgende *Forschungsbedarf* festgestellt: Der navigatorische Aspekt der sensortragenden Plattform in Verbindung mit der koordinierten sensorischen Datenerfassung und Auswertung, mit dem Ziel die Detektionsleistung auf Grundlage detaillierter Sensorleistungsmodelle zu steigern, wurde noch nicht ausreichend erforscht und bedarf daher weiterer Untersuchung. Aus dieser Forschungslücke wurde das übergeordnete Forschungsziel in Abschnitt 1.3 abgeleitet, das den Untersuchungsschwerpunkt dieser Arbeit darstellt. Die Beantwortung der Forschungsfragen und die Erreichung des Forschungsziels sollen einen Beitrag leisten, diese Forschungslücke zu schließen.

Aus der Übersicht zu den Ansätzen zur Modellierung der Detektionsleistung in Tabelle 2.2 und der Diskussion in Abschnitt 2.1.3 wird ein *weiterer Forschungsbedarf* festgestellt: Die Modellierung von Sensorleistungsmodellen für Perzeptionsketten mit Deep Learning basierten Objektdetektoren zur luftgestützten Fahrzeugdetektion wird als unzureichend bewertet. Es wird daher als sinnvoll erachtet, ein derartiges Sensorleistungsmodell für den Zweck der sensormodellbestimmten Befliegungsplanung zu entwickeln und zu untersuchen.

Aus den Zwischenzielen in Abschnitt 1.3 und dem festgestellten Forschungsbedarf werden die folgenden *Forschungsfragen* definiert. Diese sind hierarchisch den jeweiligen Zwischenzielen zugeordnet:

## 2 Stand der Forschung

**Forschungsfrage 1.1:** Wie kann das Sensor- & Perzeptionsmanagement zur automatisierten, sensormodellbestimmten Befliegungsplanung befähigt werden?

**Forschungsfrage 2.1:** Wie kann ein Sensorleistungsmodell für eine Perzeptionskette mit Deep Learning basierendem Objektdetektor modelliert werden, um die Detektionsleistung bezüglich ausgewählter perzeptiver Einflüsse abbilden zu können?

**Forschungsfrage 2.2:** Wie kann die Detektionsleistung für ein Sensorleistungsmodell in Abhängigkeit der Sensor- bzw. UAV-Position modelliert werden?

**Forschungsfrage 3.1:** Wie kann eine, bezüglich der Detektionsleistung, global optimale Flugtrajektorie generiert werden, unter Berücksichtigung sensorplattformspezifischer und missionsspezifischer Anforderungen?

**Forschungsfrage 3.2:** Wie kann eine, bezüglich der Detektionsleistung, lokal optimale Flugtrajektorie mit vorgegebener Startkonfiguration generiert werden, unter Berücksichtigung sensorplattformspezifischer und missionsspezifischer Anforderungen?

**Forschungsfrage 5.1:** Welches Potenzial bietet die sensormodellbestimmte Befliegungsplanung zur Steigerung der Detektionsleistung mit zeitkontinuierlicher Trajektorienoptimierung sowie mit zeit- und wertdiskreter Trajektorienoptimierung?

**Forschungsfrage 5.2:** Wie werden perzeptive und sensorplattformspezifische Anforderungen und Randbedingungen bei der sensormodellbestimmten Befliegungsplanung umgesetzt und eingehalten?

**Forschungsfrage 5.3:** Wie beeinflussen die zu untersuchenden Optimierungsmethoden die Aufklärungsdauer und die erforderliche Berechnungsdauer?

Die Forschungsfrage 1.1 ist abgeleitet aus dem Zwischenziel 1 in Abschnitt 1.3 und dem Fazit aus Abschnitt 2.3.3, dass das Sensor- und Perzeptionsmanagement als eine geeignete Plattform für die sensormodellbestimmte Befliegungsplanung bewertet wird. Aus dem Zwischenziel 2 und dem Forschungsbedarf, der in Abschnitt 2.1.3 bezüglich eines Sensorleistungsmodells mit Deep Learning basierender Perzeptionskette identifiziert wurde, resultiert die Forschungsfrage 2.1. Des Weiteren soll mit Hilfe der Forschungsfrage 2.2 beantwortet werden, wie die ortsabhängige Detektionsleistung durch das Sensorleistungsmodell abgebildet werden kann. Aus dem Zwischenziel 3 und dem Forschungsbedarf, der in Abschnitt 2.2.4 zur Generierung global und lokal optimaler Flugtrajektorien



## 2.4 Forschungsbedarf und Definition der Forschungsfragen

festgestellt wurde, werden die beiden Forschungsfragen 3.1 und 3.2 abgeleitet. Das Zwischenziel 5 und die Forschungsfrage 5.1 bis 5.3 beziehen sich auf die Validierung der sensormodellbestimmten Befliegungsplanung.

Die Beantwortung der einzelnen Forschungsfragen ist Voraussetzung, um die Zwischenziele 1 - 5 und das übergeordnete Forschungsziel in Abschnitt 1.3 erreichen zu können. In Tabelle 2.6 sind die Zwischenziele und die korrelierenden Forschungsfragen zusammengefasst. Zudem sind die Kapitel aufgelistet, in denen die Forschungsfragen und Ziele bearbeitet werden.

Tabelle 2.6: Übersicht zur Bearbeitung der Forschungsfragen und Forschungsziele

Zwischenziel	1	2	3	4	5
Forschungsfrage	1.1	2.1 & 2.2	3.1 & 3.2	—	5.1 - 5.3
Bearbeitung in Kapitel	3	3	3 & 4	4	6



## 3 Konzept und Methodenauswahl

In diesem Kapitel wird das funktionale Konzept zur sensormodellbestimmten Befliegungsplanung beschrieben. Im ersten Schritt wird das Prinzip zur Steigerung der Detektionsleistung durch die gezielte Beeinflussung photogrammetrischer Zustände erläutert. Anschließend werden Anforderungen an das Konzept der Befliegungsplanung definiert, um die Forderungen aus den Zwischenzielen und den Forschungsfragen umsetzen zu können. Darauf aufbauend wird das Konzept im Detail vorgestellt und dessen Einbindung in das erweiterte Sensor- und Perzeptionsmanagement erläutert. Darüber hinaus wird das Prinzip und die Entwicklung von Sensorleistungsmodellen vorgestellt, die ein wesentlicher Bestandteil für die Trajektorienoptimierung im Rahmen der Befliegungsplanung sind. Abschließend werden die Grundlagen für die Generierung der UAV-Flugtrajektorien gelegt und ausgewählte mathematische Methoden zur Trajektorienoptimierung betrachtet.

In diesem Kapitel werden die *Zwischenziele 1* und *2*, sowie Anteile am *Zwischenziel 3* aus Abschnitt 1.3 bearbeitet. Darüber hinaus werden die *Forschungsfragen 1.1*, *2.1* und *2.2* aus Abschnitt 2.4 beantwortet und es wird mit der Bearbeitung der *Forschungsfragen 3.1* und *3.2* begonnen.

### 3.1 Grundlage zur Steigerung der Detektionsleistung

Im Abschnitt 2.1 wurden bereits Ansätze vorgestellt, die zum Ziel haben, den Einfluss bestimmter Zustände auf die Detektionsleistung zu modellieren. In den Arbeiten von [30] und [50] wurde gezeigt, dass spezifische Umgebungszustände, die den photogrammetrischen, topographischen und atmosphärischen Zuständen zugeordnet werden können, die Sensordatengenerierung und letztendlich die Leistungsfähigkeit der automatisierten Objektdetektion beeinflussen.

In Tabelle 3.1 werden diese Umgebungszustände und der Zustand des aufzuklärenden Objekts (kurz: Objektzustand) in Bezug auf vier verschiedene Merkmalsausprägungen charakterisiert, wobei jeweils zwei Ausprägungen zueinander komplementär sind. Die Merkmalsausprägungen beziehen sich auf den Zustand unmittelbar vor der sensorischen Datenerfassung.

### 3 Konzept und Methodenauswahl

Tabelle 3.1: Beeinflussbarkeit relevanter Objekt- und Umgebungszustände vor der sensorischen Datenerfassung mit Auswirkung auf die Detektionsleistung.

Zustand ist ...	bekannt	unbekannt
beeinflussbar	photogrammetrische Zustände	(nicht relevant)
nicht beeinflussbar	topographische und atmosphärische Zustände	Zustand des aufzuklärenden Objekts

Mit den ersten beiden Merkmalsausprägungen wird beschrieben, ob der Zustand zum Zeitpunkt unmittelbar vor der Datenerfassung *bekannt* oder *unbekannt* ist. Ersteres trifft auf die Umgebungszustände zu. Beispielsweise können die *atmosphärischen Zustände* (z. B. Helligkeit, Bewölkung, Niederschlag) durch Expertenwissen oder aus Datenbanken für den Zeitpunkt der sensorischen Datenerfassung abgefragt werden. Die *topographischen Zustände* (u. a. Gebäude, Vegetation, Straßen, Gewässer) beziehen sich auf den Inhalt des Sensorfootprints am Boden und können vorab durch ein Geoinformationssystem ermittelt werden. Die *photogrammetrischen Zustände* umfassen den Elevationswinkel der Sensorsichtlinie und die GSD. Der Elevationswinkel resultiert aus der geometrischen Beziehung der Sensor- / UAV-Position und des Sensorfootprints am Boden. Die GSD berechnet sich aus der Länge der Sichtlinie und dem Bildwinkel des bildgebenden Sensors. Da sich die Umgebungszustände auf die sensorische Datenerfassung und Auswertung, und damit auf die Detektionsleistung auswirken können, werden sie in dieser Arbeit auch als *perzeptive Einflüsse* bezeichnet.

Im Gegensatz zu den Umgebungszuständen, ist der Zustand des aufzuklärenden Objekts zum Zeitpunkt der Messung in der Regel nicht bekannt und das eigentliche Ziel der Aufklärung. Zum Objektzustand zählt u. a. die Art des Objekts, dessen Position und Orientierung. Am Beispiel der Fahrzeugdetektion ist vorab in der Regel nicht bekannt ob, wo und wie viele Fahrzeuge sich in einem aufzuklärenden Gebiet befinden.

Die anderen beiden Merkmalsausprägungen, die in Tabelle 3.1 betrachtet werden, beziehen sich auf die *Beeinflussbarkeit* dieser Zustände. Durch die Beeinflussung der Zustände soll deren Wirkung auf die Detektionsleistung gezielt ausgenutzt werden. Bei der luftgestützten Aufklärung eignen sich im Wesentlichen nur die photogrammetrischen Zustände (in Tabelle 3.1 grün markiert), die innerhalb technischer Grenzen gezielt manipuliert werden können.

Demgegenüber können die topographischen und atmosphärischen Bedingungen nicht beeinflusst werden, da diese direkt oder indirekt (z. B. die Helligkeit und Sichtbedingungen im aufzuklärenden Gebiet) durch den Aufklärungsauftrag bestimmt werden. Der Zustand

### 3.2 Anforderungen an die sensormodellbestimmte Befliegungsplanung

des aufzuklärenden Objekts ist ebenfalls nicht beeinflussbar, da das Objekt Gegenstand der eigentlichen Aufklärung ist. Unterschiedliche Eigenschaften des Objekts, z. B. dessen Farbe, Form, Größe oder Wärmeabstrahlung, können die Korrektheit der Objektdetektion, und damit die Detektionsleistung, stark beeinflussen. Zusammenfassend kann festgehalten werden:

Die photogrammetrischen Zustände sind geeignet, durch deren gezielte Beeinflussung die Detektionsleistung zu steigern, oder einer Degradation durch Störeinflüsse entgegen zu wirken. Für das Konzept der sensormodellbestimmten Befliegungsplanung wird dieser Sachverhalt gezielt ausgenutzt, um optimale Sensor- / UAV-Positionen zu ermitteln und daraus letztendlich optimale UAV-Flugtrajektorien zu generieren.

## 3.2 Anforderungen an die sensormodellbestimmte Befliegungsplanung

Abgeleitet aus den *Forschungszielen* (Abschnitt 1.3) und dem Stand der Forschung (Kapitel 2) werden im Folgenden Anforderungen an das Konzept zur sensormodellbestimmten Befliegungsplanung festgelegt. Durch die Formulierung der Anforderungen soll ein Vorgehen definiert werden, um die Forschungsziele umsetzen und erfüllen zu können. Zur Übersicht sind diese Anforderungen in Tabelle 3.2 aufgelistet. Eine detailliertere Beschreibung findet sich in den nachfolgenden Absätzen.

Tabelle 3.2: Anforderungen an die sensormodellbestimmte Befliegungsplanung

Nr.	Die sensormodellbestimmte Befliegungsplanung soll ...	Abgeleitet aus
1	eine automatisierte Planung zur Sensorsteuerung enthalten	
2	Sensorleistungsmodelle für bildgebende Sensorik verwenden	Zwischenziel 2
3	ausgesuchte perzeptive, sensorplattformspezifische und missionsspezifische Anforderungen erfüllen	Zwischenziel 1
4	die Fähigkeit zur automatisierten Generierung optimaler Flugtrajektorien besitzen	Zwischenziel 3
5	als funktionales Konzept in das Sensor- & Perzeptionsmanagement System integriert werden	Zwischenziel 1

**Anforderung 1:** Die sensormodellbestimmte Befliegungsplanung soll eine *automatisierte Planung zur Sensorsteuerung* enthalten. In [22] wird beschrieben, dass für die Steuerung größerer UAS häufig ein UAV-Operator und ein Sensor-Operator eingesetzt werden. Die manuelle Koordination der Sensor- und Plattformbewegung erfordert eine enge Abstimmung zwischen den beiden Operateuren und bindet dadurch Ressourcen. Die automatisierte Planung zur Koordination der Bewegungsplanung des UAVs und der Steuerung der bildgebenden Sensorik soll die Operateure von dieser Tätigkeit entlasten, wodurch Ressourcen für anderweitige Aufgaben (z. B. für den Multi-UAV-Einsatz [118]) freigegeben werden. Die automatisierte Planung der Sensorsteuerung basiert auf dem Sensorleistungsmodell, wodurch zahlreiche Parameter (z. B. Sensorcharakteristik, atmosphärische Bedingungen) für die Steuerung berücksichtigt werden können.

**Anforderung 2:** Die sensormodellbestimmte Befliegungsplanung soll *Sensorleistungsmodelle* für bildgebende elektrooptische- und/oder Infrarot-Sensorik (EO/IR-Sensor) verwenden. Dies wird damit begründet, dass eine Vielzahl der UAVs, die heutzutage für Aufklärungs- und Überwachungsaufgaben eingesetzt werden, über EO- und/oder IR-Sensorsysteme verfügen und dadurch für einen großen Nutzerkreis relevant sind. Zudem soll die sensormodellbestimmte Befliegungsplanung für den Anwendungsfall der luftgestützten automatisierten Fahrzeugdetektion entwickelt werden. Die verwendeten Sensorleistungsmodelle sollen die Detektionsleistung gängiger Objektdetektoren abbilden. Darüber hinaus soll durch das Sensorleistungsmodell die zu erwartende Detektionsleistung quantifiziert werden können.

**Anforderung 3:** Die sensormodellbestimmte Befliegungsplanung soll ausgesuchte perzeptive, sensorplattformspezifische und missionsspezifische Anforderungen erfüllen.

Die **perzeptiven Anforderungen** erweitern die *Anforderung 2* um die Fähigkeit zur Adaption des *Bildwinkels* für eine EO/IR-Sensorik. Dadurch kann die *Bodenpixelauflösung* (engl.: Ground Sample Distance, GSD) für die Befliegungsplanung dynamisch an einen vorgegebenen Referenzwert angepasst werden. Dies ist erforderlich, da über die GSD maßgeblich der Detaillierungsgrad der Sensorinformation beeinflusst wird. Eine zu geringe GSD kann dazu führen, dass Objekte nicht erkannt werden, da spezifische Merkmale, die für die Detektion essentiell sind (z. B. Kanten, Muster), durch die unzureichende Auflösung verloren gehen. Demgegenüber kann aus einer sehr hohen Auflösung folgen, dass zu viele Details zu Fehldetektionen führen und dadurch die Zuverlässigkeit der Objekterkennung reduziert wird. Zudem korreliert eine hohe GSD mit einem kleinen Sensorfootprint am Boden, wodurch die *aufgeklärte Fläche pro Zeiteinheit* (Aufklärungsleistung) sinkt.

Zu den **sensorplattformspezifischen Anforderungen**, die für die Befliegungsplanung herangezogen werden sollen, gehören die *Fluggeschwindigkeit*, die *Rollrate* und das

### 3.2 Anforderungen an die sensormodellbestimmte Befliegungsplanung

*maximale Lastvielfache* des UAVs für den horizontalen Kurvenflug. Das UAV wird als Starrflügel-Flugzeug modelliert und die Fluggeschwindigkeit wird als näherungsweise konstant angenommen, wobei diese durch einen Referenzwert vorgegeben wird. Durch die Limitierung der Fluggeschwindigkeit und des maximalen Lastvielfachen soll sichergestellt werden, dass das UAV zu jeder Zeit innerhalb der zulässigen Flugbereichsgrenzen operiert. Aus der Vorgabe des maximalen Lastvielfachen und der Fluggeschwindigkeit kann der maximal zulässige *Rollwinkel* berechnet werden. Dieser geht als Randbedingung in die Trajektorienoptimierung mit ein.

Mit den **missionsspezifischen Anforderungen** sollen weitere Aspekte festgelegt werden können, die einen wesentlichen Einfluss auf die erfolgreiche Missionsdurchführung haben. Die *Minimierung der Flugdauer* während des Aufklärungsvorgangs soll als Optimierungsgröße in die Befliegungsplanung mit eingehen. Dies wird damit begründet, dass der Aufklärungsdauer in verschiedensten Missionsszenarien eine essentielle Bedeutung zukommt. Als Beispiel für eine zeitkritische Aufklärungsmission wird die luftgestützte Vermisstensuche angeführt. Hierbei ist entscheidend, eine vermisste und gegebenenfalls hilflose Person zeitnah zu finden, um sie vor einem möglichen körperlichen Schaden, z. B. durch Unterkühlung, zu bewahren. Darüber hinaus soll dem Sensor- bzw. UAV-Operateur die Möglichkeit gegeben werden, bei der Befliegungsplanung eine hohe Detektionsleistung oder eine geringe Aufklärungsdauer priorisieren zu können. Dabei ist zu berücksichtigen, dass die Detektionsleistung und die Aufklärungsdauer zwei sich gegenseitig beeinflussende Optimierungsziele sind. Dies hat zur Folge, dass aus einer Steigerung der Detektionsleistung eine längere Aufklärungsdauer resultieren kann und umgekehrt. Als eine weitere Anforderung soll für die Befliegungsplanung eine *konstante Flughöhe* vorgegeben werden können, in der sich das UAV bewegt. Das soll eine Höhenstaffelung und damit einen sicheren Betrieb mehrerer Luftfahrzeuge im selben Operationsgebiet ermöglichen. Zudem kann damit ein vertikaler Sicherheitsabstand zu Hindernissen am Boden (z. B. Antennenmasten, Türme) vorgegeben werden. Alternativ kann die Höhenvorgabe erforderlich sein, um die Geräuschsignatur des UAVs am Boden zu reduzieren und dadurch eine akustische Aufklärbarkeit zu minimieren.

**Anforderung 4:** Die sensormodellbestimmte Befliegungsplanung soll die Fähigkeit zur automatisierten Generierung *optimaler Flugtrajektorien* besitzen. Die Referenztrajektorie für die UAV-Flugbewegung soll auf die Sensorsteuerung aus *Anforderung 1* abgestimmt sein. Bei der Planung der Trajektorie sollen zudem flugdynamische Beschränkungen des UAVs berücksichtigt werden, um die zulässigen Flugbereichsgrenzen nicht zu überschreiten. Die Referenztrajektorie ist der Befliegungsplan für die automatisierte Aufklärung und dient als Sollwertvorgabe für den Autopiloten an Bord des UAVs. Die Trajektorienplanung

### 3 Konzept und Methodenauswahl

soll *deterministisch* und *reproduzierbar* erfolgen. Das Optimierungskriterium bildet die Detektionsleistung unter Einhaltung der in *Anforderung 3* aufgeführten Randbedingungen.

**Anforderung 5:** Die sensormodellbestimmte Befliegungsplanung soll in das Systemkonzept des *Sensor- und Perceptionsmanagement System* (SPMS) von Russ & Stütz [114] integriert werden. Das SPMS, das in Abschnitt 2.3.1 beschrieben wurde, stellt einen Ansatz für das Management der bildgebenden UAV-Sensorik mit integrierter Sensordatenverarbeitungskette dar. Es unterstützt bereits inhärente Fähigkeit zum auftragsbasierten Sensoreinsatz. Die Fähigkeit zur UAV-Führung, einschließlich der Bewegungsplanung und der koordinierten Sensorsteuerung, ist bislang nicht gegeben. Aus diesem Grund soll das SPMS um die Fähigkeit zur sensormodellbestimmten Befliegungsplanung erweitert werden.

### 3.3 Konzeptbeschreibung

Nach der Formulierung der Anforderungen im vorangegangenen Abschnitt wird im Folgenden ein Konzept zur Realisierung der sensormodellbestimmten Befliegungsplanung erläutert, das die genannten Anforderungen umsetzt und das *Zwischenziel 1* erfüllt. Das Resultat ist das *Erweiterte Sensor- und Perceptionsmanagement* (ESPM), welches von Zwick, Koch & Stütz entwickelt und in [31] detailliert beschrieben wurde. Das ESPM basiert auf dem Paradigma des *Sensor- und Perceptionsmanagement Systems* von Russ & Stütz [114] (vgl. Abb. 2.4 in Abschnitt 2.3.1) und erweitert dieses um inhärente und dedizierte Fähigkeiten zur hochautomatisierten sensormodellbestimmten Befliegungsplanung um ein vorgegebenes Gebiet hinsichtlich spezifischer Objekte (z. B. Fahrzeuge oder Personen) luftgestützt aufklären zu können. Es zielt darauf ab, durch die enge Verknüpfung von perzeptiven und plattformspezifischen Fähigkeiten die Aufklärungsleistung unter Einhaltung von Randbedingungen zu maximieren.

Im nächsten Abschnitt wird der Ablauf zur Planung und Durchführung eines Aufklärungsauftrags durch das ESPM beschrieben. Im Anschluss wird auf die Integration des ESPM in die Systemarchitektur des UAVs eingegangen und im letzten Schritt wird die eigentliche Systemarchitektur des ESPM vorgestellt.

Die sensormodellbestimmte Befliegungsplanung, als ein dedizierter Bestandteil des ESPM, wird in Kapitel 4 für dessen Evaluierung in ein Experimentalsystem überführt. Da der Fokus in dieser Arbeit auf der Entwicklung, Realisierung und Umsetzung der Befliegungsplanung liegt, wird für eine detaillierte Beschreibung der Fähigkeiten des



SPMS (u. a. das Perzeptionsmanagement oder die Ansteuerung und Datenerfassung über die bildgebenden Sensoren) auf [114, 115, 117] verwiesen.

### 3.3.1 Ablaufbeschreibung eines Aufklärungsauftrags durch das ESPM

In Abb. 3.1 ist der Ablauf zur automatisierten Planung und Durchführung eines Aufklärungsauftrags dargestellt, wie er durch das ESPM an Bord eines UAVs ausgeführt werden soll. Der grün hinterlegte Bereich markiert den Untersuchungsschwerpunkt dieser Arbeit, der unter dem Begriff der *sensormodellbestimmten Befliegungsplanung* zusammengefasst ist. Die Befliegungsplanung ist damit eine Teilfunktionalität für die automatisierte Abarbeitung eines Aufklärungsauftrags, eingebettet in das ESPM.

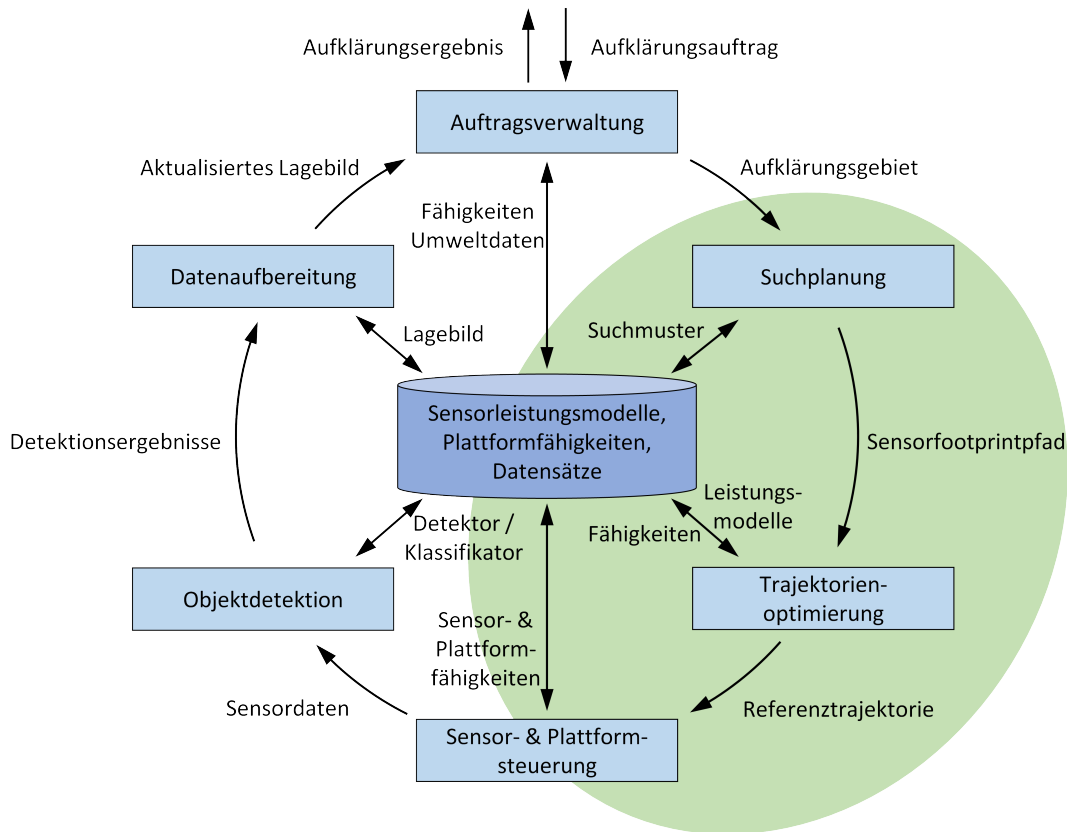


Abbildung 3.1: Ablauf zur automatisierten Planung und Durchführung von Aufklärungsaufträgen im *Erweiterten Sensor- und Perzeptionsmanagement* (ESPM). Der grün markierte Bereich umfasst die Funktionalität der sensormodellbestimmten Befliegungsplanung.

### 3 Konzept und Methodenauswahl

Der Ablauf zur automatisierten Planung und Durchführung eines Aufklärungsauftrags durch das ESPM, wie er in Abb. 3.1 abgebildet ist, wird im Folgenden zusammengefasst:

- Von einem übergeordneten Missionsmanagementsystem (nicht abgebildet) wird der Aufklärungsauftrag (z. B. die Suche nach Fahrzeugen in einem festgelegten Gebiet) an die *Auftragsverwaltung* des ESPM übermittelt.
- Mit der Information zum Aufklärungsgebiet (u. a. Ausdehnung und Lage) wird im Modul *Suchplanung* die räumlich-zeitliche Positionierung des Sensorfootprints im aufzuklärenden Gebiet geplant. Informationen, die für die Planung notwendig sind (z. B. Größe des Sensorfootprints, Footprintgeschwindigkeit am Boden), werden vor der Mission definiert und als *Datensätze* hinterlegt.
- Aus der Suchplanung resultiert der Sensorfootprintpfad, der Ausgangsbasis für die *Trajektorienoptimierung* ist. Durch die Optimierung wird dafür eine optimale UAV-Flugtrajektorie, die sogenannte Referenztrajektorie generiert. Für die Optimierung werden Fähigkeiten (u. a. flugdynamische Randbedingungen) und Leistungsmodelle berücksichtigt, um optimale und befliegbare UAV-Positionen zu ermitteln, aus denen letztendlich die Referenztrajektorie generiert wird.
- Aus der Referenztrajektorie und dem Sensorfootprintpfad wird im Modul *Sensor- & Plattformsteuerung*, unter Verwendung der Sensor- und Plattformfähigkeiten, die räumlich-zeitliche Sensorsteuerung (u. a. Sensorausrichtung, Einstellung des Bildwinkels) mit der UAV-Flugführung koordiniert. Es folgt die eigentliche Befliegung entlang der Referenztrajektorie durch das UAV, einschließlich der Erfassung der Sensordaten.
- Die Sensordaten werden während des Fluges im Modul *Objektdetektion* aufbereitet und mit Hilfe ausgewählter Objektdetektoren bzw. Klassifikatoren analysiert. Aus diesem Prozess resultieren Detektionsergebnisse, die im nächsten Schritt weiterverarbeitet werden.
- Im Modul *Datenaufbereitung* wird das Lagebild mit den Detektionsergebnissen abgeglichen und gegebenenfalls aktualisiert.
- Im letzten Arbeitsschritt wird das aktualisierte Lagebild an die *Auftragsverwaltung* weitergereicht, von wo aus es an das übergeordnete Missionsmanagementsystem gesendet wird. Damit ist der Aufklärungsauftrag abgearbeitet.

In den nachfolgenden Abschnitten wird die Integration des ESPM in das UAV näher beschrieben und die Systemstruktur des ESPM erläutert. Die Umsetzung der Funktionalität zur sensormodellbestimmten Befliegungsplanung in ein Experimentalsystem wird in Kapitel 4 vorgestellt.

### 3.3.2 Integration des ESPM in die Systemarchitektur des UAVs

Abb. 3.2 veranschaulicht die Gesamtsystemarchitektur eines *unbemannten Flugsystems* (engl.: Unmanned Aerial System, UAS), das neben dem UAV auch die Bodenstation und Peripherie zur Datenübertragung umfasst. Das ESPM ist in die Systemarchitektur des UAVs integriert. Im Gegensatz zum SPMS von Russ & Stütz [114] stellt das ESPM einen direkten Zugang zum *Flugführungs-, Navigations- und Steuerungsmodul* bereit, wodurch der Aufwand für die Übermittlung von Befehlen durch das *Missionsmanagement* reduziert wird. Diese Architektur bietet Vorteile bei der sensormodellbestimmten Befliegungsplanung, da zusätzlich zur Sensorsteuerung auch die UAV-Flugführung direkt beeinflusst werden kann. Ungeachtet dessen wird der Zugriff des ESPM auf die Flugführung und die Sensorsteuerung zur Durchführung des Aufklärungsauftrags durch das *Missionsmanagement* überwacht und zeitlich begrenzt. Bei einer Änderung von Missionsparametern, die eine Anpassung oder einen Abbruch des Aufklärungsauftrags erfordern, z. B. durch einen neuen oder höher priorisierten Aufklärungsauftrag, kann die Kontrolle über das UAV direkt vom *Missionsmanagement* übernommen werden [31].

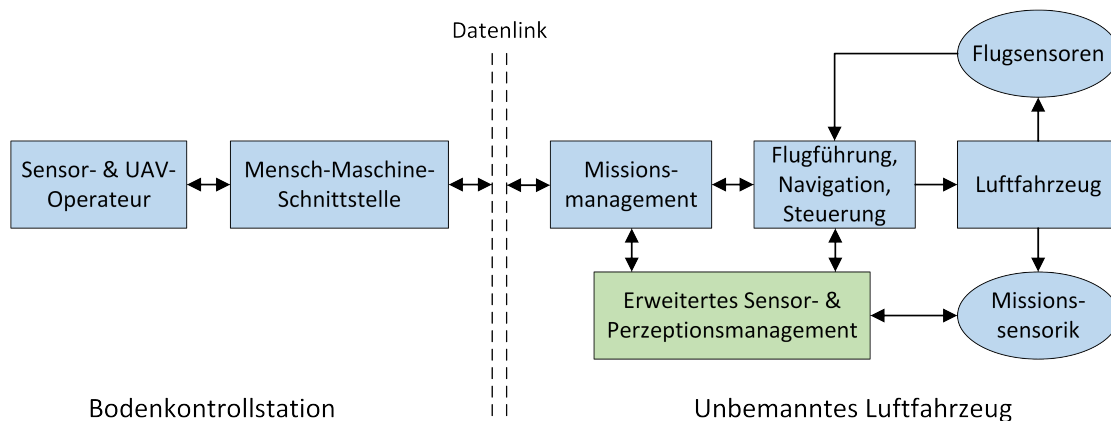


Abbildung 3.2: Integration des *Erweiterten Sensor- und Perzeptionsmanagement* in die Systemarchitektur des UAVs. Dargestellt ist das UAV im Verbund mit der Bodenkontrollstation (In Anlehnung an [31]).

Der Aufklärungsauftrag wird durch den Sensor- oder UAV-Operateur in der Bodenkontrollstation spezifiziert und per Funkübertragung an das UAV übermittelt. Der Aufklärungsauftrag beschreibt die Lage und Ausdehnung des aufzuklärenden Bereichs, der u. a. aus einem Punkt (nulldimensionale Geometrie), einer Linie (eindimensionale Geometrie) oder einer Fläche (zweidimensionale Geometrie) bestehen kann. Des Weiteren wird durch den Auftrag festgelegt, welche Objekte (z. B. Personen oder Fahrzeuge)

### 3 Konzept und Methodenauswahl

detektiert werden sollen. Zudem kann der Operateur auswählen, ob der Fokus auf einer hohen Detektionsleistung oder einer hohen Aufklärungsleistung (aufgeklärte Fläche pro Zeit) liegen soll. Hierdurch wird die Methode zur Trajektorienoptimierung festgelegt, auf die in Abschnitt 3.6 eingegangen wird. Nach Abschluss des Aufklärungsauftrags werden die Detektionsergebnisse oder ein aktualisiertes Lagebild vom UAV zurück an die Bodenstation übertragen. Die Ergebnisbewertung und die Ableitung weiterer Handlungsschritte obliegt dem Operateur.

Die einzelnen Subsysteme des UAVs, wie z. B. das *Missionsmanagement*, das Modul zur *Flugführung*, *Navigation*, *Steuerung*, sowie die *Missionssensoren*, werden in dieser Arbeit als Black Box betrachtet. Für weiterführende Informationen zu deren Funktionsweise wird auf [114, 117] verwiesen. Die interne Struktur des ESPM wird im nachfolgenden Abschnitt 3.3.3 erläutert.

#### 3.3.3 Systemarchitektur des ESPM

Nachdem im vorangegangenen Abschnitt die Integration des ESPM in das System des UAVs betrachtet wurde, wird anknüpfend auf die Systemarchitektur des ESPM eingegangen, die erstmalig in [31] beschrieben wurde. Das ESPM in Abb. 3.3 ist funktionell in eine *Auftragsebene* und eine *Unterstützungsebene* unterteilt. Diese Aufteilung wird durchgeführt, um die Komplexität des Systems zu reduzieren. Die Module der *Unterstützungsebene* führen die von den Modulen in der *Auftragsebene* gegebenen Anweisungen aus. Zu diesen Anweisungen zählen u. a. die Gimbal- und Sensorsteuerung, die Generierung und Bereitstellung von Sensordaten oder die Bearbeitung von Datenbankabfragen.

Die blauen Verbindungspfeile in der *Auftragsebene* (siehe Abb. 3.3) stellen einen Ansatz zur kooperativen und wechselseitigen Kommunikation zwischen den Modulen dar. Die schwarzen Pfeile verdeutlichen dagegen eine hierarchische Kommunikation in Form von präzisen Anweisungen, Steuerbefehlen oder die Übertragung von (Sensor-)Daten zwischen den Modulen.

Die *Auftragsebene* beinhaltet die grundlegenden Funktionsmodule zur Verwaltung und Bearbeitung eines Aufklärungsauftrags. Wenn der Operateur einen Aufklärungsauftrag erteilt, wird die Anforderung über das *Missionsmanagement* an die *Auftragsverwaltung* gesendet, die als Schnittstelle fungiert und die Aufgabenbearbeitung innerhalb des ESPM koordiniert.

Mit der Übertragung des Aufklärungsauftrags an das *Perzeptionsmanagement* werden unterschiedliche Informationen zusammengeführt und ausgewertet. Durch die Analyse der Sensorleistungsmodelle, die in der *Wissensdatenbank* abgelegt sind, wird der Res-

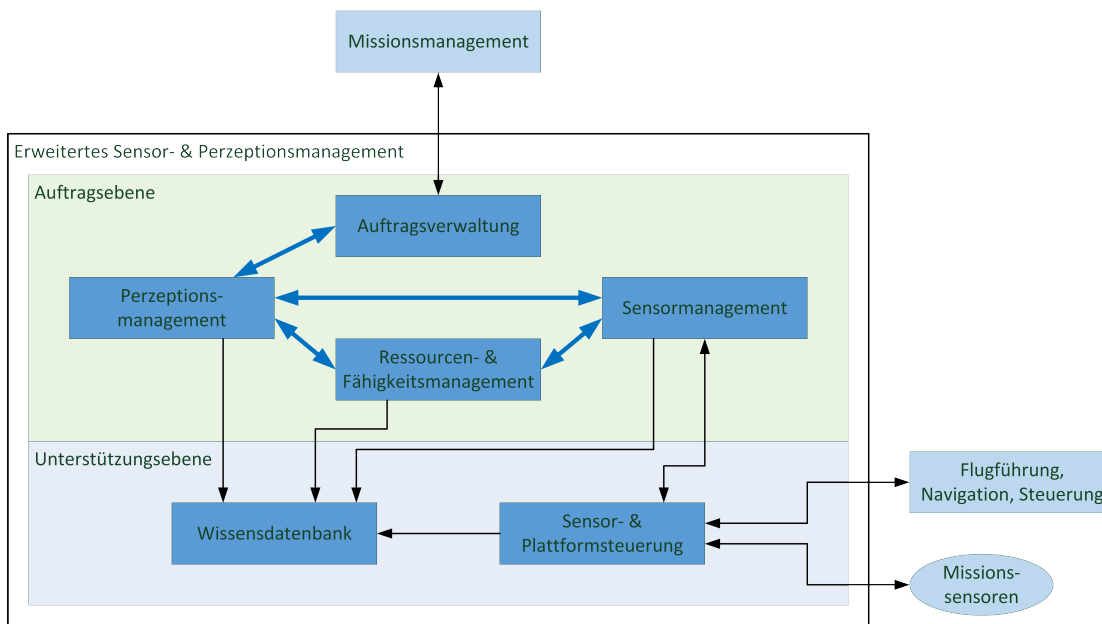


Abbildung 3.3: Systemarchitektur des *Erweiterten Sensor- und Perzeptionsmanagements* einschließlich der Schnittstellen zum Missionsmanagement, zur Flugführung und den Missionssensoren (In Anlehnung an [31])

sourcesbedarf für die Abarbeitung des Auftrags ermittelt. Darüber hinaus stellt die Datenbank dem *Perzeptionsmanagement* auch Algorithmen zur Sensordatenaufbereitung und Sensordatenauswertung (z. B. Objektdetektoren) bereit. Mit Hilfe von Sensorleistungsmodellen wählt das *Perzeptionsmanagement* die Perzeptionskette aus, für die zur Ausführung eines Aufklärungsauftrags, unter den gegebenen Umgebungsbedingungen, die beste Detektionsleistung prognostiziert wird. Für die Beschreibung eines Konzepts zur automatisierten Auswahl der optimalen Perzeptionskette wird auf [50] verwiesen. Des Weiteren werden durch das *Perzeptionsmanagement* die Rechenzeit und Rechenleistung für die zu bearbeitende Aufgabe ermittelt und eingeplant, bevor die Aufgabe an das *Sensormanagement* weitergeleitet wird.

Für den Fall, dass mehrere UAVs gemeinsam einen Aufklärungsauftrag bearbeiten sollen, kann das *Perzeptionsmanagement* das *Ressourcen- und Fähigkeitsmanagement* zur Koordinierung mehrerer, gegebenenfalls heterogen ausgestatteter UAVs für die Multi-UAV-Koordinierung und zur sensororientierten Zusammenarbeit nutzen.

Das *Sensormanagement* wird angefordert, um die räumlich-zeitliche Bewegung des Sensorfootprints im aufzuklärenden Gebiet zu planen. Diese Suchplanung (vgl. Abb. 3.1 in Abschnitt 3.3.1) erfolgt unter Berücksichtigung unterschiedlicher perzeptiver, sensorplatt-

### 3 Konzept und Methodenauswahl

formspezifischer und missionsspezifischer Randbedingungen. Die relevanten Informationen (z. B. Sensorcharakteristik, topographische Daten) die für die Bewegungsplanung erforderlich sind, werden aus der *Wissensdatenbank* abgefragt. Nach erfolgter Planung überträgt das *Sensormanagement* die Informationen zur Sensorbewegung und relevante Sensoreinstellungen (u. a. Sensortyp, Bildwinkel) über die *Sensor- und Plattformsteuerung* an die Missionssensoren.

Die *Sensor- und Plattformsteuerung* steuert die Missionssensoren (z. B. bildgebende EO/IR-Sensorik) und ruft die Sensordaten ab. Darüber hinaus dient es als Schnittstelle zum *Flugführungs-, Navigations- und Steuerungsmodul*, um das UAV entlang der Referenztrajektorie zu führen. Zu diesem Zweck erhält das Modul Anweisungen vom *Sensormanagement* und bestimmt und koordiniert die räumlich-zeitliche Bewegung sowohl der kardanischen Sensorhalterung als auch die des UAVs. Das Modul umfasst Algorithmen, um die optimale Referenztrajektorie für das UAV aus der sensormodellbestimmten Befliegungsplanung zu berechnen, wobei unterschiedliche Randbedingungen (z. B. flugdynamische Begrenzungen) berücksichtigt werden. Diese Fähigkeit ist als *Trajektorienoptimierung* in Abb. 3.1 abgebildet. Die zur Optimierung notwendigen Informationen werden aus der *Wissensdatenbank* abgerufen.

Die *Wissensdatenbank* wiederum enthält Daten, Leistungsmodelle und Dienste, auf die die verschiedenen Module des ESPM nach Bedarf zugreifen können. Die Daten und Modelle (z. B. Flugleistung, Sensoreigenschaften, Sensorleistungsmodelle) werden vorab festgelegt und in der Datenbank gespeichert. Topographische Daten werden von einem geografischen Informationssystem (GIS) bereitgestellt. Darüber hinaus stellt die Datenbank auch ein Lagebild bereit und verfügt über Informationen zum Missionsziel [31].

#### 3.3.4 Zusammenfassung

In Abschnitt 3.3 wurde ein funktionales Konzept zur sensormodellbestimmten Befliegungsplanung vorgestellt. Für die Befliegungsplanung werden perzeptive, sensorplattformspezifische und missionsspezifische Anforderungen und Randbedingungen berücksichtigt, die in Abschnitt 3.2 erläutert wurden. Die perzeptiven Anforderungen werden durch die Sensorleistungsmodelle umgesetzt, wohingegen die sensorplattformspezifischen und missionsspezifischen Randbedingungen mit Hilfe der Trajektorienoptimierung realisiert werden. Damit wurde das *Zwischenziel 1* umgesetzt und erfüllt.

Um die *Forschungsfrage 1.1* (Wie kann das Sensor- & Perzeptionsmanagement zur automatisierten, sensormodellbestimmten Befliegungsplanung befähigt werden?) zu be-

### 3.4 Generierung von Sensorleistungsmodellen

antworten, wurde das *Sensor- und Perzeptionsmanagement* um inhärente und dedizierte Fähigkeiten zur hochautomatisierten sensormodellbestimmten Befliegungsplanung erweitert, woraus das *Erweiterte Sensor- und Perzeptionsmanagement* entwickelt wurde. Darüber hinaus schafft das ESPM alle Voraussetzungen, um die *Anforderungen an die Befliegungsplanung* aus Abschnitt 3.2 umsetzen und erfüllen zu können. Nachfolgend ist der aktuelle Stand zur Bearbeitung der Zwischenziele und Forschungsfragen abgebildet.

Zwischenziel	1	2	3	4	5
Forschungsfrage	1.1	2.1 & 2.2	3.1 & 3.2	—	5.1 - 5.3
Bearbeitung in Kapitel	3	3	3 & 4	4	6

## 3.4 Generierung von Sensorleistungsmodellen

In diesem Kapitel wird auf das Konzept und ein Verfahren zur Realisierung von *Sensorleistungsmodellen* eingegangen, die in dieser Arbeit verwendet werden. Diese Leistungsmodelle bilden die Grundlage, um das *Zwischenziel 2* (Entwicklung und Untersuchung von Sensorleistungsmodellen zur Modellierung der Detektionsleistung in Abhängigkeit ausgewählter perceptiver Einflüsse und Randbedingungen.) aus Abschnitt 1.3 und die damit verbundenen *Forschungsfragen 2.1* und *2.2* aus Abschnitt 2.4 beantworten zu können. Darüber hinaus werden die *Anforderung 2* und Anteile aus *Anforderung 3* aus Abschnitt 3.2 umgesetzt.

Der Aufbau des Kapitels ist wie folgt: zu Beginn werden die Grundlagen und der allgemeine Aufbau der Sensorleistungsmodelle erläutert und deren Zweck aufgezeigt. Im nächsten Schritt wird der Prozess von der Sensordatengenerierung bis hin zur Objektdetektion anhand einer generischen Perzeptionskette dargestellt. Anschließend wird auf *Sensorleistungsmodelle mit Machine Learning basierten Klassifikatoren* (Abschnitt 3.4.3) eingegangen und die Modellierung eines *Sensorleistungsmodells für einen Deep Learning Objektdetektor* (Abschnitt 3.4.4) erläutert. Abschließend erfolgt in Abschnitt 3.4.5 die Vorstellung des Konzepts der *Perception Map*. Diese repräsentiert die lokale Ausprägung der Detektionsleistung in der Bewegungsebene des UAVs unter Anwendung des Sensorleistungsmodells.

### 3.4.1 Grundlagen der Sensorleistungsmodelle

In diesem Abschnitt wird auf die Grundlagen der in dieser Arbeit verwendeten Sensorleistungsmodelle eingegangen. Die Betrachtung erfolgt unabhängig der zu Grunde

### 3 Konzept und Methodenauswahl

liegenden Objektdetektoren bzw. Klassifikatoren. Hierfür wird zunächst der Zweck des Sensorleistungsmodells beschrieben, bevor auf einzelne Kernaspekte eingegangen wird.

Die *Sensorleistungsmodelle* werden verwendet, um eine quantitative Vorhersage der zu erwartenden *Detektionsleistung* für eine spezifische *Perzeptionskette* unter dem Einfluss relevanter *Umgebungsbedingungen* (perzeptive Einflüsse) treffen zu können. Ein Sensorleistungsmodell ist somit eine Abbildungsvorschrift, die ausgesuchte Umgebungsbedingungen auf die zu erwartende Detektionsleistung abbildet. Jedem Sensorleistungsmodell liegt eine spezifische Perzeptionskette zu Grunde, auf die in Abschnitt 3.4.2 eingegangen wird.

Das Prinzip des Sensorleistungsmodells ist in Abb. 3.4 dargestellt. Aus der Positionierung des Sensorfootprints am Boden und der Sensor- bzw. UAV-Position können die atmosphärischen, photogrammetrischen und topographischen Zustände bestimmt werden. Im Sensorleistungsmodell werden diese Umgebungs Zustände als unabhängige Variablen auf die zu erwartende Detektionsleistung als abhängige Variable abgebildet.

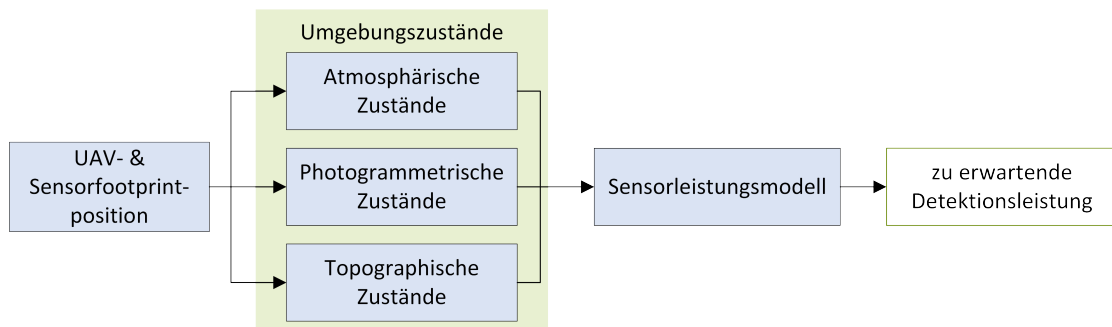


Abbildung 3.4: Das Sensorleistungsmodell bildet ausgesuchte Umgebungs Zustände einer spezifischen Perzeptionskette (nicht abgebildet) auf die zu erwartende Detektionsleistung ab

Für die Modellierung der Abbildungsvorschrift des Sensorleistungsmodells muss der Zusammenhang zwischen den Umgebungs Zuständen und der zu erwartenden Detektionsleistung ermittelt werden. Hierfür werden mit Hilfe einer vorab festgelegten Perzeptionskette Sensordaten erfasst, aufbereitet und eine automatisierte Objektdetektion durchgeführt. Die Bildverarbeitungs Algorithmen eines Sensorleistungsmodells sind für die Detektion spezifischer Objekte, z. B. Fahrzeugen oder Personen am Boden, trainiert und optimiert.

Um die Korrektheit der Objektdetektionen überprüfen und daraus die Detektionsleistung berechnen zu können, muss für die sensorisch erfassten Daten die *Ground Truth* bezüglich der zu detektierenden Objekte ermittelt werden. Dies kann auf die nachfolgenden Arten erfolgen:



### 3.4 Generierung von Sensorleistungsmodellen

- Die Bilddaten werden aus Datensätzen (z. B. der UAVDT-Datensatz [131]) entnommen, deren Bildinhalte bereits gelabelt sind.
- Es werden synthetische Bilddaten genutzt, die mit Hilfe einer Simulationsumgebung erstellt werden (siehe z. B. Hellert [50]). Ein wesentlicher Vorteil der Simulationsumgebung ist, dass die erzeugten Bilddaten und Umgebungszustände einfach variiert und automatisiert gelabelt werden können.
- Die Bilddaten werden experimentell im Realflug generiert und durch einen nachgeschalteten Prozess (manuell) gelabelt. Ein Beispiel findet sich bei Krump & Stütz [132].

Zeitgleich mit der sensorischen Bilddatenerfassung müssen auch die vorherrschenden Umgebungszustände ermittelt und aufgezeichnet werden. Beispielsweise können die atmosphärischen Bedingungen (z. B. Tageszeit, Bedeckungsgrad, Niederschlag, Sichtweite) im Operationsgebiet des UAVs mit Hilfe von Datenbanken oder Expertenwissen abgerufen werden. Für die Ermittlung der topographischen Zustände (u. a. Straßen, Gebäude, Wald, Wasser) innerhalb des Sensorfootprints am Boden kann auf ein Geoinformationssystem zurückgegriffen werden. Der Elevationswinkel und die GSD, beides Bestandteile der photogrammetrischen Zustände, können aus der Position des UAVs relativ zum Sensorfootprint am Boden sowie der Länge der Sichtlinie und dem Bildwinkel berechnet werden (siehe Gleichung (3.5) in Abschnitt 3.4.4.3).

Es kann zusammengefasst werden: für jedes erfasste Sensorbild wird eine automatisierte Objektdetektion durchgeführt und die Korrektheit der Detektion auf Grundlage der Ground Truth bestimmt. Des Weiteren werden für jedes Sensorbild die Umgebungszustände ermittelt und gespeichert. Damit liegen die unabhängigen und abhängigen Variablen für die Ermittlung der Abbildungsvorschrift des Sensorleistungsmodells vor. Aus einer großen Anzahl an Sensorbildern, bei gleichzeitiger Variation der Umgebungszustände, kann eine statistisch relevante Aussage für die prognostizierte Detektionsleistung erstellt werden. In der Praxis hat sich gezeigt, dass eine explizite Abbildungsvorschrift für die Detektionsleistung nicht oder nur näherungsweise ermittelt werden kann. Aus diesem Grund wird beispielsweise in der Arbeit von Hellert [50] ein trainiertes neuronales Netz zur Beschreibung der Abbildungsvorschrift verwendet.

Durch das Sensorleistungsmodell wird die Perzeptionskette als Black Box abstrahiert, wobei die Umgebungszustände die Eingangsgrößen und die zu erwartende Detektionsleistung die Ausgangsgröße verkörpern. Mit Hilfe des Sensorleistungsmodells kann bereits vor der eigentlichen Aufklärung eine Aussage über die zu erwartende Detektionsleistung getroffen werden, falls die Umgebungszustände vorab bekannt sind. Dies ist der Ansatz

### 3 Konzept und Methodenauswahl

der sensormodellbestimmten Befliegungsplanung, bei der eine optimale Referenztrajektorie (Sollwertvorgabe) für den Aufklärungsauftrag geplant wird, um die zu erwartende Detektionsleistung zu steigern. Die Generierung der optimalen Referenztrajektorie erfolgt hierbei bevor die eigentliche Befliegung durchgeführt wird.

#### **Definition der Detektionsleistung**

Die *Detektionsleistung*  $p$  ist in dieser Arbeit definiert als ein quantitatives Maß für die statistische Korrektheit des Ergebnisses (Erwartungswert) der maschinellen Objektdetektion, wobei die Objektdetektion die Lokalisierung und Klassifizierung eines Objekts in den Bilddaten umfasst. Die Detektionsleistung entspricht somit einem Gütemaß zur Bewertung der Korrektheit bzw. Vertrauenswürdigkeit des Messergebnisses. Ihr Wertebereich wird auf das Intervall  $p = [0, 1]$  festgelegt. Eine hohe Detektionsleistung ist gleichbedeutend mit einem hohen Erwartungswert für die Korrektheit der Objektdetektion. Entsprechend ist mit einer niedrigen Detektionsleistung eine geringe Vertrauenswürdigkeit in das Messergebnis verbunden. Die Detektionsleistungen in Abschnitt 3.4.3 und 3.4.4 werden durch unterschiedliche Metriken berechnet, worauf an der entsprechenden Stelle eingegangen wird. Für beide Varianten gelten aber weiterhin die Aussagen dieses Absatzes. Es wird angemerkt, dass die Detektionsleistung in dieser Arbeit auf die Objektdetektion in Einzelbildern bezogen ist, ohne die Berücksichtigung vorangegangener Detektionen.

#### **3.4.2 Perzeptionskette**

Die *Perzeptionskette* umfasst die wesentlichen Hard- und Softwarekomponenten von der Datenerfassung bis zur Datenauswertung und besteht aus einzelnen, aufeinander abgestimmten *Perzeptionsmodulen* die miteinander interagieren [117]. Zu diesen Modulen zählen u. a. die bildgebende Sensorik zur Datenerfassung, Filter zur Datenaufbereitung und Bildverarbeitungsalgorithmen für die automatisierte Objektdetektion. In Abb. 3.5 ist eine Perzeptionskette abstrahiert dargestellt. Die Fähigkeiten der einzelnen Perzeptionsmodule und deren Zusammenwirken in der Perzeptionskette haben einen wesentlichen Einfluss auf die erzielbare Detektionsleistung. Im Folgenden soll eine kurze physikalisch-technische Betrachtung der wesentlichen Module erfolgen, um deren Einfluss auf das Detektionsergebnis besser verstehen zu können.

Die sensorische Datenerfassung einer Perzeptionskette kann beispielsweise durch einen bildgebenden Sensor aus dem elektrooptischen- oder infraroten Spektrum (EO/IR-Sensor) erfolgen. Das Sensorsystem enthält neben dem Sensor eine Optik, die den Inhalt des Sensorfootprints am Boden auf den Sensorchip projiziert. Die Optik beeinflusst die

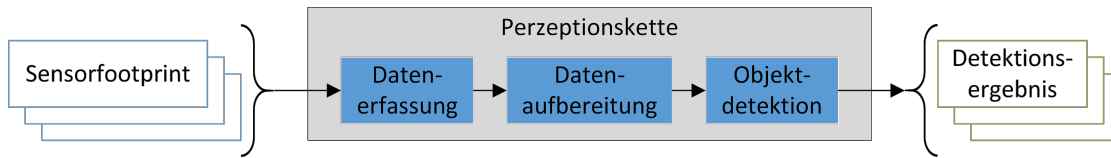


Abbildung 3.5: Abstrahierte Darstellung einer Perzeptionskette mit den unterschiedlichen Perzeptionsmodulen (blau) zur sensorischen Datenerfassung, Datenaufbereitung und Objektdetektion

räumlichen und spektralen Eigenschaften des Sensorsystems und das Sichtfeld. Der Sensor wandelt das optische Signal in ein elektronisches Signal um, wodurch der spektrale Betriebsbereich, die Empfindlichkeit und die Auflösung entscheidend durch den Sensor geprägt werden. Die Empfindlichkeit beschreibt dabei das Signal-Rausch-Verhältnis des Sensors und ist u. a. eine Funktion des Sensormaterials, der Detektorgröße, der spektralen Bandbreite und der Wellenlänge der einfallenden Strahlung [133]. Diese Eigenschaften des Sensorsystems bestimmen maßgeblich die perzeptiven Fähigkeiten und Randbedingungen der Perzeptionskette.

Wird der Modelldatensatz für die Generierung eines Sensorleistungsmodells mit synthetischen Daten erzeugt, entfällt das sensorische System aus Optik und Sensor. In diesem Fall werden die Bildinformationen vollständig durch die verwendete Simulationsumgebung erzeugt und für die weitere Datenauswertung aufbereitet. Die Leistungsfähigkeit der Simulationsumgebung zur Bereitstellung fotorealistischer Bilddaten unter Berücksichtigung relevanter physikalischer Effekte (z. B. Beleuchtung der Szenerie, realistische Topographie) ist wesentlich für die Fragestellung, ob Realdaten durch synthetische Bilddaten substituiert werden können, um damit äquivalente Sensorleistungsmodelle erstellen zu können. Für eine tiefere Untersuchung dieser Fragestellung wird auf [50, 132] und [134] verwiesen.

Ein weiteres Element in der Perzeptionskette betrifft die Aufbereitung der Sensordaten in Vorbereitung auf die Objektdetektion. Die Datenaufbereitung ist spezifisch auf den nachgeschalteten Detektionsalgorithmus angepasst und umfasst beispielsweise Funktionen zur Skalierung, Beschneidung der Bildgröße, Adaption der Auflösung, Korrektur von Verzerrungen und Abbildungsfehlern, oder die Reduktion von Rausch- und Alias-Effekten. Eine detaillierte Erläuterung dieser Funktionen findet sich u. a. in [135, 136].

Nach der Datenaufbereitung erfolgt die Objektdetektion mit Hilfe dedizierter Algorithmen und Verfahren an deren Ende ein Detektionsergebnis ermittelt wird. Die in dieser Arbeit beschriebenen Perzeptionsketten basieren auf validierten Klassifikatoren u. a. aus dem Bereich des Machine Learnings sowie einem auf Deep Learning basierten

### 3 Konzept und Methodenauswahl

Objektdetektor. Auf deren Funktionsweise wird in den nachfolgenden Abschnitten 3.4.3 und 3.4.4 eingegangen.

Jedes einzelne Modul in der Perzeptionskette hat Einfluss auf die Leistungsfähigkeit der Objektdetektion. Da die Einflüsse der einzelnen Module auf das Endergebnis nur schwer zu quantifizieren sind, wird die Perzeptionskette im Gesamten als Black Box betrachtet. Daher wird nur untersucht, welche (Sensor-)Daten erfasst werden, und wie sich diese Daten auf das Detektionsergebnis auswirken.

Unter Verweis auf den vorangegangenen Abschnitt 3.4.1 kann zusammengefasst werden: Das Sensorleistungsmodell abstrahiert die Perzeptionskette als Black Box Modell und bildet die Umgebungszustände auf die Detektionsleistung ab. Eine Veränderung eines Perzeptionsmoduls ändert die Leistungsfähigkeit der gesamten Perzeptionskette und damit auch die Detektionsleistung. Aus diesem Grund muss für jede spezifische Perzeptionskette ein individuelles Sensorleistungsmodell erstellt werden.

#### 3.4.3 Sensorleistungsmodelle mit Machine Learning basierten Klassifikatoren

Nach der Erläuterung der Grundlagen von Sensorleistungsmodellen und der Beschreibung des Funktionsprinzips einer Perzeptionskette in den vorangegangenen Abschnitten, werden nachfolgend Leistungsmodelle vorgestellt, deren Perzeptionsketten *Machine Learning* basierte *Klassifikatoren* umfassen. Die Abbildungsvorschrift zur Modellierung der Detektionsleistung wird bei diesen Modellen durch *künstliche neuronale Netze* (KNN) realisiert.

Hellert beschreibt in seiner Arbeit [50] ein verallgemeinertes Konzept zur Erzeugung von Sensorleistungsmodellen am Beispiel der luftgestützten Fahrzeugdetektion, auf das in Abschnitt 2.1.2 bereits kurz eingegangen wurde. Hierfür wurden in einer Simulationsumgebung synthetische Modelldatensätze mit visuellen und Infrarotbildern von Fahrzeugen in verschiedenen Umgebungsbedingungen erzeugt. Diese Datensätze enthalten zusätzlich zur Fahrzeugposition im Bild auch den zugehörigen Umgebungszustandsvektor, der die Ground Truth der simulierten atmosphärischen, photogrammetrischen und topographischen Bedingungen zum Zeitpunkt der Bilddatenerfassung umfasst. Der Umgebungszustandsvektor kann grundsätzlich alle messbaren Zustände beinhalten, die die sensorische Datenerfassung beeinflussen. Für die Sensorleistungsmodelle von Hellert ist der Umgebungszustandsvektor wie folgt aufgebaut:

- Atmosphärische Zustände: Bewölkung, Nebel, Niederschlag und Helligkeit, definiert durch die Tageszeit und den Monat.

### 3.4 Generierung von Sensorleistungsmodellen

- Topographische Zustände: Landbedeckung (Straßen, Ebene, Gewässer, Vegetation und Gebäude) und die Oberflächenunebenheit innerhalb des Sensorfootprints.
- Photogrammetrische Zustände: GSD und Elevationswinkel des Sensors.

Für jedes Datenset aus den Modelldatensätzen werden automatisiert Bereiche identifiziert und durch Bounding Boxen definiert, in denen sich Fahrzeuge befinden könnten (Lokalisierung). Im nächsten Schritt wird mit Hilfe ausgewählter Objektklassifikatoren geprüft, ob sich in diesen Bounding Boxen Fahrzeuge befinden. Hierzu wird für jede Bounding Box eine binäre Klassifikation (Fahrzeug erkannt bzw. kein Fahrzeug erkannt) durchgeführt. Das eigentliche Detektionsergebnis ist das Resultat aus den Ergebnissen der Lokalisierung und Klassifizierung. Aus dem Vergleich des Detektionsergebnisses mit der Ground Truth kann die Detektionsleistung berechnet werden. Die Ground Truth beschreibt hierbei die tatsächliche Position des Fahrzeugs im Sensorbild.

Bei den Sensorleistungsmodellen wird der Objektzustand (z. B. Fahrzeugtyp, Farbe, Ausrichtung) nicht explizit in dem Umgebungszustandsvektor abgebildet. Stattdessen wird der Objektzustand variiert (z. B. durch Variation des Fahrzeugtyps und der Ausrichtung) unter Beibehaltung der messbaren Umgebungszustände (siehe auch die Erläuterungen in Abschnitt 3.1). Als Konsequenz resultiert aus der Variation des Objektzustands ein statistischer Durchschnittswert für die Detektionsleistung.

Die Objektklassifikatoren, die von Hellert für die automatisierte luftgestützte Fahrzeugklassifizierung verwendet werden, sind nachfolgend aufgelistet:

- Die *Classification Cascade* (CC) [66] zählt zu den Machine Learning Verfahren.
- Das *Deformable Part Model* (DPM) [67] kann ebenfalls dem Bereich des Machine Learnings zugeordnet werden.
- Das *Template Matching* (TM) [68] ist ein Verfahren aus der gleichnamigen Domäne.
- Das *Binary Large Object* (BLOB) [69] zählt zu den wissensbasierten Methoden.

