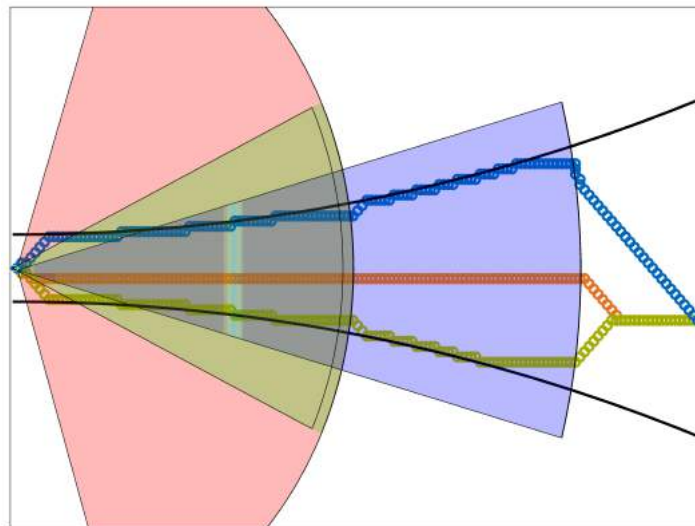


Automatische Generierung relevanter Szenarien für den Test automatisierter Fahrzeuge unter Berücksichtigung der Sensorabdeckung

Automatic generation of relevant scenarios for testing automated vehicles under consideration of the sensor coverage



Wissenschaftliche Arbeit zur Erlangung des Grades

Master of Science (M.Sc.)

an der Fakultät für Maschinenwesen der Technischen Universität München

Betreut von	Univ.-Prof. Dr.-Ing. Markus Lienkamp Thomas Ponn, M.Sc. Lehrstuhl für Fahrzeugtechnik
Eingereicht von	Thomas Lanz, B.Sc. Rolf-Pinegger-Straße 30 80689 München
Eingereicht am	Garching, den 11. Juni 2019

Aufgabenstellung

Automatische Generierung relevanter Szenarien für den Test automatisierter Fahrzeuge unter Berücksichtigung der Sensorabdeckung

Bei der Zertifizierung und Freigabe automatisiert fahrender Fahrzeuge muss aus der unendlich großen Anzahl an Verkehrsszenarien eine handhabbare und durchführbare Teilmenge ausgewählt werden. Damit eine ausreichende Aussagekraft dieser Tests gewährleistet ist, muss diese Teilmenge möglichst herausfordernd für das zu testende System sein. Eine Möglichkeit zur Erzeugung solch relevanter Szenarien besteht in der gezielten Vorgabe des Verhaltens der umgebenden Verkehrsteilnehmer, sodass ein möglichst schwieriges und komplexes Szenario entsteht. Zudem kann dieses Verhalten an die Sensorabdeckung des zu testenden Fahrzeugs adaptiert werden. So kann beispielsweise eine asymmetrische Sensorverteilung bei bestimmten Straßengeometrien (Kurve, Kuppe, . . .) dazu führen, dass Objekte erst spät vom Fahrzeug detektiert werden können. Umso später Objekte erkannt werden, desto weniger Reaktionszeit bleibt dem autonomen Fahrzeug für eine korrekte Planung und Ausführung einer sicheren Handlung übrig. Somit sind diese Szenarien für die Bewertung der Sicherheit von enormer Relevanz. Dieses Vorgehen soll im Rahmen der vorliegenden Arbeit am Use-case Autobahn durchgeführt und mittels eines numerischen Optimierungsverfahrens implementiert werden.

In einer theoretischen Masterarbeit sollen relevante Szenarien für die Zertifizierung automatisierter Fahrzeuge identifiziert werden. Dabei sollen das Sensorsetup und weitere Randbedingungen berücksichtigt werden.

Folgende Punkte sind durch Thomas Lanz zu bearbeiten:

- Einarbeitung in die szenariobasierte Absicherung von hochautomatisierten und autonomen Fahrzeugen, Einarbeitung in ein vorhandenes Sensormodell zur Darstellung der Sensorabdeckung sowie Einarbeitung in ein vorhandenes Modell zur Beschreibung der Bewegung beteiligter Verkehrsteilnehmer
- Kombination des Sensorsetups mit weiteren Randbedingungen wie Straßengeometrie, Wetterbedingungen und Beladungszustand zur Definition einer geeigneten Kostenfunktion.
- Implementierung und Vergleich zweier alternativer Optimierungsverfahren in Matlab zur Erstellung eines relativen Annäherungspfad
- Implementierung eines Verfahrens zur Erstellung relevanter Szenarien aus dem relativen Annäherungspfad
- Dokumentation der Vorgehensweise und der erzielten Ergebnisse

Die Ausarbeitung soll die einzelnen Arbeitsschritte in übersichtlicher Form dokumentieren. Der Kandidat verpflichtet sich, die Masterarbeit selbständig durchzuführen und die von ihm verwendeten wissenschaftlichen Hilfsmittel anzugeben.

Die eingereichte Arbeit verbleibt als Prüfungsunterlage im Eigentum des Lehrstuhls.

Ausgabe: 3. Dezember 2018

Abgabe: 11. Juni 2019

Univ.-Prof. Dr.-Ing. Markus Lienkamp

Thomas Ponn, M.Sc.

Geheimhaltungsverpflichtung

Herr/Frau: **Lanz, Thomas**

Gegenstand der Geheimhaltungsverpflichtung sind alle mündlichen, schriftlichen und digitalen Informationen und Materialien die der Unterzeichner vom Lehrstuhl oder von Dritten im Rahmen seiner Tätigkeit am Lehrstuhl erhält. Dazu zählen vor allem Daten, Simulationswerkzeuge und Programmcode sowie Informationen zu Projekten, Prototypen und Produkten.

Der Unterzeichner verpflichtet sich, alle derartigen Informationen und Unterlagen, die ihm während seiner Tätigkeit am Lehrstuhl für Fahrzeugtechnik zugänglich werden, strikt vertraulich zu behandeln.

Er verpflichtet sich insbesondere:

- derartige Informationen betriebsintern zum Zwecke der Diskussion nur dann zu verwenden, wenn ein ihm erteilter Auftrag dies erfordert,
- keine derartigen Informationen ohne die vorherige schriftliche Zustimmung des Betreuers an Dritte weiterzuleiten,
- ohne Zustimmung eines Mitarbeiters keine Fotografien, Zeichnungen oder sonstige Darstellungen von Prototypen oder technischen Unterlagen hierzu anzufertigen,
- auf Anforderung des Lehrstuhls für Fahrzeugtechnik oder unaufgefordert spätestens bei seinem Ausscheiden aus dem Lehrstuhl für Fahrzeugtechnik alle Dokumente und Datenträger, die derartige Informationen enthalten, an den Lehrstuhl für Fahrzeugtechnik zurückzugeben.

Eine besondere Sorgfalt gilt im Umgang mit digitalen Daten:

- Für den Dateiaustausch dürfen keine Dienste verwendet werden, bei denen die Daten über einen Server im Ausland geleitet oder gespeichert werden (Es dürfen nur Dienste des LRZ genutzt werden (Lehrstuhlaufwerke, Sync&Share, GigaMove).
- Vertrauliche Informationen dürfen nur in verschlüsselter Form per E-Mail versendet werden.
- Nachrichten des geschäftlichen E-Mail Kontos, die vertrauliche Informationen enthalten, dürfen nicht an einen externen E-Mail Anbieter weitergeleitet werden.
- Die Kommunikation sollte nach Möglichkeit über die (my)TUM-Mailadresse erfolgen.

Die Verpflichtung zur Geheimhaltung endet nicht mit dem Ausscheiden aus dem Lehrstuhl für Fahrzeugtechnik, sondern bleibt 5 Jahre nach dem Zeitpunkt des Ausscheidens in vollem Umfang bestehen. Die eingereichte schriftliche Ausarbeitung darf der Unterzeichner nach Bekanntgabe der Note frei veröffentlichen.

Der Unterzeichner willigt ein, dass die Inhalte seiner Studienarbeit in darauf aufbauenden Studienarbeiten und Dissertationen mit der nötigen Kennzeichnung verwendet werden dürfen.

Datum: 11. Juni 2019

Unterschrift: _____

Erklärung

Ich versichere hiermit, dass ich die von mir eingereichte Abschlussarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Garching, den 11. Juni 2019

Thomas Lanz, B. Sc.

Declaration of Consent, Open Source

Hereby I, Lanz, Thomas, born on February 1, 1995 , make the software I developed during my Master Thesis available to the Institute of Automotive Technology under the terms of the license below.

Garching, June 11, 2019

Thomas Lanz, B. Sc.