#### Modellierung des Leistungsmodells durch ein KNN

Jeder dieser vier Klassifikatoren ist Bestandteil einer individuellen Perzeptionskette. Für jede dieser Perzeptionsketten hat Hellert jeweils ein separates Sensorleistungsmodell auf der Basis eines trainierten KNNs modelliert. In Abb. 3.6 ist die Topologie für ein Leistungsmodell auf Basis eines KNNs dargestellt.

Das KNN beschreibt die implizite und nicht näher bekannte Abbildungsvorschrift (Stichwort: Black Box), womit für einen spezifischen Umgebungszustandsvektor (unabhängige Variable) die zu erwartende Detektionsleistung (abhängige Variable) prognostiziert werden kann. Die Topologie des neuronalen Netzes besteht jeweils aus einer Eingangs- und

### 3 Konzept und Methodenauswahl

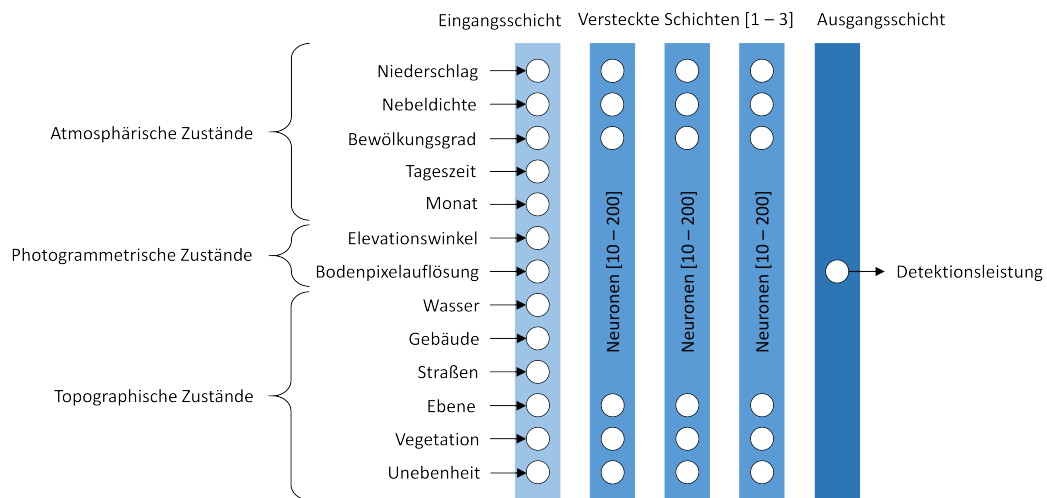


Abbildung 3.6: Darstellung eines Sensorleistungsmodells auf Basis eines neuronalen Netzes mit den atmosphärischen, photogrammetrischen und topographischen Zuständen als Eingangsgrößen und der Detektionsleistung als Ausgangsgröße (In Anlehnung an [50])

Ausgangsschicht sowie aus zwei bzw. drei versteckte Schichten, abhängig vom zu Grunde liegenden Klassifikator. Die atmosphärischen, photogrammetrischen und topographischen Zustände werden zum Umgebungszustandsvektor zusammengefasst und bilden die Eingangsgröße für das Netz. Der Umgebungszustandsvektor propagiert durch die einzelnen Schichten des Netzes, woraus letztendlich die Detektionsleistung in der Ausgangsschicht resultiert. Für weiterführende Informationen zur Entwicklung, Implementierung und Validierung dieser Leistungsmodelle wird auf [30, 50] verwiesen.

#### Metrik zur Berechnung der Detektionsleistung

Bei den Sensorleistungsmodellen von Hellert wird die Detektionsleistung über die Metrik  $F_1$ -Score bestimmt. Diese Metrik ist ein Kombinationsmaß aus dem harmonischen Mittel der *Genauigkeit* (engl.: precision) und der *Trefferquote* (engl.: recall) der Objektklassifikation. In der Praxis werden binäre Objektdetektoren häufig mit Hilfe der *Wahrheits-* oder *Konfusionsmatrix* validiert. Für eine Menge mit *positiver* (P) Merkmalsausprägung, welche die zu detektierenden Objekte (z. B. Fahrzeuge) repräsentieren und einer Menge mit *negativer* (N) Merkmalsausprägung (z. B. alles andere, außer Fahrzeuge) wird ausgedrückt, wie viele positive und negative Merkmalsausprägungen *richtig* (engl.: true, T) bzw. *falsch* (engl.: false, F) klassifiziert wurden. Hieraus können die folgenden vier Kombinationen abgeleitet werden:

### 3.4 Generierung von Sensorleistungsmodellen

- Richtig positiv (TP): die positive Merkmalsausprägung wurde korrekt klassifiziert
- Richtig negativ (TN): die negative Merkmalsausprägung wurde korrekt klassifiziert
- Falsch positiv (FP): die positive Merkmalsausprägung wurde falsch klassifiziert
- Falsch negativ (FN): die negative Merkmalsausprägung wurde falsch klassifiziert

Die Berechnung der Genauigkeit (precision) ist in Gleichung (3.1) dargestellt und gibt an, wie präzise der Detektionsalgorithmus ist. Damit wird ermittelt, wie viele der insgesamt erkannten Objekte (z. B. Fahrzeuge) tatsächlich dieser Objektklasse angehören. Die Genauigkeit beschreibt somit das Verhältnis aus der Anzahl der richtig positiv erkannten Objekte und der Gesamtzahl der detektierten Objekte bezüglich dieser Objektklasse.

$$precision = \frac{TP}{TP + FP} \quad (3.1)$$

Die Gleichung zur Ermittlung der Trefferquote (recall) ist in Gleichung (3.2) aufgetragen. Sie wird berechnet aus dem Verhältnis der Anzahl der richtig positiv erkannten Objekte einer spezifischen Objektklasse und der Gesamtzahl der Ground Truth Objekte im Bild mit eben dieser Objektklasse. Mit der Trefferquote wird angegeben, wie gut der Detektionsalgorithmus bestimmte Objektklassen in den Bilddaten erkennen kann.

$$recall = \frac{TP}{TP + FN} \quad (3.2)$$

Unter Nutzung dieser beiden Gleichungen kann anschließend der  $F_1$ -Score berechnet werden:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (3.3)$$

#### 3.4.4 Sensorleistungsmodell für einen Deep Learning Objektdetektor

Nachdem im vorangegangenen Abschnitt Sensorleistungsmodelle für unterschiedliche Objektklassifikatoren vorgestellt wurden, wird in diesem Abschnitt ein Konzept und die Generierung eines Sensorleistungsmodells für einen *Deep Learning* basierten Objektdetektor beschrieben. Damit soll die *Forschungsfrage 2.1* (Wie kann ein Sensorleistungsmodell für eine Perzeptionskette mit Deep Learning basiertem Objektdetektor modelliert werden, um die Detektionsleistung bezüglich ausgewählter perzeptiver Einflüsse abbilden zu können?) aus Abschnitt 2.4 beantwortet werden.

Das Leistungsmodell mit einem Deep Learning basierten Objektdetektor wurde erstmals in der Arbeit von Zwick, Gerdt & Stütz [33] vorgestellt und wird im Folgenden detaillierter erläutert. Das Sensorleistungsmodell bildet eine Perzeptionskette ab, welche

### 3 Konzept und Methodenauswahl

einen *YOLOv3-Objektdetektor*<sup>5</sup> verwendet, der von Redmon & Farhadi [23] entwickelt wurde. Dieser Detektor basiert auf einem trainierten und gefalteten neuronalen Netzwerk (engl.: Convolutional Neural Network, CNN). Das Ziel in diesem Abschnitt ist es, ein Sensorleistungsmodell für eine Perzeptionskette zu generieren, die einen leistungsfähigen und weit verbreiteten Deep Learning basierten Objektdetektor zur luftgestützten Fahrzeugdetektion umfasst. Hierfür wird auf den trainierten YOLOv3-Detektor aus der Arbeit von Krump, Russ & Stütz [137] zurückgegriffen.

#### 3.4.4.1 YOLOv3-Objektdetektor

Der YOLO-Objektdetektor wurde seit der ersten Version im Jahr 2015 [138] kontinuierlich weiterentwickelt, bis hin zur derzeit neuesten Version: YOLOv7 [24]. Obwohl zwischenzeitlich neuere YOLO-Versionen verfügbar sind, wird das Leistungsmodell für die Version *YOLOv3* entwickelt, welche im Jahr 2018 vorgestellt wurde. Ursächlich hierfür ist, dass das Leistungsmodell auf dem Datensatz von [132] aufbaut, für den die Objektdetektion mit diesem Objektdetektor bereits durchgeführt wurde. Das Prinzip des Leistungsmodells ist allerdings nicht auf diese spezifische Detektorversion beschränkt, worauf im weiteren Verlauf noch eingegangen wird.

Für die Wahl des YOLOv3-Objektdetektors gibt es mehrere Gründe. Eine Auswahl ist im Folgenden zusammengestellt:

- Obwohl zwischenzeitlich aktuellere Versionen des YOLO-Objektdetektors verfügbar sind, ist YOLOv3 weiterhin ein effizienter und leistungsfähiger Objektdetektor [139, 140].
- Die freie Codebasis des YOLOv3-Detektors und die Verfügbarkeit öffentlicher und annotierter Datensätze (z. B. der UAVDT-Datensatz [131]) hat zu einer großen Verbreitung dieses Detektors beigetragen.
- Die Implementierung von YOLOv3 ermöglicht die bildbasierte Objekterkennung in Echtzeit auf marktverfügbarer Hardware [23], was für den Einsatz an Bord eines UAVs von Vorteil ist.

Der YOLOv3-Objektdetektor ist eine Weiterentwicklung des YOLO-Detektors von Redmon et al. [138] aus dem Jahr 2015. Das zu Grunde liegende Funktionsprinzip beider Varianten ist ähnlich, weshalb das Funktionsprinzip anhand der Erläuterung in [138] im Folgenden kurz zusammengefasst wird:

---

<sup>5</sup>*YOLO* ist das Akronym für „you only look once“ und weist darauf hin, dass die Bildinformation nur einmal durch das CNN ausgewertet werden muss, um zu bestimmen, welche Objekte vorhanden sind und wo sie sich im Bild befinden.



### 3.4 Generierung von Sensorleistungsmodellen

Dem YOLO-Objektdetektor liegt ein CNN zu Grunde, welches in einem Durchgang mehrere Begrenzungsrahmen (engl.: Bounding Boxen) definiert, den Inhalt dieser Bounding Boxen klassifiziert und einen Zuverlässigkeitswert (Konfidenzwert) für die gemachte Klassifizierung ermittelt. Hierfür unterteilt der Detektor das zu analysierende Bild in ein Gitter aus einzelnen Zellen. Für jede Zelle werden mehrere Bounding Boxen festgelegt, deren Inhalt einer Objektklasse (z. B. Fahrzeuge) zugeordnet, und die dazugehörige Konfidenz bestimmt. Diese drückt aus, wie zuverlässig der Inhalt der Bounding Box einer bestimmten Objektklasse zugewiesen werden kann.

Das CNN des YOLO-Detektors ist aus 24 gefalteten Schichten aufgebaut, die für die Extraktion der Bildmerkmale zuständig sind. Diesen Schichten sind zwei vollständig verknüpfte Schichten nachgeschaltet, deren Aufgabe es ist, den Konfidenzwert der Objektklassifizierung und die Position der Bounding Boxen in den Bilddaten zu bestimmen.

Für das Training des YOLO-Algorithmus wird der gesamte Bildinhalt genutzt. Dadurch wird die implizite kontextuelle Information der Objektklassen und deren Erscheinungsform gespeichert. Auf diesem Weg lernt das CNN generalisierbare Repräsentationen von Objekten. YOLOv3 ist eine Weiterentwicklung des YOLO-Netzwerks bei dem u. a. ein tieferes Netz von Merkmalsdetektoren realisiert, die Genauigkeit verbessert und die Berechnungsleistung gesteigert wurden [131]. Für eine detailliertere Beschreibung zum Aufbau und zum Training des YOLO-Detektors wird auf [138] verwiesen.

#### 3.4.4.2 Vorgegangene Arbeiten

Krump, Russ & Stütz beschreiben in [137], wie ein YOLOv3-Objektdetektor für die luftgestützte Fahrzeugdetektion mit Hilfe der Trainingsdaten aus dem UAVDT-Datensatz [131] trainiert wurde. Dieser Datensatz wurde gewählt, da er eine große Anzahl an Bildern umfasst und eine hohe Variation bezüglich der Flughöhe, des Blickwinkels und der Umgebungszustände (z. B. Helligkeit, Sichtbedingungen) aufweist. In Abb. 3.7 sind beispielhaft drei Bilder mit Annotation (Bounding Boxen einschließlich Objektklassifizierung) aus dem UAVDT-Datensatz abgebildet.

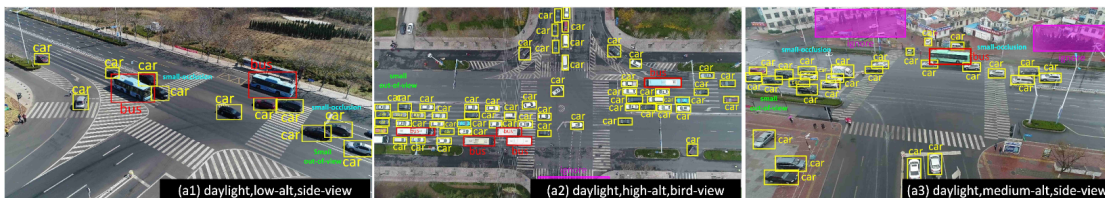


Abbildung 3.7: Beispielbilder mit Annotationen aus dem UAVDT-Datensatz [131]

### 3 Konzept und Methodenauswahl

Auf Grund der Tatsache, dass es derzeit nur sehr wenige öffentlich verfügbare Modell Datensätze zur luftgestützten Fahrzeugdetektion gibt, bei denen die photogrammetrischen Zustände, insbesondere die GSD und der Elevationswinkel des Sensors, annotiert sind, haben Krump & Stütz in [132] einen eigenen Modell Datensatz zur luftgestützten Fahrzeugdetektion erstellt, der nachfolgend als *K&S-Datensatz* bezeichnet wird. Neben den atmosphärischen Zuständen wurden zusätzlich der vertikale und horizontale Abstand zwischen dem UAV und dem Fahrzeug am Boden erfasst und im Datensatz gespeichert. Aus diesen Geometriebeziehungen können die GSD und der Elevationswinkel des Sensors berechnet werden.

Der trainierte YOLOv3-Detektor und der K&S-Datensatz sind die Grundlage für die Generierung des Sensorleistungsmodells mit Deep Learning basierendem Objektdetektor. Der K&S-Datensatz wird gewählt, da die annotierten photogrammetrischen Zustände benötigt werden, um den Einfluss der GSD und des Elevationswinkels im Leistungsmodell auf die Detektionsleistung abbilden zu können. Bevor auf die eigentliche Erstellung des Leistungsmodells in Abschnitt 3.4.4.3 eingegangen wird, erfolgt zunächst eine Beschreibung, wie der K&S-Datensatz erstellt und welche Umgebungszustände hierbei erfasst wurden.

Für die Generierung des K&S-Datensatzes haben Krump & Stütz [132] Luftbildaufnahmen von unterschiedlichen Fahrzeugen, unter Variation der Flughöhe (von 15 m bis 90 m) und des horizontalen Abstands (von 0 m bis 80 m), erstellt. In Abb. 3.8 ist das Vorgehen für die Erzeugung der Luftbildaufnahmen dargestellt. Die Luftbildaufnahmen wurden mit einem Multikopter-UAV mit kardanischem aufgehängtem EO/IR-Sensorsystem ZENMUSE XT2 [141] der Firma DJI erstellt. Über die Telemetrie des UAVs wurde die Sensorposition für jedes einzelne Bild erfasst und diesem zugeordnet [132]. Der elektrooptische Sensor des ZENMUSE XT2 besteht aus einem CMOS Sensor mit einer Auflösung von 4000 x 3000 Pixel. Die Optik hat eine feste Brennweite von 8 mm und einen Bildwinkel von  $57,12^\circ \times 42,44^\circ$ . Die feste Brennweite hat zur Folge, dass sich die GSD verändert, sobald der Abstand zwischen dem UAV und dem Fahrzeug variiert wird.

Um eine große Variation der vorherrschenden Umgebungszustände (bestehend aus atmosphärischen und photogrammetrischen Zuständen) zu erreichen, wurden die Aufnahmen zu verschiedenen Tageszeiten und bei unterschiedlichen Wetterbedingungen aufgenommen. Des Weiteren wurden für den K&S-Datensatz drei verschiedene Fahrzeuge auf unterschiedlichem Fahrbahnuntergrund und vor variierenden Hintergründen erfasst. Die Fahrzeugorientierung wurde verändert, indem das UAV auf einer halbkreisförmigen Bahn, in  $20^\circ$ -Schritten, um das Fahrzeug positioniert wurde. Die halbkreisförmige Anord-

### 3.4 Generierung von Sensorleistungsmodellen

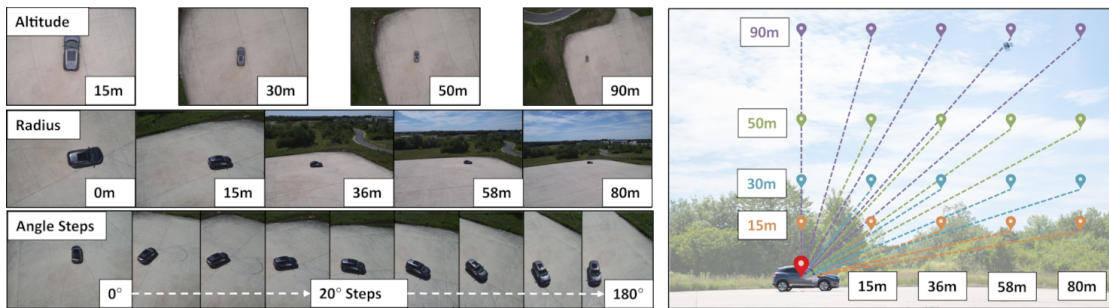


Abbildung 3.8: Links: Erzeugung der Luftbildaufnahmen unter Variation der UAV-Position durch Veränderung der Flughöhe (Altitude) und des horizontalen Abstands (Radius). Die unterschiedliche Ausrichtung des Fahrzeugs wird durch die unterschiedliche Positionierung des UAVs auf einer Kreisbahn um das Objekt erzielt. Im rechten Bild sind die horizontalen und vertikalen Abstände aufgetragen [132].

nung der Erfassungspunkte stellt sicher, dass alle Objektausrichtungen in gleicher Anzahl und ohne Duplikate aufgenommen werden [132].

Der K&S-Datensatz umfasst insgesamt ca. 3300 Luftbildaufnahmen. Für jedes Bild wurde der vertikale und horizontale Abstand zwischen dem UAV und dem Fahrzeug am Boden erfasst und gespeichert. Darüber hinaus wurden für jedes Bild die atmosphärischen Zustände, der Objektzustand sowie die Bounding Box des Fahrzeugs manuell annotiert. Die Menge der *atmosphärischen Zustände* setzt sich wie folgt zusammen:  $\{Jahreszeit, Tageszeit, Bewölkungsgrad, Sicht, Untergrund, Reflexionen\}$ . Diese Zustände beeinflussen die optische Erscheinung der Szenerie, beispielsweise durch den Sonnenstand, die Helligkeit, den Schattenwurf, oder durch Reflexionen auf Grund von Nässe. Demgegenüber werden der *vertikale Abstand* und der *horizontale Abstand* zwischen dem UAV und dem Fahrzeug den *photogrammetrischen Zuständen* zugeordnet. Die *Fahrzeugbeleuchtung* (Scheinwerfer, Rücklichter) wird dem *Objektzustand* zugeschrieben, da dieser einen Zustand des Beobachtungsobjekts (das Fahrzeug) beschreibt. Es wird angemerkt, dass die Nomenklatur der Umgebungs- und Objektzustände für diese Arbeit angepasst wurden und von den Bezeichnungen in [132] abweichen. Die eben genannten Zustände und deren Ausprägungen sind in Tabelle 3.3 aufgelistet.

In Abb. 3.9 ist die Verteilung einer Auswahl der Umgebungs- und Objektzustände im Modelldatensatz abgebildet.

### 3 Konzept und Methodenauswahl

Tabelle 3.3: Zusammenstellung der Umgebungs- und Objektzustände für den K&S-Datensatz aus [132]

Umgebungs- und Objektzustände	Ausprägung
Jahreszeit	Sommer, Herbst
Tageszeit	Tag, Nacht
Bewölkungsgrad	sonnig, bedeckt
Sicht	neblig, klar
Untergrund	nass, trocken
Reflexionen	ja, nein
Fahrzeugbeleuchtung	an, aus
vertikaler Abstand	[15 m, 90 m]
horizontaler Abstand	[0 m, 80 m]

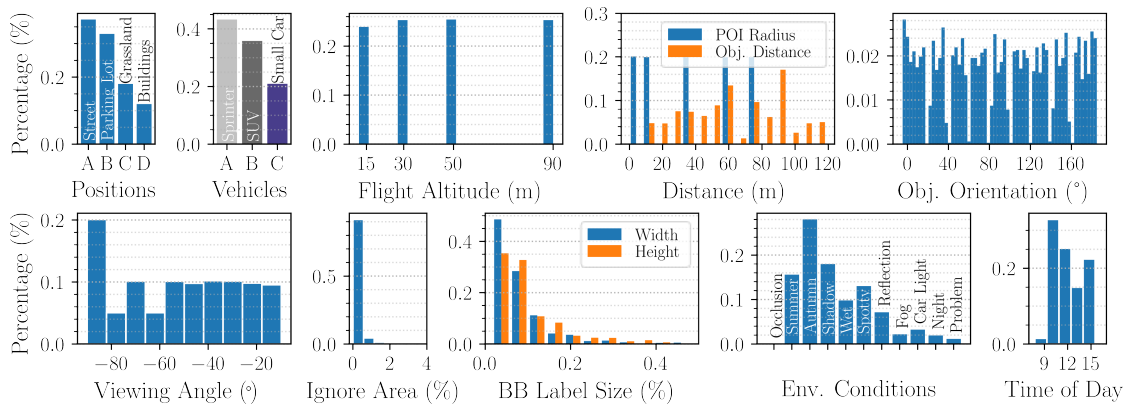


Abbildung 3.9: Übersicht zur Verteilung der Umgebungs- und Objektzustände im Modelldatensatz [132]

#### 3.4.4.3 Modellierung des Sensorleistungsmodells

Nachdem in den vorangegangenen Abschnitten der YOLOv3-Objektdetektor und der K&S-Datensatz beschrieben wurden, wird nachfolgend erläutert, wie darauf aufbauend das Sensorleistungsmodell mit Deep Learning basierendem Objektdetektor entwickelt wurde. Das Verfahren zur Generierung des Leistungsmodells wurde erstmals in Zwick, Gerdts & Stütz [33] vorgestellt und ist in Abb. 3.10 vereinfacht dargestellt.

Das zu entwickelnde Sensorleistungsmodell soll die perceptiven Eigenschaften und Fähigkeiten der Perzeptionskette widerspiegeln, die Krump & Stütz in [132] für die Erzeugung ihres K&S-Datensatzes verwendet haben. Hierbei erfolgte die Sensordatener-

### 3.4 Generierung von Sensorleistungsmodellen

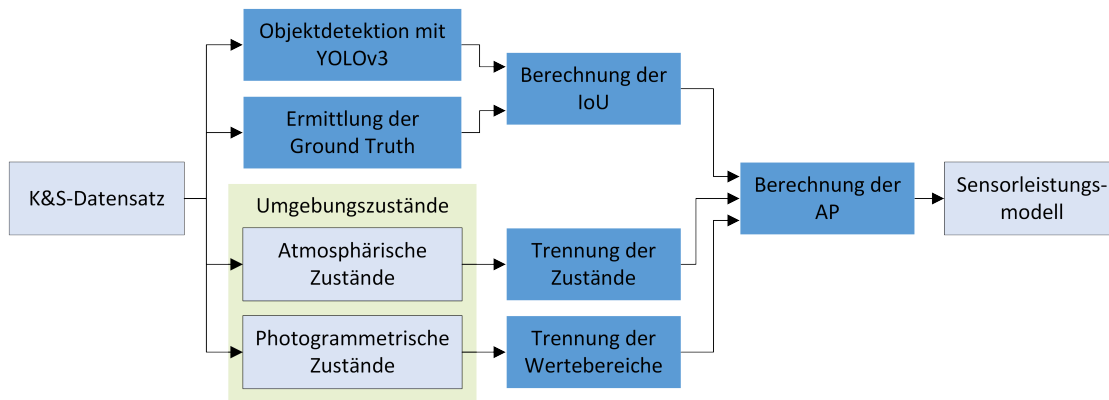


Abbildung 3.10: Verfahren zur Erstellung des Sensorleistungsmodells unter Verwendung eines YOLOv3-Objektdetektors. Der K&S-Datensatz, die Umgebungszustände und das Leistungsmodell sind hellblau dargestellt, wohingegen Aktionen dunkelblau markiert sind. Grün umrandet sind die Umgebungszustände, die für dieses Leistungsmodell genutzt werden.

fassung durch den elektrooptischen Sensor des ZENMUSE XT2 und die Objektdetektion wurde mit dem trainierten YOLOv3-Objektdetektor aus [137] durchgeführt.

Das Vorgehen zur Entwicklung des Sensorleistungsmodells wird nachfolgend in mehrere Prozessschritte unterteilt, die in Abb. 3.10 dargestellt sind: Im ersten Schritt wird aus den annotierten Daten im K&S-Datensatz die Ground Truth der Bounding Box für die Fahrzeugposition extrahiert. Des Weiteren wird mit Hilfe des YOLOv3-Objektdetektors für jedes Bild eine Objektdetektion durchgeführt, wobei kein, ein, oder mehrere Fahrzeuge detektiert werden. Zu jeder Detektion wird durch den Detektionsalgorithmus eine Bounding Box ermittelt, die die Position des detektierten Fahrzeugs im Bild beschreibt.

Auf dieser Grundlage wird das Überlappungsverhältnis (engl.: Intersection over Union, IoU) zwischen der Ground Truth Bounding Box  $BB_{GT}$  und der Bounding Box der Detektion  $BB_{Det}$  ermittelt. Die  $IoU \in [0, 1]$  ist ein quantitatives Maß für die flächige Überlappung zwischen zwei Bounding Boxen derselben Objektklasse und wird aus dem Verhältnis der Schnittmenge und der Vereinigungsmenge der beiden Boxen gebildet:

$$IoU = \frac{|BB_{GT} \cap BB_{Det}|}{|BB_{GT} \cup BB_{Det}|} \quad (3.4)$$

Die IoU ist die am weitesten verbreitete Metrik im Bereich der Objektdetektion, um die Genauigkeit einer Lokalisierung zu bewerten [142]. In der Arbeit von Padilla et al. [143] werden unterschiedliche Metriken aus verschiedenen Wettbewerben zur Objektdetektion vorgestellt und verglichen. Es hat sich gezeigt, dass der Schwellwert der IoU je nach

### 3 Konzept und Methodenauswahl

verwendeter Metrik unterschiedlich festgelegt wird. Häufig verwendete Schwellwerte sind 0,5 und 0,75. In dieser Arbeit wird für die IoU der Schwellwert auf 0,5 festgelegt. Daraus folgt, dass eine Detektion als *richtig positiv* (TP) bewertet wird, wenn dieser Schwellwert erreicht oder überschritten wird. In Abb. 3.10 ist der Vorgang zur Berechnung der IoU als blau gefärbter Block dargestellt.

Mit Hilfe der IoU wird für jedes Bild im Datensatz ermittelt, ob das Ergebnis der Objektdetektion *richtig positiv* (TP), *falsch positiv* (FP) oder *falsch negativ* (FN) ist. Hierfür sind nachfolgend die vier möglichen Kombinationen der Konfusionsmatrix zusammengefasst:

- Eine Detektion wird als *richtig positiv* (TP) klassifiziert, wenn die IoU einem vorab definierten Schwellwert gleicht oder diesen überschreitet.
- Umgekehrt wird eine Detektion als *falsch positiv* (FP) eingestuft, wenn die resultierende IoU diesen Schwellwert unterschreitet.
- Ein *falsch negatives* (FN) Ergebnis tritt ein, wenn für ein Ground Truth Objekt keine entsprechende Detektion erfolgt und somit auch keine Bounding Box generiert wird.
- Die Bedingung *richtig negativ* (TN) kann nicht eintreten. Dies wären alle möglichen Bounding Boxen für Objekte, die korrekterweise nicht erkannt werden.

#### Berechnung der GSD und des Elevationswinkels

Im nächsten Schritt werden für jedes Bild aus dem K&S-Datensatz die GSD und der Elevationswinkel des Sensors berechnet, die beide Teil der photogrammetrischen Zustände sind. Hierfür werden der vertikale und horizontale Abstand des UAVs aus den Bilddaten herangezogen. Der Elevationswinkel  $\alpha \in [0^\circ, 90^\circ]$  beschreibt den Winkel zwischen der Sichtlinie des Sensors und der horizontalen Flugbahn des UAVs (siehe Abb. 3.11). Aus einem Elevationswinkel von  $0^\circ$  folgt, dass die Sichtlinie des Sensors waagrecht verläuft und auf den Horizont gerichtet ist. Demgegenüber resultiert aus einem Elevationswinkel von  $90^\circ$ , dass der Sensor und dessen Sichtlinie senkrecht nach unten ausgerichtet sind.

Die Berechnung der GSD ist in Gleichung (3.5) beschrieben und in Abb. 3.12 sind die Geometriebeziehungen abgebildet. Um die GSD ermitteln zu können, wird vorab aus dem vertikalen und horizontalen Abstand die Länge der Sichtlinie  $s_{LoS}$  (engl.: Line of Sight, LoS) bestimmt. Die GSD kann auf unterschiedliche Arten berechnet werden, beispielsweise aus der Sichtlinie, der Brennweite  $f$  und dem Pixelabstand  $p_{Pixel}$  des Sensors. Alternativ kann die GSD mit Hilfe des Bildwinkels  $\beta$ , der Sichtlinie und der

### 3.4 Generierung von Sensorleistungsmodellen

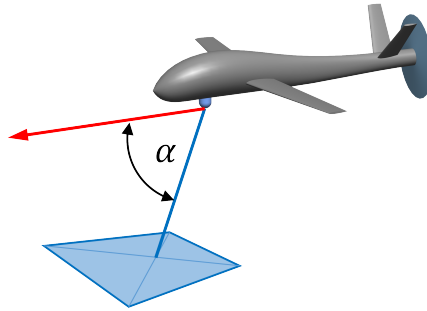


Abbildung 3.11: Der Elevationswinkel  $\alpha$  wird zwischen der horizontalen Flugbahn (rot) und der Sichtlinie (blau) aufgespannt. Die Sichtlinie ist die direkte Verbindung zwischen dem bildgebenden Sensor des UAVs und dem Mittelpunkt des Sensorfootprints am Boden

Anzahl der Sensorpixel<sup>6</sup>  $n_{Pixel}$  ermittelt werden:

$$GSD = \frac{p_{Pixel}}{f} \cdot s_{LoS} = \frac{2 \cdot s_{LoS} \cdot \tan(\beta/2)}{n_{Pixel}} \quad (3.5)$$

Es wird vereinfachend angenommen, dass die GSD im gesamten Sensorfootprint denselben Wert annimmt. Zudem werden geometrische Verzerrungen der GSD, die aus einem Elevationswinkel  $\alpha < 90^\circ$  resultieren, vernachlässigt.

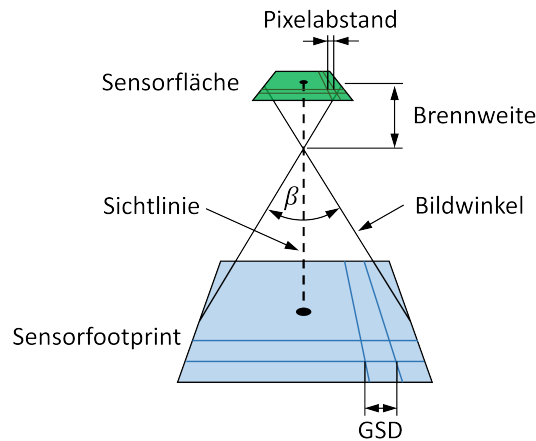


Abbildung 3.12: Darstellung der Geometriebeziehungen zwischen Sensorfläche (grün) und Sensorfootprint (hellblau) für die Berechnung der GSD

<sup>6</sup>Die Anzahl der Sensorpixel bezieht sich hierbei auf die Breite (oder Höhe) des Sensors und kann aus der Sensorauflösung entnommen werden, wie z. B.  $1920 \times 1080$  Pixel (Breite x Höhe) für einen Full-HD Sensor. Für die Berechnung der GSD ist entscheidend, dass der Bildwinkel ebenfalls auf die Breite (oder Höhe) des Sensors bezogen ist.

### Aufteilung des Datensatzes durch Trennung der Zustände und Wertebereiche

Im nächsten Schritt wird der K&S-Datensatz in Einzeldatensätze aufgeteilt, wobei jeder Einzeldatensatz eine individuelle Merkmalsausprägung der atmosphärischen Zustände in sich vereinigt. Dieser Schritt wurde in Abb. 3.10 als *Trennung der Zustände* bezeichnet. In Abb. 3.13 sind die Aufteilung, und die daraus resultierenden neun Einzeldatensätze, grafisch dargestellt. Auf der Grundlage, dass Reflexionen nur bei nassem Untergrund auftreten können und dadurch, dass diese selten in den Daten annotiert sind, wird der Zustand *Reflexion* mit der Ausprägung *ja* dem Zustand *Untergrund* und *nass* zugeordnet. Aus diesem Grund ist der Zustand *Reflexion* in Abb. 3.13 nicht explizit aufgetragen.

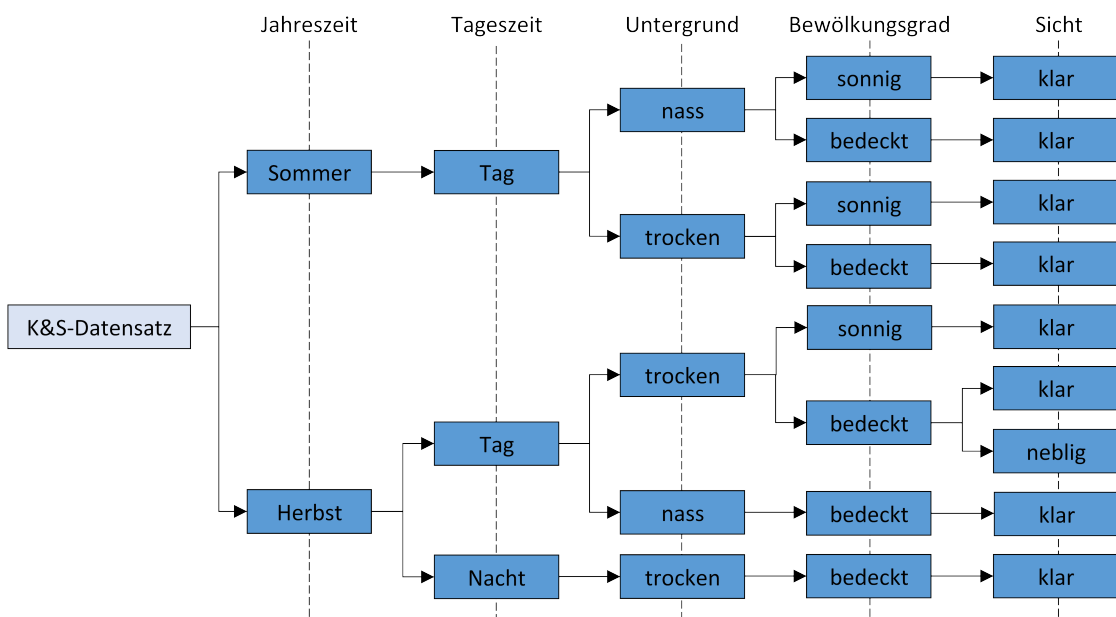


Abbildung 3.13: Darstellung der atmosphärischen Zustände und deren Ausprägungen (blau) aus dem K&S-Datensatz [132]. Fehlende Äste in der Baumstruktur verdeutlichen, dass für die jeweiligen Ausprägungen keine Daten vorliegen

Aus der grafischen Darstellung der atmosphärischen Zustände (z. B. die Jahreszeit) und deren komplementären Ausprägungen (Sommer, Herbst) geht hervor, dass die Baumstruktur nicht vollständig ist. Dies wird damit erklärt, dass nicht für alle Kombinationen Bilddaten generiert wurden. So wurden beispielsweise im Sommer keine Daten in der Nacht aufgenommen.

Im nächsten Arbeitsschritt wird der K&S-Datensatz bezüglich der photogrammetrischen Zustände aufgeteilt. Dazu wird der Wertebereich der GSD und der Wertebereich des Elevationswinkels in jeweils vier Intervalle unterteilt. Die Intervallgrößen werden so



gewählt, dass jedem Intervall etwa die gleiche Anzahl an Bildern zugeordnet werden kann. In Abb. 3.10 wurde dieser Vorgang als *Trennung der Wertebereiche* bezeichnet.

#### **Ermittlung der Average Precision**

In den vorangegangenen Arbeitsschritten wurde die IoU ermittelt und die atmosphärischen Zustände und die Wertebereiche der GSD und des Elevationswinkels aufgeteilt. Im nachfolgend letzten Arbeitsschritt wird mit Hilfe von TP, FP und FN die *Genauigkeit* (siehe Gleichung (3.1)) und die *Trefferquote* (siehe Gleichung (3.2)) für jeden Einzeldatensatz und für jeden Intervallbereich berechnet. Anschließend wird für jeden Einzeldatensatz und für jeden Intervallbereich die *Precision-Recall-Kurve* (PR-Kurve) erstellt. Mit Hilfe der PR-Kurve wird das Verhältnis zwischen der Genauigkeit und der Trefferquote bezüglich der unterschiedlichen Konfidenzwerte, die aus der Objektdetektion mit dem YOLOv3-Detektor resultieren (vgl. Abschnitt 3.4.4.1), visualisiert. Mit der PR-Kurve kann somit die Leistung binärer Klassifizierungsalgorithmen für einen spezifischen Schwellwert der IoU bewertet und grafisch dargestellt werden.

Die *Average Precision* (AP) ist eine Metrik zur Quantifizierung der Fläche unter der PR-Kurve [143, 144]. Die  $AP \in [0, 1]$  nimmt einen hohen Wert an, wenn sowohl die Genauigkeit als auch die Trefferquote für die unterschiedlichen Konfidenzwerte hoch sind. Im Umkehrschluss ist der Wert der AP niedrig, wenn die Genauigkeit und/oder die Trefferquote einen geringen Wert aufweisen. In dieser Arbeit wird die AP als quantitatives Maß für die Detektionsleistung beim Sensorleistungsmodell mit Deep Learning basierendem Objektdetektor verwendet. Ein hoher Wert für die AP korreliert somit mit einer hohen Detektionsleistung und umgekehrt. Für eine ausführliche Erläuterung zur Generierung der PR-Kurve und der Ermittlung der AP wird an dieser Stelle auf die Arbeit von Padilla et al. [143] verwiesen.

In Abb. 3.14 ist die AP in Abhängigkeit der GSD und des Elevationswinkels für das Sensorleistungsmodell mit der spezifischen Ausprägung der atmosphärischen Zustände  $\{\text{Herbst, Tag, klar, nass, bedeckt}\}$  dargestellt. Aus den jeweils vier Intervallbereichen für die GSD und für den Elevationswinkel resultieren 16 Bereiche mit einem individuellen, abschnittsweise konstanten AP-Wert. Weiße Felder deuten an, dass für diese Kombination keine Bilddaten verfügbar sind und damit auch keine AP berechnet werden kann. Aus der Abbildung geht auch hervor, dass sich die AP in Abhängigkeit der GSD und dem Elevationswinkel verändert. Ein ähnliches Verhalten wurde auch für andere Ausprägungen der atmosphärischen Zustände beobachtet. Das Sensorleistungsmodell berücksichtigt damit die perzeptiven Fähigkeiten der zu Grunde liegenden Perzeptionskette (u. a. bestehend

### 3 Konzept und Methodenauswahl

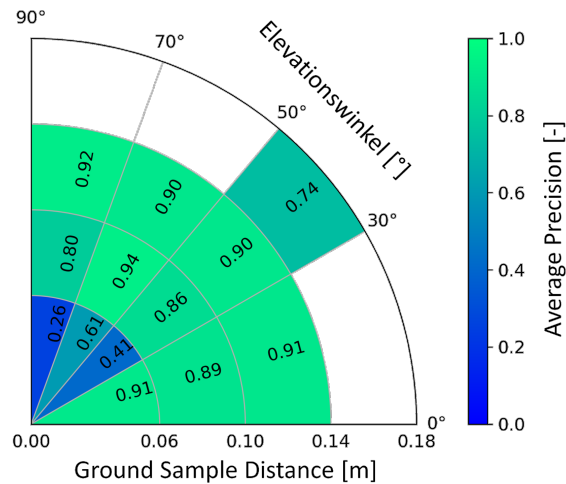


Abbildung 3.14: Darstellung der Average Precision (AP) in Abhängigkeit der Ground Sample Distance (GSD) und des Elevationswinkels für eine spezifische Ausprägung der atmosphärischen Zustände des Sensorleistungsmodells

aus dem ZENMUSE XT2 Sensorsystem und dem trainierten YOLOv3-Objektdetektor) für die Abbildung der Detektionsleistung.

Es wird erwartet, dass bei der Nutzung eines anderen bzw. neueren YOLO-Detektors (z. B. YOLOv4 [145]) die AP ebenfalls von der GSD und dem Elevationswinkel abhängt, was bedeutet, dass der vorgestellte Ansatz zur Modellierung des Sensorleistungsmodells nach wie vor Gültigkeit besitzt. Insbesondere wird bei der Verwendung einer neueren YOLO-Version erwartet, dass sich die AP in den einzelnen Intervallbereichen zu höheren Werten hin verschiebt, wie eine Untersuchung in [146] für YOLOv3, YOLOv4 und YOLO5l verdeutlicht.

Mit dem Verfahren zur Generierung eines Sensorleistungsmodells, wie es in diesem Abschnitt vorgestellt wurde, resultieren aus dem K&S-Datensatz insgesamt neun Varianten desselben Leistungsmodells. Diese unterscheiden sich durch die spezifische Ausprägung der atmosphärischen Zustände, wobei die zu Grunde liegende Perzeptionskette für alle Varianten identisch ist. Die GSD und der Elevationswinkel sind die Eingangsgrößen (unabhängige Variablen) des Leistungsmodells und die Average Precision ist die Ausgangsgröße (abhängige Variable), die die Detektionsleistung repräsentiert.

Die Modellierung des Sensorleistungsmodells durch Bereiche mit abschnittsweise konstanter Detektionsleistung wurde gewählt, da die Anzahl der Eingangsgrößen, im Vergleich zum Leistungsmodell aus Abschnitt 3.4.3, gering ist. Des Weiteren hat dieses Verfahren den Vorteil, dass das Leistungsmodell, im Vergleich zu einem neuronalen Netz, nicht

trainiert werden muss und keine Black Box darstellt. Zudem ist das Verfahren auch für Datensätze mit einer geringen Anzahl an Bildern einfach umsetzbar.

#### 3.4.5 Abbildung der Detektionsleistung durch Perception Maps

Auf Basis der in Abschnitt 3.4.3 und 3.4.4 vorgestellten bzw. entwickelten Sensorleistungsmodelle wird in diesem Abschnitt das Konzept der *Perception Map* erläutert, das erstmals von Zwick, Gerdtts & Stütz in [40] vorgestellt wurde. Mit diesem Konzept soll ein Beitrag geleistet werden, um die *Forschungsfrage 2.2* aus Abschnitt 2.4 zu beantworten.

In Zwick, Koch & Stütz [31] und in Abschnitt 3.1 wurde beschrieben, dass durch die gezielte Manipulation spezifischer Umgebungszustände die Detektionsleistung zielgerichtet beeinflusst und im besten Fall gesteigert werden kann. Auf diese Weise können negative Auswirkungen nicht kontrollierbarer Zustände (z. B. Sichtweite, Helligkeit, Niederschlag) auf die Detektionsleistung kompensiert oder abgeschwächt werden. Zu den relevanten veränderbaren Zuständen gehören der *Elevationswinkel* und die *Ground Sample Distance*. Beide Zustände können selektiv innerhalb technischer bzw. physikalischer Grenzen angepasst werden. Der Elevationswinkel kann variiert werden, indem der horizontale und/oder vertikale Abstand zwischen der Sensor- bzw. UAV-Position und dem Sensorfootprint am Boden verändert wird. Wie in Gleichung (3.5) beschrieben, kann durch eine Anpassung des Bildwinkels und/oder durch eine Variation der Länge der Sichtlinie die GSD verändert werden. Diese Zusammenhänge bilden die Grundlage zur Generierung der Perception Map.

Die in dieser Arbeit verwendeten Sensorleistungsmodelle bilden die Detektionsleistung implizit durch ein neuronales Netz (vgl. Abschnitt 3.4.3) oder als abschnittsweise definierte Funktion (vgl. Abschnitt 3.4.4) ab. Die Abbildung ist daher im Allgemeinen nicht explizit als multivariate Funktion bezüglich des Elevationswinkels und der GSD darstellbar. Aus diesem Grund wird das Konzept der Perception Map eingeführt, auf das im Folgenden eingegangen wird:

Die Perception Map stellt den räumlichen Verlauf der Detektionsleistung in Abhängigkeit der Sensorposition in einer zweidimensionalen Ebene als Potentialfeld dar. Diese Ebene ist koinzident mit der Bewegungsebene des UAVs und befindet sich in einer vorgegebenen und konstanten Flughöhe  $h_{alt}$ . Dabei bildet die Perception Map die perzeptiven Fähigkeiten und Randbedingungen des zu Grunde liegenden Sensorleistungsmodells und dessen Perzeptionskette ab.

Links in Abb. 3.15 ist eine Perception Map als dreidimensionale Grafik dargestellt. Das hierfür verwendete Sensorleistungsmodell repräsentiert eine Perzeptionskette mit dem

### 3 Konzept und Methodenauswahl

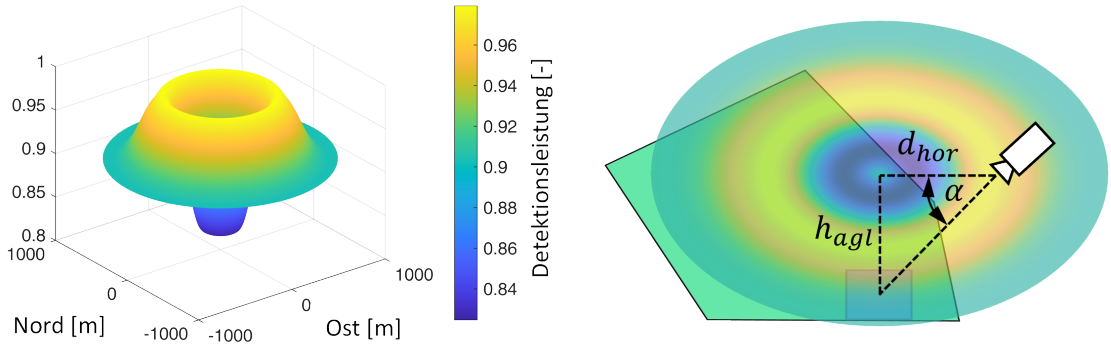


Abbildung 3.15: Abbildung einer Perception Map für das Sensorleistungsmodell mit Machine Learning basiertem Klassifikator (Zeichnung nicht maßstabsgetreu). Darstellung als dreidimensionales Diagramm (links) und dieselbe Map in einer planaren Darstellung (rechts), bezogen auf den entsprechenden Sensorfootprint (hellblaues Quadrat) am Boden. Der Elevationswinkel  $\alpha$  wird durch den horizontalen Abstand  $d_{hor}$  und die Flughöhe über Grund  $h_{agl}$  bestimmt. Die Farbkodierung der Perception Map entspricht der prognostizierten Detektionsleistung. Helle Farben stehen für hohe Leistungswerte, während dunklere Farben mit niedrigeren Werten korrelieren (übernommen und angepasst aus [33]).

Machine Learning basierten Klassifikator *Classification Cascade* aus Abschnitt 3.4.3. Die Nord-Ost-Ebene deckt sich mit der Bewegungsebene des UAVs, und die z-Richtung stellt den quantitativen Wert der prognostizierten Detektionsleistung dar, der zur besseren Veranschaulichung farblich kodiert ist. Die dargestellte Perception Map enthält Bereiche mit einer maximalen Detektionsleistung von 0,98 (gelb eingefärbt) und einem minimalen Wert von 0,83 (blau gefärbter Bereich). Rechts in Abb. 3.15 ist dieselbe Perception Map als zweidimensionale Grafik dargestellt. Das Kamerasymbol markiert die Sensorposition des UAVs, die durch den vertikalen Abstand  $h_{agl}$  und den horizontalen Abstand  $d_{hor}$  vom Zentrum des Sensorfootprints am Boden bestimmt ist. Da sich der Sensor in dem gelb markierten Bereich befindet, ist mit einer hohen Detektionsleistung für die erfassten Sensordaten zu rechnen. Mit Hilfe des vertikalen und horizontalen Abstands kann der Elevationswinkel der Sensorsichtlinie  $\alpha$  wie folgt berechnet werden:

$$\alpha = \arctan\left(\frac{h_{agl}}{d_{hor}}\right) \quad (3.6)$$

Es wird darauf hingewiesen, dass sich die vorgegebene UAV-Flughöhe  $h_{alt}$  auf einen Referenzpunkt (z. B. der Startpunkt des aufzuklärenden Bereichs) bezieht und konstant ist. Demgegenüber entspricht der vertikale Abstand  $h_{agl}$  der ortsabhängigen Flughöhe

über Grund. Die Werte von  $h_{alt}$  und  $h_{agl}$  können daher außerhalb des Referenzpunkts voneinander abweichen.

Die Perception Map wird erzeugt, in dem alle Eingangsgrößen des Sensorleistungsmodells (z. B. die atmosphärischen und topographischen Zustände), mit Ausnahme des Elevationswinkels, konstant gehalten werden. Zudem ist eine Perception Map immer einem spezifischen Sensorfootprint am Boden zugeordnet. Dies ist eine Voraussetzung für die Trajektorienoptimierung der sensormodellbestimmten Befliegungsplanung, auf die in Abschnitt 4.2 eingegangen wird.

Das Potentialfeld der kreisförmigen Perception Map ergibt sich aus der rotationssymmetrischen Darstellung der Detektionsleistung unter Variation des Elevationswinkels. Zu diesem Zweck wird der horizontale Abstand  $d_{hor}$  im Intervall  $[0, d_{pm}/2]$  variiert, wobei  $d_{pm}$  der vorgegebene Durchmesser der Perception Map ist. Der Mittelpunkt der Perception Map befindet sich somit lotrecht über dem Zentrum des zugehörigen Sensorfootprints. Die topographischen Zustände zur Berechnung der Detektionsleistung resultieren aus dem Sensorfootprint und können z. B. durch ein Geoinformationssystem ermittelt werden. Die atmosphärischen Zustände wiederum können aus einer Datenbank abgefragt oder durch Expertenwissen ermittelt werden, wobei vereinfachend in Abschnitt 1.4 festgelegt wurde, dass die atmosphärischen Zustände im gesamten aufzuklärenden Gebiet identisch und konstant sind. Dies soll an dieser Stelle auch für den Bereich innerhalb der Perception Map gelten.

Mit der Variation des horizontalen Abstands ändert sich die Länge der Sichtlinie, wodurch sich auch die GSD verändern würde (vgl. Gleichung (3.5)). Um dem zu begegnen und die Forderung nach konstanten Eingangsgrößen zu erfüllen, muss der Bildwinkel  $\beta$  des Sensorsystems dynamisch so angepasst werden, dass die GSD im Sensorfootprint konstant bleibt. Zu diesem Zweck wird der Sollwert für die GSD auf  $gsd_{ref} = 0,07$  m festgelegt. Dieser Wert stammt aus der Arbeit von [50] und hat sich als geeignet für die Fahrzeugdetektion erwiesen. Auf Grund einer festen Brennweite oder der technischen Limitierung des variablen Bildwinkels beim Sensorsystem kann es passieren, dass die GSD von der Sollwertvorgabe  $gsd_{ref}$  abweicht. In der Folge kann dies zu einer Reduktion der Detektionsleistung führen, da relevante Objektmerkmale (z. B. Kanten, Winkel) in den Bilddaten nicht mehr aufgelöst werden können. Dieser Effekt wird durch das Sensorleistungsmodell abgebildet und beeinflusst dadurch ebenfalls den Verlauf des Potentialfelds der Perception Map. Für die Evaluierung der Befliegungsplanung in Kapitel 6 wird ein UAV mit elektrooptischem Sensorsystem simuliert. Die Optik des Sensors hat einen technisch begrenzten Bildwinkel, dessen Grenzwerte  $\beta_{max}$  und  $\beta_{min}$  in Tabelle 3.4 aufgelistet sind.

### 3 Konzept und Methodenauswahl

Wie oben erwähnt, verläuft das Potentialfeld rotationssymmetrisch um den Ursprung. Ursächlich hierfür ist, dass die ortsabhängige Detektionsleistung in der Perception Map nur durch den Elevationswinkel bestimmt wird. Für ein Sensorleistungsmodell, das die Detektionsleistung in Abhängigkeit zusätzlicher Einflüsse mit Bezug auf die Himmelsrichtung / Azimut (z. B. Schattenwurf, Gegenlicht) oder die Positionierung des UAVs abbilden kann (z. B. um entlang einer Schlucht aufklären zu können), wird das Potentialfeld der Perception Map nicht mehr rotationssymmetrisch verlaufen. Auch für diesen Fall kann die Perception Map für die sensormodellbestimmte Befliegungsplanung genutzt werden und das Konzept hat weiterhin Bestand.

Der Zweck der Perception Map ist es, aus der (impliziten) Abbildung der Detektionsleistung durch das Sensorleistungsmodell die Positionen bzw. die Bereiche in der Bewegungsebene des UAVs zu identifizieren, die eine hohe bzw. die höchste zu erwartende Detektionsleistung aufweisen. Diese Bereiche sind wiederum prädestiniert für die Trajektorienoptimierung der sensormodellbestimmten Befliegungsplanung, da für die hieraus resultierende Detektionsleistung folglich ein hoher bzw. der höchste Wert prognostiziert wird.

In Tabelle 3.4 sind die verschiedenen Parameter zusammengefasst, die in diesem Abschnitt eingeführt wurden. Parameter, die in der rechten Spalte mit dem Begriff „vorgegeben“ versehen sind, wurden auf der Grundlage früherer Arbeiten oder Untersuchungen festgelegt, auf die hier nicht weiter eingegangen wird.

Tabelle 3.4: Parametereinstellung für die Generierung der Perception Maps

Parameter	Wert	Bemerkung
Sollwert der Ground Sample Distance $gsd_{ref}$	0,07 m	vorgegeben
Flughöhe des UAVs $h_{alt}$	500 m	vorgegeben
Durchmesser der Perception Map $d_{pm}$	2000 m	vorgegeben
maximaler Bildwinkel $\beta_{max}$	15°	vorgegeben
minimaler Bildwinkel $\beta_{min}$	5°	vorgegeben

Für das Sensorleistungsmodell aus Abschnitt 3.4.3, das beispielhaft eine Perzeptionskette mit dem Objektklassifikator *Classification Cascade* abbildet, resultiert ein stetiger und differenzierbarer Verlauf der Detektionsleistung unter Variation des Elevationswinkels (siehe auch Abb. 3.15). Ursächlich hierfür ist die Abbildung der Detektionsleistung durch ein neuronales Netz. Demgegenüber erhält man für das Sensorleistungsmodell mit Deep Learning basiertem Objektdetektor (vgl. Abschnitt 3.4.4) eine Perception Map mit abschnittsweise konstanter Detektionsleistung, wie in Abb. 3.16 ersichtlich wird.

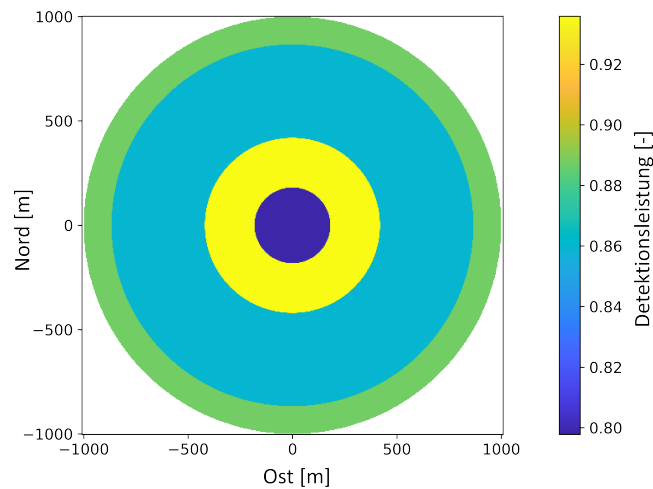


Abbildung 3.16: Perception Map, die aus dem Sensorleistungsmodell mit YOLOv3-Objektdetektor resultiert. Zu erkennen sind die farblich markierten Bereiche mit abschnittsweise konstanter Detektionsleistung (übernommen und angepasst aus [33])

Das Konzept der Perception Maps hat den Vorteil, dass es sowohl auf implizite und explizite Funktionen als auch auf differenzierbare (wie beim Sensorleistungsmodell aus Abschnitt 3.4.3) sowie auf nicht differenzierbare und diskontinuierliche Funktionen (wie z. B. für das Sensorleistungsmodell aus Abschnitt 3.4.4) angewendet werden kann. Ein weiterer entscheidender Vorteil ist, dass für jeden einzelnen Sensorfootprint die *theoretisch maximal erreichbare Detektionsleistung* bestimmt werden kann. Dies ist der höchste Wert der Detektionsleistung, der sich innerhalb der Perception Map einstellt. Dieser Wert stellt die Obergrenze für die erreichbare Detektionsleistung dar und wird in Kapitel 6 wieder aufgegriffen, um die Fähigkeit der sensormodellbestimmten Befliegungsplanung zur Steigerung der Detektionsleistung validieren zu können.

#### 3.4.6 Zusammenfassung

In Abschnitt 3.4 wurden die Grundlagen der in dieser Arbeit verwendeten Sensorleistungsmodelle vorgestellt. Darauf aufbauend wurden in Abschnitt 3.4.3 Sensorleistungsmodelle beschrieben, die für Perzeptionsketten mit Objektklassifikatoren aus unterschiedlichen Bereichen (u. a. Machine Learning, Template Matching) entwickelt wurden. Diese Modelle bilden ausgesuchte perzeptiver Einflüsse (u. a. topographische, photogrammetrische und atmosphärische Zustände) auf die zu erwartende Detektionsleistung ab. Im weiteren Verlauf dieser Arbeit wird neben dem Sensorleistungsmodell für einen Deep Learning Ob-

### 3 Konzept und Methodenauswahl

jektdetektor (vgl. Abschnitt 3.4.4) auch das Sensorleistungsmodell mit CC-Klassifikator für die Evaluierung der sensormodellbestimmten Befliegungsplanung genutzt. Die weitere Verwendung dieses Modells wird damit begründet, dass dieses Modell in der Arbeit von Hellert [50] eine hohe Genauigkeit unter den untersuchten Klassifikatoren aufwies. Ungeachtet dessen ist die hier vorgestellte Methode zur sensormodellbestimmten Befliegungsplanung ohne Einschränkungen auch für die Leistungsmodelle mit DPM-, TM- und BLOB-Klassifikator anwendbar.

In Abschnitt 3.4.4.3 wurde ein mögliches Verfahren zur Generierung von Sensorleistungsmodellen aufgezeigt, das die perzeptiven Fähigkeiten (u. a. Genauigkeit und Trefferquote des YOLOv3-Objektdetektors) und Randbedingungen (z. B. Auflösung und Signal-Rausch-Abstand des Sensorsystems ZENMUSE XT2) einer Perzeptionskette mit Deep Learning basierendem Objektdetektor abbildet. Damit wurde die *Forschungsfrage 2.1* (Wie kann ein Sensorleistungsmodell für eine Perzeptionskette mit Deep Learning basierendem Objektdetektor modelliert werden, um die Detektionsleistung bezüglich ausgewählter perzeptiver Einflüsse abbilden zu können?) aus Abschnitt 2.4 beantwortet.

Des Weiteren wurde in Abschnitt 3.4.5 das Konzept der Perception Map erläutert. Die Perception Map bildet die Detektionsleistung als Potentialfeld in der Bewegungsebene des UAVs ab, unter der Nutzung von Sensorleistungsmodellen. Damit wird die Abbildung der Detektionsleistung in Abhängigkeit der Sensor- bzw. UAV-Position ermöglicht, was für die spätere Trajektorienoptimierung erforderlich ist. Mit der Entwicklung und Umsetzung der Perception Map wird ein Beitrag zur Beantwortung der *Forschungsfrage 2.2* (Wie kann die Detektionsleistung für ein Sensorleistungsmodell in Abhängigkeit der Sensor- bzw. UAV-Position modelliert werden?) aus Abschnitt 2.4 geleistet. Darüber hinaus wurde die *Anforderung 2* und Anteile von *Anforderung 3* aus Abschnitt 3.2 umgesetzt.

Mit der Beantwortung der *Forschungsfrage 2.1* und *2.2* wurde das übergeordnete *Zwischenziel 2* (Entwicklung und Untersuchung von Sensorleistungsmodellen zur Modellierung der Detektionsleistung in Abhängigkeit ausgewählter perzeptiver Einflüsse und Randbedingungen.) aus Abschnitt 1.3 erfolgreich abgeschlossen. Nachfolgend ist der aktuelle Stand zur Bearbeitung der Zwischenziele und Forschungsfragen abgebildet.

Zwischenziel	1	2	3	4	5
Forschungsfrage	1.1	2.1 & 2.2	3.1 & 3.2	—	5.1 - 5.3
Bearbeitung in Kapitel	3	3	3 & 4	4	6



## 3.5 Grundlagen zur Generierung der UAV-Flugtrajektorie

In diesem Kapitel werden die Anforderungen und Eigenschaften der zu optimierenden UAV-Flugtrajektorie spezifiziert. Dies erfolgt in Vorbereitung für die eigentliche Trajektorienoptimierung in Abschnitt 3.6. Zu diesem Zweck werden nachfolgend die zeitkontinuierlichen und zeitdiskreten UAV-Bewegungsgleichungen hergeleitet. Diese beiden Formen der Bewegungsgleichungen werden benötigt, um in Abschnitt 3.6 unter Verwendung verschiedener Optimierungsmethoden optimale Referenztrajektorien generieren zu können. Abgeschlossen wird dieses Kapitel mit der Vorstellung der Dubins-Pfadplanung. Mit Hilfe der Dubins-Pfade werden aufeinander folgende Zustände bei der zeitdiskreten Bewegungsplanung optimal verbunden, was die Grundlage zur Generierung global optimaler Referenztrajektorien für das zeit- und wertdiskrete Optimierungsverfahren *Dynamische Programmierung* in Abschnitt 3.6.2 bildet.

Die Pfadplanung und die Trajektorienoptimierung sind elementar für die Generierung der optimalen Referenz-Flugtrajektorien in dieser Arbeit. Daher wird im Folgenden kurz auf die Differenzierung zwischen einem Pfad bzw. einer Bahn<sup>7</sup> und einer Trajektorie eingegangen: Die *Pfad-* bzw. *Bahnplanung* ist in der Regel ein rein geometrisches Verfahren zur Erzeugung einer Flugbahn zwischen einem Start- und einem Zielpunkt, ohne dass ein zeitlicher Verlauf oder dynamische Beschränkungen des UAVs berücksichtigt werden. Demgegenüber besteht die *Trajektorienoptimierung* darin, der Flugbahn einen (zeitlich) parametrisierten Geschwindigkeits- und Beschleunigungsverlauf, unter Berücksichtigung dynamischer Beschränkungen (z. B. der Rollrate), zuzuordnen. In den meisten Fällen ist die Bahnplanung eine Vorstufe für die Trajektorienoptimierung [95, 147].