Copyright 2019 Lanz, Thomas

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Inhaltsverzeichnis

Abkürzungsverzeichnis	III
Formelzeichen	V
1 Einleitung und Motivation	1
1.1 Einleitung	1
1.2 Motivation	2
1.3 Inhalt und Aufbau	3
2 Stand der Technik	5
2.1 Definitionen im Kontext von Szenarien	5
2.2 Alternative Freigabemethoden	7
2.2.1 Anforderungen an ein Testkonzept	7
2.2.2 Formale Methoden	8
2.2.3 Szenariobasierter Ansatz.....	10
2.3 Kritik am Stand der Technik	15
3 Mathematische Optimierung	19
3.1 Optimierung formal.....	19
3.2 Gradientenbasierte Verfahren	20
3.3 Routenplanung und Graphen	21
3.3.1 Bestandteile von Graphen	21
3.3.2 Erstellung eines Beispielgraphens.....	23
3.4 Optimierung auf Graphen	23
3.4.1 Formale Definitionen zu Suchalgorithmen	24
3.4.2 Tiefensuche	24
3.4.3 Breitensuche	26
3.4.4 Dijkstra-Suchalgorithmus	27
3.4.5 Informierte Suche - Der A*-Algorithmus	30
3.4.6 Vergleich der Algorithmen.....	33
3.5 Evolutionäre Algorithmen	33
3.5.1 Ameisen-Kolonie-Optimierung	34

3.5.2 Varianten und Verbesserungen der ACO	37
4 Vorgehen	41
4.1 Definition eines Minimalbeispiels	41
4.2 Entwicklung des A*-Algorithmus	42
4.3 Entwicklung des ACO-Algorithmus	45
4.4 Modellanpassungen und Randbedingungen	48
4.4.1 Modellanpassungen	49
4.4.2 Randbedingungen der Optimierung	49
4.4.3 Anpassungen der Algorithmen.....	52
4.5 Algorithmus zur Szenarioüberführung	53
4.5.1 Randbedingungen und Annahmen	53
4.5.2 Algorithmus zur Überführung in ein Basisszenario	55
5 Ergebnisse und Diskussion der Pfadfindungsalgorithmen.....	59
5.1 A*-Algorithmus.....	59
5.1.1 Ausgangsparametrierung	59
5.1.2 Variation innerhalb Sensorabdeckung.....	62
5.1.3 Variation außerhalb Sensorabdeckung	64
5.1.4 Diskussion des A*-Suchalgorithmus	67
5.2 ACO-Algorithmus	69
5.2.1 Ergebnisse der Ausgangsparametrierung.....	69
5.2.2 Ergebnisse der optimierten Parameter	70
5.2.3 Diskussion des ACO-Algorithmus.....	73
5.3 Vergleich der Algorithmen.....	74
6 Ergebnisse und Diskussion der Szenarioüberführung	77
6.1 Ergebnisse der Szenarioüberführung.....	77
6.2 Diskussion der Szenarioüberführung	80
7 Zusammenfassung und Ausblick	81
7.1 Zusammenfassung	81
7.2 Ausblick	84
Abbildungsverzeichnis.....	i
Tabellenverzeichnis	iii
Literaturverzeichnis	v
Anhang	ix

Abkürzungsverzeichnis

ACC	Adaptive Cruise Control
ACO	Ant Colony Optimization
ACS	Ant Colony System
AS	Ant System
AS _{rank}	rank-based Ant System
BAS _t	Bundesanstalt für Straßenwesen
EA	Evolutionäre Algorithmen
Ego-Fzg	Ego-Fahrzeug
EKA	Entwurfsklassen für Autobahnen
FAS	Fahrerassistenzsystem
HAF	Hochautomatisierte Fahrfunktion
Ko-HAF	Kooperatives hochautomatisiertes Fahren
LKA	Lane Keeping Assist
PEGASUS	Projekt zur Etablierung von generell akzeptierten Gütekriterien, Werkzeugen und Methoden sowie Szenarien und Situationen zur Freigabe hochautomatisierter Fahrfunktionen
PFS	Passive Friendly Safety
PRPR	Pseudo-Random-Proportional Rule
PS	Passive Safety
PSO	Particle Swarm Optimization
RAA	Richtlinie zur Anlage von Autobahnen
RPR	Random-Proportional Rule
RQ	Regelquerschnitt
RSS	Responsibility Sensitive Safety
TET	Time Exposed Time-To-Collision
TIT	Time Integrated Time-To-Collision Indicator
TSP	Traveling Salesman Problem
TTB	Time-To-Brake
TTC	Time-To-Collision
TTK	Time-To-Kickdown
TTR	Time-To-React
TTS	Time-To-Steer
TTX	Time-To-X

UNECE	United Nations Economic Commission for Europe
VDA	Verband der Automobilindustrie
WD	Wirkungsdimension

Formelzeichen

Formelzeichen	Einheit	Beschreibung
a	-	Ameise aus A
a_y	$\frac{\text{m}}{\text{s}^2}$	Querbeschleunigung
$a_{y,\text{max}}$	$\frac{\text{m}}{\text{s}^2}$	Maximale Querbeschleunigung
A	-	Menge der Ameisen
c	-	Konstante des Pfadwerts
d_{ij}	-	Länge der Kante ij
d_{jz}	-	Distanz zwischen Knoten v_j und Zielknoten v_z
$d(v_i, v_{i-1})$	m	Euklidische Distanz zwischen den Knoten v_i und v_{i-1}
$d_L(v_N, v_i)$	m	Luftlinienentfernung zwischen den Knoten v_N und v_i
e	-	Allgemeine Kantenbezeichnung
E	-	Menge der Kanten
H_K	m	Kuppenmindesthalbmesser
H_W	m	Wannenmindesthalbmesser
k_K	-	Konstante der Kostenfunktion
K	-	Kosten nach Kostenfunktion
K_{min}	-	Minimale Kosten nach Kostenfunktion
L_a	m	Pfadlänge der Ameise a
L_{best}	m	Pfadlänge der besten Ameise
L_m	m	Pfadlänge der m -besten Ameise
L_p	m	Pfadlänge
L_μ	m	Pfadlänge der μ -besten Ameise
m_{best}	-	Anzahl betrachteter bester Ameisen
n_a	-	Anzahl Ameisen
n_{Gen}	-	Anzahl Generationen
N	-	Menge der Knoten der Nachbarschaft

P_D	%	Erfassungswahrscheinlichkeit
$P_{D,i}$	%	Erfassungswahrscheinlichkeit des Knotens v_i
$P_{D,N}$	%	Erfassungswahrscheinlichkeit des Knotens v_N
$P_{D,value,path}$	-	Wert des Pfades
P_a	-	Pfad der Ameise a
P_{best}	-	Pfad der besten Ameise
P_m	-	Pfad der m -besten Ameise
P_μ	-	Pfad der μ -besten Ameise nach AS_{rank}
p_{ij}^a	%	Wahrscheinlichkeit des Übergangs von Knoten v_i zu Knoten v_j der Ameise a
q	%	Straßensteigung
q_{max}	%	Maximale Straßensteigung
q	-	Zufallszahl $\in [0, 1]$
q_0	-	Auswahlgrenze $\in [0, 1]$
Q	-	Pheromonablagewert
R	m	Kurvenradius
R_{min}	m	Kurvenmindestradius
dr	m	Inkrement in radialer Richtung
S_{AS}	-	Auswahlstrategie des AS-Algorithmus
$S_{AntStar}$	-	Auswahlstrategie des AntStar-Algorithmus
T_b	s	Eigene Bremsdauer
T_{ob}	s	Maximale Bremsdauer der umgebenden Objekte
dt	s	Zeitschrittweite
Δt	s	Zeitdauer
v	$\frac{km}{h}$	Geschwindigkeit
v	-	Allgemeine Knotenbezeichnung
v_{ch}	$\frac{km}{h}$	Geschwindigkeit des Challengers
v_{ego}	$\frac{km}{h}$	Geschwindigkeit des Ego-Fahrzeuges
Δv	$\frac{km}{h}$	Relativgeschwindigkeit
v_N	-	Nachbarknoten
v_S	-	Startknoten
v_Z	-	Zielknoten
V	-	Menge der Knoten

w	m	Fahrstreifenbreite
w_{St}	m	Fahrbahnbreite
x	m	Koordinate der longitudinalen Position oder Abstandes der Fahrzeuge
x_{max}	m	Obere longitudinale Begrenzung
x_{min}	m	Untere longitudinale Begrenzung
dx	m	Longitudinales Streckeninkrement
$\mathbf{x}_{\text{glob}}$	m	Koordinaten des globalen Koordinatensystems
$\mathbf{x}_{\text{rel}}$	m	Koordinaten des relativen Koordinatensystems
$\mathbf{x}_{\text{Ch,glob}}$	m	Aktuelle globale Position des Challengers
$\mathbf{x}_{\text{Ch,rel}}$	m	Aktuelle relative Position des Challengers
$\mathbf{x}_{\text{Ch,ref}}$	m	Positionsvorgabe des Challengers
$\Delta \mathbf{x}_{\text{Ch}}$	m	Positionsänderung des Challengers
$\Delta \mathbf{x}_{\text{Ch,soll}}$	m	Vorgabe der Positionsänderung des Challengers
$\mathbf{x}_{\text{Ego,glob}}$	m	Aktuelle globale Position des Ego-Fahrzeuges
$\mathbf{x}_{\text{Ego,rel}}$	m	Aktuelle relative Position des Ego-Fahrzeuges
$\mathbf{x}_{\text{Ego,ref}}$	m	Positionsvorgabe des Ego-Fahrzeuges
$\Delta \mathbf{x}_{\text{Ego}}$	m	Positionsänderung des Ego-Fahrzeuges
$\Delta \mathbf{x}_{\text{Ego,soll}}$	m	Vorgabe der Positionsänderung des Ego-Fahrzeuges
$\Delta \mathbf{x}_{\text{Path}}$	m	Änderung des relativen Pfades
$\mathbf{x}_{\text{M,y}}$	m	Koordinaten des Mittelpunktes der lateralen Begrenzungskreise
$\mathbf{x}_{\text{M,z}}$	m	Koordinaten des Mittelpunktes der vertikalen Begrenzungskugeln
$\mathbf{x}_{\text{Ref}}$	m	Position des nächsten Referenzknotens
$\mathbf{x}_{\text{S}}$	-	Koordinaten des Startknotens
$\mathbf{x}_{\text{Z}}$	-	Koordinaten des Zielknotens
y	m	Koordinate der lateralen Position oder Abstandes der Fahrzeuge
dy	m	Laterales Streckeninkrement
y_{max}	m	Obere laterale Begrenzung
y_{min}	m	Untere laterale Begrenzung
Δy_{max}	m	Maximaler Querversatz
z	m	Koordinate der vertikalen Position oder Abstandes der Fahrzeuge
dz	m	Vertikales Streckeninkrement
z_{max}	m	Obere vertikale Begrenzung
z_{min}	m	Untere vertikale Begrenzung