Der Zweck der Trajektorienoptimierung ist es, Referenz- bzw. Sollwertvorgaben für das Flugsteuerungssystem des UAVs zu generieren, um zu erreichen, dass das UAV eine gewünschte Flugbewegung ausführt oder einer vorgegebenen Flugbahn folgt. Zu diesem Zweck beschreibt die Referenz-Flugtrajektorie eine Folge von Sollwertvorgaben bezüglich der Position, Lage, Geschwindigkeit und Beschleunigung, um einer vorgegebenen Flugbahn, unter Einhaltung zeitlicher und/oder flugdynamischer Randbedingungen, zu folgen.

### 3.5.1 Vereinfachungen und Anforderungen an die Flugtrajektorie

Im weiteren Verlauf dieses Kapitels werden Bewegungsgleichungen für ein Starrflügel-UAV formuliert, das als Trägerplattform für einen optischen Sensor dient und zur luftgestützten

---

<sup>7</sup>Die Begriffe „Pfad“ und „Bahn“ werden in der Literatur häufig synonym verwendet, ebenso die damit verbundenen Verfahren zur „Pfad-“ bzw. „Bahnplanung“.

### 3 Konzept und Methodenauswahl

Aufklärung eingesetzt werden soll. Für die Aufstellung der Bewegungsgleichungen werden zunächst die folgenden Vereinfachungen getroffen und deren Validität begründet:

- Die Erde wird als ruhend und flach angenommen. Das erdlotfeste Koordinatensystem wird als Inertialsystem betrachtet. Diese Annahme wird als zulässig angesehen, da die Referenztrajektorie, die das Resultat der sensormodellbestimmten Befliegungsplanung ist, eine zeitlich und räumlich begrenzte Ausdehnung aufweist und dadurch Effekte der Erdkrümmung und der Erdrotation vernachlässigt werden können.
- Der Einfluss von Wind oder Turbulenzen auf die Flugbewegung des Luftfahrzeugs wird bei der Trajektorienoptimierung nicht betrachtet. Die Optimierung der Referenztrajektorie erfolgt unter Berücksichtigung der flugdynamischen Beschränkungen, die auch Leistungsreserven zum Ausgleich von Wind und Turbulenzen mit beinhalten. Dies hat zur Folge, dass der Autopilot an Bord des UAVs Störeinflüsse auf die räumlich-zeitliche Sollwertvorgabe ausgleichen und somit die Differenz zwischen der UAV-Position und der Sollwertvorgabe eigenständig ausgleichen kann.
- Das UAV bewegt sich in einer konstanten Flughöhe, wodurch die Bewegungsgleichung für die vertikale Bewegung obsolet ist. Die Vereinfachung wird damit begründet, dass beim zeitgleichen Einsatz mehrerer Luftfahrzeuge eine Höhenstaffelung eingeführt wird, um Kollisionen oder gegenseitige Beeinträchtigungen zu verhindern.
- Es wird ein idealer Kurvenflug angenommen. Dies ermöglicht eine vereinfachte Beschreibung der Bewegungsgleichungen, ohne die Berücksichtigung des Schiebewinkels. Zudem folgt daraus die eindeutige Zuordnung des Rollwinkels zum Lastvielfachen im stationären Kurvenflug. Begründet wird die Annahme eines idealen Kurvenfluges damit, dass Wind und Turbulenzen unberücksichtigt bleiben und der Autopilot an Bord des UAVs in der Lage ist, selbstständig einen koordinierten Kurvenflug durchführen zu können.
- Die Fluggeschwindigkeit wird durch ein eng begrenztes Intervall vorgegeben und kann in erster Näherung als konstant angesehen werden. Diese Vereinfachung wird damit begründet, dass für Aufklärungs- und Überwachungsmissionen eine konstante Fluggeschwindigkeit angestrebt wird, um die Stehzeit des UAVs durch eine energieeffiziente Flugführung zu maximieren.

### 3.5 Grundlagen zur Generierung der UAV-Flugtrajektorie

Die Referenz-Flugtrajektorie ist die räumlich-zeitliche Sollwertvorgabe der Flugbahn, die von einem Autopiloten an Bord des UAVs automatisiert abgeflogen werden soll. Hierfür werden an die Referenztrajektorie die folgenden Anforderungen gestellt:

- Die Referenztrajektorie definiert die Sollwerte für den Gierwinkel und die zweidimensionalen Positionskoordinaten des UAVs in der Bewegungsebene, sowie deren zeitliche Parametrisierung. Die Flughöhe des UAVs wird als konstant angenommen.
- Der räumlich-zeitliche Verlauf der Referenztrajektorie muss flugdynamische Beschränkungen des UAVs berücksichtigen.

Damit die Flugbereichsgrenzen des UAVs eingehalten werden, sollen im weiteren Verlauf die folgenden ausgesuchten flugdynamischen Beschränkungen für die Trajektorienoptimierung berücksichtigt werden:

- Der maximale Rollwinkel  $\phi_{max}$  des UAVs für den stationären Kurvenflug wird vorgegeben und resultiert aus dem zulässigen Lastvielfachen  $n_{max}$ .
- Der Rollwinkel  $\phi(t)$ , die Rollrate  $\omega(t)$ , sowie die Beschleunigung  $a(t)$  in Längsrichtung und die Fluggeschwindigkeit  $v(t)$  des UAVs sind betragsmäßig begrenzt.

Durch die betragsmäßige Begrenzung der Rollrate wird eine glatte,  $C^2$ -stetige, Trajektorie mit differenzierbarem Krümmungsverlauf forciert. Ohne Rollratenbegrenzung wäre eine sprungförmige Änderung des Rollwinkels zulässig, die physikalisch nicht realisierbar ist.

#### 3.5.2 UAV-Bewegungsgleichungen in der Ebene

Da die Trajektorienoptimierung für ein unbemanntes luftgestütztes Aufklärungssystem erfolgt, wird davon ausgegangen, dass das UAV mit einem Autopiloten (Flugregelungssystem) ausgestattet ist. Dieser berechnet aus der Referenz-Flugtrajektorie und dem aktuellen Flugzustand die notwendigen Steuereingaben für den Antriebsmotor sowie für die aerodynamischen Steuerflächen, um das UAV zeitlich und räumlich entlang der Trajektorie zu führen. Äußere Störeinflüsse auf die Flugbewegung des UAVs, die beispielsweise durch Wind und Turbulenzen hervorgerufen werden, sollen durch den Autopiloten selbstständig ausgeglichen werden. Verschiedene Softwareprodukte zur Flugsteuerung, wie z. B. *ArduPilot* [148], bieten integrierte Autopilot-Funktionen zur Entkopplung der Längs- und Querdynamik bei UAVs. Da vorausgesetzt wird, dass die Flugsteuerung eine vorgegebene Fluggeschwindigkeit und Flughöhe eigenständig regeln kann, wird in den folgenden Abschnitten nur die Querdynamik des UAVs betrachtet.

### 3.5.2.1 Zeitkontinuierliche Bewegungsgleichungen

Die Formulierung der *zeitkontinuierlichen Bewegungsgleichungen* wurde bereits in der Arbeit von Zwick, Gerds & Stütz [33] dargestellt und wird im Folgenden nochmals aufgegriffen. Diese Gleichungen bilden die Grundlage zur Modellierung der nichtlinearen UAV-Flugdynamik für die Trajektorienoptimierung mit *nichtlinearer modellprädiktiver Regelung*, auf die in Abschnitt 3.6.3 eingegangen wird.

Mit Gleichung (3.7) wird die Änderung des Systemzustands für ein allgemeines *zeitkontinuierliches dynamisches System* durch eine gewöhnliche Differentialgleichung in expliziter Form beschrieben:

$$\dot{x}(t) = F(x(t), u(t)), \quad t \in [0, T] \quad (3.7)$$

Mit dem zeitkontinuierlichen Zustandsvektor des Systems  $x(t) \in \mathbb{R}^{n_x}$ , dem Steuerungsvektor  $u(t) \in \mathbb{R}^{n_u}$  und der Zeit  $t$ .  $n_x$  und  $n_u$  bezeichnen die Dimensionen des entsprechenden Zustands- und Steuerungsvektors. Die Abbildungsvorschrift  $F : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$  beschreibt die zeitliche Änderung des Systemzustands in Abhängigkeit des eigentlichen Systemzustands, der Steuereingaben und der Zeit.

Die Differentialgleichungen (3.8) beschreiben die nichtlinearen zeitkontinuierlichen Bewegungsgleichungen für ein Starrflügel-UAV [149] im Inertialsystem einer zweidimensionalen horizontalen Ebene unter Berücksichtigung der getroffenen Vereinfachungen aus Abschnitt 3.5.1:

$$\dot{e}(t) = v(t) \cdot \sin(\psi(t)) \quad (3.8a)$$

$$\dot{n}(t) = v(t) \cdot \cos(\psi(t)) \quad (3.8b)$$

$$\dot{\psi}(t) = \frac{g}{v(t)} \cdot \tan(\phi(t)) \quad (3.8c)$$

$$\dot{v}(t) = a(t) \quad (3.8d)$$

$$\dot{\phi}(t) = \omega(t) \quad (3.8e)$$

$$x(t) = [e(t), n(t), \psi(t), v(t), \phi(t)]^T \quad (3.9)$$

Der Systemzustandsvektor (3.9) des zeitkontinuierlichen dynamischen Systems setzt sich zusammen aus den Zustandsvariablen der östlichen und nördlichen Position<sup>8</sup>  $e(t)$  und

---

<sup>8</sup>Die Systemzustandsvariablen zur Beschreibung der UAV-Position werden in dieser Arbeit mit  $n$  (north) und  $e$  (east) bezeichnet, abweichend zu der gebräuchlichen Notation mit  $x$  und  $y$ . Diese Anpassung wird gewählt, um eine Verwechslung der  $x$ -Positionskoordinate mit dem Systemzustandsvektor  $x(t)$  bzw.  $x_k$  zu vermeiden

### 3.5 Grundlagen zur Generierung der UAV-Flugtrajektorie

$n(t)$ , dem Gierwinkel  $\psi(t)$ , sowie der horizontalen Bahngeschwindigkeit  $v(t)$  und dem Rollwinkel  $\phi(t)$  des UAVs. Die Ortskoordinaten  $n$  und  $e$  definieren die UAV-Position in der Horizontalebene, dargestellt in einem erdfesten Koordinatensystem (siehe Abb. 3.17). Die als konstant angenommene Erdbeschleunigung wird mit  $g$  bezeichnet.

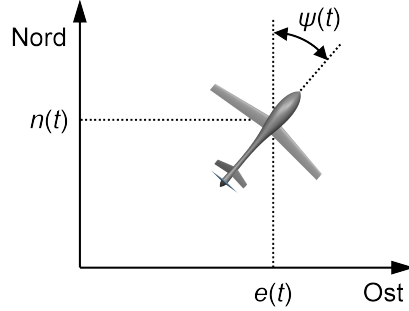


Abbildung 3.17: Position  $n(t)$ ,  $e(t)$  und Gierwinkel  $\psi(t)$  des UAVs im erdfesten Koordinatensystem

Mit dem Steuerungsvektor (3.10) wird das flugdynamische System in Gleichung (3.7) gezielt beeinflusst. Der Steuerungsvektor setzt sich zusammen aus der Sollwertvorgabe für die Beschleunigung  $a(t)$  des UAVs tangential zur Flugrichtung und der Sollwertvorgabe für die Rollrate  $\omega(t)$ .

$$u(t) = [a(t), \omega(t)]^T \quad (3.10)$$

In Gleichung (3.11) ist der Zusammenhang zwischen dem maximalen Lastvielfachen  $n_{max}$  und dem Rollwinkel  $\phi_{max}$  für den stationären, schiebefreien Kurvenflug formal beschrieben:

$$n_{max} = \frac{1}{\cos(\phi_{max})} \quad \Leftrightarrow \quad \phi_{max} = \arccos\left(\frac{1}{n_{max}}\right) \quad (3.11)$$

Um die flugdynamischen Beschränkungen des UAVs aus Abschnitt 3.5.1 berücksichtigen zu können, werden die nachfolgend aufgelisteten Zustands- und Steuerbeschränkungen festgelegt. Diese gehen als Randbedingungen in die Trajektorienoptimierung mit ein.

$$|\phi(t)| \leq \phi_{max} \quad (3.12)$$

$$|\omega(t)| \leq \omega_{max} \quad (3.13)$$

$$|a(t)| \leq a_{max} \quad (3.14)$$

$$v_{min} \leq v(t) \leq v_{max} \quad (3.15)$$

Mit  $\phi_{max}$ ,  $\omega_{max}$ ,  $v_{min/max}$  und  $a_{max} \in \mathbb{R}_+$ .

### 3 Konzept und Methodenauswahl

Neben der Begrenzung der Rollrate  $\omega(t)$ , der Beschleunigung  $a(t)$  tangential zur Flugrichtung und der Fluggeschwindigkeit  $v(t)$ , wird die Einhaltung des maximal zulässigen Lastvielfachen  $n_{max}$  durch die Limitierung des Rollwinkels  $\phi_{max}$  erreicht. Für diese Arbeit wurde das zulässige Lastvielfache auf  $n_{max} = 1, 3$  festgelegt. Die Zahlenwerte der Zustands- und Steuerbeschränkungen sind in der nachfolgenden Tabelle 3.5 zusammengefasst.

Tabelle 3.5: Zustands- und Steuerbeschränkungen für die zeitkontinuierlichen Bewegungsgleichungen

Parameter	Wert	Bemerkung
maximaler Rollwinkel $\phi_{max}$	0.7 rad	vorgegeben
maximale Rollrate $\omega_{max}$	0.5 rad/s	vorgegeben
maximale Beschleunigung $a_{max}$	0.1 m/s <sup>2</sup>	vorgegeben
minimale Fluggeschwindigkeit $v_{min}$	33 m/s	vorgegeben
maximale Fluggeschwindigkeit $v_{max}$	37 m/s	vorgegeben

#### 3.5.2.2 Zeitdiskrete Bewegungsgleichungen

Die *zeitdiskreten Bewegungsgleichungen*, auf die im Folgenden eingegangen wird, wurden bereits in der Arbeit von Zwick, Gerdts & Stütz [33] vorgestellt. Diese zeitdiskreten Gleichungen werden im weiteren Verlauf der Arbeit wieder aufgegriffen, um mit Hilfe der *dynamischen Programmierung* in Abschnitt 3.6.2 global optimale Referenz-Flugtrajektorien zu generieren. In Analogie zu den zeitkontinuierlichen Gleichungen aus dem vorangegangenen Abschnitt, ist in Gleichung (3.16) eine allgemeine Formulierung für ein *zeitdiskretes dynamisches System* mit einem Zeithorizont der Länge  $N$  beschrieben.

$$x_{k+1} = f(x_k, u_k), \quad k = 0, \dots, N - 1 \quad (3.16)$$

Mit  $x_k \in S_k$  als dem zeitdiskreten Zustandsvektor des Systems, dem Steuerungsvektor  $u_k \in U_k(x_k)$  und die diskreten Zeitschritte  $k \in \mathbb{N}_0$ . Im Gegensatz zum zeitkontinuierlichen dynamischen System in Gleichung (3.7), bei dem die Systemzustände für alle Zeitpunkte im Intervall  $[0, T]$  durch die Bewegungsgleichungen eindeutig beschrieben sind, sind die Systemzustände beim zeitdiskreten System nur für die diskreten Zeitschritte  $k$  definiert. Mit Hilfe der Abbildung  $f$  wird der Zustandsübergang von einem Zustand  $x_k$  zum darauf folgenden Zustand  $x_{k+1}$  unter dem Einfluss des zeitschrittspezifischen Steuerungsvektors  $u_k$  festgelegt.

### 3.5 Grundlagen zur Generierung der UAV-Flugtrajektorie

Nachfolgend ist ein möglicher Ansatz zur Beschreibung der Abbildungsvorschrift  $f$  durch die *zeitdiskreten Bewegungsgleichungen* (3.17) bis (3.21) dargestellt. Hierbei handelt es sich um die zeitdiskrete Formulierung der UAV-Bewegungsgleichungen aus Abschnitt 3.5.2.1, die beispielsweise mit Hilfe des Euler-Verfahrens ermittelt werden können. Der zeitliche Abstand  $\Delta t > 0$  (Zeitschrittintervall) zwischen zwei aufeinander folgender Zeitschritte (z. B. von  $k$  nach  $k + 1$ ) wird als konstant definiert.

$$e_{k+1} = e_k + v_k \cdot \sin(\psi_k) \cdot \Delta t \quad (3.17)$$

$$n_{k+1} = n_k + v_k \cdot \cos(\psi_k) \cdot \Delta t \quad (3.18)$$

$$\psi_{k+1} = \psi_k + \frac{g}{v_k} \cdot \tan(\phi_k) \cdot \Delta t \quad (3.19)$$

$$v_{k+1} = v_k + a_k \cdot \Delta t \quad (3.20)$$

$$\phi_{k+1} = \phi_k + \omega_k \cdot \Delta t \quad (3.21)$$

Sowohl der Systemzustandsvektor  $x_k$  in Gleichung (3.22), als auch der Steuerungsvektor  $u_k$  in Gleichung (3.23) sind analog zu deren zeitkontinuierlichen Pendanten aus Abschnitt 3.5.2.1 aufgebaut. Zwischen zwei Zeitschritten ist der Steuerungsvektor konstant.

$$x_k = [e_k, n_k, \psi_k, v_k, \phi_k]^T \quad (3.22)$$

$$u_k = [a_k, \omega_k]^T \quad (3.23)$$

Als Vorgriff auf Abschnitt 3.6.2 wird angemerkt, dass die eben beschriebene Abbildungsvorschrift mit fünf Zustandsgrößen für die zeit- und wertdiskrete Trajektorienoptimierung mit *dynamischer Programmierung* als nicht zielführend bewertet wird. Begründet wird diese Entscheidung durch den exponentiellen Anstieg des Berechnungsaufwands mit der Anzahl der Systemzustände  $x_k$  [150, S. 27]. Der Berechnungsaufwand und die Berechnungsdauer erschweren bzw. verhindern die Trajektorienoptimierung (ggf. unter Einhaltung von Echtzeitanforderungen) an Bord des UAVs. Weitere Informationen bezüglich des Berechnungsaufwands werden in Abschnitt 3.6.2 gegeben.

Ein vereinfachter Ansatz zur Modellierung der Zustandsübergangsfunktion  $f$  wurde in einer vorangegangenen Arbeit [40] vorgestellt. Der Zustandsvektor wurde hierbei auf drei Zustände (die diskretisierten Positionskoordinaten  $n_k$ ,  $e_k$  und den Gierwinkel  $\psi_k$ ) reduziert. Die Diskretisierung wurde so gewählt, dass der Wertebereich für  $n_k$  und  $e_k$  ein Gitter mit äquidistanten Gitterpunktabständen in der Bewegungsebene des UAVs aufspannt, was in Abb. 3.18 abstrahiert dargestellt ist. Die optimale Flugbahn

### 3 Konzept und Methodenauswahl

wurde als Streckenzug (blaue Linie) aus aneinandergereihten geraden Linien modelliert, die durch diskrete Gitterpunkte verlaufen. Unter Verwendung des Gierwinkels konnte für jeweils zwei aufeinander folgende gerade Streckenabschnitte eine Limitierung der Flugrichtungsänderung erzielt werden. Abschließend wurde der Streckenzug durch einen kubischen Spline (rote Linie) geglättet und daraus die eigentliche Flugtrajektorie generiert. Ein wesentlicher Nachteil an diesem Verfahren ist, dass auf Grund der Limitierung der Flugrichtungsänderung nur eine näherungsweise Begrenzung der Bahnkrümmung möglich ist. In Abb. 3.18 ist die Flugbahn über die diskreten Gitterpunkte 1-3 abgebildet.

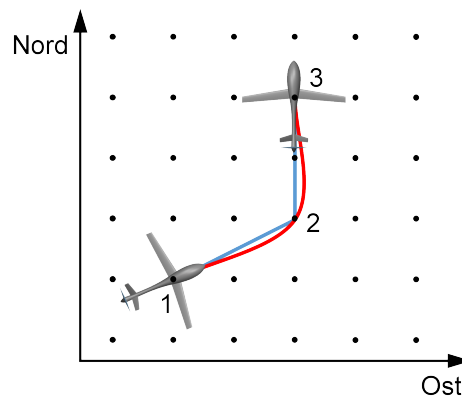


Abbildung 3.18: Modellierung der Zustandsübergänge auf einem Gitter in der Bewegungsebene des UAVs. Der Streckenzug (blau) ist die direkte Verbindung zwischen den diskreten UAV-Positionen (1-3). Mit Hilfe eines Splines (rot) wird der Streckenzug geglättet

Um das Problem der näherungsweisen Begrenzung der Bahnkrümmung lösen zu können, wurde in [33] eine Methode vorgestellt, bei dem die Zustandsübergangsfunktion für die zeitdiskrete Bewegungsgleichung des UAVs durch *Dubins-Pfade* modelliert wird. Dies ermöglicht eine explizite Berücksichtigung der flugdynamischen Beschränkungen bezüglich der Flugbahnkrümmung. Im folgenden Abschnitt 3.5.3 wird dieses Verfahren näher vorgestellt, um es im weiteren Verlauf der Arbeit für die zeit- und wertdiskrete Trajektorienoptimierung mit *dynamischer Programmierung* verwenden zu können.

#### 3.5.3 Dubins-Pfadplanung

Die *Dubins-Pfadplanung* wurde erstmals von Lester Dubins [151] formuliert und beschreibt ein Verfahren, um den kürzesten Weg zu identifizieren, der eine Startkonfiguration mit einer Zielkonfiguration in einer zweidimensionalen Ebene und unter Einhaltung einer



### 3.5 Grundlagen zur Generierung der UAV-Flugtrajektorie

Krümmungsbeschränkung miteinander verbindet. Die Konfiguration beschreibt dabei die Position des Start- bzw. Zielpunkts in der Bewegungsebene sowie die dazugehörige Richtungsvorgabe (Gierwinkel) des Geschwindigkeitsvektors.

In dieser Arbeit werden Dubins-Pfade genutzt, um die Zustandsübergänge für das zeit- und wertdiskrete Optimierungsverfahren der *dynamischen Programmierung* (siehe Abschnitt 3.6.2) zwischen zwei Zeitschritten zu modellieren. Begründet wird diese Entscheidung auf Grund von zwei wesentlichen Eigenschaften der Dubins-Pfade:

- Die Dubins-Pfade sind krümmungsbeschränkt, wodurch die Flugbereichsgrenze bezüglich des zulässigen Lastvielfachen explizit berücksichtigt werden kann.
- Aus dem Dubins-Pfad resultiert prinzipbedingt die kürzeste Flugbahn zwischen der Start- und Zielkonfiguration, wodurch die Flugdauer minimiert wird. Dies ist u. a. ein Optimierungskriterium für die sensormodellbestimmte Trajektorienoptimierung.

In Abschnitt 4.4 werden die Dubins-Pfade mit Hilfe der *dynamischen Programmierung* zu einer global optimalen Trajektorie aus aneinandergereihten Segmenten aus Dubins-Pfaden verkettet.

Für die Modellierung der Flugbahn eines Starrflügel-UAVs durch Dubins-Pfade muss eine Vorwärtsgeschwindigkeit  $v(t) > 0$  vorausgesetzt werden. Dadurch wird eine Rückwärtsbewegung ausgeschlossen, was den Dubins-Pfad von der Reeds-Shepp-Kurve [152] abgrenzt. Der Dubins-Pfad ist eine häufig verwendete Methode zur Modellierung der UAV-Flugbahn, um zeitoptimale Trajektorien unter Einhaltung von Krümmungsbeschränkungen generieren zu können. In der Literatur sind zahlreiche Anwendungsbeispiele zu finden, beispielsweise in [153–155].

Die Grundprinzipien der Dubins-Pfadplanung, die in der vorangegangenen Arbeit [33] bereits erläutert wurden, werden im Folgenden nochmals kurz dargestellt. Für die Dubins-Pfadplanung müssen die folgenden zwei Kriterien erfüllt sein [147, S. 880]:

- Die Fluggeschwindigkeit  $v$  des UAVs muss konstant gesetzt werden.
- Es muss ein maximal zulässiger Rollwinkel  $\phi_{max}$  definiert werden.

Der Dubins-Pfad wird aus der Menge der Bewegungsprimitiven  $\{L, S, R\}$  erstellt, wobei  $L$  für eine maximal gekrümmte Linkskurve,  $R$  für eine maximal gekrümmte Rechtskurve und  $S$  für ein Geradenstück steht. Dubins hat in [151] gezeigt, dass der kürzeste Weg zwischen zwei Konfigurationen aus einer Verkettung von nicht mehr als drei dieser Bewegungsprimitiven generiert werden kann.

### 3 Konzept und Methodenauswahl

Die Gleichungen (3.24) bis (3.26) wurden aus Abschnitt 3.5.2.1 übernommen und beschreiben die zeitkontinuierliche Bewegung eines UAVs, das sich in der Ebene bewegt.

$$\dot{n}(t) = v \cdot \cos(\psi(t)) \quad (3.24)$$

$$\dot{e}(t) = v \cdot \sin(\psi(t)) \quad (3.25)$$

$$\dot{\psi}(t) = \frac{g}{v} \cdot \tan(\phi(t)) \quad (3.26)$$

$$|\phi(t)| \leq \phi_{max}$$

Hierbei sind  $n(t)$  und  $e(t)$  die Positionskoordinaten des UAVs im erdfesten Koordinatensystem und  $u(t) = \phi(t)$  ist die Steuereingabe für den Rollwinkel. Die Wenderate  $\dot{\psi}(t)$  ist eine Funktion des Rollwinkels  $\phi(t)$ ,  $g$  ist die Erdbeschleunigung und  $v$  ist die konstante Bahngeschwindigkeit (Fluggeschwindigkeit).

Aus der zeitkontinuierlichen Bewegungsgleichung kann ein Dubins-Pfad generiert werden, indem die (kontinuierliche) Rollwinkelvorgabe durch die diskrete Steuereingabe  $u(t) \in U = \{(L, S, L), (L, S, R), (R, S, L), (R, S, R), (L, R, L), (R, L, R)\}$  mit  $L = -\phi_{max}$  und  $R = \phi_{max}$  ersetzt wird [156, S. 127]. Für die Bewegungsprimitiven  $L$  und  $R$  ist in Gleichung (3.27) der Zusammenhang zwischen der Fluggeschwindigkeit  $v$ , dem maximalen Rollwinkel  $\phi_{max}$  und dem resultierenden minimalen Kurvenradius  $r_{min}$ , der indirekt proportional zur maximalen Flugbahnkrümmung  $\kappa_{max}$  ist, zusammengestellt.

$$\kappa_{max} = \frac{1}{r_{min}} = \frac{g}{v^2} \cdot \tan(\phi_{max}) \quad \text{mit } v \neq 0 \quad (3.27)$$

In Tabelle 3.6 sind die relevanten Parametereinstellungen für die Dubins-Pfadplanung zusammengefasst, die im weiteren Verlauf dieser Arbeit verwendet werden.

Tabelle 3.6: Parametereinstellung für die Dubins-Pfadplanung

Parameter	Wert	Bemerkung
UAV Fluggeschwindigkeit $v$	35 m/s	vorgegeben
maximaler Rollwinkel $\phi_{max}$	0.694 rad	vorgegeben
Erdbeschleunigung $g$	9.81 m/s <sup>2</sup>	
minimaler Kurvenradius $r_{min}$	150 m	aus Gleichung (3.27)

Gleichung (3.28) beschreibt die Berechnung der Bogenlänge  $s$  des Flugpfads von der Startkonfiguration  $a$  zur Zielkonfiguration  $b$ . Die Bogenlänge wird in Abschnitt 4.4.1 als

Bewertungskriterium für die Trajektorienoptimierung wieder aufgegriffen.

$$s = \int_a^b \sqrt{\dot{n}(t)^2 + \dot{e}(t)^2} dt \quad (3.28)$$

In Abb. 3.19 ist beispielhaft dargestellt, wie die zwei Punkte (Konfigurationen)  $a$  und  $b$  mit vorgegebenem Gierwinkel  $\psi_a$  und  $\psi_b$  durch einen Dubins-Pfad verbunden werden. Die Steuereingabe  $u(t)$  resultiert aus einer spezifischen Komposition der abschnittsweise konstanten Bewegungsprimitiven  $L$ ,  $S$  und  $R$  unter Berücksichtigung der konstanten Fluggeschwindigkeit  $v$ . Es wird angemerkt, dass aus einer (Flugbahn-)Krümmung von  $\kappa < \kappa_{max}$  für  $L$  bzw.  $R$  ein längerer Pfad resultiert, der nicht dem kürzest möglichen Dubins-Pfad entspricht.

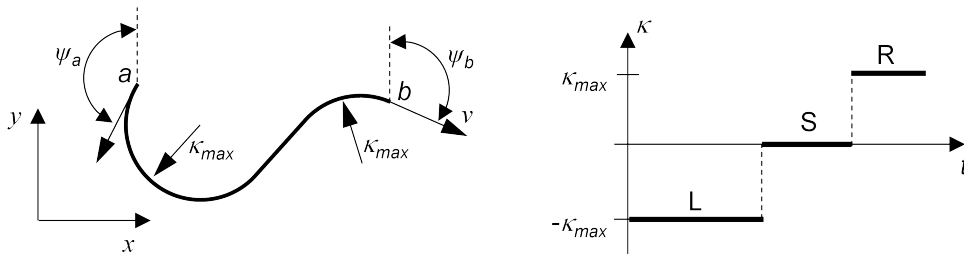


Abbildung 3.19: Beispiel für einen Dubins-Pfad (links) und zugehörigem Krümmungsverlauf (rechts) (in Anlehnung an [156, S. 127]).

Für die Lösung des Dubins-Pfadplanungsproblems wird auf die Literatur verwiesen. Lösungsansätze finden sich unter anderem in [157, 158] und [159, S. 33 ff.]. Darüber hinaus existieren fertige Software-Programmpakete für die Dubins-Pfadplanung, u. a. in der *Open Motion Planning Library*<sup>9</sup> (OMPL) oder in der *Navigation Toolbox* der proprietären Entwicklungsumgebung MATLAB<sup>10</sup>.

Abschließend wird noch darauf eingegangen, warum die Dubins-Pfadplanung mit abschnittsweise konstanter Bahnkrümmung in dieser Arbeit Verwendung findet, anstatt einem Verfahren mit kontinuierlich gekrümmten Pfaden, wie es u. a. in [160, 161] beschrieben wird. Der Vorteil kontinuierlich gekrümmter Pfade ist, dass die Rollwinkeländerung kontinuierlich erfolgt und damit eine glatte ( $C^2$ -stetige) Flugtrajektorie erzeugt werden kann. Andererseits muss die Start- und Endkonfiguration um einen weiteren Zustand

<sup>9</sup>Die *Open Motion Planning Library* ist ein quelloffenes Softwarepaket zur Berechnung von Bewegungsabläufen. Link auf die offizielle Webseite: <https://ompl.kavrakilab.org/>

<sup>10</sup>MATLAB ist eine von MathWorks entwickelte, proprietäre Multi-Paradigma-Programmiersprache und Entwicklungsumgebung für numerische Berechnungen. Link auf die offizielle Webseite: <https://www.mathworks.com/products/matlab.html>

### 3 Konzept und Methodenauswahl

für den Rollwinkel erweitert werden. In Verbindung mit der Optimierungsmethode der *dynamischen Programmierung* würde dies zu einem inakzeptablen Anstieg der Berechnungsdauer für die sensormodellbestimmte Befliegungsplanung führen. Eine detailliertere Erläuterung dieses Sachverhalts erfolgt in Abschnitt 3.6.2.

#### 3.5.4 Zusammenfassung

In diesem Abschnitt wurden die UAV-Bewegungsgleichungen sowohl in zeitkontinuierlicher als auch in zeitdiskreter Form dargestellt. Die Bewegungsgleichungen beschreiben die zeitliche Änderung des dynamischen Systems unter dem Einfluss von Steuereingaben. Sie dienen dazu, sensorplattformenspezifische Anforderungen und Beschränkungen (z. B. das maximale Lastvielfache, die Rollrate, die Fluggeschwindigkeit oder die Beschleunigung tangential zur Flugbahn) als Randbedingungen in die Trajektorienoptimierung mit einfließen zu lassen. Sie tragen somit dazu bei, die sensorplattformenspezifischen Anforderungen aus der *Anforderung 3* in Abschnitt 3.2 umzusetzen.

Die *zeitkontinuierlichen Bewegungsgleichungen* werden verwendet, um in Abschnitt 4.3 unter Verwendung der *nichtlinearen modellprädiktiven Regelung* lokal optimale Referenztrajektorien für eine gegebene Startkonfiguration zu generieren. Die zeitkontinuierlichen Bewegungsgleichungen sind daher essentiell, um die *Forschungsfrage 3.2* aus Abschnitt 2.4 beantworten zu können.

Demgegenüber werden die *zeitdiskreten Bewegungsgleichungen*, insbesondere die Dubins-Pfadplanung genutzt, um mit Hilfe der *dynamischen Programmierung* in Abschnitt 4.4 global optimale Referenztrajektorien zu generieren. Die zeitdiskrete Formulierung der Bewegungsgleichungen ist somit wesentliches Mittel, um die *Forschungsfrage 3.1* aus Abschnitt 2.4 beantworten zu können. Beide *Forschungsfragen 3.1* und *3.2* sind wiederum notwendig, um das damit verbundene *Zwischenziel 3* aus Abschnitt 1.3 erfüllen zu können.

## 3.6 Mathematische Methoden zur Trajektorienoptimierung

Ausgehend vom *Zwischenziel 3* in Abschnitt 1.3 wurden im Stand der Forschung unterschiedliche Methoden zur Trajektorienoptimierung vorgestellt. Aus diesen Optimierungsmethoden wurden in Abschnitt 2.2.4 bereits zwei Methoden ausgewählt, die einen wesentlichen Beitrag für die Beantwortung der *Forschungsfragen 3.1* und *3.2* aus Abschnitt 2.4 leisten können. Bei diesen Methoden handelt es sich um die *dynamische Programmierung*, die in Abschnitt 3.6.2 vorgestellt wird, und die *nichtlineare modellprädiktive Regelung*, auf die in Abschnitt 3.6.3 eingegangen wird.

Sowohl die modellprädiktive Regelung als auch die dynamische Programmierung gründen im Kern auf einem *Optimalsteuerungsproblem* (OSP). Aus diesem Grund werden im folgenden Abschnitt zuerst die Grundlagen der Optimalsteuerungsprobleme in der zeitkontinuierlichen und zeitdiskreten Form untersucht. Durch das OSP kann das flugdynamische System des UAVs, zusammen mit dessen Limitierungen (z. B. Fluggeschwindigkeit, Rollrate, Lastvielfaches), explizit im Optimierungsprozess zur sensormodellbestimmten Befliegungsplanung berücksichtigt werden. Dies ist entscheidend, um der *Anforderung 3* aus Abschnitt 3.2, zur Umsetzung sensorplattformspezifischer Eigenschaften, gerecht werden zu können.

In Kapitel 4 wird das funktionale Konzept der sensormodellbestimmten Befliegungsplanung in ein Experimentalsystem überführt. Zu diesem Zweck wird die dynamische Programmierung mit einem nachgeschalteten Verfahren zur Trajektorienglättung (siehe Abschnitt 4.4.3) kombiniert. Demgegenüber wird vor der nichtlinearen modellprädiktiven Regelung ein Verfahren zur UAV-Pfadplanung (siehe Abschnitt 4.3.1) durchgeführt, um die Wirkung perzeptiver Einflüsse auf die Trajektorienoptimierung berücksichtigen zu können. Die Ergebnisse aus diesem Abschnitt stellen daher nur einen Zwischenschritt auf dem Weg zur vollständigen Beantwortung der *Forschungsfragen 3.1, 3.2* und dem *Zwischenziel 3* dar.

#### 3.6.1 Optimalsteuerungsprobleme

In diesem Abschnitt werden die theoretischen Grundlagen der *optimalen Steuerung* für zeitkontinuierliche und zeitdiskrete dynamische Systeme vorgestellt. Ganz allgemein betrachtet hat die optimale Steuerung zum Ziel, eine *optimale Steuerstrategie* für ein dynamisches System zu bestimmen, wodurch eine spezifische Zielfunktion<sup>11</sup> minimiert wird und gleichzeitig ausgewählte Systemzustands- und/oder Steuerungsbeschränkungen eingehalten werden.

Der zeitliche Verlauf des Systemzustands kann durch Steuereingaben beeinflusst werden, wobei hier vorausgesetzt wird, dass die Entwicklung des Systemzustands über die Zeit deterministisch verläuft. Die Zielfunktion umfasst die erforderlichen Zielgrößen, die bei der Optimierung berücksichtigt werden sollen. Das Ergebnis der Zielfunktion ist per Konvention zu minimieren.

Bei einer Vielzahl realer dynamischer Systeme ändert sich der Systemzustand kontinuierlich über die Zeit, so auch beim flugdynamischen System des UAVs. Aus diesem Grund ist es zweckmäßig, das Optimalsteuerungsproblem zuerst zeitkontinuierlich zu

---

<sup>11</sup>In der Fachliteratur wird die *Zielfunktion* häufig auch als *Zielfunktional* oder als *Kostenfunktion* bezeichnet. Die Begriffe werden daher in dieser Arbeit als äquivalent betrachtet.

### 3 Konzept und Methodenauswahl

modellieren und dieses anschließend durch Diskretisierung in ein zeitdiskretes Optimalsteuerungsproblem zu überführen.

#### 3.6.1.1 Zeitkontinuierliche Darstellung

Die Beschreibung eines allgemeinen *zeitkontinuierlichen Optimalsteuerungsproblems* ist in den Gleichungen (3.29) aufgetragen. Die Betrachtung der Systemdynamik (3.29b) wird im Folgenden auf gewöhnliche Differentialgleichungen erster Ordnung beschränkt. Eine Beschreibung durch Differential-Algebraische-Gleichungen ist u. a. in [162] zu finden.

Die Notation für das zeitkontinuierliche OSP ist angelehnt an [163]:

$$\underset{x(\cdot), u(\cdot)}{\text{minimiere}} \quad J(x, u) = \Phi(x(t_0), t_0, x(t_f), t_f) + \int_{t_0}^{t_f} L(x(t), u(t)) dt \quad (3.29a)$$

unter den Nebenbedingungen

$$\dot{x}(t) = F(x(t), u(t)) \quad (3.29b)$$

$$x(t_0) = x_0 \quad (3.29c)$$

$$h_{min} \leq h(x(t), u(t)) \leq h_{max} \quad (3.29d)$$

$$\Phi_{min} \leq \Phi(x(t_0), t_0, x(t_f), t_f) \leq \Phi_{max} \quad (3.29e)$$

$$u(t) \in \mathbb{U} \quad (3.29f)$$

Das Optimierungsziel beim OSP ist es, eine Steuerung  $u(t) \in \mathbb{R}^{n_u}$  für das dynamische System (3.29b) zu finden, die das Ergebnis der Zielfunktion  $J(x, u) \in \mathbb{R}$  (3.29a) (hier als Bolza-Problem [164, S. 131] dargestellt) mit dem Lagrange-Term  $L : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}$  und  $\Phi : \mathbb{R}^{n_x} \times \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R} \rightarrow \mathbb{R}$  für das Zeitintervall  $[t_0, t_f] \subset \mathbb{R}$  und  $t_0 < t_f$  minimiert.

Im Fall der UAV-Trajektorienoptimierung bildet die Differentialgleichung des dynamischen Systems (3.29b) die Bewegungsgleichungen (3.8) des UAVs aus Abschnitt 3.5.2.1 ab. Der Systemzustandsvektor  $x(t) \in \mathbb{R}^{n_x}$  ist analog zur Gleichung (3.9) aufgebaut. Der Anfangszustandsvektor  $x(t_0)$  zum Anfangszeitpunkt  $t_0$  wird mit  $x_0 \in \mathbb{R}^{n_x}$  bezeichnet und  $x_f \in \mathbb{R}^{n_x}$  ist der zugehörige Endzustandsvektor  $x(t_f)$  zum Endzeitpunkt  $t_f$ .

Durch  $h(\cdot)$  in (3.29d) werden die Systemzustands- und Steuerungsbeschränkungen definiert und mit  $\Phi(\cdot)$  in (3.29a) können Randbedingungen bezüglich des Anfangs- und/oder Endzustands des Systems vorgegeben werden. Beispielsweise könnte dadurch eine präferierte Start- oder Endposition des UAVs (falls die UAV-Position Bestandteil des Systemzustands ist) in die Zielfunktion und damit in die Optimierung mit einfließen.

### 3.6 Mathematische Methoden zur Trajektorienoptimierung

Die Steuerbeschränkungen  $u(t)$  in (3.29f) können über Boxbeschränkungen der Form

$$\mathbb{U} := \{u \in \mathbb{R}^{n_u} \mid u_{i,min} \leq u_i \leq u_{i,max}, i = 1, \dots, n_u\} \quad (3.30)$$

berücksichtigt werden. Als Beispiel seien hier die Beschränkung der Rollrate  $\omega(t)$  oder der Beschleunigung  $a(t)$  des UAVs tangential zur Flugrichtung genannt.

In Abb. 3.20 ist das Prinzip des OSPs vereinfacht dargestellt. Der zeitliche Verlauf des Systemzustands  $x(t)$  wird bestimmt durch den Anfangszustand  $x_0$  und den Steuerungsvektor  $u(t)$ , wobei Steuerungs- und/oder Systemzustandsbeschränkungen (hier als gestrichelte Linie dargestellt) einzuhalten sind.

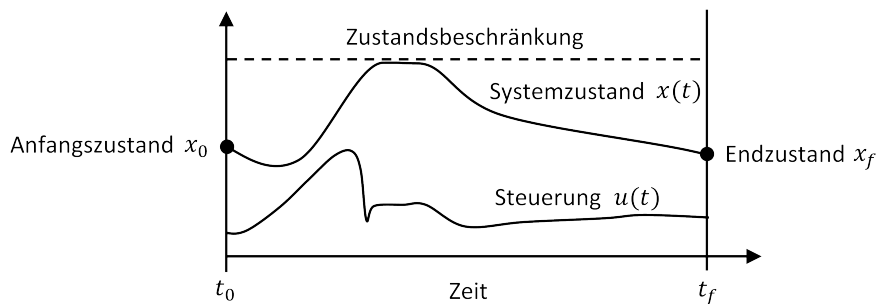


Abbildung 3.20: Prinzipdarstellung eines vereinfachten Optimalsteuerungsproblems. Der zeitliche Verlauf des Systemzustandsvektors  $x(t)$  wird bestimmt durch den Anfangszustand  $x_0$ , den Verlauf des Steuerungsvektors  $u(t)$  und ggf. durch Zustandsbeschränkungen (In Anlehnung an [165])

Durch die Zielfunktion wird das Ziel der Optimierung definiert. Beispielsweise kann eine Zielfunktionen so formuliert werden, dass die Abweichung des Systemzustandsvektors von einem vorgegebenen Referenzwert minimiert werden soll. Des Weiteren können mit ihr unterschiedliche Kriterien quantitativ gegeneinander gewichtet werden, was in Abschnitt 4.3.2 wieder aufgegriffen wird, um den Nutzen der Detektionsleistung gegenüber den Kosten für die Fluglageänderung des UAVs abwägen zu können.

Für die Definition der notwendigen und hinreichenden Optimalitätsbedingungen, sowie der Existenz- und Eindeutigkeitsätze wird auf die Fachliteratur [162, 164, 166–168] verwiesen.

#### 3.6.1.2 Zeitdiskrete Darstellung

Das *zeitdiskrete Optimalsteuerungsproblem* kann aus dem zeitkontinuierlichen OSP hergeleitet werden, in dem die zeitkontinuierlichen Bewegungsgleichungen (3.29b) geeignet

### 3 Konzept und Methodenauswahl

diskretisiert werden, z. B. mit Hilfe des Euler-Verfahrens und dem Abtastzeitintervall  $\Delta t$ . Es wird vorausgesetzt, dass die zeitkontinuierlichen Bewegungsgleichungen zweifach differenzierbar sind. Auf Grund der Analogie des zeitdiskreten zum zeitkontinuierlichen OSP werden im Folgenden nur die wesentlichen Zusammenhänge und Unterschiede erklärt, für die Grundlagen wird auf den vorangegangenen Abschnitt 3.6.1.1 verwiesen. Die separate Darstellung des OSPs in zeit- und wertdiskreter Form wird als sinnvoll erachtet, da diese Form des OSPs insbesondere für die *dynamische Programmierung* in Abschnitt 3.6.2 genutzt wird. Für die detailliertere Betrachtung der Diskretisierung zeitkontinuierlicher Systeme sowie der Stabilitätsuntersuchung zeitdiskreter Systeme wird auf Grüne & Pannek [169] verwiesen.

Das zeitdiskrete OSP in seiner allgemeinen Form kann entsprechend den Gleichungen (3.31) formuliert werden. Die Nomenklatur ist angelehnt an [170]:

$$\underset{x_0, u_0, x_1, u_1, \dots, u_{N-1}, x_N}{\text{minimiere}} \quad J(x, u) = E(x_N) + \sum_{k=0}^{N-1} L(x_k, u_k) \quad (3.31a)$$

unter den Nebenbedingungen

$$x_{k+1} - f(x_k, u_k) = 0, \quad k = 0, \dots, N-1 \quad (3.31b)$$

$$x_k \in S_k, \quad k = 0, \dots, N \quad (3.31c)$$

$$u_k \in U_k, \quad k = 0, \dots, N-1 \quad (3.31d)$$

$$r(x_0, x_N) = 0 \quad (3.31e)$$

Hierbei ist  $x = (x_0, \dots, x_N)^T \in \mathbb{R}^{(N+1)n_x}$  der zeitdiskrete Systemzustandsvektor,  $u = (u_0, \dots, u_{N-1})^T \in \mathbb{R}^{Nn_u}$  ist der Steuerungsvektor und  $k$  ist der diskrete Zeitschritt im Zeithorizont der Länge  $N$ . Das flugdynamische System des UAVs wird durch die Differenzengleichung in (3.31b) beschrieben. Hierbei wird der Zustandsübergang von einem Zustand  $x_k$  im Zeitschritt  $k$  zum nächsten Zustand  $x_{k+1}$  im darauffolgenden Zeitschritt  $k+1$  durch die Steuereingabe  $u_k$  beeinflusst. Bei der *dynamischen Programmierung* werden diese Zustandsübergänge durch Dubins-Pfade modelliert, wie sie in Abschnitt 3.5.3 vorgestellt wurden.

Gleichung (3.31a) gibt das Gütemaß der Zielfunktion  $J(x, u) \in \mathbb{R}$  an, das aus den Kosten  $E(x_N)$  im finalen Zeitschritt und der Summe der zeitschrittspezifischen Kosten  $L(x_k, u_k)$  gebildet wird. Das Ziel beim OSP ist es, den Steuerungsvektor  $u_k$  so zu wählen, dass die Zielfunktion für die diskreten Zeitschritte  $k$  minimiert wird, bei gleichzeitiger Einhaltung der Randbedingungen (3.31b) bis (3.31e). Des Weiteren können mit Hilfe von Gleichung (3.31c) Systemzustandsbeschränkungen und mit Gleichung (3.31d) Steuerungs-



### 3.6 Mathematische Methoden zur Trajektorienoptimierung

beschränkungen (z. B. eine Rollwinkel- und Rollratenbegrenzung) des OSPs realisiert werden. Durch Gleichung (3.31e) kann zudem der Anfangs- und/oder der Endzustand des Systems beeinflusst werden.

Die Abbildung  $\mu_k$  (3.32) bildet die Systemzustände  $x_k \in S_k$  auf die Steuereingaben  $u_k$  ab:

$$\begin{aligned} u_k &= \mu_k(x_k) \\ \mu_k(x_k) &\in U_k(x_k) \forall x_k \in S_k \end{aligned} \quad (3.32)$$

Mit Hilfe einer Sequenz aus zulässigen  $\mu_k$  wird die Steuerstrategie  $\pi$  (3.33) über alle Zeitschritte abgebildet.

$$\begin{aligned} \pi &= \{\mu_0, \dots, \mu_{N-1}\} \in \Pi \\ \Pi &= \{\pi = (\mu_0, \dots, \mu_{N-1}) \mid \mu_k : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_u}\} \end{aligned} \quad (3.33)$$

Aus Gleichung (3.31b), einer vorgegebenen Steuerstrategie  $\pi$  und einem spezifischen Startwert des Systemzustandsvektors  $x_0 \in S_k$  kann die Zustandsänderung (3.34) berechnet werden.

$$\begin{aligned} x_{k+1}^{\pi, x_0} &= f(x_k^{\pi, x_0}, \mu_k(x_k^{\pi, x_0})) \\ k &= 0, \dots, N-1 \end{aligned} \quad (3.34)$$

Die *optimale Steuerstrategie*<sup>12</sup>

$$\pi^* = \{\mu_0^*, \mu_1^*, \dots, \mu_{N-1}^*\} \quad (3.35)$$

zeichnet sich dadurch aus, dass die Gesamtkosten  $J_{\pi^*}(x_0)$  (3.37) der Zielfunktion (3.36) für einen gegebenen Anfangszustandsvektor  $x_0$  minimiert werden, unter Einhaltung der spezifizierten Nebenbedingungen. Die kumulierten Gesamtkosten einer zulässigen Steuerstrategie sind beschränkt auf  $J_\pi < \infty$ .

$$J_\pi(x_0) = E(x_N^{\pi, x_0}) + \sum_{k=0}^{N-1} L(x_k^{\pi, x_0}, \mu_k(x_k^{\pi, x_0})) \quad (3.36)$$

$$J_{\pi^*}(x_0) = \min_{\pi \in \Pi} J_\pi(x_0) \quad (3.37)$$

---

<sup>12</sup>Die Suche nach der *optimalen Steuerstrategie* hat den Oberbegriff *optimale Steuerung* (engl.: *Optimal Control*) für dieses Teilgebiet der mathematischen Optimierung geprägt.

### 3 Konzept und Methodenauswahl

Auf Grundlage der optimalen Steuerstrategie  $\pi^*$  kann für jeden spezifischen Anfangszustand  $x_0$  eine Sequenz (3.38) ermittelt werden, die dem zeitschrittspezifischen Systemzustand eine optimale Steuereingabe zuweist, wodurch die Gesamtkosten der Zielfunktion (3.37) minimiert werden.

$$\{(x_0, u_0), (x_1, u_1), \dots, (x_{N-1}, u_{N-1})\} \quad (3.38)$$

Im folgenden Abschnitt wird eine Übersicht verschiedener Lösungsverfahren für zeitkontinuierliche und zeitdiskrete Optimalsteuerungsprobleme gegeben.

#### 3.6.1.3 Lösungsverfahren für Optimalsteuerungsprobleme

Wie in den vorangegangenen Abschnitten bereits angesprochen wurde, ist es das Ziel des OSPs eine optimale Steuerstrategie zu finden, die eine Zielfunktion minimiert unter Einhaltung von Randbedingungen. Für die numerische Lösung von Optimalsteuerungsproblemen wird in der Literatur zwischen *direkten* und *indirekten Lösungsverfahren* sowie der *dynamischen Programmierung* unterschieden (siehe Abb. 3.21). Eine Übersicht zu diesen Verfahren ist u. a. in [163, 164, 171] zu finden.

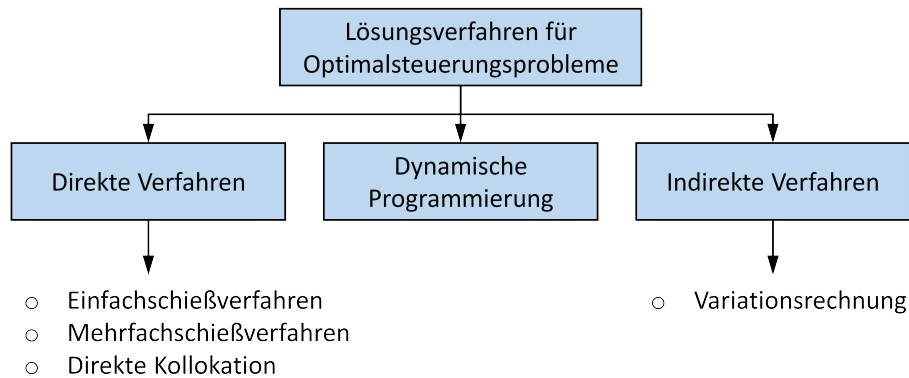


Abbildung 3.21: Numerische Verfahren zur Lösung von Optimalsteuerungsproblemen

Bei den *direkten Verfahren* zur Lösung von Optimalsteuerungsproblemen wird das zeitkontinuierliche OSP in ein endlich-dimensionales Optimierungsproblem mit diskretisierten Systemzuständen und/oder Steuereingaben überführt [172]. Die Diskretisierung kann durch einen Vertreter aus dem Gebiet der Schießverfahren oder durch Kollokation erfolgen. Zu der erst genannten Gruppe zählen das *direkte Einfachschießverfahren* [172] und das *direkte Mehrfachschießverfahren* [165, 173, 174]. Die *direkte Kollokation* [173] ist ein Vertreter der zweiten Gruppe.

### 3.6 Mathematische Methoden zur Trajektorienoptimierung

Die *dynamische Programmierung* ist ein Lösungsverfahren mit dem ein zeit- und wertdiskretes OSP numerisch gelöst werden kann. Das Verfahren wurde von Bellman [87] etabliert und basiert auf dem von ihm beschriebenen *Prinzip der Optimalität*. Hierbei wird ein komplexes Optimierungsproblem in einfachere Teilprobleme zerlegt, um dann rekursiv die optimalen Lösungen für die Teilprobleme zu finden. Ein Optimierungsproblem hat eine *optimale Substruktur*, wenn eine optimale Lösung aus den optimalen Lösungen seiner Teilprobleme konstruiert werden kann [87, 150]. Die dynamische Programmierung wird im folgenden Abschnitt 3.6.2 wieder aufgegriffen, um die UAV-Flugtrajektorienoptimierung für den zeitdiskreten Ansatz der sensormodellbestimmten Befliegungsplanung umzusetzen.

Bei den *indirekten Verfahren* wird die *Variationsrechnung* [175] genutzt, um die Optimalitätsbedingungen erster Ordnung des OSPs zu bestimmen. Hieraus resultiert ein Mehrpunkt-Randwertproblem aus dessen Lösung optimale Trajektorien resultieren, die als Extremale bezeichnet werden. Im nächsten Schritt wird jedes dieser Extremale daraufhin untersucht, ob es sich um ein lokales Minimum, Maximum oder einen Sattelpunkt handelt. Von den lokal optimalen Lösungen wird das Extremale mit den geringsten Kosten ausgewählt [163].

In Stryk & Bulirsch [176] wird zudem ein *hybrider Ansatz* aus einem direkten und indirekten Lösungsverfahren vorgestellt, worauf hier nicht weiter eingegangen wird.

#### 3.6.2 Dynamische Programmierung

Im vorangegangenen Abschnitt wurde eine kurze Übersicht zu unterschiedlichen Lösungsverfahren von Optimalsteuerungsproblemen gegeben. Im Folgenden wird näher auf das Verfahren der *dynamischen Programmierung* (DP) eingegangen. In dieser Arbeit ist DP das grundlegende Optimierungsverfahren zur sensormodellbestimmten Befliegungsplanung für zeit- und wertdiskrete Systeme, das in Abschnitt 4.4 detailliert beschrieben wird. Mit Hilfe der DP wird die Generierung *global optimaler Referenztrajektorien* ermöglicht. Das Resultat ist eine UAV-Flugtrajektorie (Referenztrajektorie) aus Dubins-Pfadsegmenten, die in Abschnitt 4.4.3 geglättet wird, um die Anforderungen an die Glattheit aus Abschnitt 3.5.1 zu erfüllen. Ein wesentlicher Vorteil von DP ist, dass für das Optimalsteuerungsproblem nicht-differenzierbare Systemdynamiken genutzt werden können, wie dies bei der Verwendung von Dubins-Pfaden erforderlich ist.

Die dynamische Programmierung wurde in den 1950er Jahren insbesondere von Richard Bellman [87] entwickelt. DP basiert auf dem *Prinzip der Optimalität*, das ebenfalls von Bellman definiert wurde und besagt:

### 3 Konzept und Methodenauswahl

„An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.“ [87, S. 83]

Das Optimalitätsprinzip kann nach [150, S. 20 f.] vereinfacht so beschrieben werden, dass jede Subtrajektorie einer optimalen Trajektorie selbst wieder eine optimale Trajektorie ist. Sei  $\pi^*$  (Gleichung (3.35) aus Abschnitt 3.6.1.2) die optimale Steuerungsstrategie für ein OSP, dann ist  $\{\mu_l^*, \mu_{l+1}^*, \dots, \mu_{N-1}^*\}$  die verkürzte optimale Steuerstrategie für das Teilproblem vom Zeitpunkt  $l$  bis zum finalen Zeitschritt  $N$ , die die Kosten der Zielfunktion  $J_{\pi^*}(x_l)$  minimiert, mit

$$J_{\pi^*}(x_l) = \min_{\pi \in \Pi} \left\{ E(x_N^{\pi, x_0}) + \sum_{k=l}^{N-1} L(x_k^{\pi, x_0}, \mu_k(x_k^{\pi, x_0})) \right\} \quad (3.39)$$

Die Optimierung bei der dynamischen Programmierung beginnt somit beim letzten Zeitschritt  $k = N$  und verläuft rückwärts bis zum ersten Zeitschritt  $k = 0$ , was auch als *Rückwärtsrechnung* bezeichnet wird. Darüber hinaus existiert eine äquivalente Methode in umgekehrter Reihenfolge, die als *Vorwärtsrechnung* bekannt ist. Für weitergehende Informationen wird diesbezüglich auf Bertsekas [150] verwiesen.

Jedem Zeitschritt  $k$  werden die Zustände in Gleichung (3.40), und dem darauf folgenden Zeitschritt  $k + 1$  werden die Zustände in Gleichung (3.41) zugewiesen:

$$x_k^i, \quad i = 1, \dots, m_{state,k} \text{ mit } m_{state,k} \in \mathbb{N} \text{ in } k \in \{0, \dots, N\} \quad (3.40)$$

$$x_{k+1}^j, \quad j = 1, \dots, m_{state,k+1} \text{ mit } m_{state,k+1} \in \mathbb{N} \text{ in } k \in \{0, \dots, N-1\} \quad (3.41)$$

$$x_k^i = [e_k^i, n_k^i, \psi_k^i]^T \quad (3.42)$$

Hierbei ist  $x_k^i$  der Zustandsvektor des zeit- und wertdiskreten Systems zum Zeitschritt  $k$ . Zum Zweck der Trajektorienoptimierung umfasst der Zustandsvektor die Positionskordinaten  $e_k^i$ ,  $n_k^i$  und den Gierwinkel  $\psi_k^i$  des UAVs.  $i$  und  $j$  sind zeitschrittabhängige Indizes zur Verdeutlichung unterschiedlicher Zustandsausprägungen. Die Anzahl der verschiedenen Zustandsausprägungen  $m_{state,k}$  zu jedem Zeitschritt ist in Gleichung (3.43) aufgetragen und resultiert aus der Diskretisierung über die Anzahl der nördlichen und östlichen Positionen ( $m_{north,k}$  und  $m_{east,k}$ ) sowie der Anzahl der unterschiedlichen Gierwinkel  $m_{\psi,k}$ . Auf die Ermittlung der Anzahl der Positionen und Gierwinkel wird in Abschnitt 4.4.1 eingegangen.

$$m_{state,k} = m_{north,k} \cdot m_{east,k} \cdot m_{\psi,k} \quad (3.43)$$

### 3.6 Mathematische Methoden zur Trajektorienoptimierung

Jedem Paar von Zuständen  $x_k^i$  in Zeitschritt  $k$  und  $x_{k+1}^j$  in Zeitschritt  $k + 1$  können die *Zustandsübergangskosten*

$$c_{trans,k}^{ij} = c_{trans}(x_k^i, x_{k+1}^j) \quad (3.44)$$

zugeordnet werden. Für den Fall, dass im Zeitschritt  $k$  keine Steuerstrategie existiert um den Zustand von  $i$  nach  $j$  zu überführen, wird  $c_{trans,k}^{ij} = \infty$  festgelegt. Im Gegensatz dazu resultieren aus einer validen Steuerstrategie Zustandsübergangskosten von  $c_{trans,k}^{ij} < \infty$ .

Weiterhin sind jedem Systemzustand  $x_k^i$  im Zeitschritt  $k$  *zustandsabhängige Kosten*

$$c_{state,k}^i = c_{state}(x_k^i) \quad (3.45)$$

zugewiesen. In Abschnitt 4.4 wird beschrieben, welchen Zweck die Zustandsübergangskosten und die zustandsabhängigen Kosten für die Trajektorienoptimierung erfüllen.

Werden die zustandsabhängigen Kosten (3.45) in die allgemeine Zielfunktion (3.31a) eingesetzt, erhält man aus der Rückwärtsrechnung die Kosten (3.46) für den letzten Zeitschritt  $k = N$ .

$$J(x_N^i) = E(x_N) = c_{state,N}^i, \quad \forall x_N^i \in S_N \quad (3.46)$$

Mit der Fortsetzung der Rückwärtsrechnung und unter Anwendung des Prinzips der Optimalität folgen aus Gleichung (3.47) die minimalen Gesamtkosten  $J(x_k^i)$  für die Zeitschritte  $k = 0, \dots, N - 1$ . Diese Gesamtkosten setzen sich zusammen aus den zustandsabhängigen Kosten  $c_{state,k}^i$  des aktuell betrachteten Zustands, den Zustandsübergangskosten  $c_{trans,k}^{ij}$  zum nachfolgenden Zustand und den minimalen Gesamtkosten  $J(x_{k+1}^j)$  von diesem Nachfolgezustand bis zum letzten Zustand.  $p_w$  ist ein Faktor zur Gewichtung der zustandsabhängigen Kosten gegenüber den Zustandsübergangskosten. Als Vorgriff auf Abschnitt 4.4.1 wird an dieser Stelle erwähnt, dass mit Hilfe des Gewichtungsfaktors der Einfluss der Detektionsleistung gegenüber der Länge der Flugtrajektorie, und damit der Aufklärungsdauer, priorisiert werden kann.

$$J(x_k^i) = L(x_k, u_k) = \min_{j=0, \dots, n_{k+1}} \left\{ p_w \cdot c_{state,k}^i + (1 - p_w) \cdot c_{trans,k}^{ij} + J(x_{k+1}^j) \right\} \quad (3.47)$$

$$\forall x_k^i \in S_k, \quad k = 0, \dots, N - 1$$

$$p_w \in [0, 1]$$

### 3 Konzept und Methodenauswahl

In Abb. 3.22 sind die Systemzustände (Kreise), die zustandsabhängigen Kosten (Index „st“) und die Zustandsübergangskosten (Index „tr“) abstrahiert für die zwei Zeitschritte von  $k$  nach  $k + 1$  in einem azyklischen Graphen aufgetragen. Die Zustandsausprägungen  $m_{state,k}$  werden durch  $m_k$  abgekürzt.