α	-	Pheromoneinfluss
β	-	Attraktivitätseinfluss
ϵ	m	Minimalabstand
η_{ij}	-	Attraktivität der Kante ij
θ	°	Nickwinkel
θ_{Ch}	°	Nickwinkel des Challengers
$\Delta\theta_{Ch}$	$\frac{°}{s}$	Nickwinkeländerung des Challengers
θ_{Ego}	°	Nickwinkel des Ego-Fahrzeugs
$\Delta\theta_{Ego}$	$\frac{°}{s}$	Nickwinkeländerung des Ego-Fahrzeugs
θ_{max}	°	Maximaler Nickwinkel
$\dot{\theta}$	$\frac{°}{s}$	Nickrate
$\dot{\theta}_{max}$	$\frac{°}{s}$	Maximale Nickrate
$d\vartheta$	°	Inkrement der Elevation
μ	-	Rang der Ameise nach AS_{rank}
ρ	-	Pheromonevaporationskoeffizient $\in (0, 1]$
σ	-	Verstärkung der besten Ameisen
τ_0	-	Initiale Pheromonmenge
τ_{ij}	-	Pheromonmenge der Kante ij
τ_j	-	Pheromonmenge des Knotens v_j
$\Delta\tau_{ij}$	-	Pheromonzuwachs der Kante ij
$\Delta\tau_j$	-	Pheromonzuwachs des Knotens v_j
$\Delta\tau_{ij}^a$	-	Pheromonablagemenge der Ameise a für Kante ij
$\Delta\tau_{ij}^{best}$	-	Pheromonablagemenge der m_{best} besten Ameisen für Kante ij nach elitärer Strategie
$\Delta\tau_j^m$	-	Pheromonablagemenge der m -besten Ameise für Knoten v_j
$\Delta\tau_{ij}^\mu$	-	Pheromonablagemenge der m -besten Ameise für Kante ij nach AS_{rank}
$d\varphi$	°	Inkrement des Azimuthwinkels
Ψ	°	Gierwinkel
Ψ_{Ego}	°	Gierwinkel des Ego-Fahrzeugs
$\Delta\Psi_{Ego}$	$\frac{°}{s}$	Gierwinkeländerung des Ego-Fahrzeugs
Ψ_{Ch}	°	Gierwinkel des Challengers
$\Delta\Psi_{Ch}$	$\frac{°}{s}$	Gierwinkeländerung des Challengers
Ψ_{max}	°	Maximaler Gierwinkel

$\Psi_{\text{max,Ch}}$	°	Maximaler Gierwinkel des Challengers
$\dot{\Psi}$	$\frac{^\circ}{\text{s}}$	Gierrate
$\dot{\Psi}_{\text{max}}$	$\frac{^\circ}{\text{s}}$	Maximale Gierrate
ω	-	Anzahl betrachteter bester Ameisen nach AS_{rank}

1 Einleitung und Motivation

Zu Beginn wird das Themenfeld der vorliegenden Arbeit präsentiert (Abschnitt 1.1), die Motivation aufgezeigt (Abschnitt 1.2) und der Inhalt vorgestellt (Abschnitt 1.3).

1.1 Einleitung

Digitalisierung und Elektromobilität werden von Dr. Ulrich Eichhorn, Geschäftsführer des Verbandes der Automobilindustrie (VDA), als Innovationstreiber für das Auto der Zukunft, ein elektrisches, vernetztes und automatisiertes Fahrzeug, identifiziert. Weiterhin kündigt er die Einführung des automatisierten Fahrens innerhalb eines zeitlichen Horizonts von zehn bis 15 Jahren an [1]. Den ersten Schritt in diese Richtung hat Audi im Jahr 2017 mit dem „Audi AI Staupilot“ getätigt [2], der nach Definition der Bundesanstalt für Straßenwesen (BASt) als „hochautomatisiert“ [3] klassifiziert wird. Dieser ist auf Autobahnen in Stau und Kolonnenverkehr bei einer Geschwindigkeit bis $60 \frac{\text{km}}{\text{h}}$ in der Lage, die Fahraufgabe komplett ohne menschliche Unterstützung zu übernehmen [2].

Hochautomatisierte Fahrfunktionen (HAF) stellen eine Erweiterung der aktuell erhältlichen und freigegebenen Fahrerassistenzsysteme (FAS), wie Abstandsregeltempomaten (ACC, engl.: Adaptive Cruise Control) und Fahrstreifenhaltesysteme (LKA, engl.: Lane Keeping Assist), dar. Diese sind aktuell bei nahezu jedem Automobilhersteller erhältlich und sind demnach zulassungsfähig. Im Kontrast dazu existieren aktuell keine Vorschriften oder Leitfäden zur Zulassung und Erprobung von Systemen, die über Stufe zwei „teilautomatisiert“ der BASt-Definition [3] hinausgehen [4, S.10]. Innerhalb der zur Typgenehmigung verpflichtenden „Einheitliche[n] Bedingungen für die Genehmigung der Fahrzeuge hinsichtlich der Lenkanlage“ der „Regelung Nr. 79 der Wirtschaftskommission der Vereinten Nationen für Europa (UN/ECE)“ [5, S.25] Revision 2 wird ausdrücklich die Gültigkeit für autonome Lenkanlagen ausgeschlossen [5, S.26]. Diese sind definiert als „[...] eine Anlage mit einer Funktion in einem komplexen elektronischen Steuersystem, die bewirkt, dass das Fahrzeug einer festgelegten Fahrspur folgt [...]“ [5, S.28], wobei „[d]er Fahrzeugführer [...] dabei nicht unbedingt die Hauptverantwortung für das Führen des Fahrzeugs“ [5, S.28] hat. Einzige Ausnahme bilden automatische Lenkfunktionen bis zu einem Grenzwert von $10 \frac{\text{km}}{\text{h}}$ [5, S.33].

Um diesen Umstand zu ändern, ist die Überarbeitung dieser Vorschrift bis 2019 das Ziel einer Arbeitsgruppe der United Nations Economic Commission for Europe (UNECE). Anschließend soll ein Leitfaden für Prüfdienste und Hersteller vorliegen, wodurch ab 2020 HAF homologationsfähig sein sollen [4, S.11]. Sobald diese Änderungen vollzogen sind, muss die Automobilindustrie in der Lage sein, nachzuweisen, dass ihre Fahrfunktionen (ausreichend) sicher sind.

1.2 Motivation

In der Automobilindustrie wird üblicherweise das V-Modell zur Entwicklung und Absicherung von Funktionen und Baugruppen verwendet [6, S.70]. Das V-Modell, beziehungsweise in seiner aktuellen Form V-ModellXT, „ist ein Vorgehensmodell zum Planen und Durchführen von Systementwicklungsprojekten“ [7, S.4]. Die grundsätzliche Idee des V-Modells ist die sukzessive Identifikation von Anforderungen mit nachfolgender Aufspaltung der Aufgabe in kleinere Teilsysteme, wobei jeder Dekomposition ein Integrations- und Prüfschritt zugeordnet wird [7, S.5]. Konventionellerweise resultiert daraus die Überprüfung von Fahrzeugen durch Tests der Hardware in definierten Szenarien. Dieser Ansatz zur Absicherung wird „Safety by Test“ genannt [8, S.171]. Durch stochastische Felderprobung und Absicherung von zusätzlichen, relevanten Szenarien wird nachgewiesen, dass das FAS den Fahrer bei der Fahraufgabe unterstützt und keine zusätzlichen kritischen Situationen in das System bringt. Der Gedanke ist, dass nach ausreichend zurückgelegter Strecke in der Erprobung alle Ereignisse und Kombinationen der Wirkungsdimensionen (WD), z.B. Geschwindigkeit und Abstand zum Vorderfahrzeug, repräsentativ abgebildet sind [8, S.171]. Aufgrund der hohen Anzahl gefahrener Kilometer in Kombination mit der Überprüfung kritischer Einzelszenarien gilt der Nachweis über die Sicherheit und Kontrollierbarkeit des Systems durch den Menschen als erbracht [9, S.444]. Dieses Verfahren wird durch die rote Linie innerhalb der WD in Abb. 1.1 symbolisiert. Gefährdungen, die zu Unfällen geführt haben, sind durch rote Kreuze repräsentiert. Grüne Kreuze stellen Gefährdungen dar, die im Rahmen der Erprobung nicht aufgedeckt wurden und somit keinen Einfluss auf die Sicherheitsbewertung des Systems haben. Für FAS ist dies ein sinnvoller Ansatz, weil der Fahrer die Verantwortung über das System trägt und dieses dauerhaft zu kontrollieren hat. Die übersehene Gefährdung entschärft der Fahrer aufgrund seiner nachgewiesenen Fahrtauglichkeit und kontrolliert die Situation wie bisher [8, S.171].

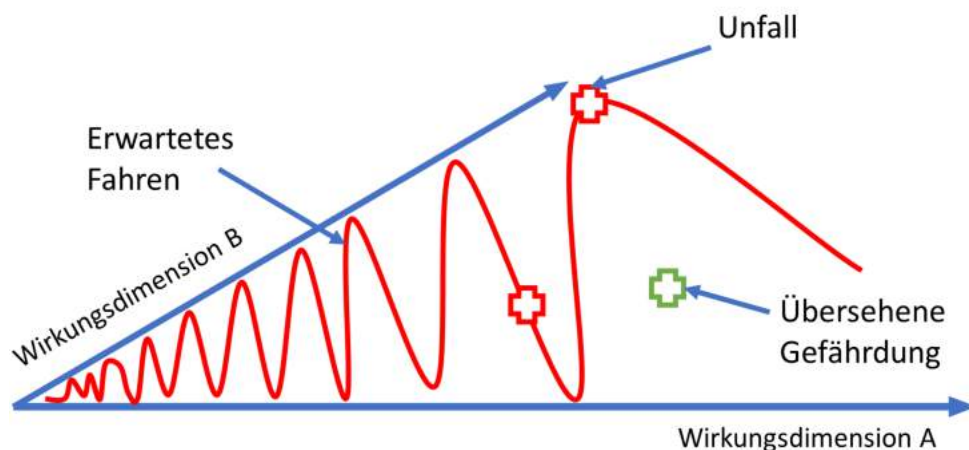


Abbildung 1.1: Absicherungsansatz Stochastische Felderprobung nach MAZZEGA et al. [8, S.172]

Im Gegensatz zu FAS übernehmen HAF die Fahraufgabe und Verantwortung innerhalb der vorgegebenen Systemgrenzen vollständig. Wird von diesem System die Gefährdung nicht erkannt, existiert keine Rückfallebene, die diesen Fehler kompensieren kann [8, S.172]. Das akzeptable Beherrschen entdeckter Gefährdungen kann nachgewiesen werden, wie soll aber die Bewertung der Systemsicherheit in einer nicht entdeckten Situation durchgeführt werden?

Theoretisch kann eine Lösung die Erhöhung der Testkilometer während der Erprobung sein, um eine höhere Durchdringung und Abdeckung der Gefahrensituationen zu erreichen. Dieses Vorgehen ist bei einer überschaubaren Anzahl kritischer Szenarien sinnvoll, wie sie bei FAS vorliegen, bei einer sehr großen Anzahl durchzuführender Versuche jedoch ökonomisch nicht mehr akzeptabel. Der Testraum autonomer Fahrzeuge ergibt sich durch die Kombination unterschiedlichster Umwelt-, Fahrzeug- und Verkehrsbedingungen zu einer unendlichen Anzahl an Testfällen. Diese müssen durch das System gänzlich und sicher kontrolliert werden. Alle Testfälle einzeln abzu prüfen ist jedoch monetär und zeitlich nicht möglich. Bereits jetzt wird enormer Aufwand betrieben, um den Nachweis sicherer Funktionen zu liefern [9, S.446]. Beispielsweise wurden für die Erprobung der 2009 zugelassenen Mercedes E-Klasse, einem konventionellen Fahrzeug ohne hochautomatisierte Fahrfunktionen, „umfassende virtuelle Tests mit digitalen Prototypen und insgesamt 36 Millionen Testkilometer“ durchgeführt [10, S.2]. Für den statistischen Nachweis, dass das Risiko durch einen hochautomatisierten Autobahnpiloten gegenüber menschlichem Fahren nicht steigt, muss man bei Festhalten am bisherigen Testkonzept eine Fahrdistanz von 6,62 Milliarden Testkilometern zurücklegen [9, S.454-458]. Diese Problematik bezeichnet WINNER als „Freigabefalle“ [11, S.31].

Für das V-Modell ergeben sich damit drei eindeutige Probleme, die ein Anwenden der Methodik unzulässig machen. Es können zu Beginn der Entwicklung keine Anforderungen definiert werden, die das System zu erfüllen hat, um als ausreichend sicher und zuverlässig eingestuft zu werden. Außerdem ergeben die Kombination unterschiedlicher Werte der Fahrzeug-, Verkehrs- und Umweltparameter einen unendlichen Zustands- und Ereignisraum, der nicht durch die Definition spezifischer Anforderungen abgedeckt werden kann. Dem entwickelten System kann zusätzlich keine valide und zuverlässige Absicherung gegenübergestellt werden, welche die Erfüllung der eingangs definierten Anforderungen sicherstellt [11, S.32]. Infolgedessen ist es nötig, den Raum der zu testenden Szenarien zu beschränken, um dennoch eine zuverlässige Aussage über Robustheit und Fehlertoleranz des Systems innerhalb geltender Normen und Gesetze mit vertretbarem Aufwand treffen zu können [8, S.173].

1.3 Inhalt und Aufbau

Aktuell findet die Diskussion und Erforschung alternativer Ansätze zur Absicherung und Reduzierung der Testfälle während der Erprobung hochautomatisierter Fahrzeuge eine hohe Beachtung in Forschung und Industrie, wie z.B. die Projekte PEGASUS und Ko-HAF zeigen (PEGASUS: Projekt zur Etablierung von generell akzeptierten Gütekriterien, Werkzeugen und Methoden sowie Szenarien und Situationen zur Freigabe hochautomatisierter Fahrfunktionen [8, S.163], Ko-HAF: Kooperatives hochautomatisiertes Fahren [12]). Diese Arbeit widmet sich der Entwicklung einer Methodik, die eine systemspezifische Auswahl der Erprobungsszenarien vornimmt. Ziel ist es, die angesprochene nötige Raffung der Testfälle durch technische Kriterien zu untermauern. Auf Basis eines bestehenden Sensormodells wird ein möglichst herausfordernder Pfad individuell für einzelne Systeme automatisch generiert. Dieser Pfad wird weiterentwickelt zu einem Basisszenario, das den Positionsverlauf zweier Fahrzeuge beinhaltet. Der Fokus liegt dabei nicht nur auf dem Beweis der Machbarkeit einer solchen Methodik, sondern auch auf einer realitätsnahen und -konformen Umsetzung durch Berücksichtigung entsprechender Randbedingungen. Die Methodik soll ein vollständiges Verfahren ausgehend von der Beschreibung der Sensorik bis hin zu fertigen Fahrzeugpositionen abdecken.

Die vorliegende Arbeit gliedert sich in sieben Kapitel (Abb. 1.2). Im Anschluss an die Einleitung (Kapitel 1) folgt die Vorstellung des gegenwärtig relevanten Standes der Technik und Forschung (Kapitel 2). Nach der Definition von Begrifflichkeiten des Szenarios im automobilen Kontext (Abschnitt 2.1) liegt der Fokus auf der Vorstellung alternativer Freigabemethoden (Abschnitt 2.2). Anschließend erfolgt die Ableitung der Themenstellung der Arbeit aus dem identifizierten Handlungsbedarf des Standes der Technik (Abschnitt 2.3).

Für die Erstellung des Pfades werden Methoden aus dem Bereich der mathematischen Optimierung benötigt. Diese werden in Kapitel 3 präsentiert, wobei auf formale Grundlagen der Optimierung (Abschnitt 3.1), gradientenbasierte Verfahren (Abschnitt 3.2), Grundlagen von Graphen und Routenplanung (Abschnitt 3.3), graphenbasierte Optimierung (Abschnitt 3.4) und evolutionäre Algorithmen (Abschnitt 3.5) eingegangen wird. Besonderer Fokus wird dabei auf Abschnitt 3.4.5 und Abschnitt 3.5.1 gelegt, die sich mit den verwendeten Algorithmen befassen. In Kapitel 4 wird deren Entwicklung vorgestellt. Dafür wird in Abschnitt 4.1 zunächst ein Minimalbeispiel definiert, anhand dessen die Entwicklung der Algorithmen A* (Abschnitt 4.2) und ACO (Abschnitt 4.3) erfolgt. Abschnitt 4.4 erklärt Modellerweiterungen und nötige Randbedingungen der Optimierung. Das jeweilige Ergebnis der beiden Pfadfindungsalgorithmen ist die Ausgangsbasis des Algorithmus, der aus dem relativen Annäherungspfad ein Szenario ableitet (Abschnitt 4.5).

Kapitel 5 präsentiert, diskutiert und vergleicht die Ergebnisse der Pfadfindungsalgorithmen. Die Ergebnisse des A*-Algorithmus (Abschnitt 5.1) umfassen die Ausgangsparametrierung (Abschnitt 5.1.1), eine Variation mit Startpunkt innerhalb der Sensorabdeckung (Abschnitt 5.1.2) und eine mit Startpunkt außerhalb der Sensorabdeckung (Abschnitt 5.1.3), sowie die Diskussion (Abschnitt 5.1.4). Anschließend werden die Ergebnisse des ACO-Algorithmus (Abschnitt 5.2) für initiale (Abschnitt 5.2.1) und optimierte (Abschnitt 5.2.2) Parameter präsentiert und diskutiert (Abschnitt 5.2.3). Den Abschluss des Kapitels bildet der Vergleich der Algorithmen inklusive der Auswahl des besser geeigneten (Abschnitt 5.3).

Kapitel 6 beinhaltet die Ergebnisse der Szenarioüberführung (Abschnitt 6.1) und deren Diskussion (Abschnitt 6.2). Zuletzt wird in Kapitel 7 das Fazit der Arbeit (Abschnitt 7.1) gezogen und ausblickend die Bedeutung und Möglichkeiten des neu entwickelten Ansatzes sowie die Ableitung zukünftiger Forschungsfelder betrachtet (Abschnitt 7.2).

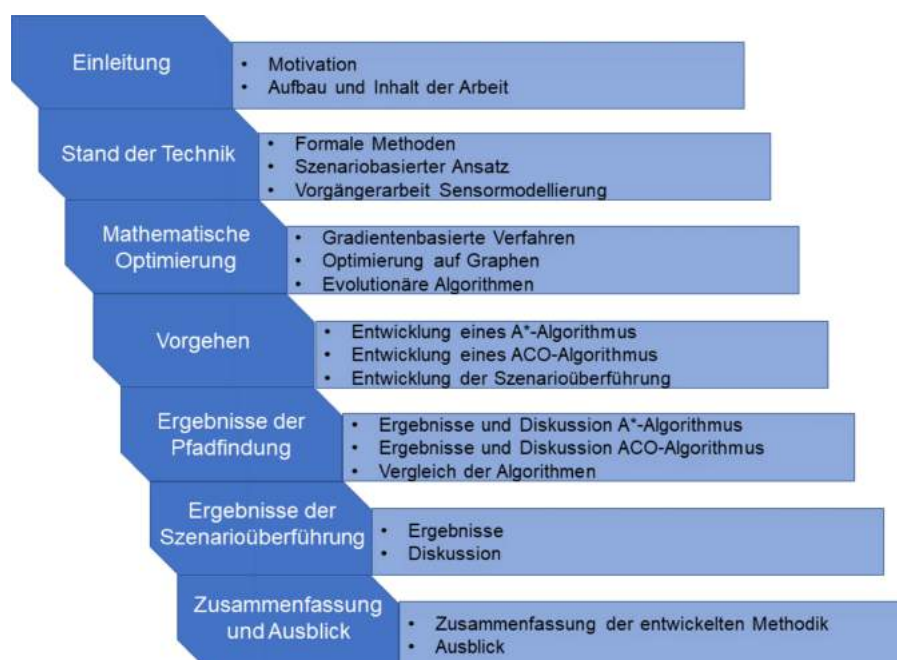


Abbildung 1.2: Inhalt und Aufbau der Arbeit

2 Stand der Technik

Dieses Kapitel beschäftigt sich mit der Vorstellung aktueller Herangehensweisen an das Lösen der Freigabefälle. Zu Beginn wird dafür eine einheitliche Basis an Begrifflichkeiten definiert (Abschnitt 2.1), anschließend werden in Abschnitt 2.2 die zu erfüllenden Anforderungen eines Testkonzeptes angeführt (Abschnitt 2.2.1) und die beiden maßgeblichen Ansätze, die formalen Methoden (Abschnitt 2.2.2) und der szenariobasierte Ansatz (Abschnitt 2.2.3), vorgestellt. Abgeschlossen wird das Kapitel mit einem Fazit und der Ableitung der bearbeiteten Fragestellung der Arbeit (Abschnitt 2.3).