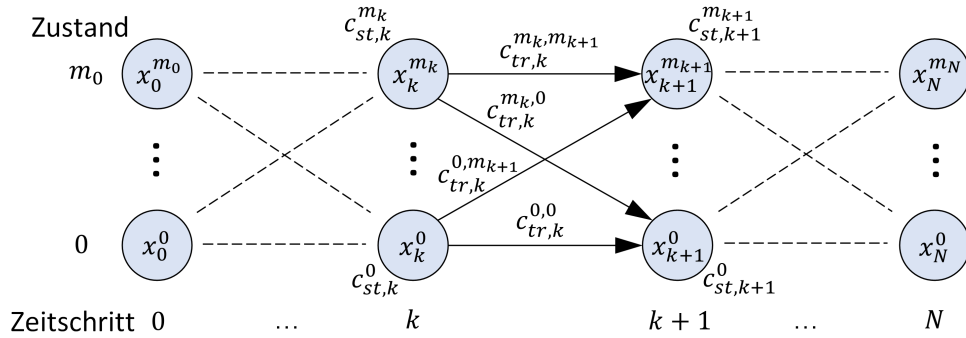


Abbildung 3.22: Darstellung der Zustände und der Zustandsübergänge in einem azyklischen Graph. Kreise repräsentieren die Zustände  $0, \dots, m_k$  in den einzelnen Zeitschritten  $0, \dots, N$ . Die Pfeile beschreiben die Zustandsübergänge zwischen zwei Zuständen. Beispielhaft sind die zustandsabhängigen Kosten  $c_{st,k}^i$  und die Zustandsübergangskosten  $c_{tr,k}^{i,j}$  von Zeitschritt  $k$  nach  $k + 1$  aufgetragen (übernommen und angepasst aus [33])

Die Rückwärtsrechnung der dynamischen Programmierung mit zustandsabhängigen Kosten und Zustandsübergangskosten läuft wie folgt ab:

1. Für jeden Zustand  $x_N^i$  im letzten Zeitschritt  $N$  werden die minimalen Gesamtkosten  $J(x_N^i)$  mit 0 initialisiert
2. Für jeden Zustand  $x_N^i$  im letzten Zeitschritt  $N$  werden die zustandsabhängigen Kosten  $c_{state,N}^i$  berechnet und den jeweiligen minimalen Gesamtkosten  $J(x_N^i)$  zugewiesen.
3. Für die Zeitschritte  $k = N - 1, \dots, 0$  werden für jeden Zustand  $x_k^i$  die zustandsabhängigen Kosten  $c_{state,k}^i$ , sowie die Zustandsübergangskosten  $c_{trans,k}^{i,j}$  von Zustand  $i$  zum nachfolgenden Zustand  $j$  berechnet. Abschließend werden für jeden Zustand  $x_k^i$  die minimalen Gesamtkosten  $J(x_k^i)$  entsprechend der Gleichung (3.47) bestimmt, woraus der optimale Nachfolgezustand resultiert.
4. Der Vorgang wird ab Punkt 3 so lange wiederholt, bis alle Zeitschritte abgearbeitet sind und  $k = 0$  erreicht ist.

### 3.6 Mathematische Methoden zur Trajektorienoptimierung

Aus der Rückwärtsrechnung und dem Prinzip der Optimalität folgt, dass jeder Zustand in den Zeitschritten  $k = 0, \dots, N - 1$  einen dezidierten optimalen (kostengünstigsten) Nachfolgezustand hat, der die minimalen Gesamtkosten aller optimalen nachfolgenden Zustände in sich vereint. Jeder Zustand ist somit Ausgangspunkt einer optimalen Subtrajektorie. Aus der iterativen Fortsetzung der Berechnung der optimalen Subtrajektorie resultiert die optimale Trajektorie für einen spezifischen Anfangszustand.

Auf Grundlage der minimalen Gesamtkosten kann eine *optimale Steuerung* (3.48)

$$u_k^*(x_k) = \mu_k^*(x_k) = \arg \min_{u_k \in U_k(x_k)} \{L(x_k, u_k) + J_{k+1}(f(x_k, u_k))\} \quad (3.48)$$

für jeden Systemzustand  $x_k$  bestimmt werden [150]. Im Fall der zeit- und wertdiskreten Trajektorienoptimierung beschreibt die optimale Steuerung eine zeitschrittabhängige spezifische Kombination aus drei Bewegungsprimitiven ( $L, S, R$ ), die einen Dubins-Pfad (vgl. Abschnitt 3.5.3) bilden.

Die global optimale Trajektorie kann ermittelt werden, indem die Gesamtkosten  $J(x_0^i)$  aller zugehörigen Anfangszustände  $x_0^i$  miteinander verglichen und die global minimalen Gesamtkosten  $J(x_0^*)$  identifiziert werden. Der Anfangszustand  $x_0^*$  ist damit Startpunkt (genauer: Startzustand) der, bezüglich der Diskretisierung, *global optimalen Trajektorie*.

Für eine ausführliche und detaillierte Beschreibung der Algorithmen zur dynamischen Programmierung wird auf Bellman [87] und Bertsekas [150] verwiesen.

Ein wesentlicher Nachteil der dynamischen Programmierung ist, dass die Zielfunktion für jede Zustandsausprägung (vgl. Gleichung (3.43)) berechnet werden muss. Der Berechnungsaufwand steigt daher proportional mit der Anzahl der Zustandsausprägungen und der Anzahl der Zeitschritte. Bellman [87] hat hierfür den treffenden Begriff „*der Fluch der Dimensionalität*“ (original: *the curse of dimensionality*) geprägt. Um die Berechnungsdauer in akzeptablen Grenzen<sup>13</sup> zu halten, ist daher eine zweckmäßige Diskretisierung des Zustandsraums zwingend erforderlich. Dies ist besonders entscheidend, wenn die Trajektorienoptimierung unter begrenzter Rechenleistung und/oder für (harte) Echtzeitanforderungen erfolgen soll.

Die Einbettung der dynamischen Programmierung in das Verfahren zur sensormodellbestimmten Befliegungsplanung zur Generierung global optimaler Referenztrajektorien wird in Abschnitt 4.4 beschrieben.

---

<sup>13</sup>Wie lange die akzeptable Berechnungsdauer ist, hängt im Wesentlichen von der Anwendung ab und muss im Rahmen der Softwareentwicklung geklärt werden.

### 3.6.3 Nichtlineare modellprädiktive Regelung

Nach der Vorstellung der dynamischen Programmierung als eine Lösungsmethode für zeit- und wertdiskrete Optimalsteuerungsprobleme wird in diesem Abschnitt auf die *nicht-lineare modellprädiktive Regelung* (engl.: Nonlinear Model Predictive Control, NMPC) eingegangen. Der Ansatz zur Nutzung von NMPC für die sensormodellbestimmte Befliehungsplanung wurde in den vorangegangenen Arbeiten von Zwick, Gerds & Stütz [33, 149] vorgestellt und wird im Verlauf dieses Abschnitts detailliert erläutert.

Im Allgemeinen wird bei der modellprädiktiven Regelung das zu regelnde dynamische System (z. B. das flugdynamische System des UAVs) durch ein Modell abstrahiert, um damit die zukünftige Entwicklung des Systemzustands, unter dem Einfluss von Steuereingaben, zu prognostizieren. Das Ziel ist es, eine optimale Steuerstrategie zu ermitteln, die die Zielfunktion des Optimalsteuerungsproblems (vgl. Abschnitt 3.6.1) für jeden Zeitschritt minimiert. Aus der optimalen Steuerstrategie kann abschließend die Referenztrajektorie generiert werden.

Dieser Abschnitt zur modellprädiktiven Regelung ist in mehrere Unterabschnitte aufgeteilt. Zu Beginn wird auf die Diskretisierung der Systemdynamik eingegangen, bevor das Funktionsprinzip von NMPC erläutert wird. Dem folgt eine Abgrenzung zwischen linearer und nichtlinearer modellprädiktiver Regelung. Abschließend wird auf die mathematischen Grundlagen von NMPC eingegangen.

#### 3.6.3.1 Diskretisierung der Systemdynamik

Die Dynamik physikalischer Systeme wird in der Regel durch zeitkontinuierliche Differentialgleichungen beschrieben. Zu diesem Zweck wurden in Abschnitt 3.6.1.1 die Grundlagen zur zeitkontinuierlichen Modellierung des OSPs gelegt. Das OSP beschreibt das flugdynamische System des UAVs, die Zielfunktion sowie die relevanten Systemzustandsbeschränkungen (z. B. Begrenzung der Fluggeschwindigkeit und des Rollwinkels) und Steuerungsbeschränkungen (u. a. die Rollrate). Die Berechnung der modellprädiktiven Regelung erfolgt auf Rechnersystemen oder eingebetteten Systemen (z. B. an Bord des UAVs), wofür das zeitkontinuierliche OSP zweckmäßig zeitlich diskretisiert werden muss. Hierfür wurde das zeitkontinuierliche OSP aus Abschnitt 3.6.1.1 durch die Diskretisierung mit dem Zeitschrittintervall  $\Delta t$  in die zeitdiskrete Darstellung aus Abschnitt 3.6.1.2 überführt. Das hierfür genutzte Zeitschrittintervall wurde experimentell ermittelt und so festgelegt, dass eine problemspezifische Balance zwischen der Genauigkeit der berechneten Trajektorie und der erforderlichen Berechnungsdauer erreicht wird.



In der Literatur wird das dynamische System der modellprädiktiven Regelung oftmals in zeitkontinuierlicher Form dargestellt, obgleich es für die eigentliche Lösung des Optimierungsproblems geeignet diskretisiert werden muss (z. B. unter Berücksichtigung des Nyquist-Shannon-Abtasttheorems [177]). Ausgehend davon wird die modellprädiktive Regelung im weiteren Verlauf dieser Arbeit als *zeitkontinuierliche Trajektorienoptimierung* bezeichnet, wohingegen die dynamische Programmierung der *zeit- und wertdiskreten Trajektorienoptimierung* zugeordnet ist.

### 3.6.3.2 Funktionsprinzip der modellprädiktiven Regelung

In Abb. 3.23 ist das Funktionsprinzip der modellprädiktiven Regelung in einem Blockschaltbild visualisiert. Das Optimierungsproblem besteht darin, mit Hilfe von MPC die *Steuereingabe* für ein *dynamisches System* so zu berechnen, dass dieses System einer *Sollwertvorgabe* optimal folgt. Hierfür nutzt die modellprädiktive Regelung ein *Modell der Systemdynamik*, um das künftige Verhalten des realen Systems prognostizieren zu können.

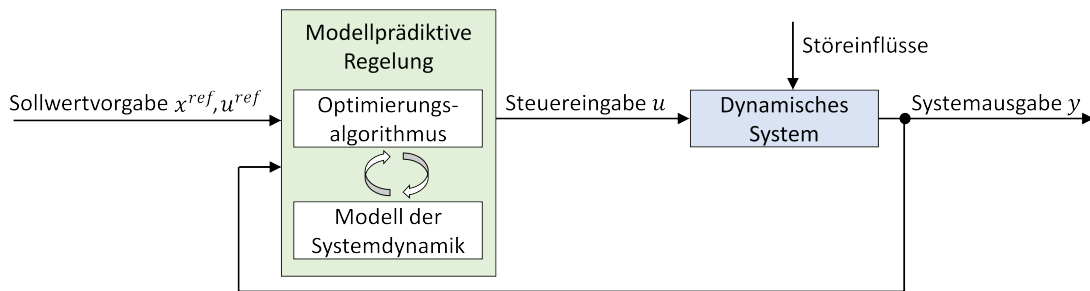


Abbildung 3.23: Blockschaltbild der modellprädiktiven Regelung und des zu regelnden dynamischen Systems

Unter Einbeziehung von Abb. 3.24 wird im Folgenden die modellprädiktive Regelung detaillierter betrachtet. Für den zum Zeitpunkt  $t_n$  gemessenen Systemzustand  $x_n$  wird das zukünftige Verhalten des Optimalsteuerungsproblems über einen endlichen Zeithorizont (auch Prädiktionshorizont) der Länge  $N$  optimiert. Das OSP wird für den Zeithorizont gelöst, so dass das dynamische System einer Sollwertvorgabe  $x_{n+k}^{ref}$  optimal folgt, unter Beachtung der Systemdynamik und der Zustands- und/oder Steuerungsbeschränkungen. Die erste Steuereingabe  $u_0^*$  der sich ergebenden optimalen Steuerstrategie  $\pi^*$  wird als rückgekoppelte Steuerung für die Dauer eines Zeitschritts  $\Delta t$  auf die Systemdynamik des Prozesses angewendet. Nach Ablauf eines Zeitschritts wird der neue Systemzustand  $x_{n+1}$

### 3 Konzept und Methodenauswahl

erfasst. Der Zeithorizont wird um einen Zeitschritt weiter in die Zukunft verschoben<sup>14</sup> und das OSP wird für  $t_{n+1}$  von neuem für den Zeithorizont gelöst. Durch die Messung und Rückkopplung des aktuellen Systemzustands wird der Regelkreis geschlossen. Dieser Vorgang wird zyklisch so oft wiederholt, bis das Optimierungsproblem für den gesamten Optimierungszeitraum  $t_n$  gelöst wurde.

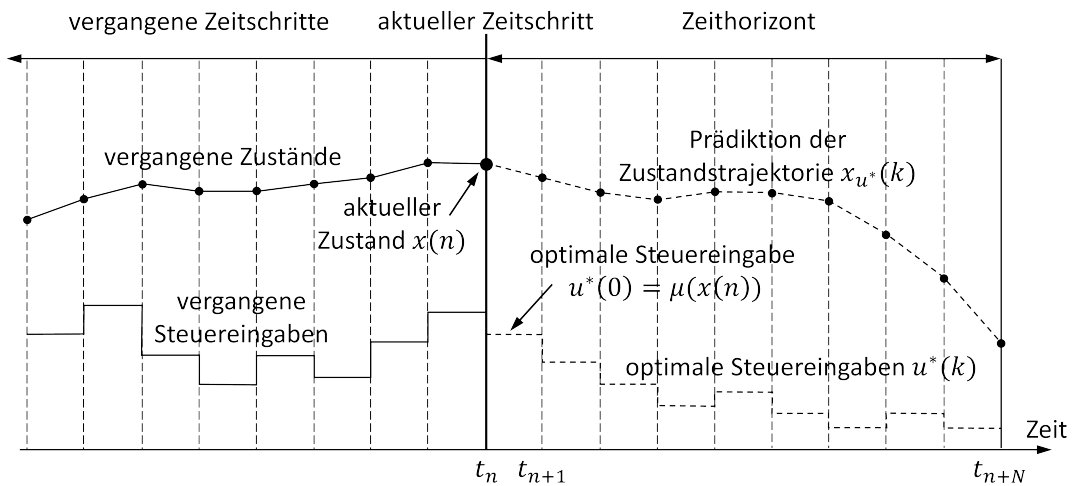


Abbildung 3.24: Prinzip der modellprädiktiven Regelung. Für den Zeithorizont werden optimale Steuereingaben berechnet, die die Zielfunktion des Optimalsteuerungsproblems minimieren (übernommen und angepasst aus [169])

#### 3.6.3.3 Lineare und nichtlineare modellprädiktive Regelung

Im Allgemeinen kann zwischen *linearer* und *nichtlinearer modellprädiktiver Regelung* unterschieden werden. Lineare MPC bezieht sich auf eine Familie von MPC-Schemata, bei denen lineare Modelle zur Vorhersage der Systemdynamik verwendet werden und lineare Beschränkungen für die Systemzustände und Eingänge berücksichtigt werden. Demgegenüber umfasst die nichtlineare modellprädiktive Regelung die MPC-Verfahren, die auf nichtlinearen Modellen beruhen und/oder eine nichtquadratische Zielfunktion verwenden sowie nichtlineare Nebenbedingungen berücksichtigen [178].

Die Theorie zur linearen MPC ist inzwischen sehr weit entwickelt. Wichtige Aspekte, wie die Online-Berechnung, das Zusammenspiel zwischen Modellierung, Identifikation und Steuerung, sowie systemtheoretische Fragen, wie z. B. die Stabilität, sind gut untersucht [179, 180]. Allerdings ist die lineare MPC ungeeignet für hochgradig nichtlineare

<sup>14</sup>Auf Grund der zyklischen Adaption des Zeithorizonts um jeweils einen Zeitschritt in die Zukunft, ist die modellprädiktive Regelung auch unter dem englischen Begriff „receding horizon control“ bekannt.

### 3.6 Mathematische Methoden zur Trajektorienoptimierung

Prozesse oder für Systeme, die eine moderate Nichtlinearität bei gleichzeitig großem Betriebsbereich aufweisen [181]. Dies ist das prädestinierte Anwendungsgebiet für die nichtlineare modellprädiktive Regelung.

Die zeitkontinuierlichen UAV-Bewegungsgleichungen (3.8) aus Abschnitt 3.5.2.1 beschreiben das Differentialgleichungssystem (3.29b) des zeitkontinuierlichen Optimalsteuerungsproblems. Auf Grund der nichtlinearen trigonometrischen Funktionen in den UAV-Bewegungsgleichungen erfolgt die Trajektorienoptimierung für zeitkontinuierliche Systeme in dieser Arbeit mit Hilfe der *nichtlinearen modellprädiktiven Regelung*. Ein weiteres Argument für die Verwendung von NMPC anstatt linearer MPC ist, dass eine Linearisierung der nichtlinearen Bewegungsgleichungen um den Arbeitspunkt, auf Grund der Länge des Prädiktionshorizonts, zu großen Abweichungen führen würde. Eine detailliertere Betrachtung bezüglich der nichtlinearen modellprädiktiven Regelung findet sich u. a. in den Arbeiten von Allgöwer et al. [178, 182] und Findeisen & Allgöwer [179].

Ein wesentlicher Vorteil von MPC ist, dass mit jedem Zeitschritt das OSP von neuem gelöst wird. Dadurch, dass zu jedem Zeitschritt der aktuelle Systemzustand in die Optimierung einfließt, erfolgt die Optimierung als geschlossener Regelkreis. Das erklärt, warum bei der modellprädiktiven Regelung von einer Regelung mit geschlossener Wirkungskette gesprochen wird. Durch den geschlossenen Regelkreis können äußere Störeinflüsse (vgl. Abb. 3.23) und Ungenauigkeiten zwischen der modellierten Systemdynamik und dem realen System ausgeglichen werden. Demgegenüber wird das eigentliche OSP zyklisch im offenen Regelkreis (als Steuerung) gelöst.

In der Regel ist ein zeitkontinuierliches OSP mit nichtlinearer Systemdynamik und/ohne Zustandsbeschränkungen nicht analytisch lösbar, da dies die explizite Lösung der Hamilton-Jacobi-Bellman Gleichungen [183] erfordert. Hierin liegt ein weiterer Vorteil von MPC. Dabei muss das OSP nur für sehr kurze Zeithorizonte gelöst werden, wofür effiziente Verfahren zur Verfügung stehen [178].

Nachfolgend ist eine kompakte Übersicht einiger weiterer Vorteile der modellprädiktiven Regelung zusammengestellt:

- MPC ist eines der wenigen Verfahren, bei dem harte Zustands- und/oder Steuerungsbeschränkungen direkt im Optimierungsprozess berücksichtigt werden können [182].
- Zukünftige Sollwertänderungen, die bekannt und innerhalb des Zeithorizonts liegen, fließen unmittelbar in die Optimierung des aktuellen Zeitschritts mit ein.
- Der zeitliche Verlauf der Sollwertvorgabe muss die Systemdynamik nicht exakt nachbilden.

### 3 Konzept und Methodenauswahl

Demgegenüber sollen auch die folgenden Nachteile erwähnt werden:

- Es muss ein geeignetes Modell gefunden und modelliert werden, um die Systemdynamik hinreichend genau abbilden zu können.
- Der zeitliche Verlauf der Sollwertvorgabe hat Einfluss auf das Optimierungsergebnis, allerdings werden die Systemdynamik, die Zielfunktion oder die Systembeschränkungen hierbei nicht explizit berücksichtigt.
- Aus der nichtlinearen Systemdynamik folgt ein im Allgemeinen nicht-konvexes Optimalsteuerungsproblem, für das nur lokal optimale Ergebnisse berechnet werden können [184].

In der Literatur finden sich zahlreiche Arbeiten, in denen die Theorie, verschiedene Algorithmen und Lösungsverfahren zur (nichtlinearen) modellprädiktiven Regelung im Detail beschrieben werden. Beispielhaft werden hier die Beiträge von Rawlings et al. [171], Grüne & Pannek [169], Camacho & Bordens [90] und Diehl et al. [185] genannt und darauf verwiesen.

#### 3.6.3.4 Mathematische Grundlagen von NMPC

Im Folgenden werden die mathematischen Grundlagen der nichtlinearen modellprädiktiven Regelung erläutert. Ausgangspunkt ist das zeitkontinuierliche Optimalsteuerungsproblem aus Abschnitt 3.6.1.1, das für die Verwendung in der modellprädiktiven Regelung mit dem Zeitschrittintervall  $\Delta t$  diskretisiert wird. Durch die Diskretisierung werden die *diskreten Zeitschritte*

$$t_k = k \cdot \Delta t, \quad \text{mit } \Delta t > 0 \text{ und } k = 0, \dots, N \quad (3.49)$$

innerhalb des Zeithorizonts der Länge  $N$  definiert. Es wird darauf hingewiesen, dass der Zeitschritt mit Index  $n$  den aktuellen Zeitpunkt der modellprädiktiven Regelung markiert und als *globaler Zeitschritt*  $t_n$  bezeichnet wird. Demgegenüber wird mit dem Index  $k$  der Zeitpunkt innerhalb des Zeithorizonts beschrieben und als *lokaler Zeitschritt*  $t_k$  definiert. Die Schrittweiten  $\Delta t$  der beiden Zeitgitter  $n$  und  $k$  sind identisch.

Die allgemeine Form des zeitdiskreten OSP aus Abschnitt 3.6.1.2 ist Ausgangspunkt für die nachfolgende Betrachtung der nichtlinearen modellprädiktiven Regelung. Um zwischen dem lokalen und globalen Zeitschritt differenzieren zu können, wird der Zeitschrittindex

### 3.6 Mathematische Methoden zur Trajektorienoptimierung

des OSPs wie folgt angepasst:

$$\underset{u_n, x_n, \dots, u_{n+N-1}, x_{n+N}}{\text{minimiere}} \quad J(x, u) = \sum_{k=0}^{N-1} L(x_{n+k}, u_{n+k}) \quad (3.50a)$$

unter den Nebenbedingungen

$$x_{n+k+1} = f(x_{n+k}, u_{n+k}) \quad (3.50b)$$

$$u_{n+k} \in U_{n+k}(x_{n+k}) \quad (3.50c)$$

$$x_{n+k} \in S_{n+k} \quad (3.50d)$$

Die Zielfunktion aus Gleichung (3.50a) ist in Gleichung (3.51) in der häufig verwendeten *quadratischen Form* aufgetragen. Die Vektoren  $x_{n+k}^{ref}$  und  $u_{n+k}^{ref}$  beschreiben zeitschrittspezifische Sollwertvorgaben für den Systemzustand und die Steuereingabe. Die Komponenten der beiden Vektoren wurden aus den Gleichungen (3.22) und (3.23) aus Abschnitt 3.5.2.2 übernommen.

$$L(x_{n+k}, u_{n+k}) = \tilde{x}_{n+k}^T Q \tilde{x}_{n+k} + \tilde{u}_{n+k}^T R \tilde{u}_{n+k} \quad (3.51)$$

$$\tilde{x}_{n+k} = x_{n+k} - x_{n+k}^{ref} \quad (3.52)$$

$$\tilde{u}_{n+k} = u_{n+k} - u_{n+k}^{ref} \quad (3.53)$$

$$x_{n+k}^{ref} = [e_{n+k}^{ref}, n_{n+k}^{ref}, \psi_{n+k}^{ref}, v^{ref}, \phi^{ref}]^T \quad (3.54)$$

$$u_{n+k}^{ref} = [a^{ref}, \omega^{ref}]^T \quad (3.55)$$

$Q$  und  $R$  in sind positiv definite, symmetrische Gewichtungsmatrizen, deren Diagonaleinträge in Tabelle 3.7 aufgelistet sind. Durch diese Matrizen kann festgelegt werden, welche Komponenten des Systemzustandsvektors und des Steuerungsvektors für die Berechnung der Zielfunktion berücksichtigt und wie stark diese gewichtet werden sollen. Dies hat unmittelbare Auswirkungen auf das Optimierungsergebnis, da hierfür die Zielfunktion minimiert werden soll. Aus den Gleichungen (3.51) bis (3.53) folgt, dass die Kosten der Zielfunktion minimal sind, wenn die Systemzustände und Steuereingaben den Sollwertvorgaben exakt entsprechen. Den Sollwertvorgaben für den Rollwinkel  $\phi^{ref}$ , die Fluggeschwindigkeit  $v^{ref}$ , die Beschleunigung  $a^{ref}$  und die Rollrate  $\omega^{ref}$  werden konstante Werte zugewiesen, die unabhängig vom aktuellen Zeitschritt sind. Die entsprechenden Werte sind in Tabelle 3.7 dokumentiert. Demgegenüber sind die Sollwertvorgaben der östlichen  $e_{n+k}^{ref}$  und nördlichen  $n_{n+k}^{ref}$  Position des UAVs sowie der Gierwinkel  $\psi_{n+k}^{ref}$  abhängig vom Zeitschritt. Auf den Ursprung und den Zweck dieser Sollwertvorgaben wird in Abschnitt 4.3.1 eingegangen.

### 3 Konzept und Methodenauswahl

In Tabelle 3.7 sind die Parametereinstellung zusammengefasst, die in dieser Arbeit für die nichtlineare modellprädiktive Regelung verwendet werden. Durch die Gewichte in den Gewichtungsmatrizen  $Q$  und  $R$  für die Sollwertvorgaben der Fluggeschwindigkeit, des Rollwinkels, der Beschleunigung in Flugrichtung und der Rollrate wird ein Flug des UAVs auf geradliniger Flugbahn und mit konstanter Fluggeschwindigkeit präferiert. Dadurch soll eine energieeffiziente Fortbewegung des UAVs erzielt werden. Demgegenüber sind in der Zielfunktion die Sollwert-Positionen ( $e_{n+k}^{ref}$  und  $n_{n+k}^{ref}$ ) höher gewichtet, um der Erzielung einer hohen, ortsabhängigen Detektionsleistung Vorrang gegenüber dem energieeffizienten UAV-Flug einzuräumen. Eine geradlinige Flugbahn stellt sich dann ein, wenn die Detektionsleistung keine oder nur eine geringe Ortsabhängigkeit aufweist.

Tabelle 3.7: Parametereinstellung für die nichtlineare modellprädiktive Regelung

Parameter	Wert	Bemerkung
Zeitschrittintervall $\Delta t$	0,5 s	vorgegeben
Zeitschritte im Zeithorizont $N$	10	vorgegeben
Sollwert für den Rollwinkel $\phi^{ref}$	0 rad	vorgegeben
Sollwert für die Geschwindigkeit $v^{ref}$	35 m/s	vorgegeben
Sollwert für die Beschleunigung $a^{ref}$	0 m/s <sup>2</sup>	vorgegeben
Sollwert für die Rollrate $\omega^{ref}$	0 rad/s	vorgegeben
Gewichtungsmatrix $Q$	1; 1; 0,1; 0,1; 0,1	vorgegeben
Gewichtungsmatrix $R$	0,5; 0,5	vorgegeben

Nach der Anpassung der Zeitschritte von  $k$  hin zu  $n + k$  für die Gleichungen (3.32) bis (3.36) können die minimalen Gesamtkosten  $J_{\pi^*}(x_n)$  für das OSP (vgl. Gleichungen (3.50)) zum Zeitschritt  $n$  über den gesamten Prädiktionshorizont berechnet werden:

$$J_{\pi^*}(x_n) = \min_{\pi \in \Pi} J_{\pi}(x_n) \quad (3.56)$$

Der Algorithmus zur Berechnung der nichtlinearen modellprädiktiven Regelung wird für jeden Abtastzeitpunkt  $t_n$  (mit  $n = 0, \dots, T$ ) des Optimierungsproblems wie folgt ausgeführt (in Anlehnung an [169]):

1. Der aktuelle Systemzustand  $x_n$  zum Zeitpunkt  $t_n$  wird gemessen.
2. Das Optimalsteuerungsproblem (3.50) wird für die quadratische Zielfunktion (3.51) und die Sollwertvorgaben  $x_{n+k}^{ref}$  und  $u_{n+k}^{ref}$  für alle lokalen Zeitschritte innerhalb des

### 3.6 Mathematische Methoden zur Trajektorienoptimierung

Zeithorizonts gelöst. Das Ergebnis ist die optimale Steuerstrategie  $\pi^*(x_n)$  bezüglich des aktuellen Systemzustands  $x_n$ .

3. Aus der optimalen Steuerstrategie  $\pi^*(x_n)$  wird die erste Steuereingabe  $u_n^* = \mu_n^*(x_n)$  für die Dauer eines Zeitschritts  $\Delta t$  auf das dynamische System angewendet.
4. Nach Ablauf des Zeitschritts wird der aktualisierte Systemzustand  $x_{n+1}$  zum Zeitpunkt  $t_{n+1}$  gemessen.
5. Der NMPC-Algorithmus beginnt wieder in Punkt 1 mit dem aktualisierten Systemzustand und wird solange fortgesetzt, bis alle Zeitschritte  $t_n$  abgearbeitet sind.

Das Ergebnis ist eine Folge optimaler Steuereingaben über alle Zeitschritte  $t_n$ , aus der die Referenztrajektorie generiert wird. Dabei wird das dynamische System unter Einhaltung von Steuer- und Zustandsbeschränkungen von einer vorgegebenen Startkonfiguration (UAV Startposition) zur Zielkonfiguration überführt, wobei die Zielfunktion minimiert wird. Die daraus resultierende Referenztrajektorie ist, im Gegensatz zur global optimalen Referenztrajektorie aus der dynamischen Programmierung (vgl. Abschnitt 3.6.2), lokal optimal. Ursächlich hierfür ist, dass einerseits eine spezifische Startkonfiguration vorgegeben werden muss, da das Verfahren nicht in der Lage ist, die optimale Startkonfiguration zu ermitteln aus der die global niedrigsten Kosten resultieren würden. Andererseits erfolgt die Optimierung des OSPs zyklisch immer nur für einen vergleichsweise kurzen Prädiktionshorizont (mit  $t_k \ll t_n$ ), wodurch auf weit in der Zukunft liegende Sollwertänderungen erst spät reagiert werden kann.

Für Untersuchungen zur Stabilität, Robustheit und Optimalität der nichtlinearen modellprädiktiven Regelung wird insbesondere auf [186] und [187] verwiesen.

In diesem Abschnitt wurden die Grundlagen der (nichtlinearen) modellprädiktiven Regelung als Vorbereitung auf Abschnitt 4.3 beschrieben. In Abschnitt 4.3 wird mit Hilfe einer Kombination aus einer Pfadplanung und der modellprädiktiven Regelung die lokal optimale Referenz-Flugtrajektorie für eine gegebene Startkonfiguration berechnet.

#### 3.6.4 Zusammenfassung

In den vorangegangenen Abschnitten wurden die mathematischen Methoden *dynamische Programmierung* und *nichtlineare modellprädiktive Regelung* für die Trajektorienoptimierung vorgestellt. Beide Methoden gründen auf einem *Optimalsteuerungsproblem*. Dies ermöglicht, dass sensorplattformenspezifische Anforderungen und Limitierungen modelliert und explizit im Optimierungsprozess berücksichtigt werden können.

### 3 Konzept und Methodenauswahl

Die dynamische Programmierung ermöglicht die Lösung des OSPs für einen zeit- und wertdiskreten Zustandsraum. Das Ergebnis ist eine global optimale Referenztrajektorie, wobei die Zustandsübergänge durch Dubins-Pfade abgebildet werden. Mit der nicht-linearen modellprädiktiven Regelung können lokal optimale Referenztrajektorien für eine spezifische Startkonfiguration berechnet werden. Bei NMPC wird das OSP zyklisch für einen vordefinierten Zeithorizont gelöst. Zu diesem Zweck wird das zeitkontinuierliche OSP problemspezifisch diskretisiert. Das Ergebnis der Optimierung ist eine Folge optimaler Steuereingaben, aus denen die UAV-Referenztrajektorie generiert wird.

In der nachfolgenden Tabelle ist der aktuelle Stand bezüglich der Bearbeitung der Zwischenziele und Forschungsfragen abgebildet. Die Fragen und Ziele, die bereits erfolgreich beantwortet wurden, sind grün dargestellt. Die gelb markierten Felder verdeutlichen, dass die Bearbeitung der *Forschungsfragen 3.1* und *3.2*, sowie des *Zwischenziels 3* begonnen, aber noch nicht abgeschlossen wurde.

Zwischenziel	1	2	3	4	5
Forschungsfrage	1.1	2.1 & 2.2	3.1 & 3.2	—	5.1 - 5.3
Bearbeitung in Kapitel	3	3	3 & 4	4	6



## 4 Umsetzung des Konzepts in ein Experimentalsystem

Das funktionale Konzept der sensormodellbestimmten Befliegungsplanung, das in Kapitel 3 beschrieben wurde, wird im Folgenden in ein Experimentalsystem überführt. Dies ist erforderlich, um die Fähigkeit der sensormodellbestimmten Befliegungsplanung zur Steigerung der Detektionsleistung in Kapitel 6 evaluieren zu können. Zudem werden in diesem Kapitel die *Forschungsfragen 3.1* und *3.2*, deren Bearbeitung im vorangegangenen Kapitel begonnen wurde, vollständig beantwortet. Damit kann das *Zwischenziel 3* aus Abschnitt 1.3 erfolgreich abgeschlossen werden. Darüber hinaus wird mit der Überführung des funktionalen Konzepts in ein Experimentalsystem das *Zwischenziel 4* erfüllt.

### 4.1 Systembeschreibung

Im vorangegangenen Kapitel wurde das funktionale Konzept der sensormodellbestimmten Befliegungsplanung vorgestellt und die Grundlagen der hierfür verwendeten Konzepte und Methoden, losgelöst voneinander, beschrieben. Auf Basis dieser Konzepte und Methoden wird nachfolgend das Experimentalsystem entwickelt und vorgestellt.

In Abb. 4.1 ist das Ablaufdiagramm zur Beschreibung der sensormodellbestimmten Befliegungsplanung für die zeitkontinuierliche Trajektorienoptimierung (links) sowie zeit- und wertdiskrete Optimierung (rechts) dargestellt. Im Ablaufdiagramm sind die einzelnen Prozessschritte abgebildet, die notwendig sind, um das funktionale Konzept in ein lauffähiges Experimentalsystem zu überführen. Die einzelnen Prozessschritte wurden aus dem Verfahren zur Planung und Durchführung von Aufklärungsaufträgen im *Erweiterten Sensor- und Perzeptionsmanagement* (siehe Abb. 3.1) aus Abschnitt 3.3.1 abgeleitet. Der Zugriff auf spezifische Daten und Modelle (z. B. Sollwertvorgaben, Leistungsmodelle, Heuristiken) erfolgt innerhalb der einzelnen Prozessschritte und ist nicht explizit abgebildet.

Bei der Befliegungsplanung wird die Planung der räumlich-zeitlichen Positionierung des Sensorfootprints am Boden von der Planung der UAV-Flugbewegung getrennt. Eine

#### 4 Umsetzung des Konzepts in ein Experimentalsystem

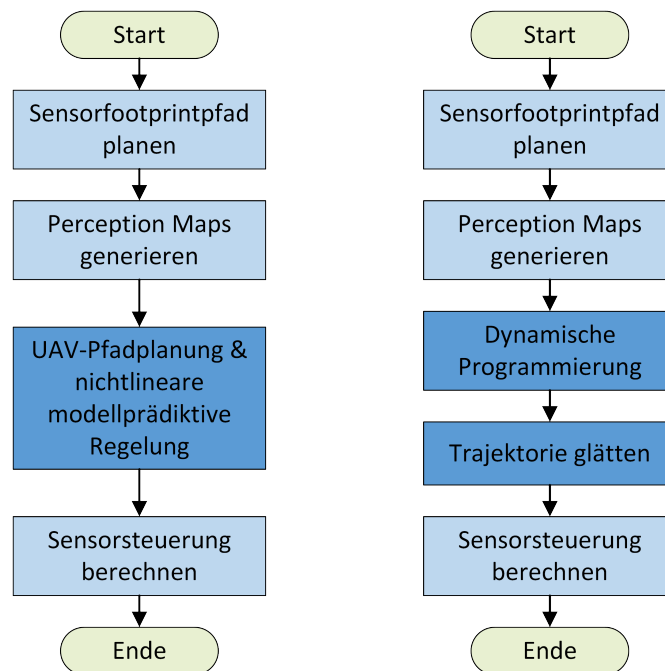


Abbildung 4.1: Ablaufdiagramm der sensormodellbestimmten Befliegungsplanung mit zeitkontinuierlicher Trajektorienoptimierung (links) sowie für die zeit- und wertdiskrete Trajektorienoptimierung (rechts). Die übereinstimmenden Prozessschritte sind hellblau markiert, wohingegen die systemspezifischen Prozesse der Trajektorienoptimierung dunkelblau dargestellt sind.

gesamtheitliche Betrachtung und Optimierung einer integrierten Sensor- und UAV-Bewegung, unter dem Einfluss zahlreicher Randbedingungen, wird als nicht sinnvoll erachtet. Diese Entscheidung wird wie folgt begründet:

Ein Grundproblem der Planungstheorie bei optimalen Ansätzen besteht darin, dass die meisten allgemeinen Planungsprobleme zur Klasse der *NP-schweren Probleme* [188] gehören, was bedeutet, dass es keine bekannten Polynomialzeit-Algorithmen zur Lösung dieser Klasse von Problemen gibt. Eine Alternative ist die Beschränkung auf suboptimale Lösungen unter Verwendung heuristischer Techniken [189].

Diese Vorgehensweise wird auch in dieser Arbeit umgesetzt. Im ersten Schritt der sensormodellbestimmten Befliegungsplanung wird auf Grundlage einer Heuristik der Sensorfootprintpfad am Boden geplant (Abschnitt 4.2). Darauf aufbauend wird für diesen Sensorfootprintpfad eine optimale Referenztrajektorie (Abschnitt 4.3 bzw. Abschnitt 4.4) für die Flugbewegung des UAVs generiert.

Bei der zeitkontinuierlichen Trajektorienoptimierung (Abschnitt 4.3) wird die Referenztrajektorie durch eine kombinierte UAV-Pfadplanung mit nichtlinearer modellprädiktiver

## 4.2 Implementierung der Sensorfootprintpfadplanung

Regelung zyklisch berechnet. Demgegenüber werden bei der zeit- und wertdiskreten Trajektorienoptimierung (Abschnitt 4.4) Dubins-Pfade mit Hilfe der dynamischen Programmierung zu einer Referenztrajektorie verbunden. Durch einen nachfolgenden Prozessschritt wird diese Trajektorie, die aus aneinandergereihten Dubins-Pfaden besteht, geglättet und die finale UAV-Referenztrajektorie erzeugt.

Der letzte Schritt besteht darin, die Sensorsteuerung (Abschnitt 4.5) zu berechnen. Die Sensorsteuerung definiert die Sollwertvorgabe für die Ausrichtung des Sensors in Abhängigkeit der räumlich-zeitlichen UAV-Position und der Sensorfootprintposition am Boden.

## 4.2 Implementierung der Sensorfootprintpfadplanung

Die Sensorfootprintpfadplanung, auch Sensorpfadplanung genannt, (engl.: Coverage Path Planning, CPP) ist einer der ersten Schritte der sensormodellbestimmten Befliegungsplanung (vgl. Abb. 4.1). Die Sensorpfadplanung soll dabei zwei Aufgaben erfüllen: zum einen soll ein Sensorfootprintpfad durch das aufzuklärende Gebiet geplant werden, zum anderen soll die räumlich-zeitliche Positionierung dedizierter Sensorfootprintpositionen auf dem Sensorfootprintpfad bestimmt werden. Die Planung des Sensorfootprintpfads erfolgt daher vor der Festlegung der dedizierten Footprintpositionen. Der Sensorfootprintpfad wird in dieser Arbeit als ein Polygonzug mit einem definierten Start- und Endpunkt beschrieben, der die geometrische Verbindung der Mittelpunkte aller Sensorfootprints repräsentiert. Die Erläuterungen zur Sensorpfadplanung, die in diesem Abschnitt gemacht werden, orientieren sich an der Arbeit von Zwick, Gerdtts & Stütz [33].

Die Systemübersicht in Abb. 4.2 stellt das Verfahren zur Sensorpfadplanung und der anschließenden Generierung der *Perception Maps* dar. Die Übersicht ist ein Ausschnitt aus Abb. 4.1 und verdeutlicht die wesentlichen Arbeitsschritte, die in den nachfolgenden Abschnitten beschrieben werden. Die sensormodellbestimmte Befliegungsplanung wird gestartet, indem eine *aufzuklärende Route bzw. Fläche* als Eingangsgröße an die Sensorpfadplanung übergeben wird. Die Ausgangsgrößen *Perception Maps* und *Sensorfootprints* sind das Ergebnis aus dem Prozessschritt zur Generierung der Perception Maps. Zusätzliche Informationen, die zur Berechnungen notwendig sind, wie z. B. das Zeitschrittintervall oder das Sensorleistungsmodell, werden aus vorab erstellten Datensätzen, die im Modul „Daten & Modelle“ hinterlegt sind, abgerufen.

#### 4 Umsetzung des Konzepts in ein Experimentalsystem

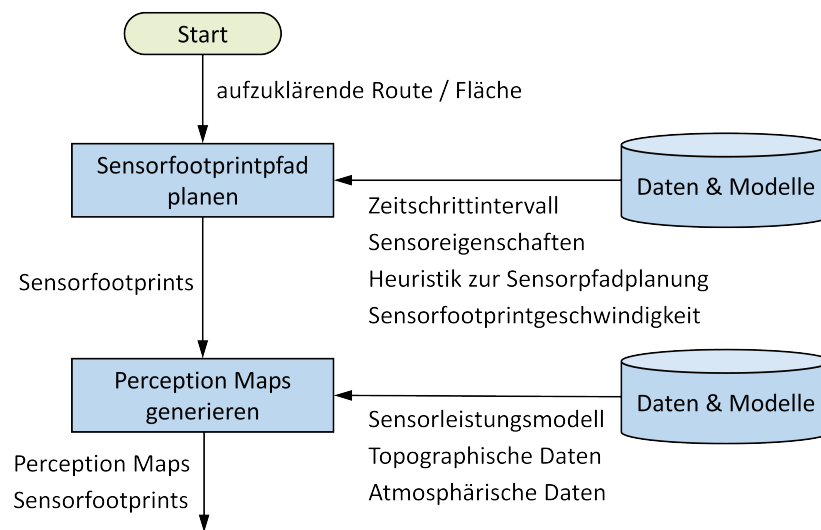


Abbildung 4.2: Systemübersicht mit den Arbeitsschritten für die Sensorpfadplanung und der anschließenden Generierung der Perception Maps.

Das aufzuklärende Gebiet kann in eine der drei folgenden Kategorien eingeteilt werden, wodurch der Prozess der Pfadplanung und der Verlauf des Sensorfootprintpfads maßgeblich beeinflusst werden:

- *Nulldimensionale Geometrie*: der aufzuklärende Bereich am Boden besteht aus einem Punkt (engl.: Point of Interest, PoI) und umfasst beispielsweise die Aufklärung eines Gebäudes. Der Sensorfootprintpfad ist somit auf einen Punkt reduziert.
- *Eindimensionale Geometrie*: der aufzuklärende Bereich nimmt die Form einer Linie an, beispielsweise um entlang einer Straße aufzuklären. Der Verlauf des Sensorfootprintpfads deckt sich hierbei mit dem Verlauf der Linie. Nachfolgend wird dies als *Routenaufklärung* bezeichnet.
- *Zweidimensionale Geometrie*: in diesem Fall hat das aufzuklärende Gebiet eine flächige Ausdehnung. Um den Verlauf des Sensorfootprintpfads ermitteln zu können, werden Heuristiken angewendet. Die Aufklärung einer Fläche wird im Folgenden als *Flächenaufklärung* bezeichnet.

Die Pfadplanung für die Aufklärung eines Punktes stellt einen trivialen Sonderfall der Sensorpfadplanung dar, da die Aufklärung mit einer einmaligen Erfassung durch den Sensorfootprint, einschließlich der nachgeschalteten Datenaufbereitung und Auswertung (Detektion), abgearbeitet ist. Die mehrmalige oder dauerhafte Erfassung des PoI entspricht einer Überwachung / einem Tracking, was in Abschnitt 1.4 bereits erwähnt wurde und

in dieser Arbeit nicht weiter betrachtet wird. Bei der Routenaufklärung deckt sich der Sensorfootprintpfad mit der Linienvorgabe, beispielsweise einem Straßenverlauf. Eine eigenständige Planung des Sensorfootprintpfads ist daher nicht erforderlich. Aus diesem Grund liegt der Fokus der Sensorpfadplanung im Folgenden auf dem Anwendungsfall für die Flächenaufklärung. Ungeachtet dessen ist die räumlich-zeitliche Positionierung der Sensorfootprints auf dem Sensorfootprintpfad sowohl für die Routen- als auch für die Flächenaufklärung relevant, worauf im weiteren Verlauf noch eingegangen wird.

Das Verfahren zur Sensorpfadplanung dient dazu, die aufzuklärende Fläche deterministisch<sup>15</sup> und vollständig mit dem Sensorfootprint zu überstreichen. Damit soll sichergestellt werden, dass die gesamte aufzuklärende Fläche sensorisch erfasst wird, um alle darin befindlichen relevanten Objekte (z. B. Fahrzeuge, Personen) detektieren zu können, was eine Forderung aus Abschnitt 1.4 ist.

### 4.2.1 Sensorpfadplanung und Bestimmung der Sensorfootprints

Das hier angewendete Verfahren zur Sensorpfadplanung basiert auf einer Heuristik, bei der eine zelluläre Diskretisierung des Aufklärungsgebiets gemäß der Boustrophedon-Zerlegung [190] vorgenommen wird. Diese wird kombiniert mit einer Vorwärts-Rückwärts-Planung des Sensorfootprints, wie sie in [191] beschrieben ist. Die Scan- oder Streifenbreite  $b_{fp}$  (4.1) wird für das CPP aus der Breite des Sensorfootprints berechnet. Dieser resultiert aus dem Referenzwert (Sollwert) der Bodenpixelauflösung (GSD)  $gsd_{ref}$  und der Anzahl der Sensorpixel  $n_{Pixel}$  bezogen auf die Breite des bildgebenden Sensors. Es wird angenommen, dass die Sensorfläche quadratisch ist. Hieraus resultiert ein ebenfalls quadratischer Sensorfootprint am Boden mit der Kantenlänge  $b_{fp}$ , wobei perspektivische Verzerrungen vernachlässigt werden.

$$b_{fp} = gsd_{ref} \cdot n_{Pixel} \quad (4.1)$$

In Abb. 4.3 ist beispielhaft die Sensorpfadplanung mit der zuvor beschriebenen Heuristik für eine aufzuklärende Fläche (grün) dargestellt. Das Ergebnis ist ein mäanderförmiger Sensorfootprintpfad (blau) auf dem die Flächenmittelpunkte aller Sensorfootprints liegen. Der detektierte Bereich umfasst die Fläche aus allen vereinigten Sensorfootprints. Befindet sich die gesamte aufzuklärende Fläche innerhalb des detektierten Bereichs, so wird die aufzuklärende Fläche vollständig vom Sensorfootprint überstrichen und erfasst.

---

<sup>15</sup>Die deterministische Aufklärung beschreibt in diesem Kontext, dass eine Fläche unter gleichbleibenden Bedingungen (u. a. bei gleicher Heuristik) immer nach dem selben Bewegungsmuster vom Sensorfootprint überstrichen wird. Demgegenüber kann bei der stochastischen Aufklärung der identischen Fläche der Verlauf des Sensorfootprintpfads zufallsbedingt variieren.

#### 4 Umsetzung des Konzepts in ein Experimentalsystem

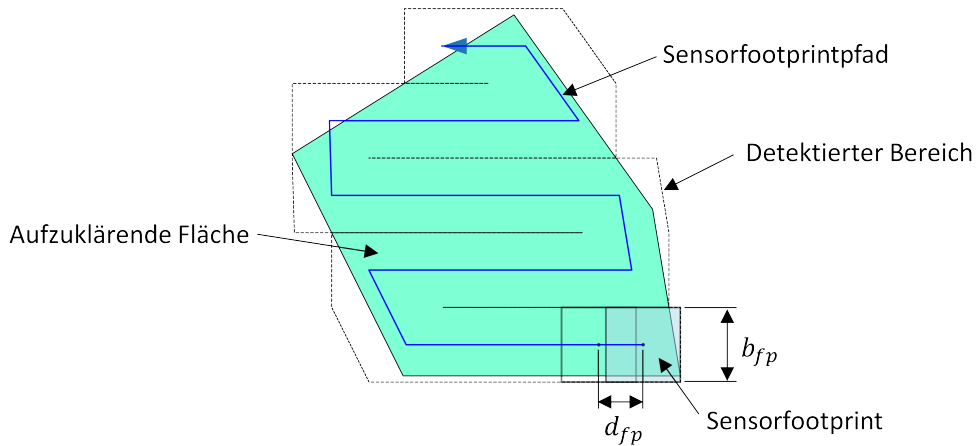


Abbildung 4.3: Prinzip der Sensorpfadplanung für eine aufzuklärende Fläche (grün). Die Flächenmittelpunkte der einzelnen Sensorfootprints (blaues Quadrat) liegen auf dem Sensorfootprintpfad (blau). Die Kantenlänge des Sensorfootprints ist definiert durch  $b_{fp}$  und der euklidische Abstand zwischen zwei aufeinander folgender Footprints wird durch  $d_{fp}$  festgelegt. Die schwarz gepunktete Linie markiert den vom Sensorfootprint überstrichenen Bereich (übernommen und angepasst aus [33]).

Die räumlich-zeitliche Positionierung der dedizierten Sensorfootprints auf dem Sensorfootprintpfad wird mit Hilfe von Gleichung (4.2) ermittelt. Hierbei beschreibt  $d_{fp}$  den euklidischen Abstand zwischen den Flächenmittelpunkten zweier aufeinander folgender Sensorfootprints. Der Abstand berechnet sich aus dem Sollwert der Sensorfootprintgeschwindigkeit  $v_{fp}$  und dem Zeitschrittintervall  $\Delta t$ . Die räumliche (örtliche) und zeitliche Positionierung des Sensorfootprints wird durch den aktuellen Zeitschritt und den Verlauf des Sensorfootprintpfads bestimmt. In Abb. 4.3 ist der erste Sensorfootprint auf dem Sensorfootprintpfad als hellblaues Quadrat abgebildet. Zusätzlich sind die Umrisse des zweiten Footprints dargestellt, der im Abstand  $d_{fp}$  zum vorhergehenden Footprint positioniert ist und diesen teilweise überlappt.

$$d_{fp} = v_{fp} \cdot \Delta t \quad (4.2)$$

Die Länge  $L$  des Sensorfootprintpfads wird bei der Routenaufklärung durch den Aufklärungsauftrag vorgegeben und bei der Flächenaufklärung mit Hilfe der verwendeten Heuristik berechnet. Aus der Länge des Sensorfootprintpfads, der Sensorfootprintgeschwindigkeit und dem Zeitschrittintervall kann die Anzahl der Zeitschritte  $K \in \mathbb{N}$  mit Gleichung (4.3) berechnet werden. Hierbei wird  $K$  auf die nächsthöhere natürliche Zahl

## 4.2 Implementierung der Sensorfootprintpfadplanung

aufgerundet.

$$K = \left\lceil \frac{L}{v_{fp} \cdot \Delta t} \right\rceil \quad (4.3)$$

In Tabelle 4.1 sind die Parametereinstellungen zusammengefasst, die für die Sensorpfadplanung in dieser Arbeit verwendet werden. Es wird darauf hingewiesen, dass sich die Zeitschrittintervalle  $\Delta t$  für die zeitkontinuierliche und die zeit- und wertdiskrete Trajektorienoptimierung unterscheiden. Dadurch ergeben sich unterschiedliche Abstände der Sensorfootprints  $d_{fp}$ .

Tabelle 4.1: Parametereinstellung für die Sensorpfadplanung

Parameter	Wert	Bemerkung
Sollwert der Bodenpixelauflösung $gsd_{ref}$	0,07 m	aus Tabelle 3.4
Anzahl der Sensorpixel $n_{Pixel}$	1920 px	vorgegeben
Breite des Sensorfootprints $b_{fp}$	134,4 m	aus Gleichung (4.1)
Geschwindigkeit des Sensorfootprints $v_{fp}$	25 m/s	vorgegeben
Zeitschrittintervall $\Delta t$ (zeitkont. Optimierung)	0,5 s	vorgegeben
Zeitschrittintervall $\Delta t$ (zeitdiskr. Optimierung)	1,5 s	vorgegeben
Distanz $d_{fp}$ (zeitkont. Optimierung)	12,5 m	aus Gleichung (4.2)
Distanz $d_{fp}$ (zeitdiskr. Optimierung)	37,5 m	aus Gleichung (4.2)

Aus der Sensorpfadplanung resultiert der geometrische Verlauf des Sensorfootprintpfads am Boden. Über das Zeitschrittintervall kann eine Sequenz von aneinander gereihter Sensorfootprints auf dem Sensorfootprintpfad definiert werden, wodurch die Anzahl, Positionierung und Reihenfolge der Footprints festgelegt wird. Im Rahmen der automatisierten Planung und Durchführung von Aufklärungsaufträgen wird für jeden einzelnen dieser Sensorfootprints eine sensorische Datenerfassung, Aufbereitung und Auswertung durch die Perzeptionskette durchgeführt. Die Datenerfassung erfolgt daher nicht als kontinuierlicher Sensorstream sondern als Einzelbild für die individuelle Sensorfootprintposition. Ein Sensorfootprint korreliert daher immer mit einer sensorischen Datenerfassung und umgekehrt. Die Sensorpfadplanung wird für die zeitkontinuierliche als auch für die zeit- und wertdiskrete Trajektorienoptimierung sowie für die Sensorsteuerung in Abschnitt 4.5 benötigt.

### 4.2.2 Generierung footprintspezifischer Perception Maps

In Abschnitt 3.4.5 wurde bereits das Konzept und die Entwicklung der Perception Map erläutert. Die Perception Map stellt die räumliche Charakteristik der Detektionsleistung in Abhängigkeit der Sensorposition in einer zweidimensionalen Ebene dar. Für jeden

dedizierten Sensorfootprint, der mit Hilfe der Sensorpfadplanung festgelegt wurde, wird eine individuelle, orts- und zeitschrittabhängige Perception Map generiert. Die Perception Maps entlang des Sensorfootprintpfads repräsentieren örtlich und zeitlich variierende Potentialfelder der Detektionsleistung in der Bewegungsebene des UAVs. Das Ziel der sensormodellbestimmten Befliegungsplanung ist es, eine optimale Referenz-Flugtrajektorie für das UAV durch die Aneinanderreihung der verschiedenen Potentialfelder zu planen, wodurch die zu erwartende Detektionsleistung maximiert wird.

Die Gesamtanzahl der Perception Maps  $N_{PM} \in \mathbb{N}$ , die für einen Aufklärungsauftrag generiert werden, entspricht der Anzahl der Zeitschritte  $K \in \mathbb{N}_0$  zuzüglich einer weiteren Perception Map zum Zeitschritt null, die sich am Startpunkt des Sensorfootprintpfads befindet:

$$N_{PM} = K + 1 \quad (4.4)$$

### 4.3 Zeitkontinuierliche Trajektorienoptimierung

In diesem Abschnitt wird auf die zeitkontinuierliche Trajektorienoptimierung eingegangen, die eine kombinierte UAV-Pfadplanung und nichtlineare modellprädiktive Regelung umfasst. Das Ziel dieses Verfahrens ist es, eine lokal optimale UAV-Referenztrajektorie für eine gegebene Startkonfiguration zu generieren, unter Berücksichtigung perzeptiver, sensorplattformspezifischer und missionsspezifischer Anforderungen und Randbedingungen. Mit der Umsetzung der zeitkontinuierlichen Trajektorienoptimierung wird die *Forschungsfrage 3.2* (Wie kann eine, bezüglich der Detektionsleistung, lokal optimale Flugtrajektorie mit vorgegebener Startkonfiguration generiert werden, unter Berücksichtigung sensorplattformspezifischer und missionsspezifischer Anforderungen?) aus Abschnitt 2.4 beantwortet und abgeschlossen. Darüber hinaus wird die *Anforderung 4* (Die sensormodellbestimmte Befliegungsplanung soll die Fähigkeit zur automatisierten Generierung optimaler Flugtrajektorien besitzen.) aus Abschnitt 3.2 umgesetzt und erfüllt.

Bei der zeitkontinuierlichen Trajektorienoptimierung dient die UAV-Pfadplanung dazu, die perzeptiven Eigenschaften und Fähigkeiten der Perzeptionskette im Optimierungsprozess berücksichtigen zu können. Darauf aufbauend werden durch die nachgeschaltete nichtlineare modellprädiktive Regelung die sensorplattformspezifischen und missionsspezifischen Anforderungen in die Optimierung mit einbezogen.

In Abb. 4.4 ist das Systemschaltbild der zeitkontinuierlichen Trajektorienoptimierung dargestellt. Die Eingangsgrößen *Perception Maps* und *Sensorfootprints* stammen aus dem vorangegangenen Prozessschritt, der in Abschnitt 4.2 beschrieben wurde. Aus vorab erstellten Daten und Modellen können zusätzliche Informationen abgerufen werden, die



für die Trajektorienoptimierung erforderlich sind. Die Ausgangsgrößen des Prozessschritts sind die *UAV-Referenztrajektorie* und die *Sensorfootprints*. Die Sensorfootprints werden dabei unverändert an den nachfolgenden Prozessschritt übermittelt.

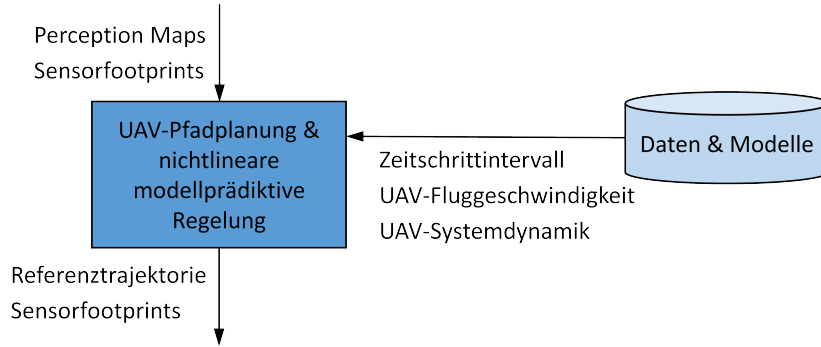


Abbildung 4.4: Systemübersicht mit den Arbeitsschritten für die kombinierte UAV-Pfadplanung und nichtlineare modellprädiktive Regelung

#### 4.3.1 UAV-Pfadplanung

Die UAV-Pfadplanung mit gekrümmten Pfadsegmenten ist ein Teil der kombinierten Pfad- und Trajektorienplanung, die bereits in Zwick, Gerds & Stütz in [33, 149] vorgestellt wurde. Die einzelnen Pfadsegmente bilden eine fächerförmige Pfadschar, wobei jedes Pfadsegment eine unterschiedliche Krümmung aufweist. Die Pfade erfüllen die folgenden zwei Aufgaben: Einerseits dient der räumlich-zeitliche Pfadverlauf als Sollwertvorgabe für die Trajektorienoptimierung mit nichtlinearer modellprädiktiver Regelung. Andererseits wird für jeden einzelnen Pfad die zu erwartende Detektionsleistung berechnet, die erzielt wird, wenn das UAV dem jeweiligen Pfadverlauf folgen würde. Dabei wird die Detektionsleistung anhand der in Abschnitt 4.2 vorgestellten zeitschrittspezifischen Perception Maps bestimmt.

Die Pfadsegmente haben ihren Ursprung an der Position des UAVs zum globalen Zeitschritt  $t_n$ . Von der UAV-Position aus breiten sich die Pfade fächerförmig mit konstanter Krümmung und gleicher Länge aus. Die Pfadlänge  $l_{path}$  wird durch Gleichung (4.5) bestimmt, wobei  $v^{ref}$  der vorgegebene Sollwert für die Geschwindigkeit des UAVs ist.  $\Delta t$  ist das Zeitschrittintervall und  $M_{prev} \in \mathbb{N}$  beschreibt die Anzahl der Zeitschritte des Vorschauhorrizonts<sup>16</sup>.

$$l_{path} = v^{ref} \cdot \Delta t \cdot M_{prev} \quad (4.5)$$

<sup>16</sup>Es wird darauf hingewiesen, dass der Vorschauhorrizont und der Zeit- bzw. Prädiktionshorrizont der modellprädiktiven Regelung zwei verschiedene Konstrukte sind.

#### 4 Umsetzung des Konzepts in ein Experimentalsystem

Der Pfadplanungsprozess lässt sich kurz wie folgt zusammenfassen: Ausgehend von der aktuellen UAV-Position im Zeitschritt  $t_n$  wird die Detektionsleistung entlang jedes einzelnen Pfads  $z \in \{1, \dots, Z\}$  mit  $Z \in \mathbb{N}$  für jeden zukünftigen Zeitschritt  $m \in \{1, \dots, M_{prev}\}$  innerhalb des Vorschauhorizonts bestimmt. Jedem Zeitschritt  $t_{n+m}$  mit  $n + m \leq K$  ist durch die Sensorpfadplanung aus Abschnitt 4.2 ein eindeutiger Sensorfootprint und eine entsprechende Perception Map zugeordnet, die den Verlauf der lokalen Detektionsleistung in der Bewegungsebene des UAVs abbildet. Die Position des UAVs resultiert aus dem Verlauf des Pfads  $z$  und der zeitschrittabhängigen Distanz  $d_{path,m}$  in (4.6), die das UAV im Vorschauhorizont entlang dieses Pfads zurücklegt.

$$d_{path,m} = v^{ref} \cdot \Delta t \cdot m \quad (4.6)$$

In Abb. 4.5 ist exemplarisch ein UAV mit fächerförmiger Pfadschar dargestellt. Die fünf Pfadsegmente weisen eine konstante Krümmung und eine identische Länge auf. Die Dreiecke markieren die zukünftigen UAV-Positionen auf den Pfadsegmenten für die Zeitschritte  $m = 1$  und  $m = 2$ .

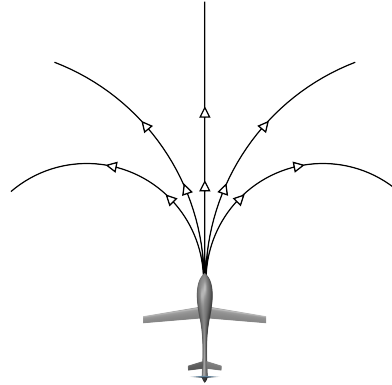


Abbildung 4.5: Darstellung des UAVs mit fächerförmiger Pfadschar, bestehend aus fünf Pfadsegmenten mit konstanter Krümmung und identischer Länge. Die Dreiecke markieren exemplarisch die UAV-Positionen auf den einzelnen Pfaden für zwei zukünftige Zeitschritte.