2.1 Definitionen im Kontext von Szenarien

Im Vorfeld der Präsentation von Ansätzen zum Auffinden valider Testkonzepte werden zur Generierung eines einheitlichen Verständnisses einige Definitionen häufig verwendeter Begriffe eingeführt. Im Kontext des automatisierten Fahrens findet für die Begriffe „Szene“ , „Situation“ und „Szenario“ die Definition von ULBRICH et al. [13] Verwendung.

Eine Szene beschreibt eine Momentaufnahme des Umfelds, welche die Szenerie, dynamische Elemente, die Selbstrepräsentation aller Akteure und Beobachter wie auch die Verknüpfung dieser Entitäten umfasst. Einzig eine Szenenrepräsentation in einer simulierten Welt kann allumfassend sein (objektive Szene, Ground Truth). In der realen Welt ist sie immer unvollständig, fehlerbehaftet, unsicherheitsbehaftet und aus der Perspektive eines oder mehrerer Beobachter (subjektive Szene) [13, S.106].

Eine Szene ist damit die stationäre Aufnahme eines spezifischen Moments inklusive Eintragungen zusätzlicher Informationen. Darauf aufbauend wird die Situation definiert:

Eine Situation beschreibt die Gesamtheit der Umstände, die für die Auswahl geeigneter Verhaltensmuster zu einem bestimmten Zeitpunkt zu berücksichtigen sind. Sie umfasst alle relevanten Bedingungen, Möglichkeiten und Determinanten von Handlungen. Eine Situation wird aus der Szene durch einen Prozess der Informationsauswahl und -augmentierung abgeleitet, basierend auf transienten (z.B. missionsspezifischen) wie auch permanenten Zielen und Werten. Folglich ist eine Situation immer subjektiv, indem sie die Sicht eines Elements repräsentiert [13, S.110].

Eine Situation erfasst die Reduktion einer Szene auf die relevanten funktionsspezifischen Aspekte und Informationen, die zur Erfüllung der zu erreichenden Ziele und Werte benötigt werden, inklusive der Erweiterung um diese Ziele und Werte. Ein Beispiel für eine Situation ist die Beschreibung der Verkehrsteilnehmer, die für das Ziel „Fahrstreifenwechsel links“ relevant sind [13, S.111]. Die verwendeten Definitionen von Szene und Situation haben gemein, dass beide als Momentaufnahme definiert sind, wodurch keine Bestimmung von Beginn und Ende benötigt wird [13, S.109]. Abschließend wird die Definition des Szenarios eingeführt:

Ein Szenario beschreibt die zeitliche Entwicklung von Szenenelementen innerhalb einer Folge von Szenen, welche mit einer Startszene beginnt. Aktionen und Ereignisse ebenso wie Ziele und Werte können spezifiziert werden, um diese zeitliche Entwicklung in einem Szenario festzulegen. Im Gegensatz zu Szenen decken Szenarien eine gewisse Zeitspanne ab [13, S.114].

Ein Szenario entsteht aus der sequentiellen Abfolge eindeutig definierter Szenen oder aus der Verknüpfung von Szenen durch Aktionen und Ereignisse. Eine beispielhafte Darstellung eines Szenarios zeigt Abb. 2.1. Das Szenario beginnt mit der Startszene ganz links und wird durch die Angabe von Aktionen („Blinken“, „Fahrstreifenwechsel“, „Fahrstreifen folgen“) in neue Szenen bis zur Endszene ganz rechts umgesetzt [13, S.115].

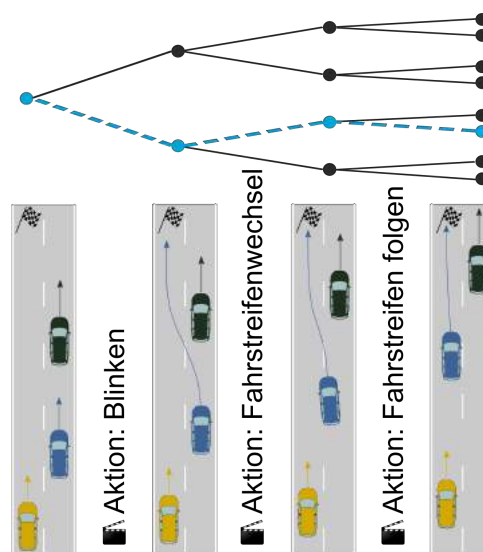


Abbildung 2.1: Darstellung einer Szenariorepräsentation nach ULBRICH et al. [13, S.115]

Nach BAGSCHIK et. al [14] liegen mehrere Konflikte zwischen den Anforderungen zur Definition von Szenarien vor [14, S.131]. Einerseits müssen die Szenarien in einer abstrakten, für den Menschen verständlichen Sprache, andererseits in einer effizient maschinenlesbaren Repräsentation vorliegen. Zustandsgrößen des Szenarios können in Wertbereichen definiert werden, entgegen der Forderung eines konkreten Testwerts in Prüfverfahren [14, S.131].

Diese Konflikte werden durch eine weitere Unterteilung des Begriffs Szenario in funktionale, logische und konkrete Szenarien adressiert. Der Begriff *funktionales Szenario* fasst eine in menschlicher Sprache ausformulierte Beschreibung der Entitäten und deren Beziehungen in für den Anwendungsfall und die -domäne spezifischem Vokabular [14, S.131]. *Logische Szenarien* identifizieren die Parameter der funktionalen Szenarien und weisen ihnen Parameterbereiche, Verteilungen oder Modellierungen in formaler Beschreibung zu [14, S.133]. Die Zuweisung spezifischer Werte und damit die Generierung abprüfbarer Einzelfälle findet in *konkreten Szenarien* statt [14, S.133]. Die Formulierung in funktionale Szenarien liefert eine verständliche und überschaubare Menge an Szenarien auf hohem Abstraktionsgrad. Durch die kontinuierliche Ausgestaltung zu logischen und anschließend konkreten Szenarien wird daraus eine hohe Anzahl an Einzelszenarien generiert. Diese Beziehung veranschaulicht Abb. 2.2 anhand eines Beispielszenarios.

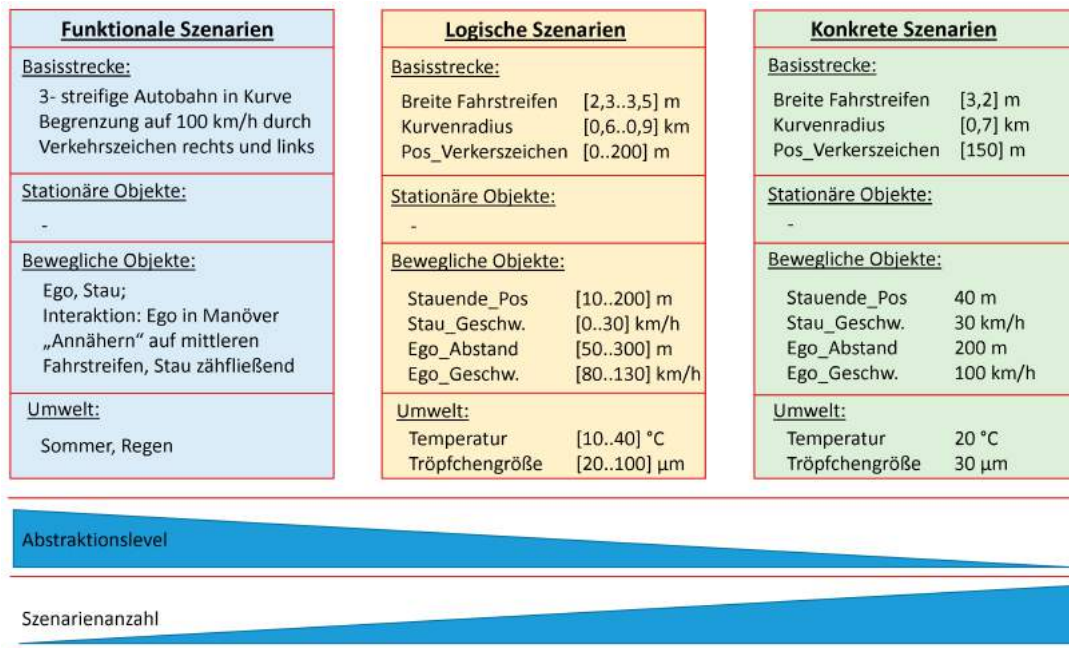


Abbildung 2.2: Abstraktionsebenen von Szenarien nach BAGSCHIK [14, S.132]

2.2 Alternative Freigabemethoden

Die bisherige Variante der Absicherung und damit erste Möglichkeit, HAF zu erproben, ist ungeeignet, sodass Alternativen benötigt werden. Aktuell existieren zwei Ansätze, um die Freigabefälle von HAF zu lösen. Ziel der *formalen Methoden* ist der Nachweis der Sicherheit eines Systems durch dessen Auslegung, indem die Sicherheit des Konzepts mathematisch bewiesen wird [9, S.460]. Der *szenariobasierte Ansatz* verwendet eine Reduktion des Testaufwandes durch Beschränkung des Testraumes auf ausgewählte Szenarien, um somit eine repräsentative aber ökonomisch beherrschbare Testabdeckung zu erhalten. WACHENFELD und WINNER [9, S.448] haben Anforderungen an ein Testkonzept definiert, die an dieser Stelle erklärt werden, um einen Vergleich der unterschiedlichen Methoden durchzuführen.