Abschließend kann die lokale pfadabhängige Detektionsleistung  $p_{det,n+m,z}$  aus der zeitschrittabhängigen UAV-Position und der zugehörigen Perception Map bestimmt werden. Aus den einzelnen Detektionsleistungen  $p_{det,n+m,z}$  wird die gewichtete *kumulative Detektionsleistung*  $p_{prev,n,z}$  nach Gleichung (4.7) berechnet, wobei eine exponentielle Gewichtung gewählt wurde, um zeitlich näherliegende Werte der Detektionsleistung zu verstärken und zeitlich weiter entfernte Werte zu schwächen. Ein großer Vorschauhorizont dient dazu, ent-

### 4.3 Zeitkontinuierliche Trajektorienoptimierung

fernere Bereiche mit höherer Detektionsleistung bei der Pfadplanung berücksichtigen zu können. Gleichzeitig soll durch die exponentielle Gewichtung verhindert werden, dass das UAV beispielsweise einen Bereich mit hoher Detektionsleistung unnötigerweise verlässt, wenn ein weit entfernter Bereich ebenfalls eine hohe Detektionsleistung aufweist. Die Wirksamkeit dieser Maßnahme wurde durch experimentelle Untersuchungen bestätigt.

$$p_{prev,n,z} = \frac{1}{M_{prev}} \sum_{m=1}^{M_{prev}} e^{-\frac{m}{M_{prev}}} \cdot p_{det,n+m,z} \quad (4.7)$$

In Gleichung (4.8) werden alle kumulativen Detektionsleistungswerte durch deren Maximalwert geteilt, um die Werte zu normieren, bevor diese im nächsten Abschnitt 4.3.2 wieder aufgegriffen und weiterverwendet werden. Aus der Normierung resultiert die *kumulierte und normierte Detektionsleistung*  $\tilde{p}_{prev,n,z} \in [0, 1]$  für den Pfad  $z$  zum Zeitschritt  $t_n$ . Der Grund für die Normierung wird in Abschnitt 4.3.2 erläutert. In Tabelle 4.2 sind die relevanten Parametereinstellung für die Pfadplanung zusammengefasst.

$$\tilde{p}_{prev,n,z} = \frac{p_{prev,n,z}}{\max_z(p_{prev,n,z})} \quad \text{mit} \quad \max_z(p_{prev,n,z}) \neq 0 \quad (4.8)$$

Die Form der einzelnen Pfadsegmente entspricht in etwa dem Flugverhalten eines Starrflügelflugzeugs. Jeder Pfad hat eine feste Krümmung, wodurch gleichmäßig verteilte fächerförmige Kurvensegmente entstehen, die den Bereich zwischen einer Linkskurve und einer Rechtskurve mit jeweils maximalem Rollwinkel abdecken. Die Anzahl der Pfade und ihr Verlauf definiert den Bereich in Flugrichtung des UAVs, der vom UAV erreicht werden kann. Die Pfadlänge hingegen bestimmt die Anzahl der zukünftigen Zeitschritte und damit den zeitlichen Vorschauhorizont.

Tabelle 4.2: Parametereinstellung für den UAV-Pfadplanungsprozess

Parameter	Wert	Bemerkung
Länge des Vorschauhorizonts $M_{prev}$	50	vorgegeben
Anzahl der Pfade $Z$	15	vorgegeben
Zeitschrittintervall $\Delta t$	0,5 s	aus Tabelle 4.1
Sollwert für die UAV-Fluggeschwindigkeit $v^{ref}$	35 m/s	aus Tabelle 3.7
Pfadlänge $l_{path}$	875 m	aus Gleichung (4.5)

Abb. 4.6 veranschaulicht das Prinzip der Pfadplanung exemplarisch anhand eines zeitlich und räumlich stationären Sensorfootprints (blaues Quadrat). Dargestellt ist die fächerförmige Pfadschar, die zur Bestimmung der zu erwartenden Detektionsleistung

#### 4 Umsetzung des Konzepts in ein Experimentalsystem

entlang jedes einzelnen Pfads genutzt wird. Die dicke schwarze Linie stellt die optimale UAV-Referenztrajektorie dar und ist das Ergebnis der Kombination aus Pfadplanung und Trajektorienplanung mit nichtlinearer modellprädiktiver Regelung, auf die in Abschnitt 4.3.2 eingegangen wird. Die Perception Map resultiert aus den atmosphärischen und topographischen Zuständen und ist als farbkodiertes Potenzialfeld dargestellt. Helle Bereiche markieren Regionen mit hoher Detektionsleistung, während dunklere Bereiche niedrige Werte kennzeichnen.

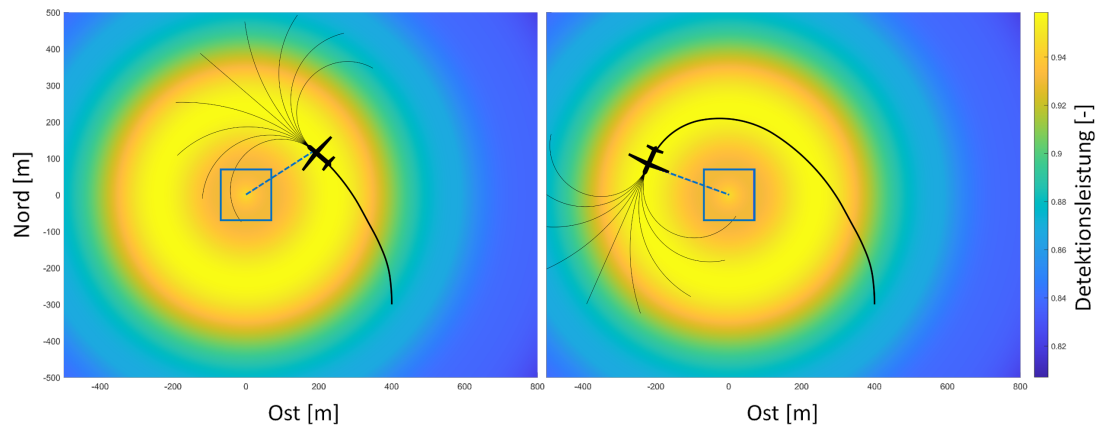


Abbildung 4.6: Prinzipdarstellung der Pfadplanung. Die fächerförmige Pfadschar ist beispielhaft durch neun unterschiedlich gekrümmte Pfadsegmente (dünne schwarze Linien) dargestellt. Die dicke schwarze Linie repräsentiert die sich aus der Trajektorienoptimierung ergebende UAV-Trajektorie. Das blaue Quadrat repräsentiert den Sensorfootprint am Boden, der zur Verdeutlichung dieses Beispiels stationär ist. Die Perception Map, die aus den atmosphärischen und topographischen Zuständen resultiert, ist farbkodiert dargestellt, wobei gelbe Bereiche Regionen mit hoher Detektionsleistung darstellen (übernommen und angepasst aus [149]).

Im Folgenden ist eine Übersicht einiger Vor- und Nachteile der UAV-Pfadplanung gegenüber der Verfahren aus Abschnitt 2.2 zusammengestellt. Zu den Vorteilen der Pfadplanung zählen:

- Das Verfahren zur Ermittlung der pfadabhängigen Detektionsleistung erfolgt unabhängig vom Verlauf der Detektionsleistung in der jeweiligen Perception Map. Die UAV-Pfadplanung kann daher sowohl für Perception Maps mit kontinuierlichem als auch mit abschnittsweise konstantem Verlauf der Detektionsleistung angewendet werden. Auf diesen Sachverhalt wird im nächsten Abschnitt nochmals eingegangen.

### 4.3 Zeitkontinuierliche Trajektorienoptimierung

- Im Gegensatz zu zahlreichen anderen Pfadplanungsalgorithmen (siehe u. a. Abschnitt 2.2) ist für die Bewegungsplanung mit einer Pfadschar kein spezifischer Endpunkt erforderlich, da dieser zum Zeitpunkt der Bewegungsplanung noch unbekannt ist.
- Der Berechnungsaufwand für die Pfadplanung kann durch die Variation der Pfadanzahl und der Pfadlänge angepasst werden.
- Über die Pfadlänge kann der Vorschauhorizont eingestellt werden, mit der die zeitliche und räumliche Entwicklung der Detektionsleistung entlang der Pfadsegmente erfasst wird.
- Über die Anzahl der Pfadsegmente kann die räumliche Auflösung der zeitschritt-spezifischen UAV-Positionen in der Perception Map angepasst werden.

Demgegenüber resultieren aus der UAV-Pfadplanung die folgenden Beeinträchtigungen:

- Die Pfadplanung mit konstant gekrümmten Pfadsegmenten ist nur für einfache Bewegungsmuster (z. B. für eine konstante Flugbahnkrümmung) und in einer konstanten Flughöhe sinnvoll. Eine diskrete Pfadplanung mit variabel gekrümmten Pfaden und/oder mit einer Variation der Flughöhe würde die Anzahl der zu untersuchenden Pfadsegmente, und den damit verbundenen Berechnungsaufwand, stark erhöhen.
- Innerhalb einer Perception Map können nur die Bereiche erfasst werden, durch die ein Pfadsegment verläuft. Eine zu grobe Diskretisierung der Pfadschar kann zur Folge haben, dass räumlich lokal begrenzte Bereiche mit hoher Detektionsleistung nicht erfasst werden.

#### 4.3.2 Kombinierte UAV-Pfadplanung und modellprädiktive Regelung

Nach der Vorstellung der nichtlinearen modellprädiktiven Regelung in Abschnitt 3.6.3 und der UAV-Pfadplanung in Abschnitt 4.3.1 werden diese kombiniert, um die optimale Steuereingabe  $u_n^*$  für jeden (globalen) Zeitschritt  $t_n$  zu berechnen. Diese optimale Steuereingabe ist das Resultat aus dem gewichteten Verhältnis zwischen *Nutzen* und *Aufwand* der zeitkontinuierlichen sensormodellbestimmten Befliegungsplanung. In diesem Kontext stellt der Nutzen die zu erwartende Detektionsleistung dar, die aus dem „Abfliegen“ der einzelnen Pfadsegmente resultieren würde, die mit Hilfe der UAV-Pfadplanung ermittelt werden. Demgegenüber entspricht der Aufwand den Kosten, die aus der Kostenfunktion der nichtlinearen modellprädiktiven Regelung stammen, wenn die einzelnen Pfadsegmente als Sollwertvorgaben im Optimierungsprozess verwendet werden. Die Kos-

#### 4 Umsetzung des Konzepts in ein Experimentalsystem

ten resultieren hierbei u. a. aus der Differenz der zeitschrittspezifischen Steuereingaben zu den Sollwert-Steuereingaben, wie dies in Gleichung (3.55) dargestellt wird.

Nachfolgend wird das Verfahren der kombinierten UAV-Pfadplanung und nichtlinearer modellprädiktiver Regelung detailliert beschrieben. Ausgehend von einem vorgegebenen Anfangszustand  $x_0$  zum Zeitpunkt  $t_0$  wird für jeden weiteren Zeitschritt  $t_n$  das folgende Vorgehen durchgeführt [149]:

Für den aktuellen Zeitschritt  $t_n$  wird die Position und der Gierwinkel des UAVs bestimmt. Ausgehend davon wird mit Hilfe der UAV-Pfadplanung der räumlich-zeitliche Verlauf der Pfadschar festgelegt. Für jedes Pfadsegment der Schar wird die *kumulative Detektionsleistung*  $p_{prev,n,z}$  nach Gleichung (4.7) berechnet. Anschließend werden die kumulativen Detektionsleistungen aller Pfadsegmente gemäß Gleichung (4.8) normiert, woraus die pfad- und zeitschrittabhängige *kumulierte und normierte Detektionsleistung*  $\tilde{p}_{prev,n,z}$  resultiert.

Im nächsten Schritt werden die Kosten für jedes Pfadsegment ermittelt, die aus der Kostenfunktion der nichtlinearen modellprädiktiven Regelung berechnet werden. Zu diesem Zweck werden aus den einzelnen Pfadsegmenten  $z$  zeitschrittspezifische Sollwertvorgaben  $x_{n+k,z}^{ref}$  für den Prädiktionshorizont der modellprädiktiven Regelung erstellt. Die Sollwertvorgabe setzt sich nach Gleichung (3.54) zusammen aus der UAV-Position  $e_{n+k,z}^{ref}$ ,  $n_{n+k,z}^{ref}$  und dem Gierwinkel  $\psi_{n+k,z}^{ref}$  für das Pfadsegment  $z$  zum Zeitschritt  $n+k$ , sowie der zeitschrittunabhängigen Vorgabe für die Fluggeschwindigkeit und den Rollwinkel. Der Gierwinkel  $\psi_{n+k,z}$  kann hierfür aus der Tangentensteigung am Ort  $e_{n+k,z}$  und  $n_{n+k,z}$  des Pfadsegments berechnet werden.

Aus der Lösung des Optimalsteuerungsproblems (3.50), und unter Einbeziehung der Sollwertvorgaben  $x_{n+k,z}^{ref}$ , ergeben sich die pfad- und zeitschrittabhängigen minimalen Gesamtkosten  $J_\pi(x_n, z)$  gemäß Gleichung (3.50a). Durch die anschließende Normierung erhält man die *normierten minimalen Gesamtkosten* in Gleichung (4.9):

$$\tilde{J}_\pi(x_n, z) = \frac{J_\pi(x_n, z)}{\max_z(J_\pi(x_n, z))} \quad \text{with} \quad \max_z(J_\pi(x_n, z)) \neq 0 \quad (4.9)$$

Abschließend wird das optimale Pfadsegment aus der Pfadschar ermittelt, welches die beste Kombination aus Aufwand und Nutzen in sich vereint. Dazu werden die normierten minimalen Gesamtkosten  $\tilde{J}_\pi(x_n, z) \in [0, 1]$  und die normierte kumulative Detektionsleistung  $\tilde{p}_{prev,n,z}$  nach Gleichung (4.10) mit  $\gamma \in [0, 1]$  gewichtet. Die Normierung der minimalen Gesamtkosten und der kumulativen Detektionsleistung stellt sicher, dass deren Wertebereiche identisch sind. Andernfalls würde die Gewichtung auf Grund unterschiedlicher Wertebereiche verzerrt werden. Der Wert für  $\gamma$  wurde in vorangegangenen

### 4.3 Zeitkontinuierliche Trajektorienoptimierung

Experimenten ermittelt und ist in Tabelle 4.3 dokumentiert.

$$c_{min,n} = \min_z \left( (1 - \gamma) \cdot \tilde{J}_\pi(x_n, z) - \gamma \cdot \tilde{P}_{prev,n,z} \right) \quad (4.10)$$

Aus Gleichung (4.10) folgt das zeitschrittabhängige kombinierte Kostenminimum  $c_{min,n}$ . Diese Kosten korrelieren mit dem optimalen Pfadsegment, das die beste Kombination aus Nutzen und Aufwand in sich vereint. Des Weiteren ist dieses Pfadsegment die Sollwertvorgabe für die nichtlineare modellprädiktive Regelung, aus der die optimale Steuereingabe  $u_n^*$ , und damit die neue UAV-Position, für den nächsten Zeitschritt resultiert. Dieses Vorgehen wird für jeden Zeitschritt einmalig durchgeführt bis alle Zeitschritte abgearbeitet sind.

Tabelle 4.3: Parametereinstellung für die kombinierte Pfadplanung und NMPC

Parameter	Wert	Bemerkung
Gewichtungsfaktor $\gamma$	0,8	vorgegeben

Aus der zyklischen Durchführung der kombinierten UAV-Pfadplanung und nichtlinearer modellprädiktiver Regelung ergibt sich die optimale Steuerstrategie  $\pi^*(x_0)$  (siehe Gleichung (3.35)) in Bezug auf den Startpunkt  $x_0$ . Diese Steuerstrategie bestimmt die räumlich-zeitliche Positionierung des UAVs und damit die (lokal) optimale Flugtrajektorie, die zugleich die Referenz-Flugtrajektorie für die sensormodellbestimmte Befliegungsplanung ist. Des Weiteren wird dadurch die eigentliche Detektionsleistung bestimmt, die aus der Anwendung der Referenztrajektorie resultieren würde.

Ein wesentlicher Vorteil dieses kombinierten Verfahrens besteht darin, dass die Pfadplanung die ortsabhängige Detektionsleistung berücksichtigt (modelliert durch die zeitschritt-spezifischen Perception Maps) und die Trajektorienoptimierung durch die modellprädiktive Regelung ausschließlich auf der Pfadplanung basiert. Dadurch kann die modellprädiktive Regelung unabhängig vom räumlich-zeitlichen Verlauf der Detektionsleistung in den Perception Maps erfolgen. Im Gegensatz zu einem gradientenbasierten Optimierungsverfahren hat die räumliche Charakteristik der Detektionsleistung (z. B. lokale Maxima<sup>17</sup>, abschnittsweise konstanter Verlauf oder Unstetigkeitsstellen) daher keinen (negativen) Einfluss auf die Robustheit der Befliegungsplanung.

In der *missionsspezifischen Anforderung* in Abschnitt 3.2 wird gefordert, dass die Minimierung der Flugdauer während des Aufklärungsvorgangs als Optimierungsgröße

<sup>17</sup>Gradientenbasierte Optimierungsverfahren können an *lokalen Minima* hängen bleiben. Im Kontext der sensormodellbestimmten Befliegungsplanung würde dieser Sachverhalt auf *lokale Maxima* der Detektionsleistung in den Perception Maps zutreffen.

#### 4 Umsetzung des Konzepts in ein Experimentalsystem

in die Befliegungsplanung mit eingehen soll. Bei der zeitkontinuierlichen Trajektorienoptimierung wird dies inhärent dadurch erreicht, dass der zurückgelegte Weg  $s_{\Delta t}$  des UAVs zwischen zwei Zeitschritten immer gleich groß ist und zugleich den minimalen Wert annimmt:

$$s_{\Delta t} = v^{ref} \cdot \Delta t \quad (4.11)$$

Daraus folgt, dass die Wegstrecke der UAV-Referenztrajektorie, die mit der zeitkontinuierlichen Trajektorienoptimierung generiert wird, per Definition minimal ist. Dies ist ein Vorteil der zeitkontinuierlichen Trajektorienoptimierung gegenüber dem Verfahren für die zeit- und wertdiskrete Optimierung, bei dem die zurückgelegte Wegstrecke zwischen zwei Zeitschritten variieren kann (siehe Abschnitt 4.4.1).

#### 4.3.3 Entwicklungsumgebung zur Implementierung der modellprädiktiven Regelung

Für die Lösung des Optimalsteuerungsproblems der nichtlinearen modellprädiktiven Regelung wird in dieser Arbeit die Entwicklungsumgebung *MATLAB*<sup>18</sup> in der Version R2021b verwendet. Die Implementierung der modellprädiktiven Regelung erfolgt hierbei mit der *Model Predictive Control Toolbox*<sup>19</sup>. Um den Algorithmus im Experimentalsystem schnell und effizient ausführen zu können, wurde dieser mit Hilfe von *MATLAB Coder*<sup>20</sup> in C++ Code konvertiert und mit einer *ROS 2*-Schnittstelle<sup>21</sup> für die Interprozesskommunikation erweitert.

### 4.4 Zeit- und wertdiskrete Trajektorienoptimierung

Die zeit- und wertdiskrete Trajektorienoptimierung mit dynamischer Programmierung stellt ein Verfahren zur Generierung global optimaler UAV-Referenztrajektorien für die sensormodellbestimmte Befliegungsplanung dar. Die diskreten Zustandsübergänge werden

---

<sup>18</sup>MATLAB ist eine von MathWorks entwickelte, proprietäre Multi-Paradigma-Programmiersprache und Entwicklungsumgebung für numerische Berechnungen. Link auf die offizielle Webseite: <https://www.mathworks.com/products/matlab.html>

<sup>19</sup>Die Model Predictive Control Toolbox bietet Funktionen für die Entwicklung modellprädiktiver Regelungen und Solver zur Lösung des Optimalsteuerungsproblems. Link auf die offizielle Webseite: <https://de.mathworks.com/products/model-predictive-control.html>

<sup>20</sup>Mit dem MATLAB Coder kann C/C++ Code automatisiert aus MATLAB-Code für unterschiedliche Hardwareplattformen generiert werden. Link auf die offizielle Webseite: <https://de.mathworks.com/products/matlab-coder.html>

<sup>21</sup>ROS (Robot Operating System) ist eine Sammlung von Software-Frameworks für die Entwicklung von Robotersoftware. ROS bietet unterschiedliche Dienste an, u. a. für die Interprozesskommunikation. Link auf die offizielle Webseite: <https://www.ros.org/>



#### 4.4 Zeit- und wertdiskrete Trajektorienoptimierung

hierbei durch Dubins-Pfade realisiert, worauf bereits in Abschnitt 3.5.3 eingegangen wurde. Mit der Umsetzung der zeit- und wertdiskrete Trajektorienoptimierung wird die *Forschungsfrage 3.1* (Wie kann eine, bezüglich der Detektionsleistung, global optimale Flugtrajektorie generiert werden, unter Berücksichtigung sensorplattformspezifischer und missionsspezifischer Anforderungen?) aus Abschnitt 2.4 abschließend beantwortet. Des Weiteren wird die *Anforderung 4* (Die sensormodellbestimmte Befliegungsplanung soll die Fähigkeit zur automatisierten Generierung optimaler Flugtrajektorien besitzen.) aus Abschnitt 3.2 umgesetzt und erfüllt.

Bei diesem Verfahren wird mit Hilfe der dynamischen Programmierung eine Referenztrajektorie aus aneinander gereihten Dubins-Pfadsegmenten generiert. Um die Anforderung an eine glatte Trajektorie aus Abschnitt 3.5.1 zu erfüllen, wird in einem nachgeschalteten Prozess die Trajektorie geglättet.

In Abb. 4.7 ist das System Schaltbild der zeit- und wertdiskreten Trajektorienoptimierung dargestellt. Die Eingangsgrößen *Perception Maps* und *Sensorfootprints* stammen aus dem vorangegangenen Prozessschritt, der in Abschnitt 4.2 beschrieben wurde. Zusätzliche Informationen, die für die dynamische Programmierung und die Glättung der Trajektorie erforderlich sind, werden vorab erstellt und sind als Daten und Modelle hinterlegt. Die Ausgangsgrößen des Prozessschritts sind die *UAV-Referenztrajektorie* und die *Sensorfootprints*. In Analogie zum vorangegangenen Prozessschritt werden auch hier die Sensorfootprints unverändert an den nachfolgenden Bearbeitungsschritt übergeben.

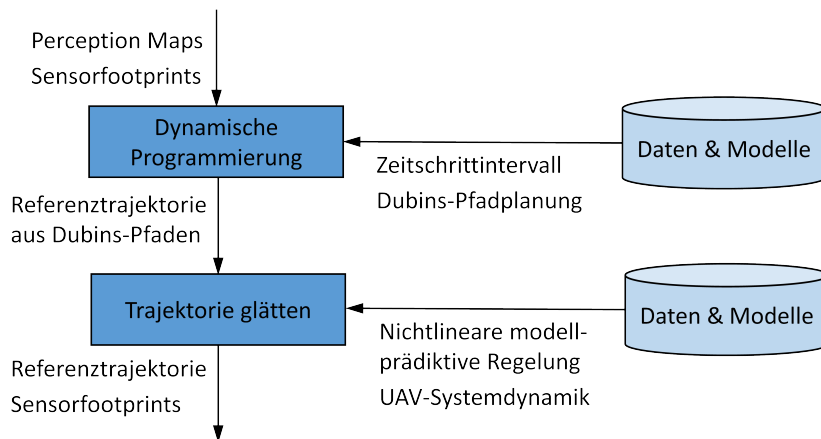


Abbildung 4.7: Systemübersicht der beiden Prozessschritte für die dynamische Programmierung und die nachgeschaltete Glättung der Trajektorie. Prozessspezifische Daten und Modelle werden vorab erstellt und während des Prozessschritts abgerufen.

#### 4.4.1 Dynamische Programmierung mit Dubins-Pfaden

In Abschnitt 3.6.2 wurde bereits das Verfahren der dynamischen Programmierung, einschließlich der Zustandsübergangskosten und der zustandsabhängigen Kosten, formal eingeführt. In diesem Abschnitt wird auf die Anpassungen eingegangen, um die dynamische Programmierung für die sensormodellbestimmte Befliegungsplanung nutzen zu können. Die Trajektorienoptimierung erfolgt für zeitlich und räumlich diskretisierte Zustände (vgl. Gleichung (3.42)).

In Abb. 4.8 ist das Prinzip der zeit- und wertdiskreten Trajektorienoptimierung mit Hilfe der dynamischen Programmierung abgebildet. Durch die Sensorpfadplanung wird der Verlauf des Sensorfootprintpfads im aufzuklärenden Gebiet definiert und die Positionierung der zeitschrittspezifischen Sensorfootprints festgelegt. Dies ist in Bild A exemplarisch durch drei Footprints dargestellt. Für jeden dieser Footprints wird eine individuelle Perception Map berechnet (Bild B). In Bild C ist die optimale Referenztrajektorie eingezeichnet, die aus verketteten Dubins-Pfaden zusammengesetzt ist und durch drei diskrete UAV- / Gitterpositionen verläuft.

Die zulässigen diskreten UAV-Positionen in der Bewegungsebene werden durch ein zweidimensionales kartesisches Gitter beschrieben. Der äquidistante Gitterpunktabstand  $d_{grid}$  nach Gleichung (4.12) ist eine Funktion der UAV-Fluggeschwindigkeit  $v^{ref}$  und dem Zeitschrittintervall  $\Delta t$  für die zeit- und wertdiskrete Optimierung. Für den Prozess der dynamischen Programmierung werden nur die Gitterpunkte berücksichtigt, die innerhalb der relevanten, zeitschrittabhängigen Perception Map liegen.

$$d_{grid} = v^{ref} \cdot \Delta t \quad (4.12)$$

In Gleichung (3.45) wurden jedem Systemzustand  $x_k^i$  für den Zeitschritt  $k$  die zustandsabhängigen Kosten  $c_{state,k}^i$  zugewiesen. Die zustandsabhängigen Kosten entsprechen der Detektionsleistung an der diskreten Gitterposition  $n_k^i$  und  $e_k^i$ . Diese resultiert aus der lokalen Detektionsleistung der zeitschrittspezifischen Perception Map und ist unabhängig vom diskreten Gierwinkel  $\psi_k^i$  des UAVs. Die Werte der Detektionsleistung, und damit die zustandsabhängigen Kosten, sind per Definition  $c_{state,k}^i \in [0, 1]$ .

Darüber hinaus können jedem Paar von Zuständen  $x_k^i$  und  $x_{k+1}^j$  die Zustandsübergangskosten  $c_{trans,k}^{ij}$  aus Gleichung (3.44) zugeordnet werden. Die Zustandsübergangskosten repräsentieren die Länge des in Gleichung (3.28) berechneten Dubins-Pfads von der Startkonfiguration  $x_k^i$  zur Zielkonfiguration  $x_{k+1}^j$  und sind in Gleichung (4.13) angegeben. Jede Startkonfiguration (gilt analog für die Zielkonfiguration mit Index  $j$ ) wird durch den diskreten Zustandsvektor  $x_k^i = [e_k^i, n_k^i, \psi_k^i]^T$  aus Gleichung (3.42) beschrieben. Hier-

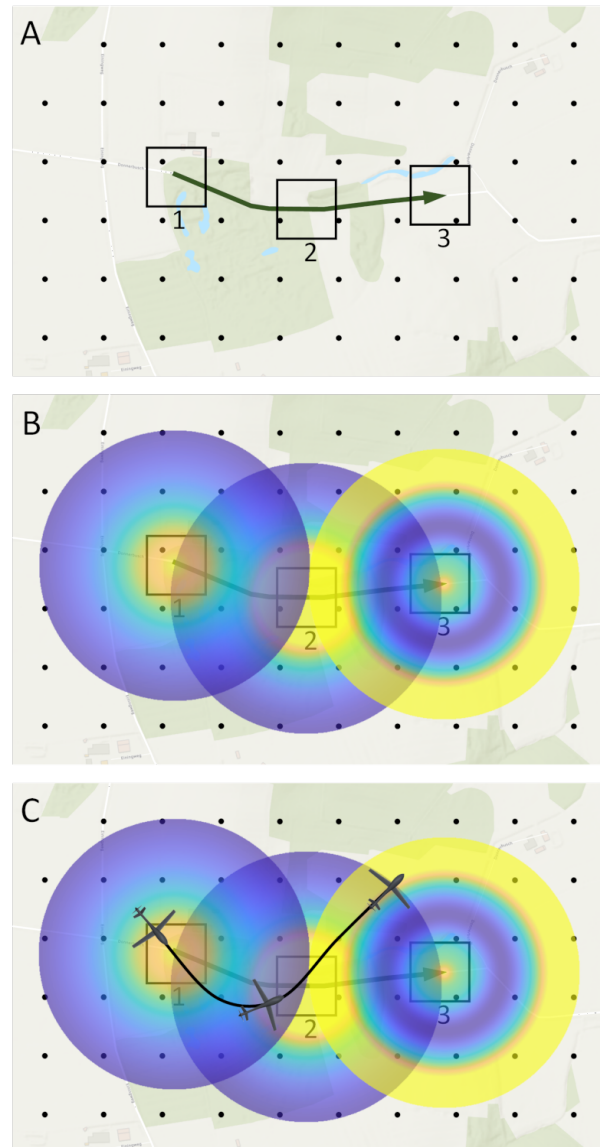


Abbildung 4.8: Prinzipdarstellung der zeit- und wertdiskreten Trajektorienoptimierung. In Bild A ist der Sensorfootprintpfad (grün), drei ausgewählte Sensorfootprints (1-3) und die Gitterpunkte für die diskreten Zustandsübergänge dargestellt. In Bild B sind die Perception Maps abgebildet, die mit den drei Sensorfootprints korrelieren. Helle bzw. gelbe Bereiche repräsentieren eine hohe zu erwartende Detektionsleistung. In Bild C ist die Flugtrajektorie, bestehend aus aneinander gereihten Dubins-Pfaden, zu sehen. Die drei UAV-Symbole markieren die diskreten UAV-Positionen auf dem Gitter. (Abbildung ist nicht maßstabsgetreu)

#### 4 Umsetzung des Konzepts in ein Experimentalsystem

bei wird die östliche und nördliche Position  $e_k^i, n_k^i$  durch die spezifische Gitterposition bestimmt. Jeder Gitterposition sind  $m_{\psi,k} = 24$  zulässige Gierwinkel  $\psi_k^i = 0^\circ, 15^\circ, \dots, 345^\circ$  zugeordnet.

$$c_{trans,k}^{ij} = \begin{cases} \infty & \text{für } s_k^{ij} < s_{min} \\ \frac{1}{s_{max}-s_{min}} \cdot s_k^{ij} + \frac{-s_{max}}{s_{max}-s_{min}} & \text{für } s_{min} \leq s_k^{ij} \leq s_{max} \\ \infty & \text{für } s_k^{ij} > s_{max} \end{cases} \quad (4.13)$$

$$s_{min} = \Delta t \cdot v^{ref} \quad (4.14)$$

$$s_{max} = 2 \cdot r_{min} \cdot \pi \quad (4.15)$$

In Gleichung (4.14) ist  $s_{min}$  die minimale Distanz, die das UAV innerhalb eines Zeitschrittes  $\Delta t$  mit der Geschwindigkeit  $v^{ref}$  zurücklegen kann. Demgegenüber ist  $s_{max}$  in Gleichung (4.15) definiert als die maximal zulässige Weglänge, die einem Vollkreis entspricht. Hierbei ist  $r_{min}$  der minimale Kurvenradius des Dubins-Pfads, der mit Gleichung (3.27) berechnet werden kann. Gleichung (4.13) verdeutlicht, dass kurze Weglängen zu geringen Übergangskosten führen und große Weglängen bestraft werden. Dubins-Pfadlängen kürzer als  $s_{min}$  sind nicht realisierbar und Pfadlängen größer als  $s_{max}$  sind unerwünscht. Um dem Rechnung zu tragen, werden diese mit unendlichen Kosten belegt. Zwischen den zulässigen Distanzen  $s_{min}$  und  $s_{max}$  ergeben sich die normierten Übergangskosten  $c_{trans,k}^{ij} \in [0, 1]$ .

Da die Weglänge  $s_k^{ij}$  direkt proportional zur Flug- / Aufklärungsdauer ist, sind kurze Weglängen ein Optimierungsziel der Befliegungsplanung, welches aus den *missionsspezifischen Anforderungen* aus Abschnitt 3.2 abgeleitet ist. Im Gegensatz zur zeitkontinuierlichen Trajektorienoptimierung aus Abschnitt 4.3 kann die Weglänge bei dem hier vorgestellten Verfahren zwischen den Zeitschritten variieren und Werte zwischen  $s_{min}$  und  $s_{max}$  annehmen. Aus diesem Grund wird die Weglänge der daraus resultierenden UAV-Referenztrajektorie in der Regel länger sein, als die Weglänge der Trajektorie, die mit nichtlinearer modellprädiktiver Regelung für einen identischen Aufklärungsauftrag generiert wird.

In Abschnitt 3.6.2 wurde bereits beschrieben, dass aus der Rückwärtsrechnung die Kosten  $J(x_N^i) = c_{state,N}^i$  für den letzten Zeitschritt  $k = N$  folgen. Für alle weiteren Zeitschritte  $k = 0, \dots, N-1$  erhält man die minimalen Gesamtkosten  $J(x_k^i)$  nach Gleichung (3.47) bzw. (4.16) unter Verwendung des Prinzips der Optimalität. Um die Zusammenhänge leichter nachvollziehen zu können, ist die Gleichung zur Berechnung der minimalen Gesamtkosten

#### 4.4 Zeit- und wertdiskrete Trajektorienoptimierung

nachfolgend nochmals dargestellt:

$$J(x_k^i) = \min_{j=0, \dots, n_{k+1}} \left( p_w \cdot c_{state,k}^i + (1 - p_w) \cdot c_{trans,k}^{ij} + J(x_{k+1}^j) \right) \quad (4.16)$$

Diese Gesamtkosten ergeben sich aus der Summe der zustandsabhängigen Kosten  $c_{state,k}^i$  des aktuell betrachteten Zustands, den Zustandsübergangskosten  $c_{trans,k}^{ij}$  zum nachfolgenden Zustand und den minimalen Gesamtkosten  $J(x_{k+1}^j)$  von diesem Nachfolgezustand bis zum letzten Zustand. Mit dem Gewichtungsfaktor  $p_w$  kann die Priorisierung der Detektionsleistung (zustandsabhängige Kosten) gegenüber der Länge der Flugtrajektorie (Zustandsübergangskosten), und damit der Aufklärungsdauer, beeinflusst werden. Ein hoher Gewichtungsfaktor bevorzugt eine hohe Detektionsleistung, mit dem Nachteil, dass sich dadurch die Länge der Trajektorie erhöhen kann. Demgegenüber begünstigt ein niedriger Gewichtungsfaktor eine kürzere Trajektorie, was zu einer reduzierten Aufklärungsdauer führt. Im Umkehrschluss kann dies allerdings zu einer Verringerung der zu erwartenden Detektionsleistung führen. Der Wert für  $p_w$  ist in Tabelle 4.4 zu finden.

Im letzten Arbeitsschritt wird die global optimale Trajektorie ermittelt, indem die Gesamtkosten  $J(x_0^i)$  aller zugehörigen Anfangszustände  $x_0^i$  miteinander verglichen und die global minimalen Gesamtkosten  $J(x_0^*)$  identifiziert werden. Der Anfangszustand  $x_0^*$  ist damit Startzustand der *global optimalen Trajektorie*. Diese global optimale Trajektorie besteht aus verketteten Dubins-Pfadsegmenten. Zwischen der Start- und Zielkonfiguration ist der Verlauf der Dubins-Pfade durchgehend stetig, allerdings nur abschnittsweise differenzierbar. Daher wird die Forderung nach einer glatten Trajektorie aus Abschnitt 3.5.1 nicht erfüllt. Aus diesem Grund wird im nächsten Abschnitt ein Verfahren vorgestellt, mit dem eine glatte Referenztrajektorie aus den Dubins-Pfaden generiert werden kann.

In Tabelle 4.4 sind die Parametereinstellungen zusammengefasst, die für die zeit- und wertdiskrete Trajektorienoptimierung mit dynamischer Programmierung verwendet werden. Das Zeitschrittintervall wurde in vorangegangenen Experimenten ermittelt und auf  $\Delta t = 1,5$  s festgelegt. Das Zeitschrittintervall beeinflusst den Gitterpunktabstand und damit die Anzahl der diskreten UAV-Positionen. In Abschnitt 3.6.2 wurde gezeigt, dass die Anzahl der Zustandsausprägungen maßgeblich die Anzahl der Rechenschritte und somit auch die Berechnungsdauer beeinflusst. Die Festlegung des Zeitschrittintervalls ist daher ein Kompromiss zwischen der erzielbaren räumlichen Auflösung (Genauigkeit) der Trajektorie und dem damit verbundenen Berechnungsaufwand.

Im Gegensatz zur Trajektorienoptimierung mit modellprädiktiver Regelung, ist das Zeitschrittintervall bei der dynamischen Programmierung eine reine Rechengröße. Auf Grund dessen, dass die Fluggeschwindigkeit des UAVs konstant ist, die zurückgelegte

Tabelle 4.4: Parametereinstellung für die zeit- und wertdiskrete Trajektorienoptimierung mit dynamischer Programmierung

Parameter	Wert	Bemerkung
UAV-Fluggeschwindigkeit $v$	35 m/s	aus Tabelle 3.6
Zeitschrittintervall $\Delta t$	1,5 s	vorgegeben
Gitterpunktabstand $d_{grid}$	52,5 m	aus Gleichung (4.12)
Gewichtungsfaktor $p_w$	0,5	vorgegeben
Anzahl der diskreten Gierwinkel $m_{\psi,k}$	24	vorgegeben
minimaler Kurvenradius $r_{min}$	150 m	aus Tabelle 3.6
minimale Dubins-Pfadlänge $s_{min}$	52,5 m	aus Gleichung (4.14)
maximale Dubins-Pfadlänge $s_{max}$	942,5 m	aus Gleichung (4.15)

Flugstrecke des UAVs aber innerhalb des Zeitschrittintervalls zwischen  $s_{min}$  und  $s_{max}$  variieren kann, muss der Zeitschritt angepasst und gegebenenfalls verlängert werden. In Folge dessen kann die Flug- bzw. Aufklärungsdauer (siehe Abschnitt 6.3.1) nicht direkt aus der Länge und der Anzahl der Zeitschrittintervalle abgeleitet werden.

#### 4.4.2 Realisierung der dynamischen Programmierung mit Dubins-Pfaden

Der Algorithmus für die dynamische Programmierung der zeit- und wertdiskreten UAV-Referenztrajektorie wurde in der Programmiersprache *Python 3.8* realisiert und auf einem PC mit Intel Core i7-6800K Prozessor (6 Kerne mit einer Taktrate von 3,4 GHz) und 32 GB Arbeitsspeicher ausgeführt. Die zeitintensive Berechnung der minimalen Gesamtkosten (siehe Gleichung (3.47)) wurde hierbei durch Multiprocessing auf mehrere Prozessorkerne verteilt. Die Zustandsübergangskosten werden aus einem Datensatz abgerufen, der im Vorfeld der Trajektorienoptimierung erstellt wurde. In diesem Datensatz sind für alle zulässigen diskreten Start- und Zielkonfigurationen die Dubins-Pfade und deren Pfadlänge hinterlegt. Durch die Nutzung dieses a priori erstellten Datensatzes kann auf die zyklische Berechnung der Dubins-Pfade verzichtet und dadurch Rechenzeit eingespart werden.

Der Fokus in dieser Forschungsarbeit liegt auf dem Nachweis der Machbarkeit der sensormodellbestimmten Befliegungsplanung. Die verwendeten Funktionalitäten und deren Algorithmen sind zweckmäßig implementiert, weshalb der Performanz bei der Ausführung der Algorithmen in dieser Arbeit eine untergeordnete Bedeutung zugemessen wurde.

### 4.4.3 Glättung der Dubins-Pfade

Im nächsten Schritt wird die UAV-Referenztrajektorie, die aus verketteten Dubins-Pfadsegmenten besteht, geglättet. Dieser Prozessschritt wird durchgeführt, um einen kontinuierlichen Rollwinkelübergang entlang der gesamten Flugbahn zu erreichen. Für die Umsetzung der Glättung wird auf das Verfahren der nichtlinearen modellprädiktiven Regelung zurückgegriffen, das bereits in Abschnitt 3.6.3 vorgestellt wurde. Zu diesem Zweck werden die Dubins-Pfadsegmente von der Start- bis zur Zielkonfiguration mit dem Zeitschrittintervall der zeitkontinuierlichen Trajektorienoptimierung diskretisiert. Die daraus resultierenden Positionen und Gierwinkel sind die Sollwertvorgaben für die Trajektorienoptimierung mit modellprädiktiver Regelung. Das Ergebnis der Optimierung ist die finale UAV-Referenztrajektorie, die über den gesamten Verlauf  $C^2$ -stetig ist (vgl. Abschnitt 3.5.1) und zudem die vorgegebenen flugdynamischen Randbedingungen (u. a. Rollrate, Rollwinkel und Geschwindigkeitsbegrenzungen) erfüllt.

## 4.5 Implementierung der Sensorsteuerung

Der letzte Prozessschritt der sensormodellbestimmten Befliegungsplanung befasst sich mit der Berechnung der Sensorsteuerung. Die Umsetzung der Sensorsteuerung resultiert aus der *Anforderung 1* (Die sensormodellbestimmte Befliegungsplanung soll eine automatisierte Planung zur Sensorsteuerung enthalten.) aus Abschnitt 3.2. Darüber hinaus ist die Entwicklung und Umsetzung der Sensorsteuerung der letzte Schritt, um das *Zwischenziel 4* (Überführung des Konzepts der sensormodellbestimmten Befliegungsplanung in ein Experimentalsystem.) aus Abschnitt 1.3 erfüllen zu können.

In Abb. 4.9 ist das Systemschaltbild für die Berechnung der Sensorsteuerung dargestellt. Die Eingangsgrößen *Referenztrajektorie* und *Sensorfootprints* stammen aus dem vorangegangenen Prozessschritt, der in Abschnitt 4.3 bzw. Abschnitt 4.4 beschrieben wurde. Zusätzliche Informationen, wie z. B. das Zeitschrittintervall oder die Sensoreigenschaften, werden als Daten und Modelle abgerufen. Das Ergebnis dieses letzten Prozessschritts sind die *UAV-Referenztrajektorie* und die *Sensorsteuerung*. Diese beiden Informationen werden im *Erweiterten Sensor- und Perzeptionsmanagement* (vgl. Abb. 3.1) benötigt, um den Aufklärungsauftrag weiter abarbeiten zu können.

Die Sensorsteuerung ist die zeitschrittabhängige Sollwertvorgabe für den Elevationswinkel  $\alpha$  und den Azimutwinkel  $\varphi$  zur Ausrichtung des Sensors, sowie den Bildwinkel  $\beta$  (Zoom) zur Adaption der Größe des Sensorfootprints am Boden. Über die Größe des Sensorfootprints wird die GSD innerhalb des Footprints bestimmt. Mit der Sensorsteuerung

#### 4 Umsetzung des Konzepts in ein Experimentalsystem

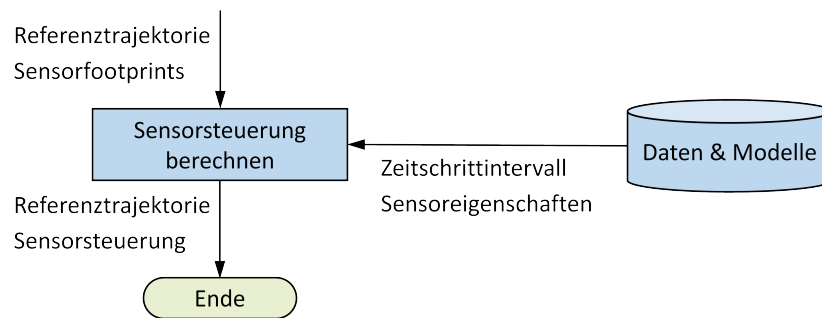


Abbildung 4.9: Systemübersicht zur Berechnung der Sensorsteuerung als den letzten Schritt in der sensormodellbestimmten Befliegungsplanung

wird eine automatisierte Sensorführung während der Abarbeitung des Aufklärungsauftrags erreicht. Die Sensorfootprintbewegung am Boden wird dabei zeitlich und räumlich mit der Flugbewegung des UAVs koordiniert. Darüber hinaus wird der Sensoroperateur von der Aufgabe der eigentlichen Sensorführung entlastet.

Bei der Sensorsteuerung wird für jeden Zeitschritt  $t_n$  der Elevations- und Azimutwinkel (siehe Abb. 4.10) ermittelt, der eingestellt werden muss, damit die Sensorsichtlinie des bildgebenden Sensors auf den korrelierenden zeitschrittspezifischen Sensorfootprint am Boden ausgerichtet ist. Die Sensorsichtlinie beschreibt dabei die direkte Verbindung des Sensors mit dem Mittelpunkt des Sensorfootprints am Boden. Die Sensorsichtlinie ist in Abb. 4.10 als blaue Linie dargestellt.

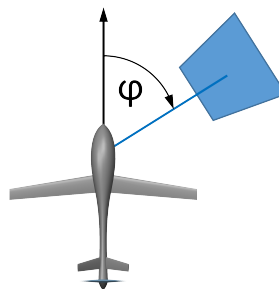


Abbildung 4.10: Darstellung des Azimutwinkels  $\varphi$ , der zwischen der Sensorsichtlinie (blaue Linie) und der Flugrichtung des UAVs aufgespannt wird. Die Sensorsichtlinie ist die direkte Verbindung zwischen dem bildgebenden Sensor am UAV und dem Mittelpunkt des Sensorfootprints (blau) am Boden.

Neben der Ermittlung der Sensorausrichtung wird bei der Sensorsteuerung auch der zeitschrittspezifische Bildwinkel  $\beta$  für das Sensorsystem berechnet. Der Bildwinkel wird



in den technischen Grenzen so angepasst, dass die Sollwertvorgabe der GSD  $gsd_{ref}$  aus Tabelle 3.4 eingehalten werden kann.

Die Sensorsteuerung ist an das Zeitschrittintervall der zeitkontinuierlichen bzw. der zeit- und wertdiskreten Trajektorienoptimierung angepasst. Das bedeutet, die Berechnung des Elevations-, Azimut- und Bildwinkels erfolgt ausschließlich für diese diskreten Zeitschritte. Zwischen den Zeitschritten werden die Winkel interpoliert.

## 4.6 Zusammenfassung

In diesem Kapitel wurde das funktionale Konzept der sensormodellbestimmten Befliegungsplanung aus Abschnitt 3.3 in ein Experimentalsystem überführt. Hierfür wurden die einzelnen Prozessschritte der Befliegungsplanung detailliert beschrieben. Mit der Überführung des Konzepts in ein Experimentalsystem wurde das *Zwischenziel 4* (Überführung des Konzepts der sensormodellbestimmten Befliegungsplanung in ein Experimentalsystem.) aus Abschnitt 1.3 umgesetzt und erfüllt.

Darüber hinaus wurden die beiden *Forschungsfragen 3.1* und *3.2* aus Abschnitt 2.4 vollständig beantwortet. Dies war eine Voraussetzung, um das damit verbundene *Zwischenziel 3* (Auswahl und Implementierung geeigneter mathematischer Optimierungsmethoden zur Generierung optimaler Flugtrajektorien in Abhängigkeit der Detektionsleistung sowie sensorplattformspezifischer und missionspezifischer Anforderungen.) abschließen zu können. Nachfolgend ist der aktuelle Stand der Untersuchungen zusammengefasst.

Zwischenziel	1	2	3	4	5
Forschungsfrage	1.1	2.1 & 2.2	3.1 & 3.2	—	5.1 - 5.3
Bearbeitung in Kapitel	3	3	3 & 4	4	6



## 5 Grundlagen der Evaluierung

In diesem Abschnitt werden die Grundlagen für die Evaluierung der sensormodellbestimmten Befliegungsplanung erarbeitet, die im nächsten Kapitel angewendet werden. Ausgehend vom *Zwischenziel 5* aus Abschnitt 1.3 und den damit verbundenen *Forschungsfragen 5.1 bis 5.3* aus Abschnitt 2.4 werden Untersuchungsgegenstände definiert, die wiederum Grundlage für die Entwicklung der Einzelexperimente in Kapitel 6 sind. Des Weiteren werden Benchmark-Trajektorien bestimmt, die zur quantitativen Bewertung der zu erwartenden Detektionsleistung<sup>22</sup> benötigt werden. Um methodisch bewerten zu können, ob durch die Befliegungsplanung die Detektionsleistung gesteigert werden kann, werden die Grundlagen des Einstichproben t-Tests erläutert. Abschließend wird auf die Experimentalumgebung eingegangen, die für die Durchführung der Experimente in der Simulation entwickelt wurde.

### 5.1 Definition der Untersuchungsgegenstände

Für die Evaluierung der sensormodellbestimmten Befliegungsplanung in der Simulation werden im Folgenden *Untersuchungsgegenstände* festgelegt. Auf deren Grundlage werden die Einzelexperimente definiert, die dazu beitragen, dass die *Forschungsfragen 5.1 bis 5.3* aus Abschnitt 2.4 beantwortet werden und das damit verbundene *Zwischenziel 5* (Validierung der sensormodellbestimmten Befliegungsplanung im Experimentalsystem anhand repräsentativer Aufklärungsmissionen unter Berücksichtigung perzeptiver und sensorplattformspezifischer Anforderungen.) erfüllt werden kann. Nachfolgend sind die einzelnen Forschungsfragen und die daraus abgeleiteten Untersuchungsgegenstände (abgekürzt: U) aufgelistet:

- *Forschungsfrage 5.1*: Welches Potenzial bietet die sensormodellbestimmte Befliegungsplanung zur Steigerung der Detektionsleistung mit zeitkontinuierlicher Trajektorienoptimierung sowie mit zeit- und wertdiskreter Trajektorienoptimierung?

---

<sup>22</sup>Es wird darauf hingewiesen, dass es sich bei der Detektionsleistung in der Regel um eine Prognose handelt, die auf Grundlage eines Sensorleistungsmodells erstellt wird. Aus diesem Grund wird mitunter der erweiterte Begriff der *zu erwartenden Detektionsleistung* verwendet, um auf diesen Sachverhalt aufmerksam zu machen.

## 5 Grundlagen der Evaluierung

- **U 1.1:** Evaluierung der Befliegungsplanung mit zeitkontinuierlicher Trajektorienoptimierung im Vergleich mit Benchmark-Trajektorien.
- **U 1.2:** Evaluierung der Befliegungsplanung mit zeit- und wertdiskreter Trajektorienoptimierung im Vergleich mit Benchmark-Trajektorien.
- *Forschungsfrage 5.2:* Wie werden perzeptive und sensorplattformsspezifische Anforderungen und Randbedingungen bei der sensormodellbestimmten Befliegungsplanung eingehalten?
  - **U 2.1:** Evaluierung der Befliegungsplanung hinsichtlich perzeptiver Anforderungen.
  - **U 2.2:** Evaluierung der Befliegungsplanung bezüglich sensorplattformsspezifischer Randbedingungen.
- *Forschungsfrage 5.3:* Wie beeinflussen die zu untersuchenden Optimierungsmethoden die Aufklärungsdauer und die erforderliche Berechnungsdauer?
  - **U 3.1:** Evaluierung der Befliegungsplanung hinsichtlich der Aufklärungsdauer.
  - **U 3.2:** Evaluierung der Befliegungsplanung hinsichtlich der erforderlichen Berechnungsdauer.

Der *Untersuchungsgegenstand 1.1* zielt darauf ab, die sensormodellbestimmte Befliegungsplanung mit zeitkontinuierlicher Trajektorienoptimierung zu evaluieren. Demgegenüber soll mit Hilfe des *Untersuchungsgegenstands 1.2* die Befliegungsplanung mit zeit- und wertdiskreter Trajektorienoptimierung untersucht werden. Zu diesem Zweck werden in Abschnitt 5.2 Benchmark-Trajektorien eingeführt, deren zu erwartende Detektionsleistungswerte als Vergleichsgrößen dienen. Für die Evaluierung werden die Detektionsleistungen ermittelt, die aus den Benchmark-Trajektorien resultieren und mit den Werten der Detektionsleistung verglichen, die aus der zeitkontinuierlichen bzw. zeit- und wertdiskreten Trajektorienoptimierung stammt. Durch den quantitativen Vergleich soll überprüft werden, ob und ggf. wie stark die Detektionsleistung durch den Ansatz der sensormodellbestimmten Befliegungsplanung gesteigert werden kann.

Mit dem *Untersuchungsgegenstand 2.1* soll evaluiert werden, ob die Sensorsteuerung fähig ist, die perzeptiven Anforderungen und Randbedingungen bezüglich der GSD und des Sensorbildwinkels umsetzen und einhalten zu können. Darüber hinaus soll mit dem *Untersuchungsgegenstand 2.2* überprüft werden, ob die sensorplattformsspezifischen Anforderungen und Randbedingungen (u. a. Rollrate, Fluggeschwindigkeit) im Rahmen der Befliegungsplanung eingehalten werden.

Der *Untersuchungsgegenstand 3.1* zielt darauf ab, die Aufklärungsdauer zu evaluieren, die aus den optimierten Referenztrajektorien resultiert. Abschließend wird mit dem *Untersuchungsgegenstand 3.2* die Berechnungsdauer untersucht, die für die numerische Berechnung der zeitkontinuierlichen sowie zeit- und wertdiskreten Referenztrajektorie aufgewendet werden muss.

## 5.2 Generierung von Benchmark-Trajektorien

Um die Befliegungsplanung durch die *Untersuchungsgegenstände 1.1* und *1.2* evaluieren zu können, werden im Folgenden drei unterschiedliche Benchmark-Trajektorien definiert. Diese Benchmark-Trajektorien wurden aus unterschiedlichen Warteschleifenmuster (engl.: Loitering Pattern) übernommen, die u. a. in [192] beschrieben werden. Diese Muster werden insbesondere in der militärischen Luftfahrt genutzt, um eine kontinuierliche Beobachtung eines (statischen) Objekts am Boden durchführen zu können.

Im Rahmen der Evaluierung wird davon ausgegangen, dass diese Benchmark-Trajektorien entweder automatisiert von einem Flugführungssystem an Bord des UAVs generiert oder von einem UAV-Operator vorgegeben werden. In beiden Fällen erfolgt die Planung ohne die Kenntnis oder explizite Berücksichtigung der Sensorleistungsmodelle aus Abschnitt 3.4.

In dieser Arbeit werden die drei folgenden Warteschleifenmuster als Benchmark-Trajektorien eingesetzt, die in Abb. 5.1 abgebildet sind. In Klammern ist die englischsprachige Bezeichnung angegeben, die im weiteren Verlauf verwendet wird:

- Kreisbahnmuster (Circle)
- Rennbahnmuster (Racetrack)
- Achtförmiges-Muster (Figure-8)

Alle drei Muster haben gemeinsam, dass sie aus einer einfach zu modellierenden Geometrie mit einem geschlossenen Linienzug bestehen. Somit kann jedes Muster beliebig oft durchlaufen werden. Die Form und Lage der Warteschleifenmuster werden durch zwei Punkte (Startpunkt  $p_{start}$  und Stützpunkt  $p_{sup}$ ), eine Richtungsangabe und einen Radius  $r_{loiter}$  spezifiziert. Es wurden diese drei Muster gewählt, da sie unterschiedliche Formeigenschaften aufweisen, wie z. B. eine konstante Bahnkrümmung für das Circle-Muster oder Abschnitte mit geraden Linienzügen beim Racetrack- und Figure-8-Muster.

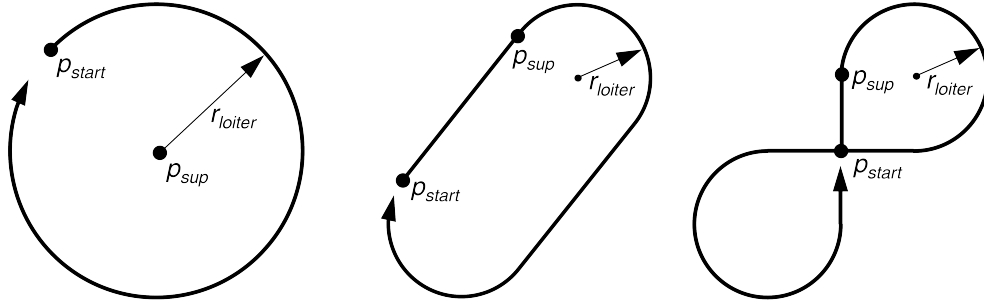


Abbildung 5.1: Darstellung der Benchmark-Trajektorien *Circle* (links), *Racetrack* (mitig) und *Figure-8* (rechts). Zusätzlich sind der Startpunkt  $p_{start}$ , der Stützpunkt  $p_{sup}$  und die Bahnrichtung skizziert. Der Radius  $r_{loiter}$  wird vorgegeben oder ergibt sich aus dem Muster.

### 5.3 Einstichproben t-Test

In den Abschnitten 6.1.2.1 und 6.1.3.1 wird die sensormodellbestimmte Befliegungsplanung mit zeitkontinuierlicher Trajektorienoptimierung evaluiert. Nach Maßgabe des *Untersuchungsgegenstands 1.1* soll hierfür die Detektionsleistung, die aus der optimierten UAV-Referenztrajektorie resultiert, mit der Detektionsleistung verglichen werden, die aus der Benchmark-Trajektorie hervorgeht. Die Werte der Detektionsleistungen, die aus den beiden unterschiedlichen Trajektorien resultieren, werden als *paarweise verbundene Stichproben* erfasst, worauf in Abschnitt 6.1.2.1 näher eingegangen wird. Um methodisch überprüfen zu können, ob die Detektionsleistung durch die Befliegungsplanung gesteigert werden kann, wird ein *rechtsseitiger Einstichproben t-Test* (Hypothesentest) durchgeführt. In diesem Abschnitt werden die Grundlagen des Hypothesentests erläutert, bevor die Methode in Kapitel 6 zur Anwendung kommt. Die Notation und Erläuterungen orientieren sich an der Arbeit von Fahrmeir et al. [193].

Um den rechtsseitigen Einstichproben t-Test durchführen zu können, wird die *Nullhypothese*  $H_0$  und die komplementäre *Alternativhypothese*  $H_1$  formuliert:

- $H_0$ : Die Detektionsleistung, die aus den UAV-Referenztrajektorien resultiert, ist kleiner oder gleich der Detektionsleistung, die aus den Benchmark-Trajektorien hervorgeht.
- $H_1$ : Die Detektionsleistung, die aus den UAV-Referenztrajektorien resultiert, übertrifft die Detektionsleistung, die aus den Benchmark-Trajektorien resultiert.

Die Werte der Detektionsleistung, die sich aus den UAV-Referenztrajektorien ergeben, werden mit  $X_i = X_1, X_2, \dots, X_n$  und  $n \in \mathbb{N}$  als Stichprobengröße bezeichnet. Analog dazu

werden die Werte der Detektionsleistung, die aus den Benchmark-Trajektorien stammen, mit  $Y_i = Y_1, Y_2, \dots, Y_n$  bezeichnet. Folglich beschreiben  $X_i$  und  $Y_i$  mit  $X_i, Y_i \in [0, 1]$  zwei paarweise verbundene Stichproben. Um die Hypothesen im t-Test überprüfen zu können, wird im Folgenden nur die paarweise Differenz der Detektionsleistungen, die nach Gleichung (5.1) ermittelt wird, untersucht:

$$D_i = X_i - Y_i \quad (5.1)$$

Für die beiden verbundenen Stichproben kann die Nullhypothese  $H_0$  überprüft werden indem die Differenzen  $D_i$  gebildet werden und der daraus berechnete Mittelwert  $\bar{x}_{Diff}$  gegen den Erwartungswert  $\mu_0 = 0$  getestet wird:

- $H_0: \bar{x}_{Diff} \leq 0$
- $H_1: \bar{x}_{Diff} > 0$

Das Signifikanzniveau des t-Tests wird auf  $\alpha = 0,01$  festgelegt. Somit liegt die Wahrscheinlichkeit, dass die Nullhypothese fälschlicherweise abgelehnt wird, bei 1%. Die Nullhypothese wird abgelehnt und die Alternativhypothese trifft zu, falls der  $p$ -Wert kleiner als  $\alpha$  ist. Der  $p$ -Wert  $p_{val}$  gibt an, wie wahrscheinlich die Ergebnisse der Stichproben sind, für den Fall, dass die Nullhypothese zutrifft [193]. Der t-Test kann mit Hilfe von Softwareprogrammen zur statistischen Datenanalyse, wie z. B. der *Statistics and Machine Learning Toolbox*<sup>23</sup> von MATLAB, durchgeführt werden. Der  $p$ -Wert wird dabei direkt berechnet.

Für die Berechnung des t-Tests wird der Mittelwert der Differenzen  $\bar{x}_{Diff}$  nach Gleichung (5.2) sowie die Standardabweichung  $s$  der Stichprobe nach Gleichung (5.3) bestimmt.

$$\bar{x}_{Diff} = \frac{1}{n} \sum_{i=1}^n D_i \quad (5.2)$$

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (D_i - \bar{x}_{Diff})^2} \quad (5.3)$$

Um den Einstichproben t-Test in der oben beschriebenen Form anwenden zu können muss vorab überprüft werden, ob die Differenzen der Detektionsleistungswerte normalverteilt sind. Bei einer Stichprobengröße von  $n > 30$  ist die Annahme einer Normalverteilung

---

<sup>23</sup>Die Statistics and Machine Learning Toolbox bietet u. a. Funktionen zur Analyse und Modellierung von Daten mit Hilfe von Statistik und maschinellem Lernen. Link auf die offizielle Webseite: <https://de.mathworks.com/products/statistics.html>

valide [193]. Für eine Stichprobengröße  $n \leq 30$  kann die Normalverteilung beispielsweise durch den Shapiro-Wilk-Test [194] überprüft werden.

## 5.4 Experimentalumgebung

Die Experimente werden in einer Simulationsumgebung durchgeführt. Zu diesem Zweck wurde eine Experimentalumgebung entwickelt, deren Systemkonzept in Abb. 5.2 dargestellt ist. Mit Hilfe einer grafischen Benutzeroberfläche (engl.: Graphical User Interface, GUI) kann die Simulation gesteuert und die Ergebnisse (u. a. Referenztrajektorie und Sensorfootprintpfad) auf einer Karte angezeigt werden. Über das Missionsskript werden die notwendigen Informationen für die Durchführung eines Aufklärungsauftrags (z. B. Routen- oder Flächenaufklärung) an die Befliegungsplanung weitergeleitet. Das Modul *Befliegungsplanung* umfasst die Funktionalitäten für die Sensorpfadplanung, die Trajektorienoptimierung und Funktionen zur Berechnung der Sensorsteuerung, wie sie in Abschnitt 4.1 beschrieben wurden. Die Ergebnisse aus der Befliegungsplanung werden an die GUI übertragen und zusätzlich zur Auswertung aufgezeichnet.

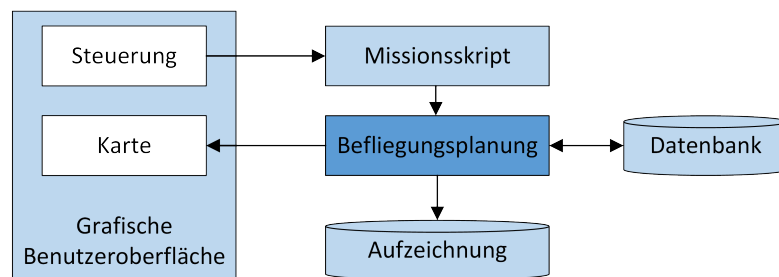


Abbildung 5.2: Aufbau der Experimentalumgebung zur Untersuchung der sensormodellbestimmten Befliegungsplanung in der Simulation



## 6 Evaluierung und Ergebnisse

In diesem Kapitel wird die sensormodellbestimmte Befliegungsplanung anhand repräsentativer Aufklärungsmissionen in der Simulation evaluiert, um das *Zwischenziel 5* aus Abschnitt 1.3 umsetzen und erfüllen zu können. Zunächst erfolgt die Untersuchung der zu erwartenden Detektionsleistung, die mit Hilfe der zeitkontinuierlichen bzw. zeit- und wertdiskreten Trajektorienoptimierung erzielt werden kann. Im nächsten Schritt wird evaluiert, ob die perzeptiven und sensorplattformspezifischen Anforderungen und Randbedingungen eingehalten werden. Abschließend wird die erforderliche Aufklärungs- und Berechnungsdauer untersucht, die mit der Trajektorienoptimierung verbunden ist.

### 6.1 Evaluierung der Befliegungsplanung durch Beispielmissionen

Das *Zwischenziel 5* aus Abschnitt 1.3 gibt vor, die sensormodellbestimmte Befliegungsplanung anhand repräsentativer Aufklärungsmissionen zu evaluieren. Zu diesem Zweck werden ein *Routenaufklärungsszenario* (Abschnitt 6.1.2) und ein *Flächenaufklärungsszenario* (Abschnitt 6.1.3) definiert. Diese beiden Aufklärungsszenarien bilden den Rahmen für die Durchführung der Einzelexperimente, die aus den einzelnen *Untersuchungsgegenständen 1.1 bis 3.2* aus Abschnitt 5.1 abgeleitet werden.

Die Untersuchung der Befliegungsplanung wird auf Basis der beiden Aufklärungsszenarien und unter Nutzung von zwei unterschiedlichen Sensorleistungsmodellen durchgeführt. Das erste Leistungsmodell, das für die Evaluierung genutzt wird, bildet eine Perzeptionskette für einen Objektklassifikator ab, der auf einer *Classification Cascade* (CC) beruht (vgl. Abschnitt 3.4.3). Dieses Sensorleistungsmodell wird im weiteren Verlauf als *CC-SLM* bezeichnet. Das zweite Sensorleistungsmodell soll die Perzeptionskette mit einem Deep Learning basierten Objektdetektor abbilden, dem ein YOLOv3-Detektor zu Grunde liegt (vgl. Abschnitt 3.4.4). Dieses Leistungsmodell wird folglich als *YOLO-SLM* bezeichnet.

Bei der sensormodellbestimmten Befliegungsplanung mit zeitkontinuierlicher Trajektorienoptimierung wird die UAV-Referenztrajektorie durch das Optimierungsverfahren der nichtlinearen modellprädiktiven Regelung (NMPC) generiert. Entsprechend wird dieser Ansatz nachfolgend als *NMPC-Trajektorienoptimierung* abgekürzt. Demgegenüber

## 6 Evaluierung und Ergebnisse

wird bei der Befliegungsplanung mit zeit- und wertdiskreter Trajektorienoptimierung die UAV-Referenztrajektorie mit Hilfe der dynamischen Programmierung (DP) ermittelt und dieses Verfahren abgekürzt als *DP-Trajektorienoptimierung* bezeichnet.

Die atmosphärischen Zustände im Operationsgebiet des UAVs werden für das Routen- und Flächenaufklärungsszenario als konstant und identisch festgelegt. Diese Entscheidung wurde bereits in Abschnitt 1.4 getroffen und erläutert. In Tabelle 6.1 sind die atmosphärischen Zustände aufgelistet, die für das Sensorleistungsmodell CC-SLM im Aufklärungsgebiet gelten. Die atmosphärischen Zustände fließen in die Berechnung des Leistungsmodells ein, wodurch die Detektionsleistung und letztendlich der Verlauf der UAV-Referenztrajektorie beeinflusst wird.

Tabelle 6.1: Parametereinstellung der atmosphärischen Zustände für das Sensorleistungsmodell CC-SLM

<b>Parameter</b>	<b>Wert</b>	<b>Bemerkung</b>
Tageszeit	16 h	vorgegeben
Monat	Juni	vorgegeben
Bewölkung	25 %	vorgegeben
Nebel	0 %	vorgegeben
Niederschlag	0 %	vorgegeben

Daneben sind in Tabelle 6.2 die atmosphärischen Zustände zusammengefasst, die für die Nutzung des Sensorleistungsmodells YOLO-SLM festgelegt wurden. Hier gilt ebenfalls die Annahme, dass die Zustände im gesamten Operationsgebiet des UAVs konstant und identisch sind.

Tabelle 6.2: Festlegung der atmosphärischen Zustände für das Sensorleistungsmodell YOLO-SLM

<b>Parameter</b>	<b>Wert</b>	<b>Bemerkung</b>
Jahreszeit	Herbst	vorgegeben
Tageszeit	Tag	vorgegeben
Bewölkungsgrad	bedeckt	vorgegeben
Sicht	klar	vorgegeben
Untergrund	nass	vorgegeben

## 6.1 Evaluierung der Befliegungsplanung durch Beispielmissionen

Für die Evaluierung der Befliegungsplanung werden perzeptive und sensorplattform-spezifische Anforderungen und Randbedingungen definiert, die im Folgenden zusammengefasst sind. Bezüglich der perzeptiven Anforderungen wird festgelegt:

- Im Rahmen der Evaluierung werden die beiden Sensorleistungsmodelle CC-SLM und YOLO-SLM genutzt.
- Die zu Grunde liegende Perzeptionskette umfasst einen elektrooptischen Sensor, dessen Auflösung in Tabelle 4.1 dokumentiert ist.
- Der Bildwinkel des Sensors und der Sollwert der Bodenpixelauflösung werden aus Tabelle 3.4 übernommen.
- Die Sensorfootprintgeschwindigkeit ist in Tabelle 4.1 definiert.

Die sensorplattform-spezifischen Randbedingungen, die für die Befliegungsplanung betrachtet werden, sind nachfolgend zusammengefasst:

- Die Grenzwerte für die Rollrate, den Rollwinkel, die Beschleunigung in Flugrichtung sowie die UAV-Fluggeschwindigkeit werden aus den Tabellen 3.5 und 3.7 übernommen.
- Das UAV bewegt sich in einer konstanten Flughöhe, die in Tabelle 3.4 definiert ist.

Für zusätzliche Informationen und Parameter, die für die Befliegungsplanung erforderlich sind, wird auf die vorangegangenen Kapitel 3 und 4 verwiesen.

### 6.1.1 Versuchsplanung

Für die Evaluierung der sensormodellbestimmten Befliegungsplanung durch die beiden *Untersuchungsgegenstände U 1.1* und *U 1.2* aus Abschnitt 5.1 wurde ein Experimentalplan erstellt, der in Abb. 6.1 dargestellt ist. Sowohl für das Routenaufklärungsszenario als auch für das Flächenaufklärungsszenario werden jeweils  $12 + 12 + 2 = 26$  Einzelexperimente in der Simulation durchgeführt.

Die Evaluierung wird wie folgt durchgeführt: Für die Untersuchung der Befliegungsplanung mit NMPC-Trajektorienoptimierung wird für jedes Einzelexperiment eine Benchmark-Trajektorie definiert und eine NMPC-optimierte Referenztrajektorie berechnet. Dabei wird festgelegt, dass die Startkonfiguration des UAVs (Startposition und Gierwinkel) für beide Trajektorien identisch ist. Dies ist eine Voraussetzung, um das Detektionsergebnis abschließend vergleichen und bewerten zu können. Zudem sind die Sensorfootprintgeschwindigkeiten beider Trajektorien identisch und konstant. Für die Untersuchung werden jeweils zwölf Experimente mit den Sensorleistungsmodellen YOLO-SLM und CC-SLM durchgeführt.

## 6 Evaluierung und Ergebnisse

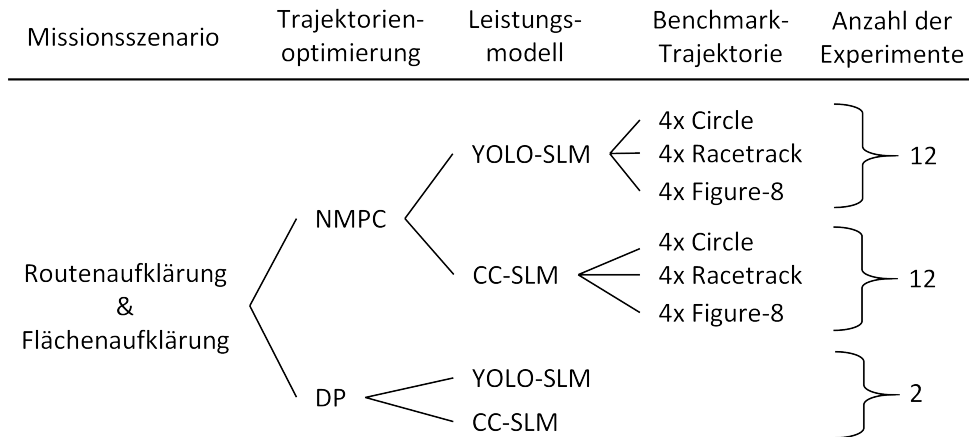


Abbildung 6.1: Experimentalplan für das Routen- und Flächenaufklärungsszenario

Für die Evaluierung der Befliegungsplanung mit NMPC-Trajektorienoptimierung erfüllen die Benchmark-Trajektorien zwei Aufgaben: Zum einen besteht ihr Zweck darin, dass die zu erwartende Detektionsleistung, die aus den Benchmark-Trajektorien resultiert, als Vergleichs- bzw. Bezugsgröße für die quantitative Bewertung der Referenztrajektorie genutzt wird. Andererseits wird durch die Positionierung und Ausrichtung der Benchmark-Trajektorie die Startkonfiguration für die Referenztrajektorie festgelegt. Dies ist notwendig, da mit Hilfe der NMPC-Trajektorienoptimierung nur lokal optimale Referenztrajektorien auf Basis einer vorgegebenen Startkonfiguration generiert werden können.

Die Trajektorienoptimierung in den Einzelexperimenten wird durch unterschiedliche Größen (unabhängige Variablen) beeinflusst. Die konstanten perzeptiven und sensorplattformenspezifischen Einflüsse wurden bereits im vorangegangenen Abschnitt 6.1 erläutert. Nachfolgend ist eine Auswahl an veränderlichen Einflussgrößen aufgelistet, die im Rahmen der Experimente als Freiheitsgrade in die Optimierung mit einfließen:

- Der variable Elevations- und Bildwinkel, hervorgerufen durch die UAV-Bewegung
- Die Variation des Sensorfootprintpfads
- Die Veränderung des topographischen Zustandsvektors entlang des Sensorfootprintpfads
- Die Variation der Referenztrajektorien und deren Startposition

Die abhängige Variable bzw. das Ergebnis der Trajektorienoptimierung ist die zu erwartende Detektionsleistung, die aus der optimierten Referenztrajektorie resultiert. Für die Evaluierung der Trajektorienoptimierung mit nichtlinearer modellprädiktiver

## 6.1 Evaluierung der Befliegungsplanung durch Beispielmissionen

Regelung (*Untersuchungsgegenstand 1.1*) wird die zu erwartende Detektionsleistung mit der Detektionsleistung verglichen, die aus den Benchmark-Trajektorien hervorgeht. Um eine etwaige Steigerung der Detektionsleistung durch die sensormodellbestimmte Befliegungsplanung statistisch und methodisch belegen zu können, wird der t-Test aus Abschnitt 5.3 durchgeführt.

Im Experimentalplan (vgl. Abb. 6.1) wurde die Anzahl der Einzelerperimente für die NMPC-optimierte Referenztrajektorie mit dem Sensorleistungsmodell YOLO-SLM bzw. CC-SLM auf jeweils zwölf Experimente festgelegt. Dies wird damit begründet, dass die Anzahl ausreichend groß sein muss, um eine statistische Relevanz für die Auswertung mit dem t-Test zu erreichen. Die zwölf Einzelerperimente resultieren aus jeweils vier verschiedenen Konfigurationen der drei unterschiedlichen Benchmark-Trajektorien (Circle, Racetrack und Figure-8) aus Abschnitt 5.2. Die unterschiedlichen Konfigurationen der Benchmark-Trajektorien unterscheiden sich in der Ausdehnung, der Positionierung, der Durchlaufrichtung und der Orientierung (gilt nur für Racetrack- und Figure-8-Muster) der Trajektorien. In Abb. 6.4 ist eine Auswahl der Trajektorien abgebildet.

Für die Trajektorienoptimierung mit dynamischer Programmierung (*Untersuchungsgegenstand 1.2*) werden zwei Simulationen für das Routenaufklärungsszenario und zwei für das Flächenaufklärungsszenario durchgeführt, wobei jeweils eine Simulation mit dem Sensorleistungsmodell YOLO-SLM und dem Leistungsmodell CC-SLM erfolgt. Das Ergebnis ist in beiden Fällen eine global optimale Referenztrajektorie, wobei die zu erwartende Detektionsleistung mit der theoretisch maximal erreichbaren Detektionsleistung verglichen wird. Zusätzlich wird der Mittelwert der Detektionsleistung berechnet, der aus mehreren Benchmark-Trajektorien gebildet wird. Dieser Mittelwert dient als Bezugsgröße um eine mögliche Steigerung der Detektionsleistung durch die DP-optimierten Referenztrajektorien quantifizieren zu können.

Für die Durchführung der Experimente wird Folgendes angenommen: Die aufzuklärende Route bzw. die aufzuklärende Fläche wurde vorab festgelegt und ist bekannt. Die Sensorpfadplanung wurde im Vorfeld mit Hilfe des in Abschnitt 4.2 beschriebenen Verfahrens durchgeführt und ist identisch sowohl für die optimierte UAV-Referenztrajektorie als auch für die Benchmark-Trajektorie. Es wird davon ausgegangen, dass die Benchmark-Trajektorien von einem UAV-Operator oder einem Flugführungssystem an Bord des UAVs festgelegt werden, ohne dass das entsprechende Sensorleistungsmodell bekannt ist oder berücksichtigt wird. Die Detektionsleistung, die aus dem Abfliegen der beiden Trajektorien resultiert, wird allerdings auf der Grundlage desselben Leistungsmodells bestimmt. Dies ist notwendig, um die Detektionsergebnisse abschließend miteinander vergleichen und bewerten zu können.

### 6.1.2 Beispielmission Routenaufklärungsszenario

Beim Routenaufklärungsszenario, das in Abb. 6.2 abgebildet ist, folgt der Sensorfootprintpfad (grüne Linie) einem Straßenzug. Die aufzuklärende Fläche, die dabei vom Sensorfootprint überstrichen wird, umfasst sowohl ländliche (Wiese, Vegetation, Wasser) als auch urbane Bereiche (Straßen, Gebäude). Der Verlauf des Sensorfootprintpfads wurde so gewählt, dass dieser verschiedenen starke Richtungswechsel in unterschiedliche Richtungen aufweist. Damit soll die Anpassungsfähigkeit der UAV-Referenztrajektorie an den Sensorfootprintpfad aufgezeigt werden. Der Sensorfootprintpfad hat eine Länge von ca. 2,2 km und verläuft in Pfeilrichtung. In der Abbildung ist der erste Sensorfootprint maßstabsgetreu als türkisfarbenes Quadrat skizziert.

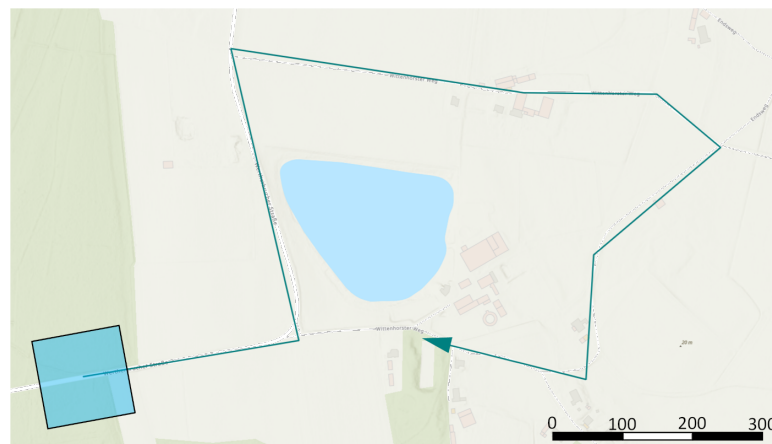


Abbildung 6.2: Routenaufklärung entlang einer Straße. Die grüne Linie markiert den Verlauf des Sensorfootprintpfads, wobei der erste Sensorfootprint als türkisfarbenes Quadrat visualisiert ist.

In Abb. 6.3 sind die topographischen Zustände (Vegetation, Wiese, Straße, Wasser, Gebäude, Geländeunebenheit) in Abhängigkeit der Länge des Sensorfootprintpfads aufgetragen. Das Diagramm beschreibt die anteilige Zusammensetzung des topographischen Zustandsvektors im Sensorfootprint entlang des Footprintpfads. Mit Ausnahme der Geländeunebenheit ergibt die Summe der einzelnen Zustände den Wert Eins. Beim Routenaufklärungsszenario verläuft der Sensorfootprintpfad entlang einer Straße. Aus diesem Grund ist der topographische Zustand *Straße* ein durchgehender Bestandteil des Zustandsvektors über die gesamte Länge des Footprintpfads. Des Weiteren dominiert der Zustand *Wiese* über weite Bereiche entlang des Pfads. Nach ca. 1200 m und gegen Ende der Route überstreicht der Sensorfootprint urbanes Gebiet mit einer hohen Gebäudedichte. Auf Grund des flachen Terrains nimmt der Zustand *Geländeunebenheit*

## 6.1 Evaluierung der Befliegungsplanung durch Beispielmissionen

einen Wert ein, der gleich oder nahezu Null beträgt. Durch das Diagramm wird verdeutlicht, dass der topographische Zustandsvektor entlang der aufzuklärenden Route stark variiert: von bewaldeter Fläche (Vegetation) über Wiesen bis hin zu bebautem Gebiet. Diese Variation beeinflusst auch die Ausprägung der Perception Maps, die mit Hilfe des Sensorleistungsmodells CC-SLM generiert werden und in Abb. 6.8 dargestellt sind.

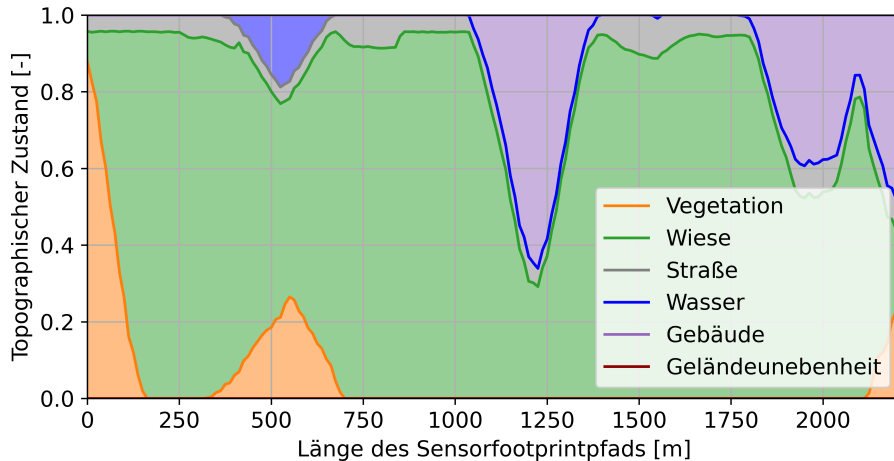


Abbildung 6.3: Verlauf der topographischen Zustände entlang der aufzuklärenden Route

### 6.1.2.1 Befliegungsplanung mit NMPC-Trajektorienoptimierung

Die Perception Maps, die aus dem YOLO-SLM resultieren, sind identisch entlang der gesamten Aufklärungsroute. Der Grund hierfür ist, dass für die Berechnung der Perception Maps durch das YOLO-SLM nur atmosphärische Zustände betrachtet werden und diese für den gesamten aufzuklärenden Bereich als konstant und homogen vorausgesetzt wurden. In Abb. 3.16 ist die Perception Map dargestellt, die aus der Variation des Elevationswinkels und den atmosphärischen Zuständen resultiert, die im vorangegangenen Abschnitt 6.1 festgelegt wurden. Aus der Abbildung wird zudem ersichtlich, dass diese Perception Map Bereiche mit abschnittsweise konstanter Detektionsleistung aufweist. Demgegenüber ist der Verlauf der Detektionsleistung in den Perception Maps, die aus dem CC-SLM resultieren, stetig und differenzierbar.

### Ergebnisdarstellung für das Sensorleistungsmodell YOLO-SLM

Das Ergebnis der sensormodellbestimmten Befliegungsplanung ist beispielhaft in Abb. 6.4 zu sehen. Hierbei sind vier der zwölf Einzelexperimente zum Routenaufklärungsszenario mit dem Sensorleistungsmodell YOLO-SLM abgebildet. In den vier Einzelbil-

## 6 Evaluierung und Ergebnisse

dem ist jeweils der Sensorfootprintpfad (dunkelgrün), die NMPC-optimierte UAV-Referenztrajektorie (blau) und die Benchmark-Trajektorie (hellgrün) eingezeichnet. Die Startkonfiguration der Referenztrajektorie ist identisch mit der Benchmark-Trajektorie und zur Verdeutlichung durch ein UAV-Symbol markiert. In Bild a) ist eine Ausprägung der Circle-Benchmark-Trajektorie und in b) eine Variante des Racetrack-Musters zu sehen. In c) und d) sind zwei verschiedene Ausprägungen des Figure-8-Musters dargestellt.

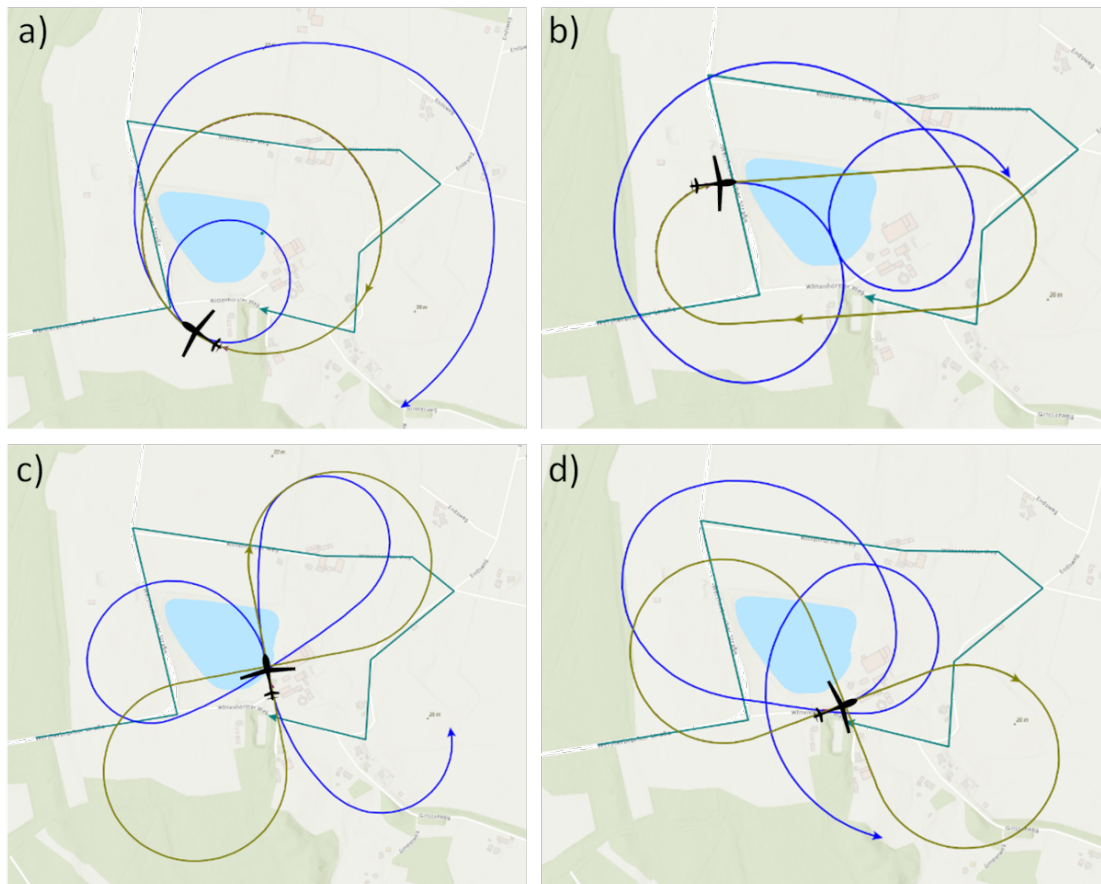


Abbildung 6.4: Trajektorienoptimierung für das Routenaufklärungsszenario mit dem Sensorleistungsmodell YOLO-SLM. Die blaue Linie zeigt die NMPC-optimierte Referenztrajektorie und die hellgrüne Linie repräsentiert die Benchmark-Trajektorie. Die Startpunkte beider Trajektorien sind identisch und durch ein schwarzes UAV-Symbol gekennzeichnet. In a) bis d) sind unterschiedliche Benchmark-Trajektorien dargestellt: das Circle-Muster in a), das Racetrack-Muster in b) sowie zwei verschiedene Varianten des Figure-8-Musters in c) und d). Aus Gründen der Übersichtlichkeit ist der Sensorfootprint nicht dargestellt.



## 6.1 Evaluierung der Befliegungsplanung durch Beispielmissionen

In Abb. 6.5 ist der Verlauf der Detektionsleistung für den Datensatz „route-yolo-fig8-2“ aus Tabelle 6.3 in Abhängigkeit des Sensorfootprintpfads abgebildet. Das Routenaufklärungsszenario, das hierfür exemplarisch verwendet wurde, ist in Abb. 6.4 in Bild c) zu sehen. Die blaue Linie in Abb. 6.5 markiert die zu erwartende Detektionsleistung, die aus der NMPC-optimierten Referenztrajektorie resultiert. Demgegenüber ist die Detektionsleistung der Benchmark-Trajektorie rot markiert. Die schwarze Linie stellt die *theoretisch maximal erreichbare Detektionsleistung* als Obergrenze dar, die im Mittel bei  $p_{max} = 0,936$  liegt. Die kurzzeitigen Einbrüche der Detektionsleistung bei der NMPC-optimierten Referenztrajektorie werden damit begründet, dass die starken Richtungswechsel des Sensorfootprintpfads zu einer großen Verschiebung der Perception Maps und damit der Bereiche mit hoher Detektionsleistung führen. Unter Berücksichtigung der flugdynamischen Beschränkungen resultiert aus der NMPC-Optimierung eine Referenztrajektorie, die nicht vollständig in dem Bereich mit hoher Detektionsleistung verläuft.

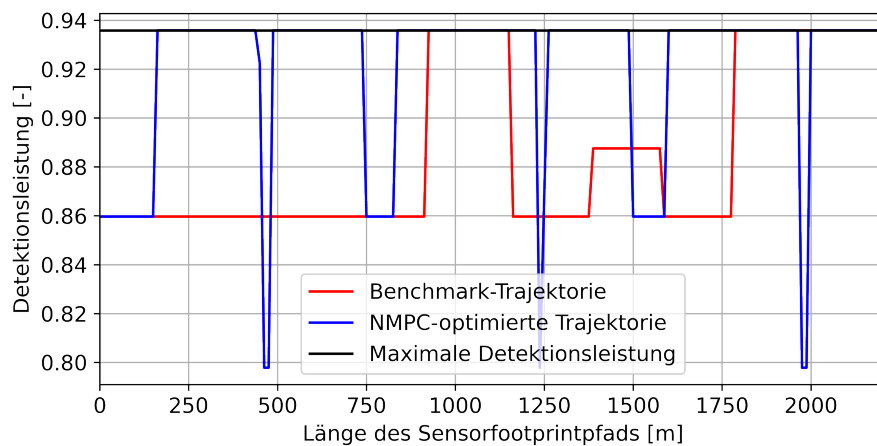


Abbildung 6.5: Ergebnis der Detektionsleistung für die NMPC-optimierte Referenztrajektorie (blau) und die Benchmark-Trajektorie (rot) entlang des Sensorfootprintpfads. Die schwarze Linie markiert die theoretisch maximal erreichbare Obergrenze der Detektionsleistung mit Leistungsmodell YOLO-SLM.

Die Ergebnisse der Einzelerperimente für das Routenaufklärungsszenario mit nichtlinearer modellprädiktiver Regelung sind in Tabelle 6.3 zusammengefasst. Die obere Hälfte der Tabelle (grau hinterlegt) enthält die Ergebnisse für die Trajektorienoptimierung mit dem Sensorleistungsmodell YOLO-SLM. In der unteren Hälfte sind die Daten für das Leistungsmodell CC-SLM dargestellt (blau hinterlegt).

Die Bezeichnung der Einzelerperimente ist in der ersten Spalte *Datensätze* zu finden. In der zweiten Spalte, mit der Bezeichnung *Max. DL*, ist die *theoretisch maximal erreichbare*

Tabelle 6.3: Ergebnisdarstellung der Detektionsleistung (DL) für das Routenaufklärungsszenario mit Sensorleistungsmodell YOLO-SLM (grau hinterlegter Datensatz) und CC-SLM (blau hinterlegter Datensatz)

Datensatz	Max. DL (abs.)	DL aus NMPCC-Optimierung (abs.)	DL aus NMPCC-Optimierung (rel.)	$\bar{x}$	s	DL aus Benchmark-Trajektorie (abs.)	DL aus Benchmark-Trajektorie (rel.)	$\bar{x}$	s	DL Differenz (abs.)	DL Differenz (rel.)	$\bar{x}$	s
route-yolo-circle-1		0,931	99,47%			0,892	95,30%			0,039	4,17% <sub>P</sub>		
route-yolo-circle-2		0,912	97,44%			0,860	91,88%			0,052	5,56% <sub>P</sub>		
route-yolo-circle-3		0,916	97,86%			0,871	93,06%			0,045	4,81% <sub>P</sub>		
route-yolo-circle-4		0,931	99,47%			0,881	94,12%			0,050	5,34% <sub>P</sub>		
route-yolo-racetr-1		0,933	99,68%			0,854	91,24%			0,079	8,44% <sub>P</sub>		
route-yolo-racetr-2	0,936	0,936	100,00%	98,87%	0,85%	0,879	93,91%	93,96%	1,38%	0,057	6,09% <sub>P</sub>	4,91% <sub>P</sub>	1,40% <sub>P</sub>
route-yolo-racetr-3		0,931	99,47%			0,886	94,66%			0,045	4,81% <sub>P</sub>		
route-yolo-racetr-4		0,929	99,25%			0,905	96,69%			0,024	2,56% <sub>P</sub>		
route-yolo-fig8-1		0,918	98,08%			0,877	93,70%			0,041	4,38% <sub>P</sub>		
route-yolo-fig8-2		0,920	98,29%			0,885	94,55%			0,035	3,74% <sub>P</sub>		
route-yolo-fig8-3		0,916	97,86%			0,880	94,02%			0,036	3,85% <sub>P</sub>		
route-yolo-fig8-4		0,932	99,57%			0,884	94,44%			0,048	5,13% <sub>P</sub>		
route-cc-circle-1		0,842	86,71%			0,768	79,09%			0,074	7,62% <sub>P</sub>		
route-cc-circle-2		0,838	86,30%			0,786	80,95%			0,052	5,36% <sub>P</sub>		
route-cc-circle-3		0,795	81,87%			0,738	76,00%			0,057	5,87% <sub>P</sub>		
route-cc-circle-4		0,797	82,08%			0,751	77,34%			0,046	4,74% <sub>P</sub>		
route-cc-racetr-1		0,841	86,61%			0,765	78,78%			0,076	7,83% <sub>P</sub>		
route-cc-racetr-2		0,813	83,73%			0,775	79,81%			0,038	3,91% <sub>P</sub>		
route-cc-racetr-3	0,971	0,790	81,36%	84,19%	1,92%	0,772	79,51%	79,37%	2,27%	0,018	1,85% <sub>P</sub>	4,82% <sub>P</sub>	2,96% <sub>P</sub>
route-cc-racetr-4		0,808	83,21%			0,771	79,40%			0,037	3,81% <sub>P</sub>		
route-cc-fig8-1		0,802	82,60%			0,797	82,08%			0,005	0,51% <sub>P</sub>		
route-cc-fig8-2		0,821	84,55%			0,785	80,84%			0,036	3,71% <sub>P</sub>		
route-cc-fig8-3		0,822	84,65%			0,810	83,42%			0,012	1,24% <sub>P</sub>		
route-cc-fig8-4		0,841	86,61%			0,730	75,18%			0,111	11,43% <sub>P</sub>		

## 6.1 Evaluierung der Befliegungsplanung durch Beispielmissionen

*Detektionsleistung*  $p_{max}$  aufgetragen, die in Abschnitt 3.4.5 bereits beschrieben wurde. Dieser Wert entspricht dem arithmetischen Mittelwert der maximalen Detektionsleistung aus jeder einzelnen Perception Map entlang der aufzuklärenden Route.  $p_{max}$  stellt damit die obere Grenze der zu erwartenden Detektionsleistung für das Einzelexperiment dar. Der Wert ist nur abhängig vom verwendeten Sensorleistungsmodell und dem Verlauf des Umgebungszustandsvektors entlang des Sensorfootprintpfads. Aus diesem Grund ist der Wert der theoretisch maximal erreichbaren Detektionsleistung für alle Einzelexperimente desselben Sensorleistungsmodells identisch.

In den nächsten beiden Spalten sind die Detektionsleistungswerte der NMPC-optimierten Referenztrajektorien und der Benchmark-Trajektorien aufgetragen. In den Spalten sind die Absolutwerte als auch die Relativwerte mit Bezug auf  $p_{max}$  angegeben. Die Werte der Detektionsleistung sind wiederum die arithmetischen Mittelwerte der Detektionsleistung zu jedem Zeitschritt  $\Delta t$  im Einzelexperiment. Zudem sind der Mittelwert  $\bar{x}$  und die Standardabweichung  $s$  für die Relativwerte der Detektionsleistung aus den einzelnen Experimenten angegeben. Die farbliche Markierung verdeutlicht, dass der Mittelwert der Detektionsleistung aus der NMPC-optimierten UAV-Referenztrajektorie höher ist als der Mittelwert der Detektionsleistung, der aus den Benchmark-Trajektorien resultiert.

Die Messung der Detektionsleistung, die aus der UAV-Referenztrajektorie und der Benchmark-Trajektorie resultiert, wird als *paarweise verbundene Stichprobe* angesehen. Begründet wird dies damit, dass die Startkonfiguration, der Sensorfootprintpfad und das zu Grunde liegende Sensorleistungsmodell sowohl für die Referenztrajektorie als auch für die Benchmark-Trajektorie identisch sind. Ein potenzieller Unterschied in der Detektionsleistung ist daher ausschließlich auf den Verlauf der UAV-Flugbahn zurückzuführen, wobei diese durch die sensormodellbestimmte Befliegungsplanung bzw. die Benchmark-Trajektorie bestimmt wird.

In der letzten Spalte von Tabelle 6.3 ist die paarweise Differenz der Detektionsleistungswerte nach Gleichung (5.1) sowie deren Mittelwert und Standardabweichung in Prozentpunkten (%P) aufgetragen. Der Mittelwert verdeutlicht, dass eine Steigerung der Detektionsleistung durch die NMPC-Trajektorienoptimierung gegenüber der Detektionsleistung aus den Benchmark-Trajektorien erreicht wurde, auf die im weiteren Verlauf noch genauer eingegangen wird.

In den nachfolgenden Abbildungen sind die Ergebnisse aus Tabelle 6.3 grafisch dargestellt. In Abb. 6.6 sind die Werte der Detektionsleistung aus der NMPC-optimierten Referenztrajektorie (blau) und die Werte aus den Benchmark-Trajektorien (orange) zusammen aufgetragen. Links sind die Ergebnisse als Absolutwerte und rechts als Relativwerte dargestellt. Es zeigt sich, dass die Werte der Detektionsleistung, die aus den

## 6 Evaluierung und Ergebnisse

Referenztrajektorien resultieren, zwischen  $p = 97,44\%$  und  $p = 100,00\%$  liegen und damit der theoretisch maximal erreichbaren Detektionsleistung sehr nahekommen bzw. diese erreichen.

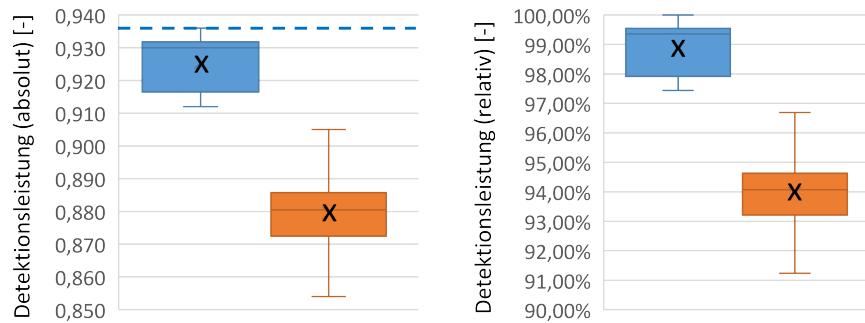


Abbildung 6.6: Ergebnisdarstellung des Routenaufklärungsszenarios mit dem Sensorleistungsmodell YOLO-SLM. Die Detektionsleistungswerte aus der NMPC-optimierten UAV-Referenztrajektorie sind blau dargestellt, die Detektionsleistungswerte aus den Benchmark-Trajektorien sind orange hinterlegt. Die Absolutwerte sind links und die Relativwerte im rechten Diagramm zu sehen. Die blau gestrichelte Linie markiert die theoretisch maximal erreichbare Detektionsleistung  $p_{max}$ .

In Abb. 6.7 ist die paarweise Differenz der zu erwartenden Detektionsleistung  $D_i$  aus den zwölf Einzelexperimenten zur Routenaufklärung mit dem Leistungsmodell YOLO-SLM als Boxplot aufgetragen. Die Differenzwerte wurden in Übereinstimmung mit Gleichung (5.1) ermittelt. Ein Differenzwert größer Null korreliert demzufolge mit einer Steigerung der Detektionsleistung durch die sensormodellbestimmte Befliegungsplanung.

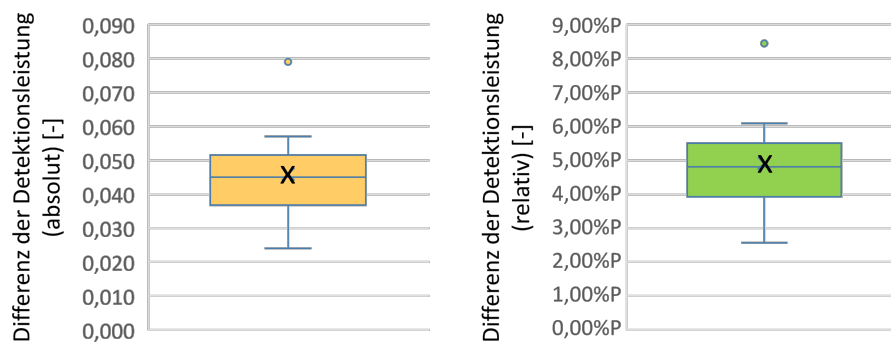


Abbildung 6.7: Paarweise Differenz der Detektionsleistungswerte aus den Einzelexperimenten für das Routenaufklärungsszenario mit dem Sensorleistungsmodell YOLO-SLM. Absolutwerte sind links und Relativwerte in Prozentpunkten (%P) sind rechts dargestellt.

## 6.1 Evaluierung der Befliegungsplanung durch Beispielmissionen

Im linken Diagramm ist die Differenz der Detektionsleistung in Absolutwerten und rechts die dazugehörigen Relativwerte in Prozentpunkten dargestellt. Das schwarze „X“ im Boxplot markiert den arithmetischen Mittelwert (hier:  $\bar{x} = 4,91\%$ P), der aus Tabelle 6.3 entnommen ist. Die Abbildung verdeutlicht, dass alle Werte für die Differenz der Detektionsleistung größer als Null sind. Daraus folgt, dass für jedes Stichprobenpaar eine Steigerung der Detektionsleistung durch die Befliegungsplanung erreicht wurde. Die Differenz bzw. Steigerung der Detektionsleistung liegt dabei in einem Bereich von  $p = 0,024$  (2,56 %P) bis  $p = 0,079$  (8,44 %P), wobei die Relativwerte als Prozentpunkte in Klammern angegeben sind.

Um die Differenz der zu erwartenden Detektionsleistung aus der sensormodellbestimmten Befliegungsplanung und den Benchmark-Trajektorien für das YOLO-SLM methodisch bewerten zu können, wird der t-Test aus Abschnitt 5.3 durchgeführt. Vor dem t-Test wurde mit Hilfe des Shapiro-Wilk-Tests [194] überprüft und nachgewiesen, dass die Differenzwerte normalverteilt sind. Das Ergebnis aus dem t-Test stellt sich wie folgt dar:

Die Steigerung der durchschnittlichen Detektionsleistung von  $\bar{x} = 4,91\%$ P durch die sensormodellbestimmte Befliegungsplanung mit nichtlinearer modellprädiktiver Regelung und dem Sensorleistungsmodell YOLO-SLM ist *signifikant* ( $p_{val} = 7,81 \cdot 10^{-8} < \alpha = 0,01$ ) gegenüber der durchschnittlichen Detektionsleistung, die aus den Benchmark-Trajektorien resultiert. Die Nullhypothese  $H_0$  aus Abschnitt 5.3 muss daher verworfen werden und die Alternativhypothese  $H_1$  wird angenommen.

### Ergebnisdarstellung für das Sensorleistungsmodell CC-SLM

Nachdem die Befliegungsplanung für das Sensorleistungsmodell YOLO-SLM untersucht wurde, wird die Evaluierung mit dem Fokus auf das Leistungsmodell CC-SLM fortgesetzt. Ein Teil der vorangegangenen Erläuterungen sind auch für das CC-SLM gültig, weshalb im Folgenden hauptsächlich auf die Unterschiede und Besonderheiten eingegangen wird.

In Abb. 6.8 ist die aufzuklärende Route (grüne Linie) dargestellt und zusätzlich sind links und rechts exemplarisch einzelne Perception Maps (nicht maßstabsgetreu) abgebildet. Die dargestellten Perception Maps wurden mit Hilfe des Sensorleistungsmodells CC-SLM generiert. Der Verlauf der Detektionsleistung in den Perception Maps wird maßgeblich durch den topographischen Zustandsvektor bestimmt, der sich an der jeweiligen Sensorfootprintposition einstellt. Die stetige Veränderung des topographischen Zustandsvektors entlang des Sensorfootprintpfads hat zur Folge, dass der Verlauf der Detektionsleistung in den unterschiedlichen Perception Maps stark variiert. Die Detektionsleistung in den Perception Maps ist farblich codiert dargestellt. Helle Flächen (gelb markiert) repräsentieren

## 6 Evaluierung und Ergebnisse

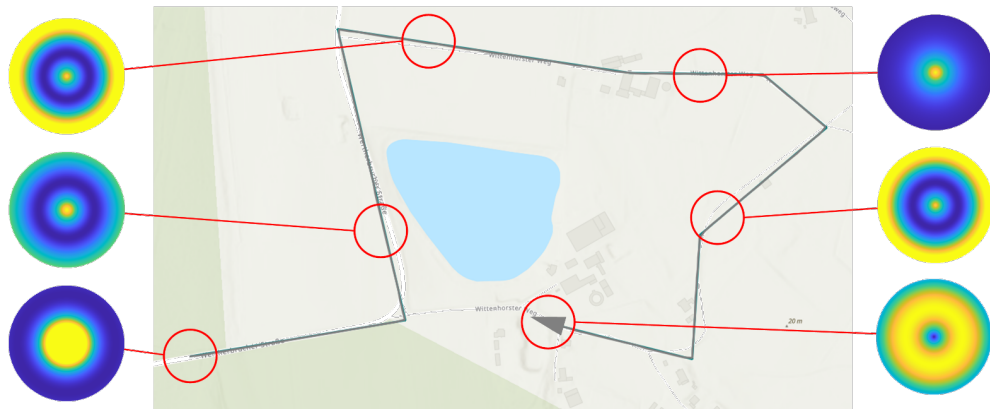


Abbildung 6.8: Darstellung des Routenaufklärungsszenarios. Die grüne Linie markiert die aufzuklärende Route. Links und rechts sind exemplarisch verschiedene Perception Maps (nicht maßstabsgetreu) aufgetragen, die mit Hilfe des Sensorleistungsmodells CC-SLM generiert werden. Helle Bereiche repräsentieren eine hohe Detektionsleistung, wohingegen dunkle Bereiche mit einer niedrigen Detektionsleistung verbunden sind [33].

Bereiche mit hoher Detektionsleistung, wohingegen dunkle Bereiche (blau dargestellt) mit einer niedrigen Detektionsleistung korrelieren.



Abbildung 6.9: Trajektorienoptimierung für das Routenaufklärungsszenario mit dem Sensorleistungsmodell CC-SLM. Die blaue Linie zeigt die NMPC-optimierte Referenztrajektorie und die hellgrüne Linie repräsentiert die Benchmark-Trajektorie. Im linken Bild ist die Benchmark-Trajektorie mit Kreisbahnmuster und rechts mit Rennbahnmuster abgebildet.

Zwei der zwölf durchgeführten Einzelexperimente zur Routenaufklärung mit dem Sensorleistungsmodell CC-SLM sind beispielhaft in Abb. 6.9 dargestellt. Der Sensorfootprintpfad am Boden ist dunkelgrün markiert, die NMPC-optimierte UAV-Referenztrajektorie ist als

## 6.1 Evaluierung der Befliegungsplanung durch Beispielmissionen

blaue Linie und die Benchmark-Trajektorie als hellgrüne Linie eingezeichnet. Die Startkonfiguration der beiden Trajektorien ist wieder identisch und durch das UAV-Symbol gekennzeichnet.

In Abb. 6.10 ist der Verlauf der Detektionsleistung exemplarisch für den Datensatz „route-cc-racetr-1“ aus Tabelle 6.3 aufgetragen, dessen Trajektorienverlauf im rechten Bild von Abb. 6.9 zu sehen ist. Die Detektionsleistung der NMPC-optimierten Referenztrajektorie ist als blaue Linie eingezeichnet. Daneben sind die Benchmark-Trajektorie (rote Linie) und die theoretisch maximal erreichbare Detektionsleistung (schwarze Linie) als Obergrenze abgebildet. Im Gegensatz zum Sensorleistungsmodell YOLO-SLM ist der Verlauf der Obergrenze inhomogen. Dies wird mit der großen Variation der Detektionsleistung in den Perception Maps entlang des Sensorfootprintpfads begründet, was in Abb. 6.8 verdeutlicht wurde. Die Detektionsergebnisse aller zwölf Einzelexperimente sind in der unteren Hälfte von Tabelle 6.3 zusammengefasst.

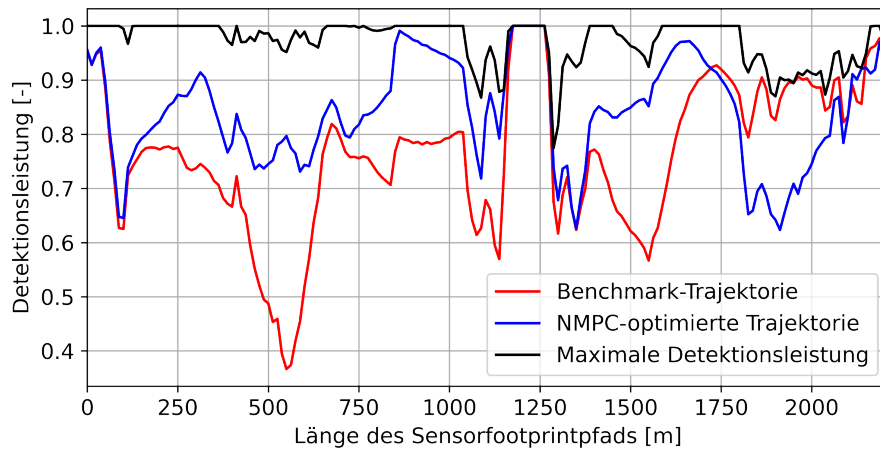


Abbildung 6.10: Darstellung der Detektionsleistung für die NMPC-optimierte Referenztrajektorie (blau) und die Benchmark-Trajektorie (rot) entlang des Sensorfootprintpfads. Die schwarze Linie markiert die theoretisch maximal erreichbare Detektionsleistung als Obergrenze.

Auf Basis der Detektionsergebnisse aus Tabelle 6.3 wurden die nachfolgenden Boxplots zur NMPC-Trajektorienoptimierung mit dem Sensorleistungsmodell CC-SLM erstellt. In Abb. 6.11 sind die Werte der Detektionsleistung aufgetragen, die aus den NMPC-optimierten Referenztrajektorien (blau) und den Benchmark-Trajektorien (orange) resultieren. Im linken Diagramm sind die Ergebnisse als Absolutwerte und rechts als Relativwerte dargestellt. Die Relativwerte der Detektionsleistung, die aus den Referenztrajektorien resultieren, liegen zwischen  $p = 81,36\%$  und  $p = 86,71\%$  bei einem

## 6 Evaluierung und Ergebnisse

Mittelwert von  $\bar{x} = 84,19\%$ . Demgegenüber liegt der Mittelwert der Detektionsleistung aus den Benchmark-Trajektorien bei  $\bar{x} = 79,37\%$ .

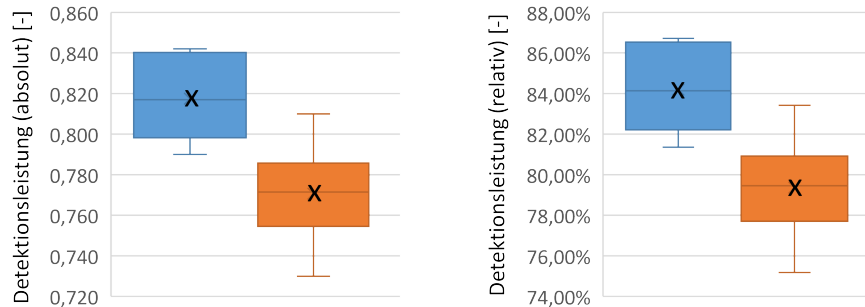


Abbildung 6.11: Ergebnisdarstellung des Routenaufklärungsszenarios mit dem Sensorleistungsmodell CC-SLM. Die Detektionsleistungswerte aus der NMPC-optimierten UAV-Referenztrajektorie sind blau dargestellt und die Werte aus den Benchmark-Trajektorien sind orange hinterlegt. Die Absolutwerte sind links, die Relativwerte rechts abgebildet.

Für die genauere Betrachtung der Ergebnisse ist in Abb. 6.12 die paarweise Differenz der zu erwartenden Detektionsleistung aus den Einzelexperimenten zur Routenaufklärung mit dem Leistungsmodell CC-SLM als Boxplot aufgetragen. Ein Differenzwert größer Null korreliert mit einer Steigerung der Detektionsleistung durch die sensormodellbestimmte Befliegungsplanung. Aus der Abbildung und der Tabelle 6.3 geht hervor, dass für jedes Stichprobenpaar eine Steigerung der Detektionsleistung durch die sensormodellbestimmte Befliegungsplanung erreicht wurde. Die Steigerung liegt dabei zwischen  $p = 0,51\%P$  und  $p = 11,43\%P$  und im Mittel bei  $\bar{x} = 4,82\%P$ .

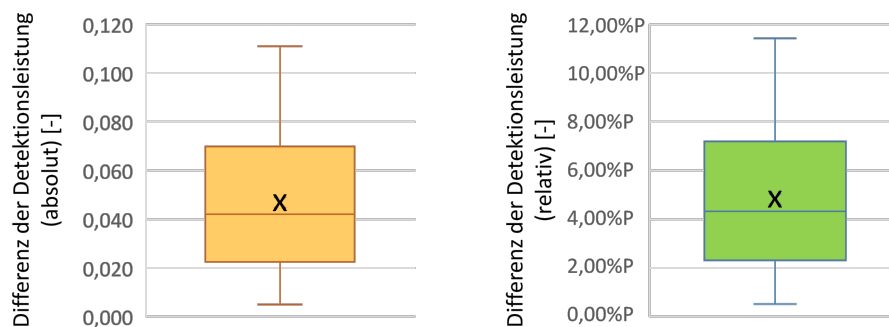


Abbildung 6.12: Paarweise Differenz der Detektionsleistungswerte aus den Einzelexperimenten für das Routenaufklärungsszenario mit dem Sensorleistungsmodell CC-SLM. Absolutwerte sind links und Relativwerte rechts dargestellt.



## 6.1 Evaluierung der Befliegungsplanung durch Beispielmissionen

Um die Differenz der zu erwartenden Detektionsleistung für das Sensorleistungsmodell CC-SLM methodisch bewerten zu können, wurden der Shapiro-Wilk-Test und der t-Test in analoger Weise zur vorangegangenen Untersuchung für das YOLO-SLM durchgeführt. Das Ergebnis des t-Tests ist wie folgt:

Die Steigerung der durchschnittlichen Detektionsleistung um  $\bar{x} = 4,82$  Prozentpunkte durch die sensormodellbestimmte Befliegungsplanung mit nichtlinearer modellprädiktiver Regelung und dem Sensorleistungsmodell CC-SLM ist *signifikant* ( $p_{val} = 1,08 \cdot 10^{-4} < \alpha = 0,01$ ) gegenüber der durchschnittlichen Detektionsleistung, die aus den Benchmark-Trajektorien resultiert. Die Nullhypothese  $H_0$  wird daher, zugunsten der Alternativhypothese  $H_1$ , verworfen.

### 6.1.2.2 Befliegungsplanung mit DP-Trajektorienoptimierung

Nachfolgend wird die sensormodellbestimmte Befliegungsplanung für die zeit- und wertdiskrete Trajektorienoptimierung anhand des Routenaufklärungsszenarios evaluiert (*Untersuchungsgegenstand 1.2*). Die Werte für die Detektionsleistung entlang des Sensorfootprintpfads werden nur für die diskreten Zeitschritte untersucht. Insbesondere bei der DP-optimierten Trajektorie werden zwischen den Zeitschritten mitunter längere Flugstrecken (z. B. beim Kurvenflug) ohne sensorische Datenerfassung (Detektionen) zurückgelegt. In diesen Fällen würde eine kontinuierliche Bewertung der Detektionsleistung zu einer Verfälschung der Ergebnisse führen.

In Abb. 6.13 sind die beiden DP-optimierten und geglätteten Referenztrajektorien für das Routenaufklärungsszenario mit dem Sensorleistungsmodell YOLO-SLM (links) und CC-SLM (rechts) abgebildet. Dabei kann man deutlich den Einfluss der Dubins-Pfadplanung erkennen, weshalb die Referenztrajektorie aus Geraden- und Kurvensegmenten besteht. Insbesondere die geraden Abschnitte der Trajektorie verlaufen überwiegend parallel zur Ost-West- bzw. Nord-Süd-Hauptachse des Gitters (nicht abgebildet, siehe hierfür Abb. 4.8). Dies wird damit begründet, dass bei dieser Konstellation die Referenztrajektorie durch diejenigen Gitterpunkte verläuft, die räumlich den geringsten Abstand  $d_{grid}$  (vgl. Gleichung (4.12)) zueinander haben, wodurch die Länge der Trajektorie minimiert wird. Die geraden Trajektoriensegmente werden durch Kurvensegmente miteinander verbunden, um eine Richtungsänderung der Flugtrajektorie zu erzielen. Durch die Einbindung der Kurvensegmente wird die Länge der UAV-Trajektorie deutlich gesteigert. Mit der Länge der Flugbahn steigt auch die Aufklärungsdauer, worauf in Abschnitt 6.3.1 gesondert eingegangen wird.

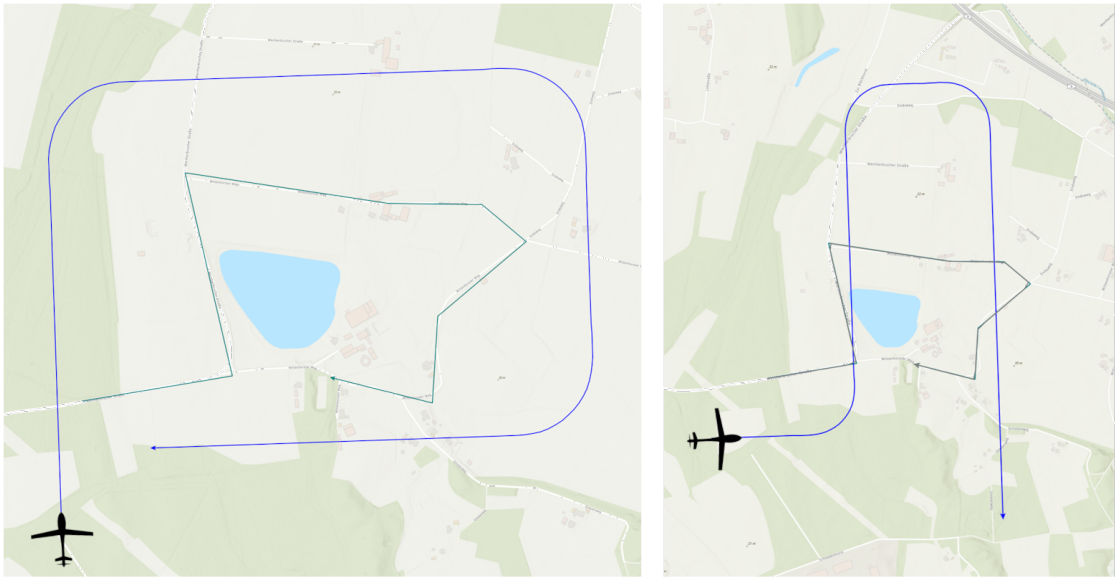


Abbildung 6.13: Sensormodellbestimmte Befliegungsplanung für das Routenaufklärungszenario mit DP-optimierter Referenztrajektorie (blaue Linie). Der Sensorfootprintpfad ist als dunkelgrüne Linie eingezeichnet und die global optimale Startposition ist durch das UAV-Symbol markiert. Links ist die Referenztrajektorie für das Sensorleistungsmodell YOLO-SLM und rechts für das Leistungsmodell CC-SLM dargestellt.

In der nachfolgenden Tabelle 6.4 sind die zu erwartenden Detektionsleistungswerte aus der Trajektorienoptimierung mit dynamischer Programmierung im Vergleich zum Mittelwert aus den Benchmark-Trajektorien zusammengefasst, wobei zwischen den beiden Sensorleistungsmodellen YOLO-SLM und CC-SLM unterschieden wird. Die theoretisch maximal erreichbare Detektionsleistung stellt die Obergrenze der erreichbaren Detektionsleistung dar. Darüber hinaus ist die zu erwartende Detektionsleistung als Absolut- und Relativwert angegeben. Die Benchmark-Trajektorien werden, analog zur NMPC-Trajektorienoptimierung, als Vergleichswert für die quantitative Evaluierung der Referenztrajektorien genutzt.

Aus der DP-Trajektorienoptimierung resultiert jeweils *eine* global optimale UAV-Referenztrajektorie für das Routenaufklärungszenario mit Sensorleistungsmodell YOLO-SLM und *eine* für das Leistungsmodell CC-SLM. Aus diesem Grund entfällt eine statistische Betrachtung der Detektionsleistung, die aus der UAV-Referenztrajektorie folgt.

Die zu erwartende Detektionsleistung der Benchmark-Trajektorie in Tabelle 6.4 ist ein Mittelwert, der aus den Detektionsleistungswerten von zwölf Benchmark-Trajektorien mit dem Sensorleistungsmodell YOLO-SLM bzw. zwölf Trajektorien mit dem Leistungs-

## 6.1 Evaluierung der Befliegungsplanung durch Beispielmissionen

Tabelle 6.4: Vergleich der Detektionsleistung mit DP-optimierter Referenztrajektorie und Benchmark-Trajektorien für das Routenaufklärungsszenario mit Sensorleistungsmodell YOLO-SLM und CC-SLM

	YOLO-SLM		CC-SLM	
	DP-Traj.	Benchm.	DP-Traj.	Benchm.
Theoretisch maximal erreichbare Detektionsleistung	0,936	0,936	0,970	0,970
Erwartete Detektionsleistung (abs.)	0,936	0,879	0,833	0,771
Erwartete Detektionsleistung (rel.)	100,00 %	93,92 %	85,88 %	79,52 %

modell CC-SLM ermittelt wird. Die  $12 + 12 = 24$  Benchmark-Trajektorien, die zu diesem Zweck genutzt werden, sind identisch mit den Benchmark-Trajektorien aus Tabelle 6.3 in Abschnitt 6.1.2. Um den Mittelwert der Detektionsleistung bestimmen zu können, wurden die UAV-Positionen entlang der Benchmark-Trajektorien mit dem Zeitschrittintervall  $\Delta t = 1,5$  s der zeit- und wertdiskreten Trajektorienoptimierung (vgl. Tabelle 4.4) diskretisiert. Durch die Diskretisierung konnte jeder UAV-Position eine zeitschrittspezifische Perception Map zugeordnet werden. Darüber hinaus wurde für jede Perception Map ein Gitter definiert, das kongruent mit dem Gitter aus der DP-Trajektorienoptimierung (vgl. Abb. 4.8) ist. Abschließend wurde die Detektionsleistung der zeitschrittabhängigen UAV-Position aus den Detektionsleistungswerten der räumlich nächsten Gitterpunkte interpoliert.

Aus den Benchmark-Trajektorien mit dem Leistungsmodell YOLO-SLM resultiert der Mittelwert der Detektionsleistung von  $\bar{x} = 93,92$  % (Relativwert) mit der Standardabweichung  $s = 1,35$  %. Demgegenüber kann durch die DP-optimierte Referenztrajektorie eine Detektionsleistung von  $100,00$  % erzielt werden, die den Mittelwert der Detektionsleistung aus den Benchmark-Trajektorien um  $6,08$  Prozentpunkte übersteigt und zudem identisch mit der theoretisch maximal erreichbaren Detektionsleistung ist.

Beim Routenaufklärungsszenario mit dem Sensorleistungsmodell CC-SLM erhält man für die Benchmark-Trajektorien einen Mittelwert der Detektionsleistung von  $\bar{x} = 79,52$  % (Relativwert) bei einer Standardabweichung von  $s = 2,35$  %. Der Relativwert der Detektionsleistung, der aus der DP-Trajektorienoptimierung resultiert, liegt bei  $85,88$  % und ist damit um  $6,36$  Prozentpunkte höher, als der Mittelwert der Detektionsleistung aus den Benchmark-Trajektorien.

Abschließend wird darauf hingewiesen, dass sich die theoretisch maximal erreichbare Detektionsleistung zwischen der DP-optimierten Trajektorie und der NMPC-optimierten Trajektorie geringfügig unterscheiden kann. Dies wird insbesondere bei der Nutzung

des Sensorleistungsmodells CC-SLM ersichtlich. Die Ursache kann auf die Verwendung unterschiedlicher Zeitschrittintervalle und die kontinuierliche bzw. diskrete (gitterbasierte) Trajektorienoptimierung zurückgeführt werden. Dadurch werden räumlich und/oder zeitlich lokal begrenzte Veränderungen der Detektionsleistung in den Perception Maps unterschiedlich genau erfasst.

### 6.1.3 Beispielmission Flächenaufklärungsszenario

Die Evaluierung der sensormodellbestimmten Befliegungsplanung für das Flächenaufklärungsszenario verläuft analog zum Routenaufklärungsszenario. Aus diesem Grund wird im Folgenden nur noch auf Kernaspekte eingegangen, um eine Wiederholung der Erläuterungen aus Abschnitt 6.1.2.1 zu vermeiden.

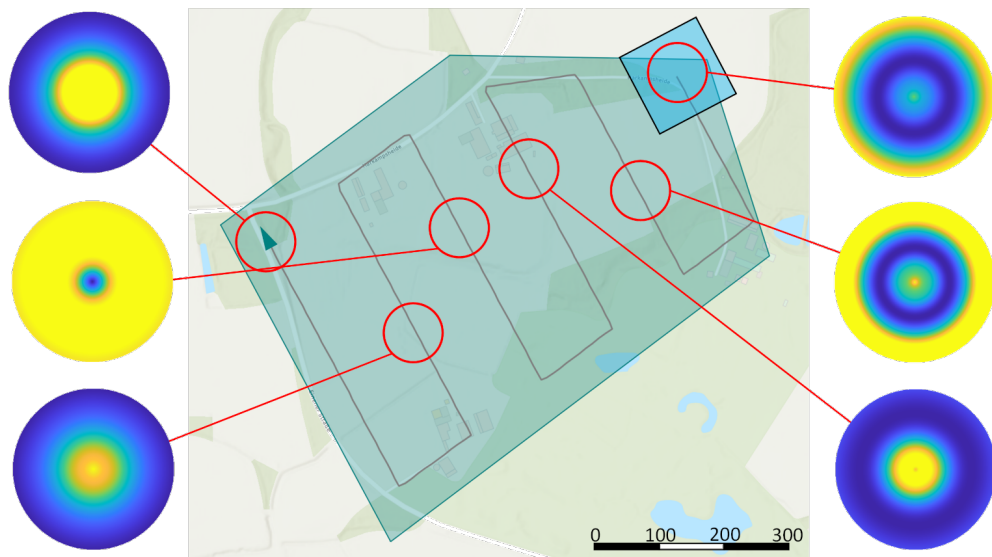


Abbildung 6.14: Darstellung der aufzuklärenden Fläche (grün) und der meanderförmige Verlauf des Sensorfootprintpfads (grüne Linie). Der erste Sensorfootprint ist als türkisfarbenes Quadrat eingezeichnet. Links und rechts sind exemplarisch verschiedene Perception Maps (nicht maßstabsgetreu) aufgetragen, die mit Hilfe des Sensorleistungsmodells CC-SLM generiert werden. Helle Bereiche repräsentieren eine hohe Detektionsleistung, wohingegen dunkle Bereiche mit einer niedrigen Detektionsleistung korrelieren.