2.2.1 Anforderungen an ein Testkonzept

Ein Testkonzept muss nach WACHENFELD und WINNER [9, S.448] zweierlei Arten von Kriterien erfüllen, um eine Aussage über ein Bewertungsobjekt treffen zu können. Diese sind die Effektivitäts- und Effizienzkriterien.

- Effektivitätskriterien
 - Repräsentativ - valide: Ausreichende Testabdeckung und Realitätsnähe
 - Variierbar: Umsetzung aller von der Testfallgenerierung definierten Testfälle
 - Beobachtbar: Analyse der Parameter zur Bewertung des Bestehens des Tests
- Effizienzkriterien
 - Ökonomisch: Schnelle und kostengünstige Testdurchführung
 - Reproduzierbar: Identische Tests bei verändertem Modell durchführbar
 - Frühzeitig: Test früh im Entwicklungsprozess möglich
 - Sicher: Reduktion des Risikos auf annehmbares Maß

Mithilfe dieser Kriterien lässt sich die eingangs formulierte und argumentierte These, dass das aktuelle Testkonzept nicht für HAF übertragbar sei, formalisieren. Keines der Effizienzkriterien kann eingehalten werden. Das Befahren von Milliarden Kilometern liegt weit außerhalb des ökonomischen Rahmens. Die Reproduzierbarkeit kann nur indirekt angenommen werden, weil nicht sämtliche Verkehrssituationen erneut identisch auftreten werden, sondern nur ähnliche im Rahmen statistischer Wahrscheinlichkeit. Eine frühzeitige Aussage kann nicht vorliegen, weil zunächst ein vollständiges Auto entwickelt werden muss, dass anschließend als Gesamtsystem getestet werden muss. Die Sicherheit von Realfahrten eines zu testenden Systems zu beurteilen, bevor der Test zur Bewertung der Systemsicherheit erfolgt, schließt sich per Definition aus. Eine bessere Abdeckung kann bei den Effektivitätskriterien erreicht werden. Valide sind die Tests in jedem Fall, weil sie unter Realbedingungen ausgeführt werden. Die Repräsentativität ist bei abgeschlossenem Test vorhanden. Ändert sich das Modell im Laufe des Tests, muss erneut begonnen werden. Die Beobachtbarkeit ist durch Datenlogging-Systeme vollumfänglich gewährleistet.

2.2.2 Formale Methoden

Das Ziel formaler Methoden kann formuliert werden als vollständige Programmverifikation für nichttriviale Programme durch den Beweis der mathematischen Korrektheit. HALANG und KONAKOVSKY [15] sehen jedoch für die Softwareentwicklung den Nachweis über die Anforderungserfüllung als kritische und häufig nicht erfüllbare Aufgabe [15, S.293]. Der Grundgedanke ist für jede Operation die möglichen Ein- und Ausgabebereiche zu begrenzen, sodass zum Ende der Sequenz an Operationen der Ergebnisbereich des Programms eindeutig definierbar ist. Anhand der Methode der Vor- und Nachbedingungen erläutern sie das grundsätzliche Vorgehen vollständiger Verifikation. Jeder Anweisung (S) wird eine Vorbedingung (P) und Nachbedingung (Q) zugewiesen, sodass Q nach der Ausführung von S unter P erfüllt ist. Die äquivalente Schreibweise ist $\{P\}S\{Q\}$ bzw. für $P: x > 3$ und $S: x = x + 5$ kann $Q: x > 8$ verifiziert werden, in Kurzschreibweise $\{x > 3\}x := x + 5\{x > 8\}$. Für nachgeschaltete Operationen ist dadurch $P: x > 8$ verifiziert.

Die Absicherung durch formale Methoden greift diesen Gedanken auf und liefert so einen Paradigmenwechsel weg vom für die Automobilindustrie üblichen „Safety by Test“-Konzept hin zum „Safety by Design“-Konzept. Abstrakt betrachtet, müssen für die Anweisung „sicher fahren“ entsprechende Vorbedingungen gefunden werden, sodass aus den Nachbedingungen keine kritischen Situationen entstehen können.

MITSCH et al. [16, S.1] präsentieren einen Algorithmus, der einen Roboter autonom und nachgewiesen sicher agieren lässt. Sie geben die beiden Bedingungen an, die ein sicheres System erfüllen muss. Diese sind die Eigenschaften „Passive Safety“ (PS, dt.: passive Sicherheit) und „Passive Friendly Safety“ (PFS, dt.: passiv freundliche Sicherheit) nach MACEK et al. [17].

Sei T_b die eigene Bremsdauer und T_{ob} die maximale Bremsdauer der umgebenden Objekte, so ist PS gegeben, falls ein kollisionsfreies Bremsmanöver im wahrgenommenen Raum auf T_b existiert. Allerdings wird dabei das aufgezwungene Verhalten für andere Verkehrsteilnehmer bzw. die Dauer bis zur Kollision nach Stillstand nicht berücksichtigt. PFS hingegen ist gegeben, wenn freundlich gesinnte Objekte ausreichend Zeit $T_b + T_{ob}$ für Ausweich- und Bremsmanöver haben und ihnen eine unfallfreie Fahrt ermöglicht wird [17, S.487].

MITSCH et al. [16] leiten für PS den Beweis absoluter Sicherheit für alle eigenen Entscheidungen und für PFS zusätzlich die Absicherung für jeden weiteren Verkehrsteilnehmer ab [16, S.3]. Für PS muss das Fahrzeug in den Stillstand überführbar sein, bevor Kontakt zu anderen Objekten hergestellt wird.

Für das Verfahren bedeutet dies, dass das Fahrzeug niemals von einer sicheren zu einer unsicheren Trajektorie wechselt [16, S.4]. Eine sichere Trajektorie liegt vor, wenn innerhalb der definierten Parametergrenzen (z.B. Bremsleistung, Beschleunigung, Winkelgeschwindigkeit, Ansprechzeit) des Fahrzeuges bei, im schlechtesten Fall, gerader Annäherung der bewegten Objekte mit Maximalgeschwindigkeit die Distanz zwischen Fahrzeug und Objekt größer ist als die Bremsstrecke [16, S.6]. Die stärkere Anforderung der PFS beweisen sie, indem für sämtliche bewegte Objekte eine worst-case Bremsstrecke berechnet und kollisionsfrei berücksichtigt werden muss. Zusätzlich wird die Möglichkeit aufgezeigt, das Modell probabilistisch zu implementieren und damit Sicherheitswahrscheinlichkeit zu erhalten, sowie PS bei Unsicherheit in der Bestimmung der aktuellen Position und der Aktuatorik bewiesen [16, S.7].

SHALEV-SHWARTZ et al. [18] folgen dem Ansatz formaler Methoden, um ein sicheres Konzept zu erstellen, das für die Gesellschaft akzeptabel und für Millionen Fahrzeuge anwendbar ist [18, S.2]. Sie weisen ebenfalls darauf hin, dass die statistische Absicherung von autonomen Fahrzeugen nicht möglich ist [18, S.4]. In ihrem Modell verwenden sie den Ansatz „Responsibility Sensitive Safety“ (RSS, dt.: verantwortungssensitive Sicherheit), der nicht nur die Sicherheit von Aktionen, sondern auch die Schuld potentieller Unfälle betrachtet [18, S.4]. Das ist nötig, weil es Zustände und Verkehrssituationen gibt, in denen die einzige absolut sichere Aktion gewesen wäre, das Auto in der Garage stehen zu lassen, jedoch das kritische Ereignis völlig unverschuldet eintritt [18, S.5]. Ähnlich zur PFS definieren sie sicheren longitudinalen Abstand als den Abstand, bei dem die Bremsung eines Fahrzeugs nicht zur Kollision mit einem vorausfahrenden, maximal verzögernden Fahrzeug führt [18, S.6]. Ausgehend von dieser Distanz wird der erste Zeitpunkt, an dem ein Unfall unausweichlich ist, als Betrachtungspunkt verwendet, um Verantwortung und Schuld in Szenarien zu definieren. Durch weitere formale Definitionen wird eine Metrik für Autobahnsituationen generiert, die in Worten folgendermaßen zusammengefasst werden kann: „Handelt es sich um eine Folgefahrt, ist das hintere Fahrzeug schuld. Handelt es sich um einen Einschervorgang, sind beide Fahrzeuge schuld, außer ein Fahrzeug ist (signifikant) näher in der Fahrstreifenmitte und gleichzeitig in Querrichtung nicht (signifikant) schneller“ [18, S.7].

Weiterhin definieren SHALEV-SHWARTZ et al. [18, S.8f] formale Regeln, die ein Auflösen der Schuldfrage autonomer Fahrzeuge auch im innerstädtischen Kontext für Okklusionen ermöglichen. Diese sind zusammengefasst:

- Verletzt (nur) ein Fahrzeug das Geschwindigkeitslimit, so trägt es (alleinig) Schuld [18, S.8].
- Bei einem Fußgänger-Unfall hat immer das Fahrzeug Schuld, außer es steht, der Fußgänger bewegt sich schneller als für Menschen üblich oder der Fußgänger läuft seitlich in das Fahrzeug [18, S.9].

Diese Regeln können für sämtliche Verkehrsteilnehmer und Straßenbedingungen erweitert werden. Hielten sich sämtliche Fahrzeuge an diese Regeln und würden sicherstellen, dass sie nie die Schuld an einem Unfall tragen, kann es keinen Unfall geben [18, S.9]. Anschließend werden formale Algorithmen präsentiert, die eine Schuldzuweisung durch Kontrolle der Manöver und somit den Ausschluss selbstverschuldeter Unfälle ermöglichen [18, S.12ff].