In Abb. 6.14 ist das Flächenaufklärungsszenario abgebildet. Der aufzuklärende Bereich ist als grün eingefärbte Fläche dargestellt und wird durch ein Polygon begrenzt. Dieses Gebiet umfasst, ähnlich dem Routenaufklärungsszenario aus Abschnitt 6.1.2, sowohl

## 6.1 Evaluierung der Befliegungsplanung durch Beispielmissionen

ländliche als auch urbane Regionen. Mit Hilfe der Sensorpfadplanung aus Abschnitt 4.2 wurde der meanderförmige Verlauf des Sensorfootprintpfads (grüne Linie) geplant, dessen Länge ca. 3 km beträgt. Der Pfadverlauf stellt sicher, dass das Gebiet vollständig und deterministisch durch den Sensorfootprint (türkisfarbenes Quadrat) überstrichen und erfasst wird. Der meanderförmige Verlauf des Sensorfootprintpfads ist dadurch gekennzeichnet, dass der Pfad durch mehrere Richtungswechsel in abwechselnd lange und kurze gerade Abschnitte unterteilt ist. Die langen Pfadabschnitte verlaufen parallel und in entgegengesetzter Richtung. Die Herausforderung beim Flächenaufklärungsszenario besteht darin, die UAV-Flugtrajektorie optimal an die zahlreichen aufeinander folgenden Richtungswechsel des Sensorfootprintpfads anzupassen.

Der topographische Zustandsvektor, der den Inhalt des Sensorfootprints für das Flächenaufklärungsszenario repräsentiert, ist in Abb. 6.15 dargestellt. Die topographischen Zustände sind hierfür in Abhängigkeit des Verlaufs des Sensorfootprintpfads aufgetragen. Hierbei ist zu erkennen, dass ländliche und urbane Bereiche abwechselnd dominieren.

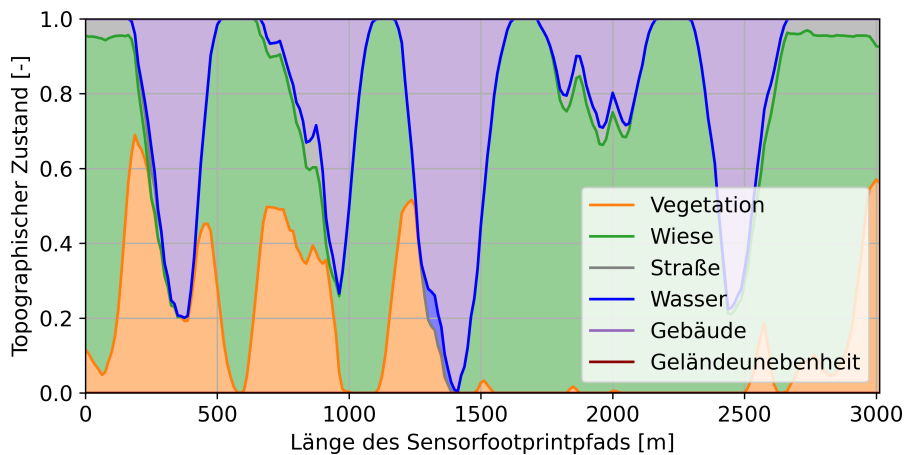


Abbildung 6.15: Verlauf der topographischen Zustände entlang des Sensorfootprintpfads in der aufzuklärenden Fläche

### 6.1.3.1 Befliegungsplanung mit NMPC-Trajektorienoptimierung

Das Ergebnis der sensormodellbestimmten Befliegungsplanung mit NMPC-optimierter Referenztrajektorie ist exemplarisch in Abb. 6.16 für zwei Einzelexperimente abgebildet. Im linken Bild ist die optimale Referenztrajektorie (blaue Linie) zur Benchmark-Trajektorie mit Racetrack-Muster (grüne Linie) für das Sensorleistungsmodell YOLO-SLM dargestellt. Demgegenüber ist im rechten Bild die Befliegungsplanung für die Benchmark-Trajektorie mit Kreismuster und dem Leistungsmodell CC-SLM zu sehen.

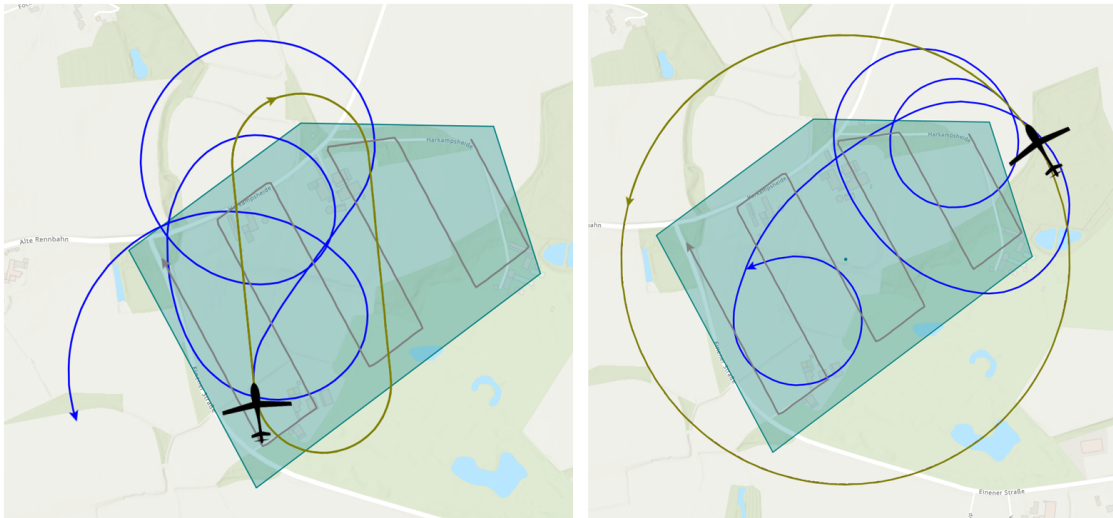


Abbildung 6.16: Trajektorienoptimierung für das Flächenaufklärungsszenario mit dem Sensorleistungsmodell YOLO-SLM (links) und dem Leistungsmodell CC-SLM (rechts). Die blaue Linie zeigt die NMPC-optimierte Referenztrajektorie und die hellgrüne Linie repräsentiert die Benchmark-Trajektorie als Rennbahnmuster (links) und als Kreismuster (rechts). Die Startpunkte der Trajektorien sind durch das UAV-Symbol gekennzeichnet.

In Tabelle 6.5 sind die Ergebnisse der zu erwartenden Detektionsleistungswerte aus den  $12 + 12 = 24$  Einzelexperimenten für das Flächenaufklärungsszenario zusammengefasst. In der ersten Spalte sind die einzelnen Datensätze für das Sensorleistungsmodell YOLO-SLM (grau hinterlegt) und CC-SLM (blau hinterlegt) angegeben. Die theoretisch maximal erreichbare Detektionsleistung (Max. DL) ist in der zweiten Spalte hinterlegt. Dem folgen die Werte der zu erwartenden Detektionsleistung für die NMPC-optimierten Trajektorien sowie für die Benchmark-Trajektorien. In der letzten Spalte ist die Differenz der Detektionsleistungswerte aufgetragen. Neben den Absolutwerten (abs.) sind die Relativwerte (rel.), der arithmetische Mittelwert  $\bar{x}$  und die Standardabweichung  $s$  in Prozentpunkten angegeben.

### Ergebnisdarstellung für das Sensorleistungsmodell YOLO-SLM

Die Simulationsergebnisse aus Tabelle 6.5 sind in den folgenden Abbildungen grafisch aufbereitet. In Abb. 6.17 sind die Detektionsleistungswerte für die paarweise verbundenen Stichproben der NMPC-optimierten Referenztrajektorien (blau) und der Benchmark-Trajektorien (orange) zusammen aufgetragen. Im linken Diagramm sind die Ergebnisse als Absolutwerte und rechts als Relativwerte dargestellt. Die Detektionsleistungswerte, die

## 6.1 Evaluierung der Befliegungsplanung durch Beispielmissionen

Tabelle 6.5: Ergebnisdarstellung der Detektionsleistung (DL) für das Flächenaufklärungsszenario mit Sensorleistungsmodell YOLO-SLM (grau hinterlegter Datensatz) und CC-SLM (blau hinterlegter Datensatz)

Datensatz	Max. DL		DL aus NMPC-Optimierung		DL aus Benchmark-Trajektorie		DL Differenz	
	(abs.)	(rel.)	(abs.)	(rel.)	(abs.)	(rel.)	(abs.)	(rel.)
area-yolo-circle-1		0,914	0,864	92,31 %	0,050	5,34 %P		
area-yolo-circle-2		0,922	0,892	95,30 %	0,030	3,21 %P		
area-yolo-circle-3		0,921	0,869	92,84 %	0,052	5,56 %P		
area-yolo-circle-4		0,922	0,897	95,83 %	0,025	2,67 %P		
area-yolo-racetr-1		0,933	0,870	92,95 %	0,063	6,73 %P		
area-yolo-racetr-2	0,936	0,930	0,896	95,73 %	0,034	3,63 %P	4,04 %P	1,21 %P
area-yolo-racetr-3		0,924	0,896	95,73 %	0,028	2,99 %P		
area-yolo-racetr-4		0,925	0,900	96,15 %	0,025	2,67 %P		
area-yolo-fig8-1		0,932	0,901	96,26 %	0,031	3,31 %P		
area-yolo-fig8-2		0,925	0,888	94,87 %	0,037	3,95 %P		
area-yolo-fig8-3		0,931	0,892	95,30 %	0,039	4,17 %P		
area-yolo-fig8-4		0,936	0,896	95,73 %	0,040	4,27 %P		
area-cc-circle-1		0,874	0,794	83,76 %	0,080	8,44 %P		
area-cc-circle-2		0,855	0,817	86,18 %	0,038	4,01 %P		
area-cc-circle-3		0,858	0,777	81,96 %	0,081	8,54 %P		
area-cc-circle-4		0,857	0,800	84,39 %	0,057	6,01 %P		
area-cc-racetr-1		0,865	0,845	89,14 %	0,020	2,11 %P		
area-cc-racetr-2	0,948	0,866	0,806	85,02 %	0,060	6,33 %P	4,96 %P	1,96 %P
area-cc-racetr-3		0,851	0,827	87,24 %	0,024	2,53 %P		
area-cc-racetr-4		0,859	0,822	86,71 %	0,037	3,90 %P		
area-cc-fig8-1		0,852	0,817	86,18 %	0,035	3,69 %P		
area-cc-fig8-2		0,867	0,823	86,81 %	0,044	4,64 %P		
area-cc-fig8-3		0,852	0,812	85,65 %	0,040	4,22 %P		
area-cc-fig8-4		0,859	0,811	85,55 %	0,048	5,06 %P		



## 6 Evaluierung und Ergebnisse

aus den Referenztrajektorien resultieren, liegen zwischen  $p = 97,65\%$  und  $p = 100,00\%$ , wohingegen der Maximalwert der Detektionsleistung aus den Benchmark-Trajektorien bei  $p = 96,26\%$  liegt. Damit wird der bereits hohe Maximalwert aus den Benchmark-Trajektorien durch die Detektionsleistungswerte der Befliegungsplanung übertroffen und noch weiter gesteigert.

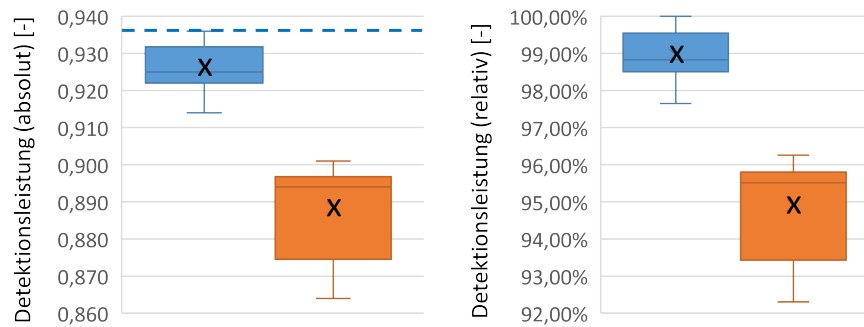


Abbildung 6.17: Ergebnisdarstellung des Flächenaufklärungsszenarios mit dem Sensorleistungsmodell YOLO-SLM. Im linken Diagramm sind die Absolutwerte der Detektionsleistung für die NMPC-optimierten Referenztrajektorien (blau) und die Benchmark-Trajektorien (orange) aufgetragen. Die Relativwerte sind im rechten Diagramm zu sehen. Die blau gestrichelte Linie markiert die theoretisch maximal erreichbare Detektionsleistung.

Dieser Sachverhalt wird in Abb. 6.18 verdeutlicht. Darin ist die paarweise Differenz der zu erwartenden Detektionsleistung  $D_i$  (siehe Gleichung (5.1)) aus den zwölf Einzelexperimenten und dem Sensorleistungsmodell YOLO-SLM als Boxplot aufgetragen. Die relative Steigerung der Detektionsleistung liegt zwischen  $p = 2,67\%$  und  $p = 6,73\%$  bei einem Mittelwert von  $\bar{x} = 4,04$  Prozentpunkten.

Im linken Diagramm von Abb. 6.18 ist die Differenz der Detektionsleistung in Absolutwerten und rechts die dazugehörigen Relativwerte dargestellt. Die Abbildung verdeutlicht, dass alle Werte für die Differenz der Detektionsleistung größer als Null sind. Daraus folgt, dass für jedes Stichprobenpaar eine Steigerung der Detektionsleistung durch die Befliegungsplanung erreicht wurde.

Um die Differenz der zu erwartenden Detektionsleistung aus der sensormodellbestimmten Befliegungsplanung und den Benchmark-Trajektorien für das Flächenaufklärungsszenario mit dem Sensorleistungsmodell YOLO-SLM methodisch bewerten zu können, wird wieder der t-Test aus Abschnitt 5.3 durchgeführt. Vor dem t-Test wurde ebenfalls mit Hilfe des Shapiro-Wilk-Tests [194] überprüft und nachgewiesen, dass die Differenzwerte normalverteilt sind. Das Ergebnis aus dem t-Test stellt sich wie folgt dar:



## 6.1 Evaluierung der Befliegungsplanung durch Beispielmissionen

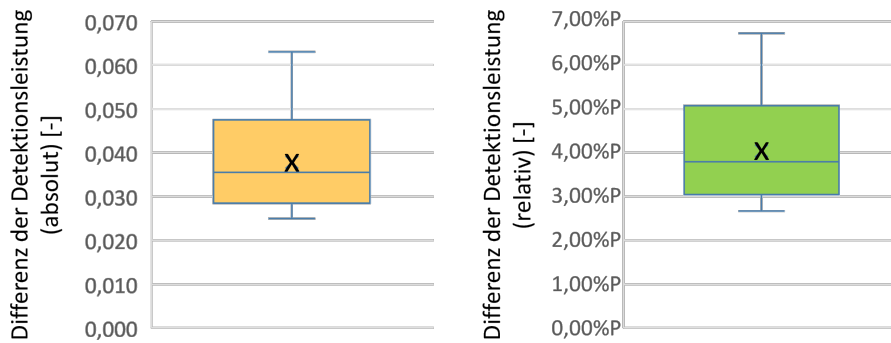


Abbildung 6.18: Paarweise Differenz der Detektionsleistungswerte aus den Einzelerperimenten für das Flächenauflärungsszenario mit Sensorleistungsmodell YOLO-SLM. Absolutwerte sind links und Relativwerte rechts dargestellt.

Die Steigerung der durchschnittlichen Detektionsleistung um  $\bar{x} = 4,04$  Prozentpunkte durch die sensormodellbestimmte Befliegungsplanung mit nichtlinearer modellprädiktiver Regelung und dem Sensorleistungsmodell YOLO-SLM ist *signifikant* ( $p_{val} = 1,31 \cdot 10^{-7} < \alpha = 0,01$ ) gegenüber der durchschnittlichen Detektionsleistung, die aus den Benchmark-Trajektorien resultiert. Die Nullhypothese  $H_0$  aus Abschnitt 5.3 muss daher verworfen werden und die Alternativhypothese  $H_1$  wird angenommen.

### Ergebnisdarstellung für das Sensorleistungsmodell CC-SLM

Nach der Evaluierung der sensormodellbestimmten Befliegungsplanung mit NMPC-optimierter Referenztrajektorie und Sensorleistungsmodell YOLO-SLM erfolgt abschließend die Untersuchung für das Leistungsmodell CC-SLM. In Abb. 6.14 wurde bereits der Verlauf des Sensorfootprintpfads und der daraus resultierenden Perception Maps dargestellt. Ähnlich zum Routenaufklärungsszenario ist auch hier wieder eine hohe Variation bezüglich des Verlaufs der Detektionsleistung in den einzelnen Perception Maps zu erkennen. Dies stellt eine Herausforderung für die Generierung einer optimalen Trajektorie dar, die durch die Bereiche mit hoher Detektionsleistung verlaufen soll.

Auf Grundlage der Detektionsergebnisse aus Tabelle 6.5 wurden die nachfolgenden Boxplots zum Flächenauflärungsszenario mit NMPC-optimierter Trajektorie und Sensorleistungsmodell CC-SLM erstellt. In Abb. 6.19 sind die Werte der Detektionsleistung aufgetragen, die aus den NMPC-optimierten Referenztrajektorien (blau hinterlegt) und den Benchmark-Trajektorien (orange hinterlegt) stammen. Die Relativwerte der Detektionsleistung, die aus den Referenztrajektorien resultieren, liegen zwischen  $p = 89,77\%$  und

## 6 Evaluierung und Ergebnisse

$p = 92,19\%$  bei einem arithmetischen Mittelwert von  $\bar{x} = 90,67\%$ . Demgegenüber liegt der Mittelwert der Detektionsleistung aus den Benchmark-Trajektorien bei  $\bar{x} = 85,72\%$ .

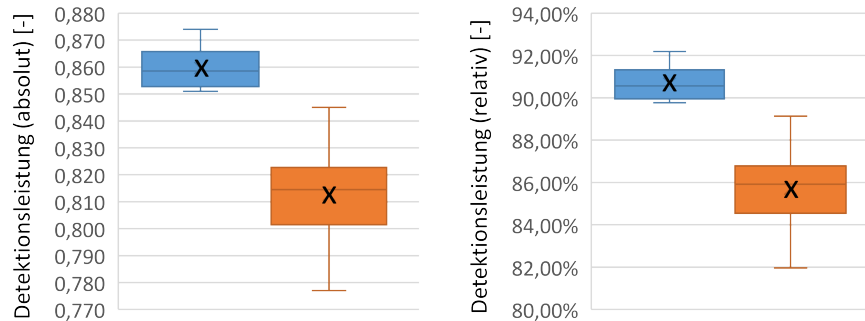


Abbildung 6.19: Ergebnisdarstellung des Flächenauflärungsszenarios mit Sensorleistungsmodell CC-SLM. Die Detektionsleistungswerte aus der NMPC-optimierten UAV-Referenztrajektorie sind blau und die Werte aus den Benchmark-Trajektorien orange hinterlegt. Die Absolutwerte sind links, die Relativwerte rechts abgebildet.

Aus Abb. 6.20 geht hervor, dass für jede paarweise verbundene Stichprobe eine Steigerung der Detektionsleistung durch die UAV-Referenztrajektorie gegenüber der Benchmark-Trajektorie erreicht wird. Die Steigerung liegt dabei zwischen  $p = 2,11\%P$  und  $p = 8,54\%P$  bei einem arithmetischen Mittelwert von  $\bar{x} = 4,96$  Prozentpunkten.

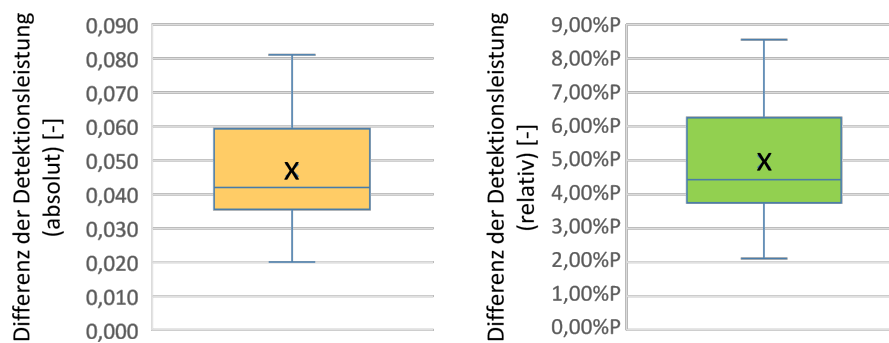


Abbildung 6.20: Paarweise Differenz der Detektionsleistungswerte aus den Einzelexperimenten für das Flächenauflärungsszenario mit dem Sensorleistungsmodell CC-SLM. Absolutwerte sind links und Relativwerte rechts dargestellt.

Die methodische Bewertung der Differenz der zu erwartenden Detektionsleistung aus den NMPC-optimierten Referenztrajektorien und den Benchmark-Trajektorien für das Flächenauflärungsszenario mit Sensorleistungsmodell CC-SLM erfolgt analog zur

## 6.1 Evaluierung der Befliegungsplanung durch Beispielmissionen

Bewertung des Leistungsmodells YOLO-SLM mit Hilfe des t-Tests. Hierfür wurde ebenfalls vorab durch den Shapiro-Wilk-Test nachgewiesen, dass die Differenzwerte normalverteilt sind. Das Ergebnis des t-Test lautet wie folgt:

Die Steigerung der durchschnittlichen Detektionsleistung um  $\bar{x} = 4,96$  Prozentpunkte durch die sensormodellbestimmte Befliegungsplanung mit nichtlinearer modellprädiktiver Regelung und dem Sensorleistungsmodell CC-SLM ist *signifikant* ( $p_{val} = 2,10 \cdot 10^{-6} < \alpha = 0,01$ ) gegenüber der durchschnittlichen Detektionsleistung, die aus den Benchmark-Trajektorien resultiert. Die Nullhypothese  $H_0$  aus Abschnitt 5.3 wird daher verworfen und die Alternativhypothese  $H_1$  wird angenommen.

### 6.1.3.2 Befliegungsplanung mit DP-Trajektorienoptimierung

In diesem Abschnitt wird die sensormodellbestimmte Befliegungsplanung mit DP-optimierter Referenztrajektorie anhand des Flächenaufklärungsszenario evaluiert und dem *Untersuchungsgegenstand 1.2* zugeordnet. Die nachfolgende Untersuchung verläuft analog zu Abschnitt 6.1.2.2, in dem bereits die Grundlagen der Evaluierung erläutert wurden. In Abb. 6.21 ist die DP-optimierte UAV-Referenztrajektorie (blaue Linie) für das Flächenaufklärungsszenario mit dem Sensorleistungsmodell YOLO-SLM (links) und dem Leistungsmodell CC-SLM (rechts) dargestellt. Die Startposition der Trajektorie wird durch das UAV-Symbol markiert.



Abbildung 6.21: Sensormodellbestimmte Befliegungsplanung für das Flächenaufklärungsszenario mit DP-optimierter Referenztrajektorie (blaue Linie). Der Sensorfootprintpfad ist als dunkelgrüne Linie eingezeichnet und die global optimale Startposition ist durch das UAV-Symbol markiert. Links ist die Referenztrajektorie für das Sensorleistungsmodell YOLO-SLM und rechts für das Leistungsmodell CC-SLM dargestellt.

In Abb. 6.22 ist exemplarisch die zeitschrittspezifische Detektionsleistung der DP-optimierten Referenztrajektorie als blaue Linie, und die theoretisch maximal erreichbare Detektionsleistung als schwarze Linie in Abhängigkeit des Sensorfootprintpfads dargestellt.

## 6 Evaluierung und Ergebnisse

Die Abbildung soll die hohe Variation der Detektionsleistung entlang des Sensorfootprintpfads verdeutlichen. Mit der hohen Variation ist gleichzeitig die Herausforderung verbunden, eine global optimale Referenztrajektorie zu ermitteln, aus der eine hohe Detektionsleistung resultiert, unter Einhaltung perzeptiver und sensorplattformspezifischer Randbedingungen.

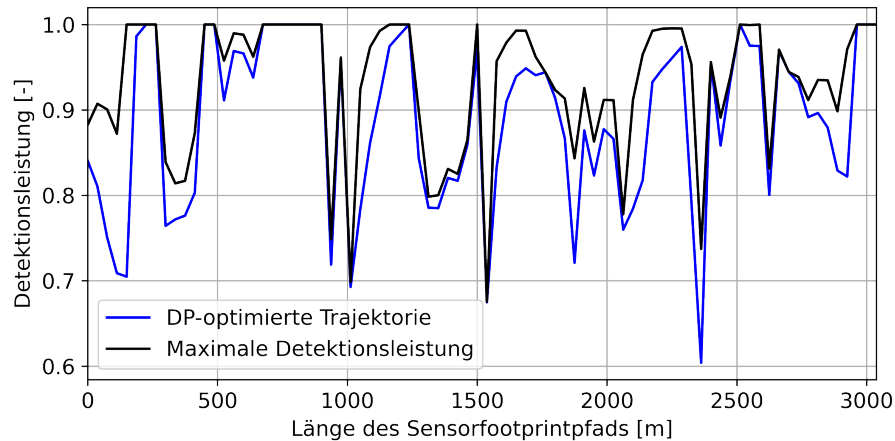


Abbildung 6.22: Zeitschrittspezifische Detektionsleistung der DP-optimierten Referenztrajektorie (blaue Linie) und der theoretisch maximal erreichbaren Detektionsleistung (schwarze Linie) für das Flächenauflärungsszenario mit Sensorleistungsmodell CC-SLM

In Tabelle 6.6 ist die zu erwartende Detektionsleistung für die Trajektorienoptimierung mit dynamischer Programmierung im Vergleich zu den Benchmark-Trajektorien zusammengefasst, wobei zwischen den beiden Sensorleistungsmodellen YOLO-SLM und CC-SLM unterschieden wird.

Tabelle 6.6: Vergleich der Detektionsleistung mit DP-optimierter Referenztrajektorie und Benchmark-Trajektorien für das Flächenauflärungsszenario mit Sensorleistungsmodell YOLO-SLM und CC-SLM

	YOLO-SLM		CC-SLM	
	DP-Traj.	Benchm.	DP-Traj.	Benchm.
Theoretisch maximal erreichbare Detektionsleistung	0,936	0,936	0,944	0,944
Erwartete Detektionsleistung (abs.)	0,933	0,888	0,888	0,813
Erwartete Detektionsleistung (rel.)	99,68 %	94,88 %	94,07 %	86,13 %

## 6.1 Evaluierung der Befliegungsplanung durch Beispielmissionen

Für das Sensorleistungsmodell YOLO-SLM resultiert aus den Benchmark-Trajektorien ein Mittelwert für die Detektionsleistung von  $\bar{x} = 94,88\%$  (Relativwert) mit der Standardabweichung von  $s = 1,33\%$ . Demgegenüber kann durch die DP-optimierte Referenztrajektorie eine Detektionsleistung von  $99,68\%$  erzielt werden, die den Mittelwert der Detektionsleistung aus den Benchmark-Trajektorien um  $4,80$  Prozentpunkte übersteigt. Zudem wird die theoretisch maximal erreichbare Detektionsleistung annähernd erzielt.

Beim Flächenaufklärungsszenario mit dem Sensorleistungsmodell CC-SLM wird für die Benchmark-Trajektorien ein Mittelwert der Detektionsleistung von  $\bar{x} = 86,13\%$  (Relativwert) prognostiziert, bei einer Standardabweichung von  $s = 1,72\%$ . Der Relativwert der Detektionsleistung, der aus der DP-Trajektorienoptimierung resultiert, liegt bei  $94,07\%$  und ist damit  $7,94$  Prozentpunkte höher, als der Mittelwert der Detektionsleistung aus den Benchmark-Trajektorien.

### **Abweichung zwischen Dubins-Pfadsegmenten und geglätteter Trajektorie**

Um den Einfluss der Trajektorienglättung (vgl. Abschnitt 4.4.3) auf die zu erwartende Detektionsleistung quantifizieren zu können, wird im Folgenden die geometrische Abweichung zwischen der Sollwerttrajektorie, die aus verketteten Dubins-Pfadsegmenten zusammengesetzt ist, und der geglätteten Referenztrajektorie untersucht. Die maximale Abweichung  $\Delta r$  zwischen der Sollwerttrajektorie und der Referenztrajektorie wird auf Basis der euklidischen Distanz in der horizontalen Ebene berechnet.

Die Untersuchung der maximalen Abweichung für die DP-optimierten Referenztrajektorien in den vier Einzelexperimenten (Routen- und Flächenaufklärungsszenario jeweils mit CC-SLM und YOLO-SLM) hat ergeben, dass die Abweichung zwischen der Sollwert- und Referenztrajektorie einen maximalen Wert von  $\Delta r = 1,72$  m nicht überschreitet. Die geglättete Referenztrajektorie bildet die Sollwerttrajektorie damit sehr genau ab.

Des Weiteren wurde untersucht, ob die Abweichung der Trajektorien zu einem Unterschied der zu erwartenden Detektionsleistung führt: Sowohl für das Routen- als auch für das Flächenaufklärungsszenario mit YOLO-SLM konnte kein Unterschied bei der Detektionsleistung festgestellt werden. Dies wird damit begründet, dass die Sollwerttrajektorie und die Referenztrajektorie durch dieselben Bereiche der abschnittsweise konstanten Perception Maps verlaufen. Beim Routen- und Flächenaufklärungsszenario mit CC-SLM resultiert aus der Sollwert- und der Referenztrajektorie ein Unterschied in der Detektionsleistung von maximal  $0,0021$  Prozentpunkten. Aus diesem Grund wird die Abweichung nicht weiter betrachtet.

### 6.1.4 Zusammenfassung

In diesem Abschnitt wurde für die sensormodellbestimmte Befliegungsplanung mit NMPC- bzw. DP-optimierter Referenztrajektorie und den Sensorleistungsmodellen YOLO-SLM und CC-SLM nachgewiesen, dass mit diesem Ansatz die durchschnittlich zu erwartende Detektionsleistung gesteigert werden kann, im Vergleich zur Detektionsleistung, die aus Benchmark-Trajektorien resultiert. Darüber hinaus können mit DP-optimierten Referenztrajektorien nochmals höhere Detektionsleistungswerte erzielt werden, im Vergleich zu den Durchschnittswerten, die sich aus den NMPC-optimierten Trajektorien ergeben.

Für das Routenaufklärungsszenario konnte mit den NMPC-optimierten Referenztrajektorien eine Detektionsleistung erreicht werden, die die Leistungswerte der Benchmark-Trajektorien durchschnittlich um 4,91 Prozentpunkte (YOLO-SLM) bzw. 4,82 Prozentpunkte (CC-SLM) übersteigen. Für das Flächenaufklärungsszenario wurde die durchschnittliche Detektionsleistung der Benchmark-Trajektorien um 4,04 Prozentpunkte (YOLO-SLM) bzw. 4,96 Prozentpunkte (CC-SLM) übertroffen.

Für die DP-optimierten Referenztrajektorien ergab sich eine Steigerung der zu erwartenden Detektionsleistung für das Routenaufklärungsszenario von 6,08 Prozentpunkte (YOLO-SLM) bzw. 6,36 Prozentpunkte (CC-SLM) gegenüber der durchschnittlichen Detektionsleistung der Benchmark-Trajektorien. Für das Flächenaufklärungsszenario lag die Steigerung der Detektionsleistung gegenüber der durchschnittlichen Detektionsleistung der Benchmark-Trajektorien bei 4,80 Prozentpunkte (YOLO-SLM) bzw. 7,94 Prozentpunkte (CC-SLM).

Damit wurden die *Untersuchungsgegenstände 1.1* und *1.2* aus Abschnitt 5.1 sowie die korrelierende *Forschungsfrage 5.1* (Welches Potenzial bietet die sensormodellbestimmte Befliegungsplanung zur Steigerung der Detektionsleistung mit zeitkontinuierlicher Trajektorienoptimierung sowie mit zeit- und wertdiskreter Trajektorienoptimierung?) erfolgreich abgearbeitet und beantwortet. Nachfolgend ist der aktuelle Stand der Untersuchung abgebildet.

Zwischenziel	1	2	3	4	5
Forschungsfrage	1.1	2.1 & 2.2	3.1 & 3.2	—	5.1 - 5.3
Bearbeitung in Kapitel	3	3	3 & 4	4	6

## 6.2 Untersuchung von Anforderungen und Randbedingungen bei der Befliegungsplanung

Nachfolgend werden die *Untersuchungsgegenstände 2.1* und *2.2* aus Abschnitt 5.1 bearbeitet. Es wird untersucht, ob im Rahmen der sensormodellbestimmten Befliegungsplanung perzeptive und sensorplattformenspezifische Anforderungen und Randbedingungen umgesetzt und eingehalten werden. Nach der Bearbeitung der beiden Untersuchungsgegenstände kann die *Forschungsfrage 5.2* aus Abschnitt 2.4 abgeschlossen werden.

### 6.2.1 Untersuchung perzeptiver Anforderungen

Der *Untersuchungsgegenstand 2.1* hat zum Ziel, die Sensorsteuerung der sensormodellbestimmten Befliegungsplanung hinsichtlich der Umsetzung der perzeptiven Anforderungen und Randbedingungen zu untersuchen. Die Anforderung an die Sensorsteuerung besteht darin, den Bildwinkel des bildgebenden Sensors dynamisch und in einer Art anzupassen, sodass ein vorgegebener GSD-Sollwert für jeden diskreten Zeitschritt entlang des Sensorfootprintpfads eingehalten wird. Die Randbedingungen werden durch das verwendete Sensorsystem der Perzeptionskette vorgegeben, insbesondere durch die technischen Grenzwerte des Sensorbildwinkels.

In Abb. 6.23 ist exemplarisch der Verlauf des Bildwinkels und die damit korrelierende GSD in Abhängigkeit von der Länge des Sensorfootprintpfads dargestellt. Der abgebildete Verlauf resultiert aus der DP-optimierten Referenztrajektorie für das Flächenaufklärungszenario mit dem Sensorleistungsmodell CC-SLM. Der Verlauf der Trajektorie ist im rechten Bild von Abb. 6.21 zu sehen.

Im oberen Teil von Abb. 6.23 ist der dynamische Verlauf des Bildwinkels (schwarze Linie) über die Länge des Sensorfootprintpfads aufgetragen. Die technischen Grenzwerte des Bildwinkels (rote Linien) werden durch den minimalen und den maximalen Bildwinkel  $\beta_{min}$  bzw.  $\beta_{max}$  definiert und stammen aus Tabelle 3.4. Im unteren Teil der Abbildung ist der Verlauf der GSD (schwarze Linie) und die GSD-Sollwertvorgabe  $gsd_{ref} = 0,07$  m (blau gepunktete Linie) aufgetragen, deren Werte ebenfalls in Tabelle 3.4 zu finden sind.

Der geometrische Zusammenhang zwischen der GSD und dem Bildwinkel wurde bereits in Gleichung (3.5) beschrieben und ist u. a. von der Länge der Sensorsichtlinie  $s_{LoS}$  (Abstand zwischen dem Sensor am UAV und dem Sensorfootprint am Boden) abhängig. Wird die Länge der Sichtlinie immer weiter vergrößert oder verkleinert, wird ein Punkt erreicht, an dem der Bildwinkel nicht weiter adaptiert und damit der vorgegebene GSD-Sollwert nicht weiter eingehalten werden kann. Dies wird in Abb. 6.23 ersichtlich. Dabei

## 6 Evaluierung und Ergebnisse

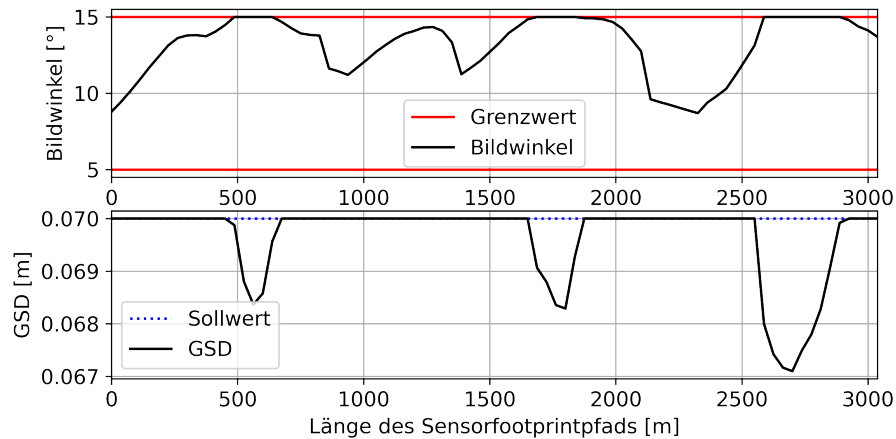


Abbildung 6.23: Dynamische Adaption des Bildwinkels um die GSD-Sollwertvorgabe zu erfüllen. Erreicht der Bildwinkel den Grenzwert, hat dies eine Veränderung der GSD zur Folge.

erreicht der Bildwinkel dreimal den Grenzwert und infolgedessen weicht die GSD von ihrem Sollwert ab. Davon ausgenommen ist die Sensorsteuerung innerhalb der Grenzwerte des Bildwinkels zu jeder Zeit in der Lage, dem GSD-Sollwert exakt zu folgen. Dieses Verhalten der Sensorsteuerung konnte in allen Einzelexperimenten beobachtet werden.

### 6.2.2 Untersuchung sensorplattformspezifischer Randbedingungen

Die Einhaltung sensorplattformspezifischer Randbedingungen (*Untersuchungsgegenstand 2.2*) wird exemplarisch an der NMPC-optimierten Referenztrajektorie für das Flächenaufklärungsszenario mit dem Sensorleistungsmodell YOLO-SLM evaluiert. In Abb. 6.24 sind die Steuereingaben und Systemzustände für den zu untersuchenden Datensatz „area-yolocratr-4“ aus Tabelle 6.5 abgebildet. Der Verlauf der zugehörigen Referenztrajektorie ist im linken Bild von Abb. 6.16 zu sehen.

In Gleichung (3.10) wurden die Steuereingaben  $u(t)$  des UAVs (die *Rollrate*  $\omega(t)$  und die *Beschleunigung* tangential zur Flugrichtung  $a(t)$ ) festgelegt. Darüber hinaus wurden in Abschnitt 3.5.2.1 Steuerbeschränkungen definiert, deren Zahlenwerte in Tabelle 3.5 zu finden sind. Mit der dynamischen Anpassung der *Rollrate* soll das UAV der Sollwertvorgabe in Querrichtung und mit der Adaption der *Beschleunigung* in Flugrichtung soll das UAV der Sollwertvorgabe in Längsrichtung folgen.

In Abb. 6.24 ist zu erkennen, dass die Steuereingaben die festgelegten Grenzwerte (rote Linien) temporär erreichen, diese aber nicht überschreiten. Dies ist insbesondere für die Steuereingabe *Beschleunigung* im Zeitintervall zwischen 50 s und 60 s ersichtlich. Die



## 6.2 Untersuchung von Anforderungen und Randbedingungen bei der Befliegungsplanung

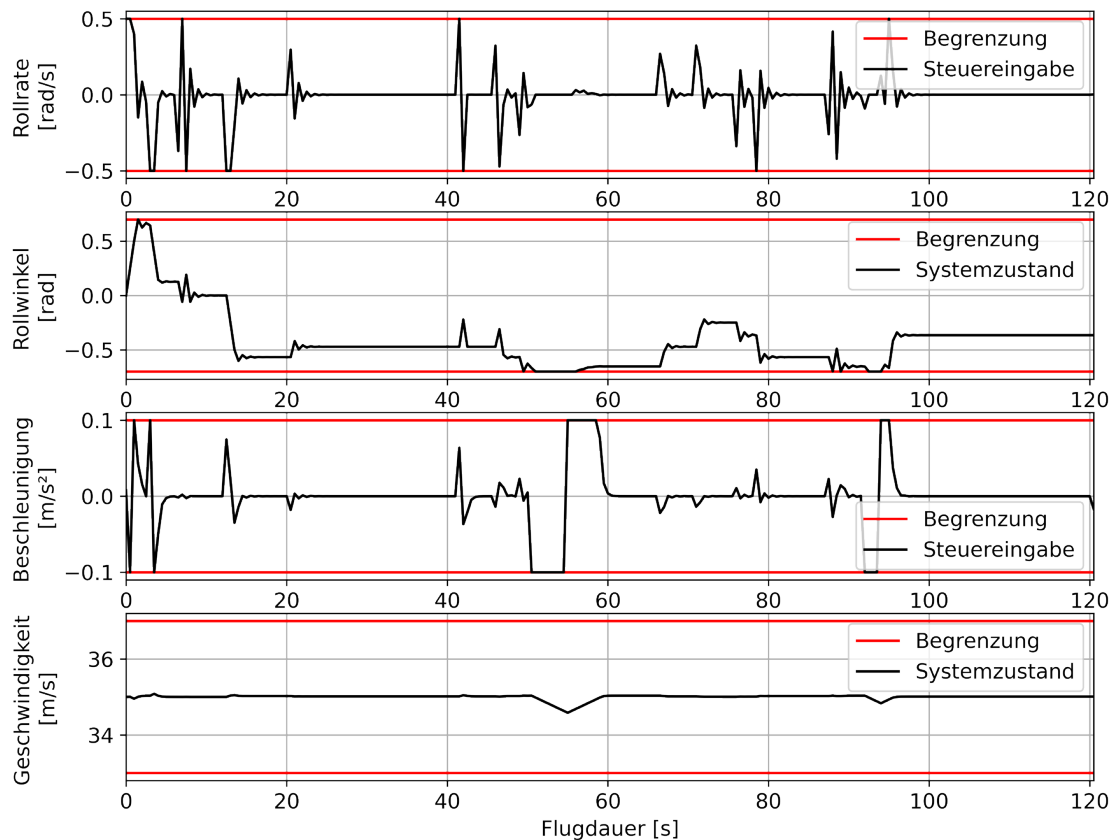


Abbildung 6.24: Darstellung der Steuereingaben *Rollrate* und *Beschleunigung* tangential zur Flugrichtung für die NMPC-optimierte Trajektorie mit Sensorleistungsmodell YOLO-SLM. Die Steuereingaben führen zu Änderungen der Systemzustände *Geschwindigkeit* und *Rollwinkel* des UAVs in Abhängigkeit der Flugdauer. Die roten Linien markieren die festgelegten Grenzwerte.

Ursache für die Spitzen und Fluktuationen in den Steuereingaben kann im Wesentlichen auf die Parametrisierung der Gewichtungsmatrizen  $Q$  und  $R$  aus Gleichung (3.51) zurückgeführt werden. Deren Diagonaleinträge wurden so gewählt, dass das UAV möglichst genau der Sollwertvorgabe für die nördliche  $n^{ref}(t)$  und östliche  $e^{ref}(t)$  Position folgt. Daraus resultieren Steuereingaben, die eine hohe Dynamik und hohe Amplituden aufweisen. Für den Fall, dass die Steuereingaben gedämpft werden sollen, kann dies durch eine Anpassung der Gewichtungsmatrizen erreicht werden.

Die Steuereingaben bewirken u. a. eine Änderung der beiden UAV-Systemzustände: *Rollwinkel*  $\phi(t)$  und *Geschwindigkeit*  $v(t)$  in Längsrichtung. Für diese Systemzustände wurden ebenfalls Systemzustandsbeschränkungen definiert, die sicherstellen sollen, dass

das UAV nur innerhalb der zulässigen flugdynamischen und technischen Grenzen betrieben wird. In Abb. 6.24 sind diese Beschränkungen der beiden Systemzustände als rote Linien eingezeichnet. Die dazugehörigen Zahlenwerte können der Tabelle 3.5 entnommen werden. Die Adaption der Fluggeschwindigkeit hat zum Ziel, die Sollwertvorgabe für die zeitschrittspezifische Position des UAVs in Längsrichtung bestmöglich zu erfüllen. Daher wird das UAV zeitweise verzögert bzw. beschleunigt. Abschließend kann festgehalten werden: auf Grund der modellprädiktiven Regelung werden zu jeder Zeit die vorgegebenen Steuerungs- und Zustandsbeschränkungen eingehalten und das UAV operiert demzufolge durchweg innerhalb der festgelegten Grenzwerte.

Die Untersuchung auf Einhaltung der sensorplattformenspezifischen Randbedingungen wird in diesem Abschnitt am Beispiel einer NMPC-optimierten Referenztrajektorie, nicht aber für eine DP-optimierte Trajektorie, durchgeführt. Dies wird als hinreichend betrachtet, da die DP-optimierte Trajektorie aus Dubins-Pfaden zusammengesetzt ist, die in einem nachgeschalteten Arbeitsschritt geglättet werden (siehe hierzu Abschnitt 4.4.3). Für die Glättung der Trajektorie wird ebenfalls die NMPC-Trajektorienoptimierung verwendet. Die daraus resultierende Referenztrajektorie erfüllt somit alle Anforderungen und Randbedingungen einer NMPC-optimierten Referenztrajektorie.

### 6.2.3 Zusammenfassung

In den vorangegangenen Abschnitten wurde untersucht, ob ausgewählte perzeptive Anforderungen, wie z. B. die Einhaltung der Sollwertvorgabe für die Bodenpixelauflösung (GSD), durch die Sensorsteuerung der sensormodellbestimmten Befliegungsplanung umgesetzt werden. Dies wurde erfüllt, solange der Bildwinkel innerhalb der technischen Grenzen variiert wurde. Des Weiteren wurde überprüft und nachgewiesen, dass ausgewählte sensorplattformenspezifische Randbedingungen eingehalten werden. Hierfür wurden die Rollrate, der Rollwinkel, die Beschleunigung in Flugrichtung und die Geschwindigkeit des UAVs untersucht.

Damit wurden die beiden *Untersuchungsgegenstände 2.1* und *2.2* aus Abschnitt 5.1 sowie die damit verbundene *Forschungsfrage 5.2* (Wie werden perzeptive und sensorplattformenspezifische Anforderungen und Randbedingungen bei der sensormodellbestimmten Befliegungsplanung umgesetzt und eingehalten?) erfolgreich abgearbeitet und beantwortet. Nachfolgend ist der aktuelle Stand der Untersuchung abgebildet:

Zwischenziel	1	2	3	4	5
Forschungsfrage	1.1	2.1 & 2.2	3.1 & 3.2	—	5.1 - 5.3
Bearbeitung in Kapitel	3	3	3 & 4	4	6

## 6.3 Evaluierung der Aufklärungs- und Berechnungsdauer

In diesem Abschnitt werden die beiden letzten *Untersuchungsgegenstände* 3.1 und 3.2 aus Abschnitt 5.1 bearbeitet. Zu diesem Zweck werden die Aufklärungs- und Berechnungsdauer zur Planung und Durchführung der sensormodellbestimmten Befliegungsplanung für die NMPC- und DP-Trajektorienoptimierung evaluiert. Durch die Bearbeitung der beiden Untersuchungsgegenstände soll die *Forschungsfrage* 5.3 aus Abschnitt 2.4 beantwortet und abgeschlossen werden.

### 6.3.1 Evaluierung der Aufklärungsdauer

Nachfolgend wird auf den *Untersuchungsgegenstand* 3.1 zur Evaluierung der Aufklärungsdauer eingegangen. Die Aufklärungsdauer beschreibt die Zeitdauer, die das UAV für das Abfliegen der Referenztrajektorie benötigt. Sie ist damit äquivalent zur Flugdauer. Auf Grund der näherungsweise konstanten UAV-Fluggeschwindigkeit  $v^{ref} = 35$  m/s (aus Tabelle 3.7) ist die Aufklärungsdauer in erster Näherung proportional zur Länge der Referenztrajektorie.

Die Länge der NMPC-optimierten Referenztrajektorie, die für die Aufklärung eines Sensorfootprintpfads generiert wird, ist prinzipbedingt minimal. Ursächlich hierfür ist, dass der Flugweg zwischen zwei Zeitschritten  $s_{\Delta t}$  aus Gleichung (4.11) genau der Strecke entspricht, die das UAV mit der Sollwertgeschwindigkeit  $v^{ref}$  in einem Zeitschrittintervall  $\Delta t$  zurücklegt. Demgegenüber liegt die zurückgelegte Flugstrecke innerhalb eines Zeitschrittintervalls bei der DP-optimierten Referenztrajektorie zwischen  $s_{min}$  und  $s_{max}$  (siehe Tabelle 4.4 und die Erläuterungen in Abschnitt 4.4.1). Für die Bearbeitung eines Aufklärungsauftrags folgt daraus: die DP-optimierte Referenztrajektorie ist gleich lang oder länger als die Trajektorie, die aus der NMPC-Optimierung resultiert, bezogen auf den selben Aufklärungsauftrag.

Da die Aufklärungsdauer proportional zur Länge der Flugtrajektorie ist, wird die Trajektorienlänge in den nachfolgenden Tabellen zweckmäßigerweise mit angegeben. In Tabelle 6.7 sind der Mittelwert  $\bar{x}$  und die Standardabweichung  $s$  für die Trajektorienlänge sowie die korrelierende Aufklärungsdauer der NMPC-optimierten Referenztrajektorien

## 6 Evaluierung und Ergebnisse

aufgetragen. Die Mittelwerte und Standardabweichungen umfassen jeweils die Ergebnisse der 24 Einzelerperimente für das Routen- und Flächenaufklärungsszenario, wobei die Sensorleistungsmodelle YOLO-SLM und CC-SLM zusammengefasst sind. Die Mittelwerte wurden hierbei auf ganzzahlige Werte gerundet.

Tabelle 6.7: Trajektorienlänge und korrelierende Aufklärungsdauer der NMPC-optimierten Referenztrajektorien

Missionsszenario	Trajektorienlänge		Aufklärungsdauer	
	$\bar{x}$	s	$\bar{x}$	s
Routenaufklärung	3092 m	5,4 m	88 s	0,2 s
Flächenaufklärung	4203 m	13,9 m	120 s	0,4 s

Bei der NMPC-Trajektorienoptimierung ist die Aufklärungsdauer beim Flächenaufklärungsszenario um 36 % länger, als die Aufklärungsdauer für das Routenaufklärungsszenario. Ursächlich hierfür ist, dass die Aufklärungsdauer direkt proportional zur Anzahl der Zeitschritte  $N$  ist und diese wiederum proportional zur Länge des Sensorfootprintpfads sind. Ein Längenvergleich der Sensorfootprintpfade der beiden Aufklärungsszenarien bestätigt, dass die Pfadlänge des Flächenaufklärungsszenarios mit ca. 3 km ebenfalls 36 % länger ist als die ca. 2,2 km des Routenaufklärungsszenarios.

Ursächlich für die (Standard-)Abweichung der Trajektorienlänge und der Aufklärungsdauer ist, dass die UAV-Fluggeschwindigkeit durch einen Sollwert  $v^{ref}$  vorgegeben wird, der durch  $v_{min}$  und  $v_{max}$  (vgl. Tabelle 3.5) begrenzt wird. Aus der NMPC-Trajektorienoptimierung resultiert daher eine Referenztrajektorie, die in ihrer Länge von einer Trajektorie mit fest vorgegebener Fluggeschwindigkeit in Grenzen abweichen kann. Die Untersuchung der Standardabweichung zeigt, dass diese mit 0,17 % bzw. 0,33 %, bezogen auf die Routen- bzw. Flächenaufklärung, sehr gering ausfällt und daher im Folgenden vernachlässigt wird.

Die Trajektorienlänge und Aufklärungsdauer, die aus den Einzelerperimenten mit DP-Trajektorienoptimierung resultieren, sind in Tabelle 6.8 zusammengefasst. Im Gegensatz zur NMPC-optimierten Referenztrajektorie wird der Verlauf der Referenztrajektorie für den gesamten diskreten Zustandsraum optimiert und ist daher maßgeblich von der Ausprägung der Detektionsleistungen in den Perception Maps abhängig. Aus diesem Grund sind die Ergebnisse in der Tabelle entsprechend der verwendeten Sensorleistungsmodelle YOLO-SLM und CC-SLM aufgeteilt. Die Trajektorienlänge und die Aufklärungsdauer sind dabei auf die geglättete Referenztrajektorie (vgl. Abschnitt 4.4.3) bezogen.

### 6.3 Evaluierung der Aufklärungs- und Berechnungsdauer

Tabelle 6.8: Trajektorienlänge und korrelierende Aufklärungsdauer der DP-optimierten Referenztrajektorien mit Sensorleistungsmodell YOLO-SLM und CC-SLM

Missionsszenario	YOLO-SLM		CC-SLM	
	Länge	Dauer	Länge	Dauer
Routenaufklärung	3682 m	105 s	3689 m	105 s
Flächenaufklärung	5808 m	166 s	5731 m	164 s

Der direkte Vergleich der missionsspezifischen Aufklärungsdauer der NMPC-optimierten und der DP-optimierten Referenztrajektorie zeigt, dass die Aufklärungsdauer beim Routenaufklärungsszenario mit DP-Optimierung ca. 19 % über dem Wert der NMPC-Optimierung liegt. Beim Flächenaufklärungsszenario liegt die Aufklärungsdauer der DP-optimierten Trajektorienoptimierung, abhängig vom Sensorleistungsmodell, zwischen 36 % und 38 % über der NMPC-optimierten Trajektorie.

Der Grund für die gesteigerte Trajektorienlänge bzw. Aufklärungsdauer bei der DP-optimierten Trajektorie besteht darin, dass die Referenztrajektorie auf Basis eines kartesischen Gitters generiert wird. Die Parametereinstellung und die verwendete Kostenfunktion der DP-Trajektorienoptimierung begünstigen gerade Trajektoriensegmente, die entlang der Gitterhauptachsen ausgerichtet sind. Der Gitterpunktabstand parallel zu den Hauptachsen ist minimal und entspricht  $s_{min}$  (vgl. Abschnitt 4.4.1). Die Länge einer Trajektorie, die parallel zu den Hauptachsen verläuft, ist dadurch minimal. Die Richtungsänderung der Trajektoriensegmente wird auf Grund der Parametereinstellung der DP-Trajektorienoptimierung bevorzugt durch Kreisbogen realisiert, die die geraden Trajektoriensegmente zwischen zwei Zeitschritten miteinander verbinden. Die Länge eines Kreisbogens ist dabei (wesentlich) größer als der minimale Gitterpunktabstand, der zwischen zwei Zeitschritten vom UAV zurückgelegt wird (siehe auch Abb. 6.21). Mit jedem Richtungswechsel bzw. jedem Kreisbogen vergrößert sich dadurch die Gesamtlänge der Trajektorie.

Im Rahmen dieser Arbeit wurde durch eine Parameterstudie die zu erwartende Detektionsleistung in Abhängigkeit der Trajektorienlänge untersucht. Zu diesem Zweck wurden der Gewichtungsfaktor  $p_w$ , die Anzahl der diskreten Gierwinkel  $m_{\mu,k}$  und die maximale Pfadlänge  $s_{max}$  variiert (vgl. Tabelle 4.4). Das Ziel war es, eine zweckmäßige Parametrisierung zu finden, aus der eine hohe zu erwartende Detektionsleistung in Verbindung mit einer geringen Trajektorienlänge resultiert. Die Parameterwerte aus dem Ergebnis der Parameterstudie sind in Tabelle 4.4 zusammengefasst. Auf die Details der Studie wird in dieser Arbeit nicht weiter eingegangen.

### 6.3.2 Untersuchung der Berechnungsdauer

Bei der Ermittlung der Zeitdauer für die Berechnung der Referenztrajektorien (*Untersuchungsgegenstand 3.2*) wird ausschließlich das Verfahren der *kombinierten UAV-Pfadplanung und modellprädiktive Regelung* aus Abschnitt 4.3.2 bzw. die *dynamische Programmierung mit Dubins-Pfaden* aus Abschnitt 4.4.1 betrachtet. Die Berechnungsdauer der vorgelagerten Prozessschritte, wie die Planung des Sensorfootprintpfads (Abschnitt 4.2) und die Generierung der Perception Maps (Abschnitt 4.2.2) werden dabei nicht untersucht. Der Grund hierfür ist, dass diese Prozesse identisch für beide Optimierungsverfahren sind und der Fokus der nachfolgenden Evaluierung die Unterschiede der beiden Verfahren beleuchten soll.

Für die Berechnung der NMPC- und DP-optimierten Referenztrajektorien wurde ein PC mit Intel Core i7-6800K Prozessor (6 Kerne mit einer Taktrate von 3,4 GHz) und 32 GB Arbeitsspeicher genutzt. In Abschnitt 4.3.3 und 4.4.2 wurde bereits auf die Implementierung der modellprädiktiven Regelung bzw. der dynamischen Programmierung eingegangen.

In Tabelle 6.9 sind die Mittelwerte  $\bar{x}$  für die Zeitdauer zur Berechnung der Referenztrajektorien zusammengefasst, wobei die Mittelwerte auf ganzzahlige Werte gerundet wurden. Für die Berechnungsdauer der NMPC-optimierten Referenztrajektorien ist zusätzlich zum Mittelwert auch die Standardabweichung  $s$  aufgeführt.

Tabelle 6.9: Zeitdauer für die Berechnung der Referenztrajektorien

Missionsszenario	Berechnungsdauer		
	NMPC-Optimierung $\bar{x}$	$s$	DP-Optimierung $\bar{x}$
Routenaufklärung	52 s	5,8 s	2732 s
Flächenaufklärung	98 s	18,0 s	3719 s

Die verwendeten Sensorleistungsmodelle (YOLO-SLM und CC-SLM) beeinflussen wesentlich die zu erwartende Detektionsleistung, allerdings haben sie keinen Einfluss auf den eigentlichen Prozess der Trajektorienoptimierung. Aus diesem Grund werden die Einzelexperimente für die Leistungsmodelle YOLO-SLM und CC-SLM zusammen untersucht. Die Berechnungsdauer der NMPC-Trajektorienoptimierung umfasst jeweils 24 Einzelexperimente für das Routen- und Flächenaufklärungsszenario. Demgegenüber resultiert der Mittelwert für die Berechnungsdauer der DP-Optimierung aus jeweils zwei Experimenten, weshalb keine Standardabweichung angegeben wird.

### 6.3 Evaluierung der Aufklärungs- und Berechnungsdauer

Die Berechnungsdauer der DP-Trajektorienoptimierung ist für die Flächenaufklärung ca. 36 % größer, als für das Routenaufklärungsszenario. Dies wird damit erklärt, dass der Sensorfootprintpfad für das Flächenaufklärungsszenario mit ca. 3 km ebenfalls um 36 % länger ist als der Footprintpfad der Routenaufklärung (ca. 2,2 km). Da die Anzahl der diskreten Zeitschritte  $N$  (vgl. Abschnitt 3.6.2) linear mit der Länge des Sensorfootprintpfads skaliert, verlängert sich damit auch die Berechnungsdauer proportional mit der Anzahl der Zeitschritte.

Bei der NMPC-Optimierung konnte keine ähnlich deutliche Abhängigkeit der Berechnungsdauer von der Länge des Sensorfootprintpfads, und letztendlich der Anzahl der Zeitschritte, beobachtet werden. Dies wird einerseits auf die hohe Standardabweichung der Messungen zurückgeführt. Andererseits wurde bei der Durchführung der Experimente in der Simulation beobachtet, dass der Ressourcenbedarf des Arbeitsspeichers mit der Anzahl der Zeitschritte überproportional stark angestiegen ist, was auf einen gesteigerten Verwaltungsaufwand bei der Programmausführung schließen lässt. Demgegenüber kann die große Standardabweichung damit erklärt werden, dass die Berechnungsdauer, die für die Lösung des Optimalsteuerungsproblems erforderlich ist, stark von der Anzahl der erforderlichen Iterationsschritte, dem Konvergenzverhalten und dem Abbruchkriterium des Algorithmus bestimmt wird und damit zwischen den einzelnen Zeitschritten stark variieren kann.

Die Berechnungsdauer für die Trajektorienoptimierung mit modellprädiktiver Regelung beträgt ca. 0,9 min für das Routenaufklärungsszenario und ca. 1,6 min für das Flächenaufklärungsszenario. Demgegenüber liegt die Berechnungsdauer beim Routenaufklärungsszenario mit DP-optimierter Referenztrajektorie im Mittel bei ca. 45 min und damit 52-mal höher als bei der NMPC-Optimierung. Die Zeitdauer für die Berechnung der DP-optimierten Referenztrajektorie im Flächenaufklärungsszenario beträgt im Mittel 62 min und damit 40-mal mehr, als für die Berechnung der NMPC-optimierten Trajektorie notwendig ist. Der hohe Berechnungsaufwand für die DP-Optimierung kann damit erklärt werden, dass die Anzahl der Rechenschritte exponentiell mit der Anzahl der Zustandsausprägungen  $m_{state,k}$  (vgl. Gleichung (3.43)) steigt. Auf diesen Aspekt wurde bereits in Abschnitt 3.6.2 hingewiesen. Die Entscheidung, ob der hohe Berechnungsaufwand eine mögliche Steigerung der zu erwartenden Detektionsleistung rechtfertigt, obliegt dem UAV- / Sensoroperateur und ist u. a. vom Missionsziel abhängig. Eine abschließende Bewertung kann daher in dieser Arbeit nicht getroffen werden.

An dieser Stelle soll nochmals erwähnt werden, dass durch eine effizientere Code-Implementierung und eine umfassendere Parallelisierung der Berechnungsschritte eine weitere Reduzierung der Berechnungsdauer zu erwarten ist. Da dies allerdings nicht im

Fokus dieser Forschungsarbeit lag, wurden diesbezüglich keine weiteren Untersuchungen durchgeführt.

### 6.3.3 Zusammenfassung

In diesem letzten Abschnitt von Kapitel 6 wurden die *Untersuchungsgegenstände 3.1* und *3.2* bearbeitet. Zu diesem Zweck wurden die Aufklärungsdauer und die Berechnungsdauer der Referenztrajektorien untersucht, die mit Hilfe der NMPC- bzw. DP-Trajektorienoptimierung generiert wurden. Es hat sich gezeigt, dass sowohl die Berechnungsdauer als auch die Aufklärungsdauer der DP-optimierten Referenztrajektorien höher ist, als bei den NMPC-optimierten Trajektorien.

Die Aufklärungsdauer für das Routenaufklärungsszenario mit DP-optimierten Trajektorien ist ca. 19% höher als die Aufklärungsdauer, die für die NMPC-optimierten Trajektorien benötigt wird. Für das Flächenaufklärungsszenario resultierte aus den DP-optimierten Trajektorien eine Steigerung der Aufklärungsdauer um 36% bzw. 38%.

Des Weiteren war die Berechnungsdauer der DP-optimierten Referenztrajektorie für das Routenaufklärungsszenario ca. 52-mal und für das Flächenaufklärungsszenario ca. 40-mal höher, im Vergleich zur NMPC-Trajektorienoptimierung.

Mit der Bearbeitung der beiden *Untersuchungsgegenstände 3.1* und *3.2* kann die damit verbundene *Forschungsfrage 5.3* (Wie beeinflussen die zu untersuchenden Optimierungsmethoden die Aufklärungsdauer und die erforderliche Berechnungsdauer?) beantwortet und erfolgreich abgeschlossen werden. Damit wurden alle Forschungsfragen und Zwischenziele beantwortet.

Zwischenziel	1	2	3	4	5
Forschungsfrage	1.1	2.1 & 2.2	3.1 & 3.2	—	5.1 - 5.3
Bearbeitung in Kapitel	3	3	3 & 4	4	6



## 7 Diskussion und Ausblick

In diesem Kapitel werden die Ergebnisse der Zwischenziele zusammengefasst, diskutiert und daraus die Erfüllung des Forschungsziels begründet. Anschließend wird auf Kritikpunkte der verwendeten Konzepte eingegangen. Dem folgt eine Übersicht der technisch-wissenschaftlichen Beiträge, die im Rahmen dieser Forschungsarbeit geleistet wurden. Abschließend werden Anknüpfungspunkte für die weitere Forschung vorgestellt.

### 7.1 Zusammenfassung der Zwischenziele und des Forschungsziels

Im Folgenden werden alle *Zwischenziele* aus Abschnitt 1.3 zusammenfassend betrachtet. Zu diesem Zweck werden diese nachfolgend nochmals aufgelistet:

- Zwischenziel 1:** Entwicklung eines funktionalen Konzepts zur sensormodellbestimmten Befliegungsplanung. Die Befliegungsplanung soll ausgewählte perzeptive, sensorplattformenspezifische und missionsspezifische Anforderungen und Randbedingungen berücksichtigen.
- Zwischenziel 2:** Entwicklung und Untersuchung von Sensorleistungsmodellen zur Modellierung der Detektionsleistung in Abhängigkeit ausgewählter perzeptiver Einflüsse und Randbedingungen.
- Zwischenziel 3:** Auswahl und Implementierung geeigneter mathematischer Optimierungsmethoden zur Generierung optimaler Flugtrajektorien in Abhängigkeit der Detektionsleistung sowie sensorplattformenspezifischer und missionsspezifischer Anforderungen.
- Zwischenziel 4:** Überführung des Konzepts der sensormodellbestimmten Befliegungsplanung in ein Experimentalsystem.
- Zwischenziel 5:** Validierung der sensormodellbestimmten Befliegungsplanung im Experimentalsystem anhand repräsentativer Aufklärungsmissionen unter Berücksichtigung perzeptiver und sensorplattformenspezifischer Anforderungen.

## 7 Diskussion und Ausblick

Das *Zwischenziel 1* bezieht sich auf die Entwicklung eines funktionalen Konzepts zur sensormodellbestimmten Befliegungsplanung. Zu diesem Zweck wurden in Abschnitt 3.2 relevante perzeptive, sensorplattformspezifische und missionsspezifische Anforderungen und Randbedingungen definiert, die bei der Befliegungsplanung zu berücksichtigen sind. Die perzeptiven Anforderungen werden durch die Sensorleistungsmodelle umgesetzt und die sensorplattformspezifischen und missionsspezifischen Randbedingungen werden mit Hilfe der Trajektorienoptimierung realisiert. Darüber hinaus wurde das *Sensor- und Perzeptionsmanagement* um inhärente und dedizierte Fähigkeiten zur hochautomatisierten sensormodellbestimmten Befliegungsplanung erweitert, woraus das *Erweiterte Sensor- und Perzeptionsmanagement* (ESPM) entwickelt wurde. Zudem schafft das ESPM alle Voraussetzungen, um die in Abschnitt 3.2 festgelegten Anforderungen und Randbedingungen umsetzen und erfüllen zu können. Damit wurde das *Zwischenziel 1* erfolgreich abgeschlossen.

Für die Umsetzung und Erfüllung des *Zwischenziels 2* wurde in Abschnitt 3.4.4.3 ein Verfahren zur Generierung von Sensorleistungsmodellen aufgezeigt, das die perzeptiven Fähigkeiten (u. a. Genauigkeit und Trefferquote des Objektdetektors) und Randbedingungen (z. B. Sensorauflösung, Bildwinkel) einer Perzeptionskette mit Deep Learning basierendem Objektdetektor abbildet. Darauf aufbauend wurde in Abschnitt 3.4.5 das Konzept der Perception Map vorgestellt. Die Perception Map bildet die Detektionsleistung als Potentialfeld in der Bewegungsebene des UAVs ab, unter Verwendung von Sensorleistungsmodellen. Damit wird die Abbildung der Detektionsleistung in Abhängigkeit der Sensor- bzw. UAV-Position ermöglicht. Auch hier wurde das *Zwischenziel* erreicht.

Im Rahmen des *Zwischenziels 3* wurden in Abschnitt 3.6 die beiden mathematischen Methoden *dynamische Programmierung* und *nichtlineare modellprädiktive Regelung* für die Trajektorienoptimierung vorgestellt. Beide Methoden gründen auf einem *Optimalsteuerungsproblem*. Dies ermöglicht, dass sensorplattformspezifische Anforderungen und Limitierungen modelliert und explizit im Optimierungsprozess berücksichtigt werden können. Das Ergebnis der Optimierung ist eine Folge optimaler Steuereingaben, aus denen die optimale UAV-Referenztrajektorie in Abhängigkeit der Detektionsleistung generiert wird. Damit wurde das *Zwischenziel 3* erfolgreich umgesetzt.

Mit dem *Zwischenziel 4* wurde die sensormodellbestimmte Befliegungsplanung aus Abschnitt 3.3 in ein Experimentalsystem überführt. Hierfür wurden die einzelnen Prozessschritte, die in Kapitel 3 beschrieben wurden, zu einem ablauffähigen System miteinander verknüpft, wodurch dieses *Zwischenziel* ebenfalls als erfüllt betrachtet werden kann.

Das Erreichen des *Zwischenziels 5* wurde bereits in den vorangegangenen Abschnitten 6.1.4, 6.2.3 und 6.3.3 anhand der *Forschungsfragen 5.1 bis 5.3* und den dazugehörigen

*Untersuchungsgegenständen* bewertet. Es konnte gezeigt werden, dass das Zwischenziel durch die durchgeführten Experimente und der daraus folgenden Ergebnisse erfüllt wird.

Mit der vollständigen Bearbeitung der *Zwischenziele* und der Beantwortung der damit verbundenen *Forschungsfragen* aus Abschnitt 2.4 wurde das *übergeordnete Forschungsziel* dieser Arbeit (Entwicklung und Validierung eines funktionalen Konzepts zur Steigerung der Detektionsleistung unbemannter luftgestützter Aufklärungssysteme durch die Generierung optimaler, sensormodellbestimmter UAV-Flugtrajektorien. Für die Trajektorienoptimierung sind ausgewählte perzeptive, sensorplattformspezifische und missionsspezifische Anforderungen zu berücksichtigen.) aus Abschnitt 1.3 vollständig erreicht und erfolgreich umgesetzt.

## 7.2 Diskussion

Im vorangegangenen Abschnitt wurde begründet, dass alle Zwischenziele erfüllt und das übergeordnete Forschungsziel dieser Arbeit erreicht wurde. Dennoch sollen im Folgenden Kritikpunkte der verwendeten Konzepte diskutiert werden.

### 7.2.1 Defizite bei der Modellierung perzeptiver Einflussfaktoren

Bei den Sensorleistungsmodellen, die in dieser Arbeit Anwendung finden, werden unterschiedliche Umgebungszustände (u. a. atmosphärische- und photogrammetrische Zustände) als Eingangsgrößen verwendet, um daraus die zu erwartende Detektionsleistung als Ausgangsgröße zu ermitteln. Die Sonne ist die dominierende Lichtquelle im visuellen Spektrum, weshalb der Sonnenstand, beschrieben durch den Höhen- und Azimutwinkel, und die damit verbundenen optischen Effekte (z. B. Schattenwurf, Gegenlicht) einen Einfluss auf die erfassten Bilddaten und letztendlich auf die Detektionsleistung haben. Da der Höhen- und Azimutwinkel der Sonne im Sensorleistungsmodell nicht betrachtet werden, ist der Verlauf der Detektionsleistung in den Perception Maps rotationssymmetrisch zum Ursprung. Da insbesondere durch die optischen Effekte eine Beeinflussung der Detektionsleistung zu erwarten ist, wird empfohlen, bei einer Weiterentwicklung der Leistungsmodelle den Höhen- und Azimutwinkel als zusätzliche Eingangsgrößen mit in das Leistungsmodell aufzunehmen. Dadurch kann die Detaillierung und Genauigkeit des Leistungsmodells weiter gesteigert werden. Die explizite Berücksichtigung des Sonnenstands im Leistungsmodell hätte zur Folge, dass die damit generierten Perception Maps nicht länger rotationssymmetrisch verlaufen. Ungeachtet dessen hätte das Konzept der sensormodellbestimmten Befliegungsplanung weiterhin Bestand und würde zu validen Re-

ferenztrajektorien führen. Dies wird damit begründet, dass bei der zeit- und wertdiskreten Trajektorienoptimierung der gesamte diskrete Zustandsraum betrachtet wird, der durch die Perception Maps abgedeckt wird. Demgegenüber wird bei der zeitkontinuierlichen Trajektorienoptimierung durch die UAV-Pfadplanung (siehe Abschnitt 4.3.1) sichergestellt, dass der relevante (erfliegbare) Bereich der Perception Maps für die Optimierung berücksichtigt wird.

### 7.2.2 Einschränkungen bei der Modellierung sensorplattformenspezifischer Randbedingungen

Für die Modellierung der flugdynamischen Eigenschaften wurde im Rahmen der sensormodellbestimmten Befliegungsplanung eine konstante bzw. näherungsweise konstante Fluggeschwindigkeit vorausgesetzt. Dies wurde damit begründet, dass in dieser Arbeit ein Starrflügel-UAV mit definierter Fluggeschwindigkeit untersucht wird. Demgegenüber werden für die unbemannte luftgestützte Aufklärung häufig Multikopter-UAV eingesetzt, deren horizontale Fluggeschwindigkeit stark variieren kann: vom stationären Schwebeflug bis hin zur maximalen Fluggeschwindigkeit. Diesem Sachverhalt wird weder in der zeitkontinuierlichen noch in der zeit- und wertdiskreten Trajektorienoptimierung Rechnung getragen. Die explizite Variation der Fluggeschwindigkeit hätte zur Folge, dass zusätzliche Systemzustände berücksichtigt werden müssten, die insbesondere bei der zeit- und wertdiskreten Trajektorienoptimierung zu einer deutlichen Steigerung des Berechnungsaufwands und der damit verbundenen Berechnungsdauer führen würden. Für den Fall, dass eine in weiten Bereichen variable Fluggeschwindigkeit betrachtet werden soll, müssten weitere und umfassende Maßnahmen zur Optimierung und/oder Parallelisierung der Trajektorienoptimierung umgesetzt werden.

## 7.3 Technisch-wissenschaftliche Beiträge

Im Folgenden wird auf die technisch-wissenschaftlichen Beiträge eingegangen, die im Zuge dieser Forschungsarbeit erarbeitet wurden. In dieser Arbeit wurde ein Ansatz zur sensormodellbestimmten Befliegungsplanung entwickelt und untersucht, mit dem Ziel, die Detektionsleistung unbemannter luftgestützter Aufklärungssysteme zu steigern. Die Befliegungsplanung erfolgt hierbei unter Berücksichtigung ausgewählter perzeptiver, sensorplattformenspezifischer und missionsspezifischer Anforderungen. Für die Zielerreichung wurden verschiedene innovative Ansätze entwickelt, untersucht und die Ergebnisse in mehreren internationalen Veröffentlichungen dem Fachpublikum vorgestellt. Die Ergeb-

nisse aus diesen Arbeiten sollen einen Beitrag leisten, um den Stand der Forschung auf dem Gebiet der sensormodellbestimmten Befliegungsplanung zu erweitern:

- In der Arbeit von Ruf, Zwick, Morawietz & Stütz [195] wurden durch eine Literaturrecherche relevante Umgebungszustände identifiziert, die die Detektionsleistung einer Bildverarbeitungskette mit automatisierter Objektdetektion beeinflussen. Um die Detektionsleistung der automatisierten Detektion zu maximieren, wurde eine Leistungsbewertung mittels Pareto-Optimierung durchgeführt. Das Resultat waren optimale und räumlich-zeitlich koordinierte Sollwertvorgaben für die Sensorsteuerung (u. a. Bildwinkel und Elevationswinkel) sowie die Sensor- bzw. UAV-Positionierung.
- In der Arbeit von Zwick, Koch & Stütz [31] wurde das Paradigma des Sensor- und Perzeptionsmanagements [114] um inhärente Fähigkeiten zur automatisierten, sensormodellbestimmten Befliegungsplanung erweitert, indem die Sensorführung eng an die räumlich-zeitliche Plattformbewegung gekoppelt wurde. Durch die Generierung optimaler sensorspezifischer Flugtrajektorien wurde die sensorische Bilddatenerfassung derart beeinflusst, dass die Detektionsleistung der nachgeschalteten Bildverarbeitungskette gesteigert werden konnte.
- Die Generierung global optimaler Referenztrajektorien zur Steigerung der Detektionsleistung unter Verwendung detaillierter Sensorleistungsmodelle zur luftgestützten Fahrzeugdetektion wurde erstmals in Zwick, Gerdts & Stütz [40] vorgestellt. Unter Verwendung der dynamischen Programmierung wurden optimale Sensor- bzw. UAV-Positionen ermittelt, die wiederum Grundlage für die Generierung global optimaler UAV-Referenztrajektorien waren.
- In einer weiteren Arbeit von Zwick, Gerdts & Stütz [149] wurde ein neues Verfahren zur Generierung optimaler Referenztrajektorien für eine vorgegebene Startkonfiguration vorgestellt. Die Referenztrajektorie wurde mit Hilfe der nichtlinearen modellprädiktiven Regelung unter Berücksichtigung unterschiedlicher perzeptiver, sensorplattformspezifischer und missionsspezifischer Randbedingungen generiert. Hierbei war wiederum das Ziel, die Detektionsleistung der luftgestützten Aufklärung zu steigern.
- Ein Sensorleistungsmodell für eine Perzeptionskette mit Deep Learning basiertem Objektdetektor für den Einsatz in der luftgestützten Detektion wurde erstmals in der Arbeit von Zwick, Gerdts & Stütz [33] entwickelt und untersucht. Unter

Anwendung der Optimierungsmethoden der nichtlinearen modellprädiktiven Regelung und der dynamischen Programmierung wurden Referenz-Flugtrajektorien für die luftgestützte Befliegungsplanung geniert. Der Nachweis zur Steigerung der Detektionsleistung wurde durch Experimente in der Simulation erbracht.

### 7.4 Ausblick und Anknüpfungspunkte für die weitere Forschung

Auf Basis der erfolgten Untersuchungen zur sensormodellbestimmten Befliegungsplanung wurden die folgenden Anknüpfungspunkte für weitere Forschungsarbeiten identifiziert:

- Nach der Entwicklung des Konzepts zur sensormodellbestimmten Befliegungsplanung und dessen Umsetzung und Evaluierung durch ein Experimentalsystem, wird die Untersuchung des Konzepts durch UAVs mit bildgebender Sensorik im Realflug vorgeschlagen. Zu diesem Zweck müssten die entwickelten Algorithmen bezüglich Robustheit und Echtzeitfähigkeit hin optimiert werden. Durch den Einsatz der Befliegungsplanung im Realflug könnten zusätzliche Anforderungen und Randbedingungen identifiziert und untersucht werden, die in dem gegenwärtigen Konzept nicht oder nur ansatzweise modelliert sind.
- Ein weiterer Untersuchungsaspekt der identifiziert wurde, bezieht sich auf die automatisierte sensormodellbestimmte Befliegungsplanung mit zeitlich sequentieller oder paralleler Nutzung unterschiedlicher Aufklärungssensorik (z. B. elektrooptischer Sensor und Lidar), um einen Aufklärungsauftrag (z. B. Landezonenaufklärung) abzuarbeiten. Hierfür könnte die Trajektorienoptimierung für die gleichzeitige Nutzung verschiedener Sensorleistungsmodelle untersucht werden.
- Es hat sich gezeigt, dass die Zeitdauer für die Abarbeitung eines Aufklärungsauftrags näherungsweise proportional mit der Länge des Sensorfootprintpfads am Boden skaliert. Aus diesem Grund wird empfohlen, die sensormodellbestimmte Befliegungsplanung für den kooperativen Multi-UAV-Ansatz (siehe hierzu u. a. [43, 196]) zu analysieren. Durch den gleichzeitigen und koordinierten Einsatz mehrerer UAVs sollte es möglich sein, die Aufklärungsdauer signifikant zu reduzieren, indem das aufzuklärende Gebiet aufgabenspezifisch unterteilt und zeitgleich durch mehrere UAVs aufgeklärt wird.

# Literatur

- [1] NATO Standardization Agency, Hrsg. *STANAG 4671: UAV Systems Airworthiness Requirements (USAR) for North Atlantic Treaty Organization (NATO) Military UAV Systems*. 2007.
- [2] H. Shakhatreh u. a. „Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges“. In: *IEEE Access* 7 (2019), S. 48572–48634. DOI: 10.1109/ACCESS.2019.2909530.
- [3] J. Zhang und H. Huang. „Occlusion-Aware UAV Path Planning for Reconnaissance and Surveillance“. In: *Drones* 5.3 (2021). DOI: 10.3390/drones5030098.
- [4] X. Zhao u. a. „Detection, Tracking, and Geolocation of Moving Vehicle From UAV Using Monocular Camera“. In: *IEEE Access* 7 (2019), S. 101160–101170. DOI: 10.1109/ACCESS.2019.2929760.
- [5] C. Kyrkou u. a. „Optimized vision-directed deployment of UAVs for rapid traffic monitoring“. In: *2018 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2018, S. 1–6. DOI: 10.1109/ICCE.2018.8326145.
- [6] S. A. Cambone u. a. „Unmanned aircraft systems roadmap 2005-2030“. In: *Office of the Secretary of Defense* (2005). URL: [https://fas.org/irp/program/collect/uav\\_roadmap2005.pdf](https://fas.org/irp/program/collect/uav_roadmap2005.pdf) (besucht am 20.09.2023).
- [7] V. A. Feraru, R. E. Andersen und E. Boukas. „Towards an Autonomous UAV-based System to Assist Search and Rescue Operations in Man Overboard Incidents“. In: *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2020, S. 57–64. DOI: 10.1109/SSRR50563.2020.9292632.
- [8] L. Qingqing u. a. „Towards Active Vision with UAVs in Marine Search and Rescue: Analyzing Human Detection at Variable Altitudes“. In: *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2020, S. 65–70. DOI: 10.1109/SSRR50563.2020.9292596.

## Literatur

- [9] D. Erdos, A. Erdos und S. E. Watkins. „An Experimental UAV System for Search and Rescue Challenge“. In: *IEEE Aerospace and Electronic Systems Magazine* 28.5 (2013), S. 32–37. DOI: 10.1109/MAES.2013.6516147.
- [10] S. Sambolek und M. Ivasic-Kos. „Automatic Person Detection in Search and Rescue Operations Using Deep CNN Detectors“. In: *IEEE Access* 9 (2021), S. 37905–37922. DOI: 10.1109/ACCESS.2021.3063681.
- [11] E. Lygouras u. a. „Unsupervised Human Detection with an Embedded Vision System on a Fully Autonomous UAV for Search and Rescue Operations“. In: *Sensors* 19 (2019). DOI: 10.3390/s19163542.
- [12] D. Avola u. a. „Aerial video surveillance system for small-scale UAV environment monitoring“. In: *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. Piscataway, NJ: IEEE, 2017, S. 1–6. DOI: 10.1109/AVSS.2017.8078523.
- [13] S. Manfreda u. a. „On the Use of Unmanned Aerial Systems for Environmental Monitoring“. In: *Remote Sensing* 10 (2018). DOI: 10.3390/rs10040641.
- [14] M. A. Akhloufi, A. Couturier und N. A. Castro. „Unmanned Aerial Vehicles for Wildland Fires: Sensing, Perception, Cooperation and Assistance“. In: *Drones* 5 (2021). DOI: 10.3390/drones5010015.
- [15] M. Gašparović u. a. „An automatic method for weed mapping in oat fields based on UAV imagery“. In: *Computers and Electronics in Agriculture* 173 (2020). DOI: 10.1016/j.compag.2020.105385.
- [16] J. Cukor u. a. „Use of aerial thermography to reduce mortality of roe deer fawns before harvest“. In: *PeerJ* 7 (2019). DOI: 10.7717/peerj.6923.
- [17] N. Guimarães u. a. „Forestry Remote Sensing from Unmanned Aerial Vehicles: A Review Focusing on the Data, Processing and Potentialities“. In: *Remote Sensing* 12 (2020). DOI: 10.3390/rs12061046.
- [18] C. Zhang und J. M. Kovacs. „The application of small unmanned aerial systems for precision agriculture: a review“. In: *Precision Agriculture* 13 (2012), S. 693–712. DOI: 10.1007/s11119-012-9274-5.
- [19] F. Vanegas u. a. „A Novel Methodology for Improving Plant Pest Surveillance in Vineyards and Crops Using UAV-Based Hyperspectral and Spatial Data“. In: *Sensors* 18 (2018). DOI: 10.3390/s18010260.



- [20] J. Langhammer und T. Vacková. „Detection and Mapping of the Geomorphic Effects of Flooding Using UAV Photogrammetry“. In: *Pure and Applied Geophysics* 175 (2018), S. 3223–3245. DOI: 10.1007/s00024-018-1874-1.
- [21] F. Nex und F. Remondino. „UAV for 3D mapping applications: a review“. In: *Applied Geomatics* 6 (2014). DOI: 10.1007/s12518-013-0120-x.
- [22] J. M. Peschel und R. R. Murphy. „On the Human–Machine Interaction of Unmanned Aerial System Mission Specialists“. In: *IEEE Transactions on Human-Machine Systems* 43.1 (2013), S. 53–62. DOI: 10.1109/TSMCC.2012.2220133.
- [23] J. Redmon und A. Farhadi. „Yolov3: An incremental improvement“. In: *arXiv* (2018). DOI: 10.48550/arXiv.1804.02767. URL: <https://arxiv.org/pdf/1804.02767.pdf>.
- [24] D. Spodarets. *A Guide to the YOLO Family of Computer Vision Models*. URL: <https://dataphoenix.info/a-guide-to-the-yolo-family-of-computer-vision-models/> (besucht am 21.06.2023).
- [25] Y. Li und F. Ren. „Light-Weight RetinaNet for Object Detection“. In: *arXiv* (2019). DOI: 10.48550/arXiv.1905.10011. URL: <https://arxiv.org/pdf/1905.10011.pdf>.
- [26] M. Y. Arafat, M. M. Alam und S. Moh. „Vision-Based Navigation Techniques for Unmanned Aerial Vehicles: Review and Challenges“. In: *Drones* 7.2 (2023), S. 89. DOI: 10.3390/drones7020089.
- [27] M. Liu u. a. „UAV-YOLO: Small Object Detection on Unmanned Aerial Vehicle Perspective“. In: *Sensors* 20.8 (2020). DOI: 10.3390/s20082238.
- [28] A. Jain u. a. „AI-Enabled Object Detection in UAVs: Challenges, Design Choices, and Research Directions“. In: *IEEE Network* 35.4 (2021), S. 129–135. DOI: 10.1109/MNET.011.2000643.
- [29] J. Sandino u. a. „UAV Framework for Autonomous Onboard Navigation and People/Object Detection in Cluttered Indoor Environments“. In: *Remote Sensing* 12.20 (2020). DOI: 10.3390/rs12203386.
- [30] C. Hellert, S. Koch und P. Stütz. „Using Algorithm Selection for Adaptive Vehicle Perception Aboard UAV“. In: *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2019, S. 1–8. DOI: 10.1109/AVSS.2019.8909862.

## Literatur

- [31] M. Zwick, S. Koch und P. Stütz. „Enhancing Detection and Tracking Performance Using Sensor-specific Flight Trajectory Generation for UAVs: A Conceptual Approach“. In: *AIAA Scitech 2020 Forum*. Reston, Virginia, 2020. DOI: 10.2514/6.2020-1454.
- [32] C. Hellert und P. Stütz. „Performance prediction and selection of aerial perception functions during UAV missions“. In: *2017 IEEE Aerospace Conference*. IEEE, 2017, S. 1–12. DOI: 10.1109/AERO.2017.7943734.
- [33] M. Zwick, M. Gerds und P. Stütz. „Sensor-Model-Based Trajectory Optimization for UAVs to Enhance Detection Performance: An Optimal Control Approach and Experimental Results“. In: *Sensors* 23 (2023). DOI: 10.3390/s23020664.
- [34] J. J. Kiam, E. Besada-Portas und A. Schulte. „Hierarchical Mission Planning with a GA-Optimizer for Unmanned High Altitude Pseudo-Satellites“. In: *Sensors* 21 (2021). DOI: 10.3390/s21051630.
- [35] S. Hosseinianaki und M. Mesbahi. „Energy Aware Aerial Surveillance for a Long Endurance Solar-Powered UAV“. In: *Guidance, Navigation, and Control and Co-located Conferences*. 2013. DOI: 10.2514/6.2013-4552.
- [36] S. Koch und P. Stütz. „Sensor Scheduling for Airborne Multi-target Tracking with Limited Sensor Resources“. In: *Advances in Computer Vision and Computational Biology*. Transactions on Computational Science and Computational Intelligence. Cham: Springer International Publishing, 2021, S. 211–220. DOI: 10.1007/978-3-030-71051-4\_16.
- [37] P. Skoglar u. a. „Road Target Search and Tracking with Gimbaled Vision Sensor on an Unmanned Aerial Vehicle“. In: *Remote Sensing* 4 (2012), S. 2076–2111. DOI: 10.3390/rs4072076.
- [38] E. Semsch u. a. „Autonomous UAV Surveillance in Complex Urban Environments“. In: *2009 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*. Piscataway, NJ: IEEE, 2009, S. 82–85. DOI: 10.1109/WI-IAT.2009.132.
- [39] P. Skoglar. *Tracking and Planning for Surveillance Applications*. Bd. 1432. Linköping: Department of Electrical Engineering, Linköping University, 2012. ISBN: 978-91-7519-941-2.

- [40] M. Zwick, M. Gerdtts und P. Stütz. „Enhancing Detection Performance through Sensor Model-based Trajectory Optimization for UAVs“. In: *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*. 2021. DOI: 10.1109/DASC52595.2021.9594460.
- [41] J. Tisdale, Z. Kim und J. K. Hedrick. „Autonomous UAV path planning and estimation“. In: *IEEE Robotics & Automation Magazine* 16 (2009), S. 35–42. DOI: 10.1109/MRA.2009.932529.
- [42] E. G. Hemingway und O. M. O’Reilly. „Perspectives on Euler angle singularities, gimbal lock, and the orthogonality of applied forces and applied moments“. In: *Multibody System Dynamics* 44 (2018), S. 31–56. DOI: 10.1007/s11044-018-9620-0.
- [43] H.-X. Chen, Y. Nan und Y. Yang. „Multi-UAV Reconnaissance Task Assignment for Heterogeneous Targets Based on Modified Symbiotic Organisms Search Algorithm“. In: *Sensors* 19 (2019). DOI: 10.3390/s19030734.
- [44] Y. Wang u. a. „Reconnaissance Mission Conducted by UAV Swarms Based on Distributed PSO Path Planning Algorithms“. In: *IEEE Access* 7 (2019), S. 105086–105099. DOI: 10.1109/ACCESS.2019.2932008.
- [45] A. Gupta, D. Bessonov und P. Li. „A decision-theoretic approach to detection-based target search with a UAV“. In: *IROS Vancouver 2017*. Piscataway, NJ: IEEE, 2017. DOI: 10.1109/IROS.2017.8206423.
- [46] S. Hansen, T. McLain und M. Goodrich. „Probabilistic Searching Using a Small Unmanned Aerial Vehicle“. In: *AIAA Infotech@Aerospace 2007 Conference and Exhibit*. Reston, Virginia, 2007. DOI: 10.2514/6.2007-2740.
- [47] L. Evers u. a. „Online stochastic UAV mission planning with time windows and time-sensitive targets“. In: *European Journal of Operational Research* 238 (2014), S. 348–362. DOI: 10.1016/j.ejor.2014.03.014.
- [48] S.-l. Liao u. a. „Path planning for moving target tracking by fixed-wing UAV“. In: *Defence Technology* 16 (2020), S. 811–824. DOI: 10.1016/j.dt.2019.10.010.
- [49] B. Li und Y. Wu. „Path Planning for UAV Ground Target Tracking via Deep Reinforcement Learning“. In: *IEEE Access* 8 (2020), S. 29064–29074. DOI: 10.1109/ACCESS.2020.2971780.
- [50] C. Hellert. „Algorithmenauswahl für den adaptiven Sensoreinsatz an Bord unbemannter Luftfahrzeuge“. Dissertation. Universität der Bundeswehr München, 2019.

- [51] Z. Shiller. „Off-Line and On-Line Trajectory Planning“. In: *Motion and Operation Planning of Robotic Systems*. Bd. 29. Mechanisms and Machine Science. Springer, 2015, S. 29–62. DOI: 10.1007/978-3-319-14705-5\_2.
- [52] Z. Wu u. a. „Delving Into Robust Object Detection From Unmanned Aerial Vehicles: A Deep Nuisance Disentanglement Approach“. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), S. 1201–1210. DOI: 10.1109/ICCV.2019.00129.
- [53] H. Caesar u. a. „nuScenes: A multimodal dataset for autonomous driving“. In: *arXiv* (2020). DOI: 10.48550/arXiv.1903.11027. URL: <https://arxiv.org/pdf/1903.11027.pdf>.
- [54] J. Geyer u. a. „A2D2: Audi Autonomous Driving Dataset“. In: *arXiv* (2020). DOI: 10.48550/arXiv.2004.06320. URL: <https://arxiv.org/pdf/2004.06320.pdf>.
- [55] Q. Fan, L. Brown und J. Smith. „A closer look at Faster R-CNN for vehicle detection“. In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. Piscataway, NJ: IEEE, 2016, S. 124–129. DOI: 10.1109/IVS.2016.7535375.
- [56] B. Cheng u. a. „Revisiting RCNN: On Awakening the Classification Power of Faster RCNN“. In: *arXiv* (2018). DOI: 10.48550/arXiv.1803.06799. URL: <https://arxiv.org/pdf/1803.06799.pdf>.
- [57] R. Jin, Z. Jia und Z. Chu. „Efficient Aerial Image Object Detection with Imaging Condition Decomposition“. In: *2023 IEEE International Conference on Image Processing (ICIP)*. 2023, S. 620–624. DOI: 10.1109/ICIP49359.2023.10222553.
- [58] H.-K. Jung und G.-S. Choi. „Improved YOLOv5: Efficient Object Detection Using Drone Images under Various Conditions“. In: *Applied Sciences* 12 (2022). DOI: 10.3390/app12147255.
- [59] B. Kiefer, M. Messmer und A. Zell. „Diminishing Domain Bias by Leveraging Domain Labels in Object Detection on UAVs“. In: *2021 20th International Conference on Advanced Robotics (ICAR)*. Piscataway, NJ: IEEE, 2021. DOI: 10.1109/ICAR53236.2021.9659357.
- [60] O. Acatay u. a. „Comprehensive Evaluation of Deep Learning based Detection Methods for Vehicle Detection in Aerial Imagery“. In: *Proceedings of AVSS 2018*. Piscataway, NJ: IEEE, 2018. DOI: 10.1109/AVSS.2018.8639127.
- [61] W. Liu u. a. „SSD: Single Shot MultiBox Detector“. In: *Computer vision - ECCV 2016*. Bd. 9905. Lecture Notes in Computer Science. Cham: Springer, 2016, S. 21–37. DOI: 10.1007/978-3-319-46448-0\_2.

- [62] J. Sang u. a. „An Improved YOLOv2 for Vehicle Detection“. In: *Sensors* 18 (2018). DOI: 10.3390/s18124272.
- [63] G.-S. Xia u. a. „DOTA: A Large-Scale Dataset for Object Detection in Aerial Images“. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2018. DOI: 10.1109/cvpr.2018.00418.
- [64] E.-J. Lee u. a. „Validation of object detection in UAV-based images using synthetic data“. In: *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications III*. Proceedings of SPIE. Bellingham, Washington, USA: SPIE, 2021. DOI: 10.1117/12.2586860.
- [65] Z. Yi, S. Yongliang und Z. Jun. „An improved tiny-yolov3 pedestrian detection algorithm“. In: *Optik* 183 (2019), S. 17–23. DOI: 10.1016/j.ijleo.2019.02.038.
- [66] P. Viola und M. Jones. „Fast and Robust Classification Using Asymmetric AdaBoost and a Detector Cascade“. In: *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*. NIPS’01. Cambridge, MA, USA: MIT Press, 2001, S. 1311–1318.
- [67] P. Felzenszwalb, D. McAllester und D. Ramanan. „A discriminatively trained, multiscale, deformable part model“. In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, S. 1–8. DOI: 10.1109/CVPR.2008.4587597.
- [68] R. Brunelli und T. Poggio. „Template matching: matched spatial filters and beyond“. In: *Pattern Recognition* 30.5 (1997), S. 751–768. DOI: 10.1016/S0031-3203(96)00104-5.
- [69] T. B. Moeslund. „BLOB Analysis“. In: *Introduction to Video and Image Processing*. Undergraduate Topics in Computer Science. London: Springer London, 2012, S. 103–115. DOI: 10.1007/978-1-4471-2503-7\_7.
- [70] N. O’Mahony u. a. „Deep Learning vs. Traditional Computer Vision“. In: *Advances in Computer Vision*. Bd. 943. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing AG, 2020, S. 128–144. DOI: 10.1007/978-3-030-17795-9\_10.
- [71] R. I. Dancila und R. M. Botez. „New flight trajectory optimisation method using genetic algorithms“. In: *The Aeronautical Journal* 125 (2021), S. 618–671. DOI: 10.1017/aer.2020.138.

## Literatur

- [72] M. Zhang u. a. „A Multi-Strategy Improved Differential Evolution algorithm for UAV 3D trajectory planning in complex mountainous environments“. In: *Engineering Applications of Artificial Intelligence* 125 (2023). DOI: 10.1016/j.engappai.2023.106672.
- [73] T. Zhang u. a. „Trajectory Optimization for UAV Emergency Communication With Limited User Equipment Energy: A Safe-DQN Approach“. In: *IEEE Transactions on Green Communications and Networking* 5 (2021), S. 1236–1247. DOI: 10.1109/TGCN.2021.3068333.
- [74] Y. Liu u. a. „Control-oriented UAV highly feasible trajectory planning: A deep learning method“. In: *Aerospace Science and Technology* 110 (2021). DOI: 10.1016/j.ast.2020.106435.
- [75] H. Teng u. a. „3D Optimal Surveillance Trajectory Planning for Multiple UAVs by Using Particle Swarm Optimization With Surveillance Area Priority“. In: *IEEE Access* 8 (2020), S. 86316–86327. DOI: 10.1109/ACCESS.2020.2992217.
- [76] N. Ahmed, C. J. Pawase und K. Chang. „Distributed 3-D Path Planning for Multi-UAVs with Full Area Surveillance Based on Particle Swarm Optimization“. In: *Applied Sciences* 11 (2021). DOI: 10.3390/app11083417.
- [77] S. Pérez-Carabaza u. a. „UAV trajectory optimization for Minimum Time Search with communication constraints and collision avoidance“. In: *Engineering Applications of Artificial Intelligence* 85 (2019), S. 357–371. DOI: 10.1016/j.engappai.2019.06.002.
- [78] H. Tian u. a. „Design and validation of a multi-objective waypoint planning algorithm for UAV spraying in orchards based on improved ant colony algorithm“. In: *Frontiers in plant science* 14 (2023). DOI: 10.3389/fpls.2023.1101828.
- [79] A. Ghorbani, A. Abid und J. Zou. „Interpretation of Neural Networks Is Fragile“. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33 (2019), S. 3681–3688. DOI: 10.1609/aaai.v33i01.33013681.
- [80] G. Montavon, W. Samek und K.-R. Müller. „Methods for interpreting and understanding deep neural networks“. In: *Digital Signal Processing* 73 (2018). DOI: 10.1016/j.dsp.2017.10.011.
- [81] R. C. Fong und A. Vedaldi. „Interpretable Explanations of Black Boxes by Meaningful Perturbation“. In: *2017 IEEE International Conference on Computer Vision (ICCV)* (2017). DOI: 10.1109/ICCV.2017.371.

- [82] X. Yu, X. Zhou und Y. Zhang. „Collision-Free Trajectory Generation and Tracking for UAVs Using Markov Decision Process in a Cluttered Environment“. In: *Journal of Intelligent & Robotic Systems* 93 (2019), S. 17–32. DOI: 10.1007/s10846-018-0802-z.
- [83] C. M. Eaton, E. K. Chong und A. A. Maciejewski. „Robust UAV path planning using POMDP with limited FOV sensor“. In: *2017 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2017, S. 1530–1535. DOI: 10.1109/CCTA.2017.8062674.
- [84] A. Gasparetto u. a. „Path Planning and Trajectory Planning Algorithms: A General Overview“. In: *Motion and Operation Planning of Robotic Systems*. Bd. 29. Cham: Springer International Publishing, 2015, S. 3–27. DOI: 10.1007/978-3-319-14705-5\_1.
- [85] Q. Lin u. a. „A shortest path routing algorithm for unmanned aerial systems based on grid position“. In: *Journal of Network and Computer Applications* 103 (2018), S. 215–224. DOI: 10.1016/j.jnca.2017.08.008.
- [86] G. Farid u. a. „Modified A-Star (A\*) Approach to Plan the Motion of a Quadrotor UAV in Three-Dimensional Obstacle-Cluttered Environment“. In: *Applied Sciences* 12.12 (2022), S. 5791. DOI: 10.3390/app12125791.
- [87] R. Bellman. *Dynamic programming*. 6. Printing. Princeton, N.J.: Princeton University Press, 1972. ISBN: 0-691-07951-X.
- [88] E. Bulut und I. Guevenc. „Trajectory Optimization for Cellular-Connected UAVs with Disconnectivity Constraint“. In: *2018 IEEE International Conference on Communications workshops (ICC workshops)*. Piscataway, NJ: IEEE, 2018. DOI: 10.1109/ICCW.2018.8403623.
- [89] S. A. P. Quintero u. a. „Optimal UAV coordination for target tracking using dynamic programming“. In: *49th IEEE Conference on Decision and Control (CDC)*. IEEE, 2010, S. 4541–4546. DOI: 10.1109/CDC.2010.5717933.
- [90] E. F. Camacho und C. Bordons. *Model Predictive control*. 2. Aufl. Advanced textbooks in control and signal processing. London: Springer London, 2007. ISBN: 978-1-85233-694-3. DOI: 10.1007/978-0-85729-398-5.
- [91] R. Ghaemi, J. Sun und I. V. Kolmanovsky. „An integrated perturbation analysis and Sequential Quadratic Programming approach for Model Predictive Control“. In: *Automatica* 45 (2009), S. 2412–2418. DOI: 10.1016/j.automatica.2009.06.028.

## Literatur

- [92] T. Wakabayashi, Y. Nunoya und S. Suzuki. „Dynamic Obstacle Avoidance of Multi-Rotor UAV using Chance Constrained MPC“. In: *2021 21st International Conference on Control, Automation and Systems (ICCAS)*. Piscataway, NJ: IEEE, 2021, S. 412–417. DOI: 10.23919/ICCAS52745.2021.9649942.
- [93] T. J. Stastny, A. Dash und R. Siegwart. „Nonlinear MPC for Fixed-wing UAV Trajectory Tracking: Implementation and Flight Experiments“. In: *AIAA Guidance Navigation and Control Conference 2017*. 2017. DOI: 10.2514/6.2017-1512.
- [94] J. D. Lee u. a. „Terrain Following of Tiltrotor UAV using MPC and Nonlinear Disturbance Observer-Based Backstepping“. In: *2022 22nd International Conference on Control, Automation and Systems (ICCAS)*. Piscataway, NJ: IEEE, 2022, S. 852–857. DOI: 10.23919/ICCAS55662.2022.10003812.
- [95] C. Goerzen, Z. Kong und B. Mettler. „A Survey of Motion Planning Algorithms from the Perspective of Autonomous UAV Guidance“. In: *Journal of Intelligent & Robotic Systems* (2010), S. 65–100. DOI: 10.1007/s10846-009-9383-1.
- [96] M. G. Park, J. H. Jeon und M. C. Lee. „Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing“. In: *2001 IEEE International Symposium on Industrial Electronics proceedings*. Piscataway, NJ: IEEE, 2001, S. 1530–1535. DOI: 10.1109/ISIE.2001.931933.
- [97] A. Batinovic u. a. „Path Planning with Potential Field-Based Obstacle Avoidance in a 3D Environment by an Unmanned Aerial Vehicle“. In: *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*. Piscataway, NJ: IEEE, 2022, S. 394–401. DOI: 10.1109/ICUAS54217.2022.9836159.
- [98] G.-c. Luo u. a. „UAV Path Planning in Mixed-Obstacle Environment via Artificial Potential Field Method Improved by Additional Control Force“. In: *Asian Journal of Control* 17.5 (2015), S. 1600–1610. DOI: 10.1002/asjc.960.
- [99] S. M. LaValle. „Rapidly-Exploring Random Trees: A New Tool for Path Planning“. In: *Research Report 9811* (1998).
- [100] S. M. LaValle und J. J. Kuffner. „Randomized Kinodynamic Planning“. In: *The International Journal of Robotics Research* 20.5 (2001), S. 378–400. DOI: 10.1177/02783640122067453.
- [101] D. J. Webb und J. van den Berg. „Kinodynamic RRT\*: Asymptotically optimal motion planning for robots with linear dynamics“. In: *2013 IEEE International Conference on Robotics and Automation (ICRA 2013)*. Piscataway, NJ: IEEE, 2013, S. 5054–5061. DOI: 10.1109/ICRA.2013.6631299.



- [102] B. Donald u. a. „Kinodynamic motion planning“. In: *Journal of the ACM* 40.5 (1993), S. 1048–1066. DOI: 10.1145/174147.174150.
- [103] J. Ge u. a. „Trajectory Planning of Fixed-wing UAV Using Kinodynamic RRT Algorithm“. In: *10th International Conference on Information Science and Technology*. Piscataway, NJ: IEEE, 2020, S. 44–49. DOI: 10.1109/ICIST49303.2020.9202213.
- [104] M. Yavari, K. Gupta und M. Mehrandezh. „Interleaved Predictive Control and Planning for an Unmanned Aerial Manipulator With on-the-Fly Rapid Re-Planning in Unknown Environments“. In: *IEEE Transactions on Automation Science and Engineering* 20.3 (2023), S. 1690–1705. DOI: 10.1109/TASE.2022.3184458.
- [105] L. Lu u. a. „Fixed-Wing UAV Path Planning in a Dynamic Environment via Dynamic RRT Algorithm“. In: *Mechanism and machine science*. Bd. 408. Lecture Notes in Electrical Engineering. Singapore: Springer, 2017, S. 271–282. DOI: 10.1007/978-981-10-2875-5\_23.
- [106] R. Penicka und D. Scaramuzza. „Minimum-Time Quadrotor Waypoint Flight in Cluttered Environments“. In: *IEEE Robotics and Automation Letters* 7.2 (2022), S. 5719–5726. DOI: 10.1109/LRA.2022.3154013.
- [107] 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.
- [108] V. Darbari, S. Gupta und O. P. Verma. „Dynamic motion planning for aerial surveillance on a fixed-wing UAV“. In: *2017 International Conference on Unmanned Aircraft Systems (ICUAS'17)*. Piscataway, NJ: IEEE, 2017, S. 488–497. DOI: 10.1109/ICUAS.2017.7991463.
- [109] W. Li u. a. „Path Planning for UAV Based on Improved PRM“. In: *Energies* 15.19 (2022). DOI: 10.3390/en15197267.
- [110] G. Ng und K. Ng. „Sensor management – what, why and how“. In: *Information Fusion* 1.2 (2000), S. 67–75. DOI: 10.1016/S1566-2535(00)00009-9.
- [111] P. Skoglar. *Planning Methods for Aerial Exploration and Ground Target Tracking*. Bd. 1420. Linköping studies in science and technology. Thesis. Linköping: Department of Electrical Engineering, Linköping University, 2009. ISBN: 978-91-7393-509-8.
- [112] A. O. Hero und D. Cochran. „Sensor Management: Past, Present, and Future“. In: *IEEE Sensors Journal* 11.12 (2011), S. 3064–3075. DOI: 10.1109/JSEN.2011.2167964.

## Literatur

- [113] A. Mahajan und D. Teneketzis. „Multi-Armed Bandit Problems“. In: *Foundations and applications of sensor management*. Signals and communication technology. New York, NY: Springer Science + Business Media, 2008, S. 121–151. DOI: 10.1007/978-0-387-49819-5\_6.
- [114] M. Russ und P. Stütz. „Airborne sensor and perception management: A conceptual approach for surveillance UAS“. In: *2012 15th International Conference on Information Fusion*. 2012, S. 2444–2451.
- [115] M. Russ. „Realisierung eines Sensor- & Perzeptionsmanagement Systems für unbemannte fliegende Plattformen“. Dissertation. Universität der Bundeswehr München, 2023.
- [116] D. Smirnov und P. Stütz. „Use case driven approach for ontology-based modeling of reconnaissance resources on-board UAVs using OWL“. In: *2017 IEEE Aerospace Conference*. IEEE, 2017, S. 1–17. DOI: 10.1109/AERO.2017.7943606.
- [117] C. Hellert u. a. „A High Level Active Perception Concept for UAV Mission Scenarios“. In: *Deutsche Gesellschaft für Luft- und Raumfahrt-Lilienthal-Oberth eV* (2012).
- [118] J. Uhrmann und A. Schulte. „Task-based Guidance of Multiple UAV Using Cognitive Automation“. In: *COGNITIVE 2011, The Third International Conference on Advanced Cognitive Technologies and Applications* (2011).
- [119] A. Gupta und X. Fernando. „Simultaneous Localization and Mapping (SLAM) and Data Fusion in Unmanned Aerial Vehicles: Recent Advances and Challenges“. In: *Drones* 6.4 (2022), S. 85. DOI: 10.3390/drones6040085.
- [120] I. Cvišić u. a. „SOFT-SLAM: Computationally efficient stereo visual simultaneous localization and mapping for autonomous unmanned aerial vehicles“. In: *Journal of Field Robotics* 35.4 (2018), S. 578–595. DOI: 10.1002/rob.21762.
- [121] A. Rauf u. a. „A State of the Art in Simultaneous Localization and Mapping (SLAM) for Unmanned Ariel Vehicle (UAV): A Review“. In: *Electrical, Control and Communication Engineering* 18.1 (2022), S. 50–56. DOI: 10.2478/ecce-2022-0007.
- [122] N. Patrizi u. a. „A UAV-enabled Dynamic Multi-Target Tracking and Sensing Framework“. In: *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*. Institute of Electrical and Electronics Engineers, 2020, S. 1–6. DOI: 10.1109/GLOBECOM42002.2020.9322567.

- [123] S. Ragi und E. K. P. Chong. „Dynamic UAV path planning for multitarget tracking“. In: *2012 American Control Conference (ACC 2012)*. Piscataway, NJ: IEEE, 2012, S. 3845–3850. DOI: 10.1109/ACC.2012.6314653.
- [124] Y. Zhao u. a. „Systemic design of distributed multi-UAV cooperative decision-making for multi-target tracking“. In: *Autonomous Agents and Multi-Agent Systems* 33.1-2 (2019), S. 132–158. DOI: 10.1007/s10458-019-09401-5.
- [125] N. Yifeng u. a. „A method of visual sensor employment planning for UAV target detection“. In: *Proceedings of the 28th Chinese Control and Decision Conference (2016 CCDC)*. Piscataway, NJ: IEEE, 2016, S. 784–789. DOI: 10.1109/CCDC.2016.7531091.
- [126] T. Sjaardema, C. Smith und G. Birch. *History and Evolution of the Johnson Criteria*. United States, 2015. DOI: 10.2172/1222446.
- [127] Y. M. Yoo und C. G. Park. „Improvement of terrain referenced navigation using a Point Mass Filter with grid adaptation“. In: *International Journal of Control, Automation and Systems* 13.5 (2015), S. 1173–1181. DOI: 10.1007/s12555-013-0410-4.
- [128] S. Ivić u. a. „Multi-UAV trajectory planning for 3D visual inspection of complex structures“. In: *Automation in Construction* 147 (2023). DOI: 10.1016/j.autcon.2022.104709.
- [129] A. Bircher u. a. „Receding horizon path planning for 3D exploration and surface inspection“. In: *Autonomous robots* 42.2 (2018), S. 291–306. DOI: 10.1007/s10514-016-9610-0.
- [130] M. Qi, L. Dou und B. Xin. „3D Smooth Trajectory Planning for UAVs under Navigation Relayed by Multiple Stations Using Bézier Curves“. In: *Electronics* 12.11 (2023). DOI: 10.3390/electronics12112358.
- [131] D. Du u. a. „The Unmanned Aerial Vehicle Benchmark: Object Detection and Tracking“. In: *arXiv* (2018), S. 370–386. DOI: 10.48550/arXiv.1804.00518.
- [132] M. Krump und P. Stütz. „UAV Based Vehicle Detection with Synthetic Training: Identification of Performance Factors Using Image Descriptors and Machine Learning“. In: *Modelling and Simulation for Autonomous Systems*. Cham: Springer International Publishing, 2021, S. 62–85. DOI: 10.1007/978-3-030-70740-8\_5.
- [133] R. G. Driggers. *Introduction to Infrared and Electro-Optical Systems*. 3rd ed. Norwood: Artech House, 2022. ISBN: 978-1-63081-832-6.

## Literatur

- [134] G. Hummel. „On synthetic datasets for development of computer vision algorithms in airborne reconnaissance applications“. Dissertation. Universitätsbibliothek der Universität der Bundeswehr München und Universität der Bundeswehr München, Fakultät für Luft- und Raumfahrttechnik, 2017.
- [135] R. Szeliski. *Computer Vision*. Cham: Springer International Publishing, 2022. ISBN: 978-3-030-34371-2. DOI: 10.1007/978-3-030-34372-9.
- [136] H. Süße und E. Rodner. *Bildverarbeitung und Objekterkennung*. Wiesbaden: Springer Fachmedien Wiesbaden, 2014. ISBN: 978-3-8348-2605-3. DOI: 10.1007/978-3-8348-2606-0.
- [137] M. Krump, M. Ruß und P. Stütz. „Deep Learning Algorithms for Vehicle Detection on UAV Platforms: First Investigations on the Effects of Synthetic Training“. In: *Modelling and Simulation for Autonomous Systems*. Cham: Springer International Publishing, 2020, S. 50–70. DOI: 10.1007/978-3-030-43890-6\_5.
- [138] J. Redmon u. a. „You Only Look Once: Unified, Real-Time Object Detection“. In: *arXiv* (2016). DOI: 10.48550/arXiv.1506.02640. URL: <http://arxiv.org/pdf/1506.02640v5>.
- [139] L. Jiao u. a. „A Survey of Deep Learning-Based Object Detection“. In: *IEEE Access* 7 (2019), S. 128837–128868. DOI: 10.1109/ACCESS.2019.2939201.
- [140] Z.-Q. Zhao u. a. „Object Detection With Deep Learning: A Review“. In: *IEEE transactions on neural networks and learning systems* 30.11 (2019), S. 3212–3232. DOI: 10.1109/TNNLS.2018.2876865.
- [141] DJI, Hrsg. *ZENMUSE XT2: User Manual V1.0*. 2018. URL: <https://www.dji.com/de/zenmuse-xt2> (besucht am 23.09.2022).
- [142] H. Rezatofighi u. a. „Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression“. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), S. 658–666. DOI: 10.1109/CVPR.2019.00075.
- [143] R. Padilla u. a. „A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit“. In: *Electronics* 10.3 (2021), S. 279. DOI: 10.3390/electronics10030279.
- [144] K. Boyd, K. H. Eng und C. D. Page. „Area under the Precision-Recall Curve: Point Estimates and Confidence Intervals“. In: *Advanced information systems engineering*. Bd. 7908. Lecture Notes in Computer Science. Berlin und Heidelberg: Springer, 2013, S. 451–466. DOI: 10.1007/978-3-642-40994-3\_29.

- [145] A. Bochkovskiy, C.-Y. Wang und H.-Y. M. Liao. „YOLOv4: Optimal Speed and Accuracy of Object Detection“. In: *arXiv* (2020). DOI: 10.48550/arXiv.2004.10934. URL: <https://arxiv.org/pdf/2004.10934>.
- [146] U. Nepal und H. Eslamiat. „Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs“. In: *Sensors* 22.2 (2022). DOI: 10.3390/s22020464.
- [147] S. M. LaValle. *Planning algorithms*. Cambridge: Cambridge University Press, 2006. ISBN: 9780511546877. DOI: 10.1017/CB09780511546877.
- [148] *ArduPilot*. URL: <https://ardupilot.org/> (besucht am 02.07.2023).
- [149] M. Zwick, M. Gerdtts und P. Stütz. „Sensor Model-Based Trajectory Optimization for UAVs Using Nonlinear Model Predictive Control“. In: *AIAA SCITECH 2022 Forum*. Reston, Virginia, 2022. DOI: 10.2514/6.2022-1286.
- [150] D. P. Bertsekas. *Dynamic programming and optimal control*. Fourth edition. Bd. 1. Athena scientific optimization and computation series. Belmont, Massachusetts: Athena Scientific, 2017. ISBN: 978-1886529434.
- [151] L. E. Dubins. „On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents“. In: *American Journal of Mathematics* 79.3 (1957). DOI: 10.2307/2372560.
- [152] J. Reeds und L. Shepp. „Optimal paths for a car that goes both forwards and backwards“. In: *Pacific Journal of Mathematics* 145.2 (1990), S. 367–393. DOI: 10.2140/pjm.1990.145.367.
- [153] H. Chitsaz und S. M. LaValle. „Time-optimal paths for a Dubins airplane“. In: *2007 46th IEEE Conference on Decision and Control*. 2007, S. 2379–2384. DOI: 10.1109/CDC.2007.4434966.
- [154] I. Lugo-Cardenas u. a. „Dubins path generation for a fixed wing UAV“. In: *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2014, S. 339–346. DOI: 10.1109/ICUAS.2014.6842272.
- [155] M. Owen, R. W. Beard und T. W. McLain. „Implementing Dubins Airplane Paths on Fixed-Wing UAVs“. In: *Handbook of Unmanned Aerial Vehicles*. Dordrecht: Springer Netherlands, 2015, S. 1677–1701. DOI: 10.1007/978-90-481-9707-1\_120.
- [156] W. Fichter und J. Stephan. *Flugregelung*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2020. ISBN: 978-3-662-60906-4. DOI: 10.1007/978-3-662-60907-1.

## Literatur

- [157] A. M. Shkel und V. Lumelsky. „Classification of the Dubins set“. In: *Robotics and Autonomous Systems* 34.4 (2001), S. 179–202. DOI: 10.1016/S0921-8890(00)00127-5.
- [158] J.-D. Boissonnat, A. Cerezo und J. Leblond. „Shortest paths of bounded curvature in the plane“. In: *Proceedings 1992 IEEE International Conference on Robotics and Automation*. 1992, S. 2315–2320. DOI: 10.1109/ROBOT.1992.220117.
- [159] David A. Anisi u. a. *Optimal Motion Control of a Ground Vehicle*. 2003. URL: <https://api.semanticscholar.org/CorpusID:14854743>.
- [160] K. Yang und S. Sukkarieh. „An Analytical Continuous-Curvature Path-Smoothing Algorithm“. In: *IEEE Transactions on Robotics* 26.3 (2010), S. 561–568. DOI: 10.1109/TR0.2010.2042990.
- [161] T. Fraichard und A. Scheuer. „From Reeds and Shepp’s to Continuous-Curvature Paths“. In: *IEEE Transactions on Robotics* 20.6 (2004), S. 1025–1035. DOI: 10.1109/TR0.2004.833789.
- [162] M. Gerdts. *Optimal Control of ODEs and DAEs*. 1. Aufl. De Gruyter, 2012. ISBN: 9783110249958. DOI: 10.1515/9783110249996.
- [163] A. Rao. „A Survey of Numerical Methods for Optimal Control“. In: *Advances in the Astronautical Sciences* 135 (2010).
- [164] J. T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. 2. Aufl. Society for Industrial and Applied Mathematics, 2010. ISBN: 978-0-89871-688-7. DOI: 10.1137/1.9780898718577.
- [165] M. Diehl u. a. „Fast Direct Multiple Shooting Algorithms for Optimal Robot Control“. In: *Fast motions in biomechanics and robotics*. Bd. 340. Lecture Notes in Control and Information Sciences. Berlin und Heidelberg: Springer, 2006, S. 65–93. DOI: 10.1007/978-3-540-36119-0\_4.
- [166] G. Giorgi, B. Jiménez und V. Novo. „Approximate Karush–Kuhn–Tucker Condition in Multiobjective Optimization“. In: *Journal of Optimization Theory and Applications* 171.1 (2016), S. 70–89. DOI: 10.1007/s10957-016-0986-y.
- [167] D. E. Kirk. *Optimal control theory: An introduction*. Dover Publications, 2004. ISBN: 0-486-43484-2.
- [168] W. Alt. *Nichtlineare Optimierung*. Wiesbaden: Vieweg+Teubner Verlag, 2002. ISBN: 978-3-322-84904-5. DOI: 10.1007/978-3-322-84904-5.

- [169] L. Grüne und J. Pannek. *Nonlinear Model Predictive Control*. Cham: Springer International Publishing, 2017. ISBN: 978-3-319-46023-9. DOI: 10.1007/978-3-319-46024-6.
- [170] M. Diehl. *Lecture Notes on Optimal Control and Estimation*. 2014. URL: <https://www.syscop.de/files/2015ss/events/tempo/optimalcontrolDiehl.pdf> (besucht am 25.07.2022).
- [171] J. B. Rawlings, D. Q. Mayne und M. Diehl. *Model predictive control: Theory, computation, and design*. 2. Aufl. Madison, Wisconsin: Nob Hill Publishing, 2017. ISBN: 978-0-97593773-0.
- [172] D. Kraft. „On Converting Optimal Control Problems into Nonlinear Programming Problems“. In: *Computational Mathematical Programming. NATO ASI Series* (1985), S. 261–280. DOI: 10.1007/978-3-642-82450-0\_9.
- [173] H. G. Bock und K. J. Plitt. „A Multiple Shooting Algorithm for Direct Solution of Optimal Control Problems“. In: *IFAC Proceedings Volumes* 17.2 (1984), S. 1603–1608. DOI: 10.1016/S1474-6670(17)61205-9.
- [174] M. Gerdts. „Direct Shooting Method for the Numerical Solution of Higher-Index DAE Optimal Control Problems“. In: *Journal of Optimization Theory and Applications* 117.2 (2003), S. 267–294. DOI: 10.1023/A:1023679622905.
- [175] G. Leitmann. *The calculus of variations and optimal control: An introduction*. 3. printing. Bd. 24. Mathematical concepts and methods in science and engineering. New York, NY: Plenum Press, 1986. ISBN: 0306407078.
- [176] O. von Stryk und R. Bulirsch. „Direct and indirect methods for trajectory optimization“. In: *Annals of Operations Research* 37.1 (1992), S. 357–373. DOI: 10.1007/BF02071065.
- [177] A. J. Jerri. „The Shannon sampling theorem - Its various extensions and applications: A tutorial review“. In: *Proceedings of the IEEE* 65.11 (1977), S. 1565–1596. DOI: 10.1109/PROC.1977.10771.
- [178] F. Allgöwer, R. Findeisen und Z.K. Nagy. „Nonlinear model predictive control: From theory to application“. In: *Journal of the Chinese Institute of Chemical Engineers* 35 (2004), S. 299–315. URL: <https://api.semanticscholar.org/CorpusID:16943674>.
- [179] R. Findeisen und F. Allgöwer. „An Introduction to Nonlinear Model Predictive Control“. In: *21st Benelux meeting on systems and control*. Bd. 11. 2002, S. 119–141.

## Literatur

- [180] M. Morari und J. H. Lee. „Model predictive control: past, present and future“. In: *Computers & Chemical Engineering* 23.4-5 (1999), S. 667–682. DOI: 10.1016/S0098-1354(98)00301-9.
- [181] M. A. Henson. „Nonlinear model predictive control: current status and future directions“. In: *Computers & Chemical Engineering* 23.2 (1998), S. 187–202. DOI: 10.1016/S0098-1354(98)00260-9.
- [182] F. Allgöwer u. a. „Nonlinear Predictive Control and Moving Horizon Estimation — An Introductory Overview“. In: *Advances in Control*. London: Springer, 1999, S. 391–449. DOI: 10.1007/978-1-4471-0853-5\_19.
- [183] N. Caroff und H. Frankowska. „Optimality and Characteristics of Hamilton-Jacobi-Bellman Equations“. In: *Optimization, Optimal Control and Partial Differential Equations*. Basel: Birkhäuser, 1992, S. 169–180. DOI: 10.1007/978-3-0348-8625-3\_16.
- [184] D. Mayne. „Nonlinear Model Predictive Control: Challenges and Opportunities“. In: *Nonlinear Model Predictive Control*. Birkhäuser, Basel, 2000, S. 23–44. DOI: 10.1007/978-3-0348-8407-5\_2.
- [185] M. Diehl, H. J. Ferreau und N. Haverbeke. „Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation“. In: *Nonlinear model predictive control*. Bd. 384. Lecture Notes in Control and Information Sciences. Berlin und Heidelberg: Springer, 2009, S. 391–417. DOI: 10.1007/978-3-642-01094-1\_32.
- [186] G. de Nicolao, L. Magni und R. Scattolini. „Stability and Robustness of Nonlinear Receding Horizon Control“. In: *Nonlinear Model Predictive Control*. Birkhäuser, Basel, 2000, S. 3–22. DOI: 10.1007/978-3-0348-8407-5\_1.
- [187] D. Mayne u. a. „Constrained model predictive control: Stability and optimality“. In: *Automatica* 36.6 (2000), S. 789–814. DOI: 10.1016/S0005-1098(99)00214-9.
- [188] L. Erickson und S. LaValle. „A Simple, but NP-Hard, Motion Planning Problem“. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 27.1 (2013), S. 1388–1393. DOI: 10.1609/aaai.v27i1.8545.
- [189] P. R. Chandler und M. Pachter. „Research issues in autonomous control of tactical UAVs“. In: *Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No.98CH36207)*. 1998, S. 394–398. DOI: 10.1109/ACC.1998.694698.
- [190] H. Choset und P. Pignon. „Coverage Path Planning: The Boustrophedon Cellular Decomposition“. In: *Field and Service Robotics*. London: Springer, 1998, S. 203–209. DOI: 10.1007/978-1-4471-1273-0\_32.



- [191] H. Choset. „Coverage for robotics – A survey of recent results“. In: *Annals of Mathematics and Artificial Intelligence* 31.1/4 (2001), S. 113–126. DOI: 10.1023/A:1016639210559.
- [192] NATO Standardization Agency, Hrsg. *STANAG 4586 (Edition 3) - Standard Interfaces of UAV Control System (UCS) for NATO UAV Interoperability*. 2012.
- [193] L. Fahrmeir u. a. *Statistik: Der Weg zur Datenanalyse*. 8. Aufl. Springer-Lehrbuch. Berlin und Heidelberg: Springer Spektrum, 2016. ISBN: 978-3-662-50371-3. DOI: 10.1007/978-3-662-50372-0.
- [194] P. Royston. „Approximating the Shapiro-Wilk W-test for non-normality“. In: *Statistics and Computing* 2.3 (1992), S. 117–119. DOI: 10.1007/BF01891203.
- [195] C. Ruf u. a. „Enhancing Automated Aerial Reconnaissance Onboard UAVs Using Sensor Data Processing-Characteristics and Pareto Front Optimization“. In: *AIAA Scitech 2019 Forum*. Reston, Virginia, 2019. DOI: 10.2514/6.2019-1541.
- [196] N. Liao u. a. „Joint mission planning and spectrum resources optimization for multi-UAV reconnaissance“. In: *IET Communications* 17.3 (2023), S. 324–335. DOI: 10.1049/cmu2.12541.