Im weiteren Verlauf ihrer Arbeit wird von SHALEV-SHWARTZ et al. [18] ein Reinforcement Learning Algorithmus als Regelung RSS-konform implementiert, der die Regeln und Algorithmen der Schuldidentifikation einhält und sicherstellt, dass keine Unfälle selbstverschuldet entstehen [18, S.18ff]. Abschließend wird ein Verfahren definiert, in dem mit Unsicherheiten und Fehlern im Sensorsystem umgegangen wird [18, S.18ff].

Dieser Ansatz lässt sich nicht durch die Anforderungen an ein Testkonzept von WACHENFELD und WINNER [9, S.448] beurteilen, weil nicht durch den Test oder die Erprobung die Sicherheit bewiesen wird, sondern durch Verifikation. Dadurch ist der Test des Systems nicht mehr nötig. Jedoch ist fraglich, ob selbst bei bewiesener Sicherheit eines Systems nach formalen Methoden dieses ohne reale Erprobung zugelassen wird. Zusätzlich ergibt sich die Problematik, dass die Absicherung auf Programmebene erfolgt und dieses somit für Freigabe- und Haftungsfragen zugänglich gemacht werden muss. Ob diese Offenlegung von Funktionen und Know-How durch die Automobilhersteller akzeptiert werden würde, ist fraglich. Zumal die bisherige Absicherung ein reines Black-Box-Testing ohne Kenntnisse über die hinterlegten Programme ist. Dennoch ist diese Variante der Absicherung bzw. Auslegung von Fahrzeugen attraktiv, um die Freigabefälle autonomer Fahrzeuge zu lösen. Nach aktuellem Stand haben die formalen Methoden noch keinen praxistauglichen Algorithmus für den Betrieb in Fahrzeugen geliefert und wie von SHALEV-SHWARTZ et al. [18] bemerkt, benötigt es einen generellen Konsens, um die maximale Effizienz des Ansatzes auszunutzen. In dieser Arbeit wird nach einem anderen Ansatz vorgegangen, der im nächsten Abschnitt präsentiert wird.

2.2.3 Szenariobasierter Ansatz

Das Vorgehen der Arbeit erfolgt entsprechend dem *szenariobasierten Ansatz*, wie er unter anderem in dem Projekt PEGASUS [8, S.163] verwendet wird. Die Methodik rafft den Testraum auf ausgewählte Szenarien, um den Testaufwand zu reduzieren. Die Testgenerierung findet häufig durch eine Analyse von Szenarien anhand definierter Qualitätsmaße statt. Ziel ist es, alle relevanten statt aller möglichen Szenarien zu identifizieren und die Qualität des Systems auf Basis dieser zu ermitteln [19, S.50]. Die identifizierten Szenarien liefern die Grundlage der Freigabe, die durch die hohe Kritikalität der Szenarien einen großen Anteil am Unfallgeschehen, jedoch nur einen geringen Streckenanteil erfassen. Umgekehrt hat der Großteil der Fahrkilometer nahezu keinen Einfluss [20, S.149].

Die Bewertung der Szenarien erfolgt anhand einer zulässigen und repräsentativen Kritikalitätsmetrik. Unkritische, sowie weniger kritische, gleichartige Szenarien werden gerafft. Die so ermittelten Szenarien können simulativ getestet werden, um die Funktionsgrenze zu ermitteln. Anschließend kann diese in Realerprobung validiert und durch Szenarien, die simulativ nicht abzu prüfen sind, ergänzt werden. Dies bietet die Möglichkeit, den Testraum systematisch einzugrenzen und nachweislich kritische und relevante Testfälle zum Abtesten der Funktionsgrenze zu erhalten, die in der Realerprobung einen beherrschbaren Aufwand benötigen [19, S.50-51].

Im Folgenden werden unterschiedliche Varianten zur Szenarioidentifikation als Hinführung zum Ansatz der vorliegenden Arbeit erläutert. Zunächst wird auf augmentierte Realdaten und ein kombinatorisches Verfahren als Quelle für Szenarien eingegangen. Danach wird die Ausgangsbasis dieser Arbeit erläutert, ein Modell zur Generierung von Erfassungswahrscheinlichkeiten innerhalb der Sensorfelder eines Fahrzeuges. Abschließend werden unterschiedliche Möglichkeiten aufgezeigt, wie Kritikalitätsmetriken definiert sein können.

Realdaten und Erweiterungen

Es existieren unterschiedliche Möglichkeiten, Szenarien zu identifizieren und zu bewerten. Die erste Variante, mit der Testfälle für hochautomatisierte Fahrfunktionen ermittelt werden können, ist die Auswertung des realen Verkehrsgeschehens. Kritische Szenarien werden in Feldstudien und Recherchen in Unfalldatenbanken identifiziert und in eingangs definierte Ereignisklassen eingeteilt. Die Relevanz für den Straßenverkehr resultiert aus der Anzahl der Eintragungen in den unterschiedlichen Klassen [9, S.459]. In logischer Konsequenz stellt bei diesem Vorgehen auch jeder Unfall ein kritisches Szenario dar, weil die tatsächliche Kritikalität des Szenarios die beherrschbare Kritikalität der fahrzeugführenden Einheit (Mensch) überschreitet und somit der Unfall geschehen kann [21, S.162]. Ein starker Kritikpunkt bei alleiniger Verwendung dieser Quelle für Szenarien ist, dass zur Absicherung hochautomatisierter Funktionen nicht, wie bisher, die Kontrollierbarkeit durch den Menschen als Referenz verwendet werden kann. Stattdessen muss ein Nachweis über die Funktionalität und Zuverlässigkeit eines technischen Systems erfolgen [9, S.449]. Eine Abdeckung von 99,06 % der Ereignisse einer Unfallstudie [22, S.5987] kann nur eine Aussage über die Veränderung der für den Menschen kritischen Ereignisse, nicht über die Sicherheit des Systems bieten. Durch die Automatisierung zusätzlich entstehende Risiken werden nicht erfasst.

PÜTZ et al. [21] schlagen neben Unfalldatenbanken weitere Realdaten-Quellen, nämlich Prüfge-ländeabsicherungen, Feld- und Feldversuchsdaten, vor. Diese unterliegen derselben Problematik. Durch die Ergänzung um die virtuellen Quellen Fahr Simulator und Verkehrssimulation sowie die verbale Quelle Expertenwissen liegt eine breitere und fundierte Datenbasis vor [21, S.164]. Gerade in Verkehrssimulationen können Aspekte des automatisierten Systems getestet werden. Aus der Vielfalt der Quellen können logische Szenarien mit Häufigkeitsverteilungen der einzelnen Parameter bzw. funktionale Szenarien generiert werden [21, S.165]. Über die angewandte Kritikalitätsmetrik können wiederum relevante konkrete Szenarien als Testspezifikationen für die Absicherung ausgewählt werden [21, S.166], indem eine Raffung durch Ausschluss der unkritischen und innerhalb derselben Kategorie auch der weniger kritischen Testfälle erfolgt [9, S.459]. Alle vorgestellten Verfahren nutzen jedoch kein Wissen über das automatisierte System, sondern nutzen den Menschen als Referenz. Das deckt eine wichtige Klasse an Szenarien ab, jedoch nicht den vollständigen zu überprüfenden Raum möglicher Szenarien.

Kombinatorische Verfahren

Ein Testkonzept, das auf die repräsentative Abdeckung des Testraumes abzielt, ist der *modulare virtuelle Testbaukasten* von SCHULDT [23, S.71]. Er besteht aus den vier Elementen Auswahl und Analyse von Einflussparametern, Testfallgenerierung, Testdurchführung und Testauswertung [23, S.79]. Ziel des Verfahrens ist es, ein effizientes und valides Testverfahren mit einer möglichst geringen Anzahl an Testfällen bei ausreichender Testtiefe zu erzeugen [24, S.2]. Es integriert sich als Variante in das Konzept der szenariobasierten Absicherung durch Auffinden relevanter Szenarien und basiert auf Anforderungen und Szenarienkatalogen [24, S.5]. Im Gegensatz zu anderen Ansätzen des szenariobasierten Ansatzes wird für die Auswahl der zu testenden Szenarien auf die Bewertung der Kritikalität verzichtet und stattdessen der mögliche Zustandsraum durch Kombinatorik abgedeckt. Zu den beiden Punkten Auswahl und Analyse von Einflussparametern sowie Testauswertung wird auf SCHULDT et al. [24] verwiesen.

Ursprünglich war das Konzept für Fahrerassistenzsysteme vorgesehen, doch das methodische Vorgehen und vor allem die Ausgestaltung der Szenarien lassen sich auf automatisierte Fahrfunktionen übertragen.

Die Ausgestaltung eines Szenarios erfolgt nach dem Ansatz von SCHULDT [23] auf Basis eines 4-Ebenenmodells, wobei auf jeder der vier Ebenen: Basisstreckennetzwerk, situationspezifische Anpassung am Basisstreckennetzwerk, Beschreibung und Regelung der Akteure sowie Umweltbedingungen unterschiedliche Aspekte definiert werden können [23, S.107]. Als Weiterentwicklung zu diesem Ansatz wird von BAGSCHIK [25] eine zweite Ebene eingefügt, welche die Verkehrs- oder Leitinfrastruktur gesondert aufnimmt, um Regeln und Semantik vom Aufbau des Szenarios zu trennen [25, S.1816f]. In einer zusätzlichen Erweiterung fügen SAUERBIER et al. [26] eine sechste Ebene ein, die Ebene Digitale Informationen. Diese enthält sämtliche digital erhaltenen Daten, beispielsweise aus Digitalkarten oder V2X-Kommunikation [26, S.43]. Das „Sechs-Ebenen-Modell“ nach SAUERBIER et al. [26, S.43] wird in Abb. 2.3 dargestellt.

Anschließend werden die Szenarien systematisch definiert. Die anfänglich unendlich große Anzahl diskreter Wertepaarungen der relevanten Parameter wird durch Gruppierung und kombinatorische Testverfahren innerhalb ihrer Wertebereiche variiert und auf valide und redundanzarme Testfälle reduziert [24, S.11-13]. Das Vorgehen weicht damit von dem Gedanken der Raffung unkritischer Testfälle ab und minimiert die Anzahl der Testfälle kombinatorisch.

Wie bereits WACHENFELD und WINNER [9, S.461] als Kernaspekt einer Testdurchführung fordern, besteht die Testdurchführung nach SCHULDT [24] aus virtuellen und künstlichen Elementen. Ausgehend von Simulationen werden reale Elemente schrittweise bis hin zu Realtests in einer Abfolge unterschiedlicher Erprobungsarten integriert, sodass zu Beginn die Vorteile der Simulation (Reproduzierbarkeit, Sicherheit, Dauer, Kosten) und bei hohem Systemreifegrad die Vorteile der Realerprobung (Realitätsnähe, Use-Case) zum Tragen kommen [24, S.17, 18].

Die vorliegende Arbeit befasst sich mit der Generierung von Szenarien aus der Sensorabdeckung eines Fahrzeugs. Dafür wird in Anlehnung an das Sechs-Ebenen-Modell nach SAUERBIER [26] die erste Ebene definiert, sowie in der vierten Ebene die Bewegung der Objekte.

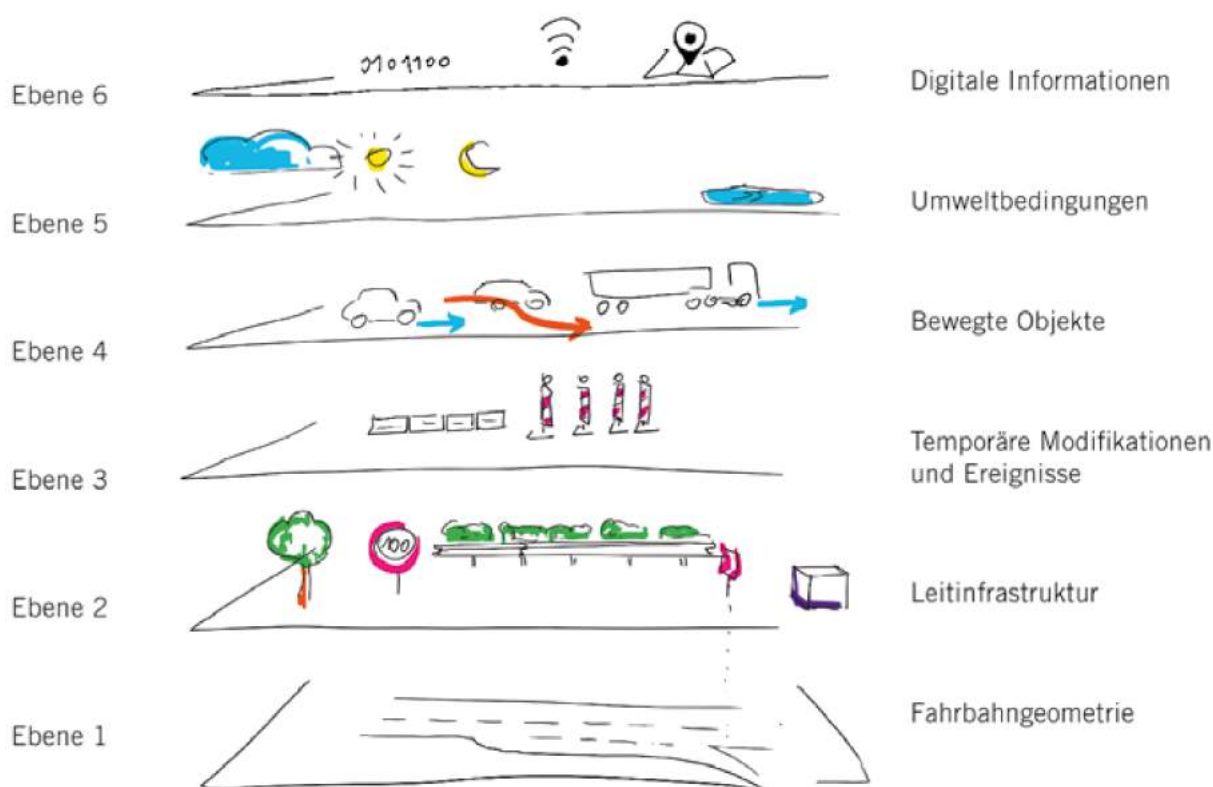


Abbildung 2.3: Sechs-Ebenen-Modell nach SAUERBIER [26, S.43]

Modellierung der Sensorabdeckung

Eine weitere Variante, Szenarien zu generieren, wird aktuell am Lehrstuhl für Fahrzeugtechnik der Technischen Universität München untersucht. Ein Aspekt, der in bisherigen Arbeiten und Projekten wenig Beachtung gefunden hat, ist die Ausprägung des abzusichernden Systems, genauer gesagt, die Ausstattung an umgebungserfassender Sensorik des Fahrzeuges. Ein hochautomatisiertes Fahrzeug kann unabhängig von der Kritikalität der Verkehrssituation oder den verwendeten Algorithmen nur auf das reagieren, was es durch seine Sensorik wahrnimmt. Infolgedessen ist es Ziel des Konzepts, die Definition von Szenarien unter Berücksichtigung der sensorischen Umfelderkennung durchzuführen.

MÜLLER [27] baut im Rahmen seiner Arbeit am Beispiel des Audi A8 [27, S.34] ein Modell auf, mit dem optimale, kritische Annäherungspfade an ein Fahrzeug berechnet werden [27, S.3] (Abb. 2.4). Diese Pfade bilden die Grundlage der Annäherung zweier Fahrzeuge aus. Je geringer die Erfassungswahrscheinlichkeit des Pfades ist, desto kritischer bzw. relevanter ist er. Aus der fahrzeugspezifischen Sensortopologie und deren Eigenschaften, den Umweltbedingungen und dem Detektionsobjekt erstellt das Modell eine mathematische Modellierung der Sensorabdeckung zur Ermittlung der Erfassungswahrscheinlichkeit in einer definierten Schnittebene [27, S.34]. Es definiert dafür Erfassungsbereiche und -wahrscheinlichkeiten der einzelnen Sensoren in sensorspezifischen Kugelkoordinatensystemen. Anschließend wird aus den Einzelwahrscheinlichkeiten eine fusionierte Wahrscheinlichkeit in einem globalen Kugelkoordinatensystem gebildet. Für die Modellierung der einzelnen Sensoren und des Gesamtsystems werden dr als radiales Inkrement, $d\varphi$ als Inkrement des Azimutwinkels und $d\theta$ als Inkrement der Elevation verwendet. Die so ermittelten Werte werden in ein kartesisches Koordinatensystem mit Inkrementen dx , dy und dz interpoliert, das zur Pfadfindung verwendet wird. In Abb. 2.5 zeigt der farbige Bereich den gesamten Erfassungsbereich des Fahrzeugs.

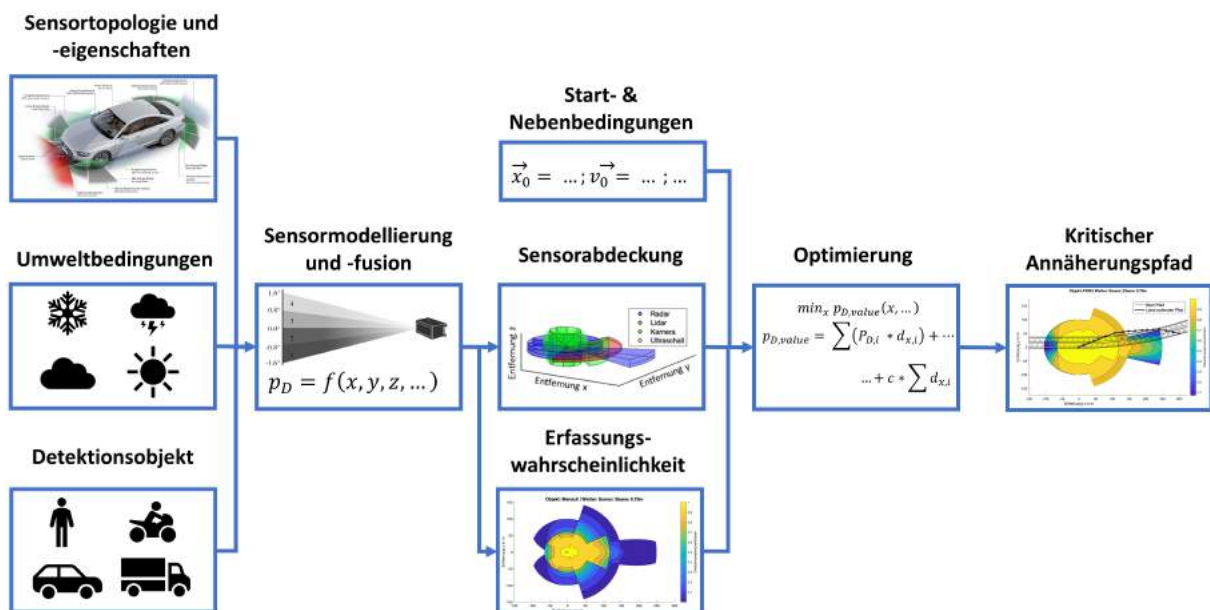


Abbildung 2.4: Vorgehen und Aufbau des Modells von MÜLLER [27], Abbildungen aus MÜLLER [27] und [28]

Zur Ermittlung optimaler Annäherungspfade wird die fusionierte Detektionswahrscheinlichkeit in einer x - y -Ebene auf Höhe z verwendet. Als Startbedingungen werden je ein Start- und Endpunkt, die Höhe der Schnittebene, das Erfassungsobjekt sowie die Wetterbedingungen vordefiniert [27, S.48]. Zu Beginn wird eine Linie zwischen den Punkten gebildet und mittels Gleichung (2.1) als Kostenfunktion der Optimierung der Wert dieses Pfades bestimmt [27, S.49].

$$p_{D,value,path} = \sum (P_{D,i} * dx_i) + c * \sum dx_i \quad (2.1)$$

Der erste Summand erfasst für den Bereich innerhalb der Sensorabdeckung die Detektionswahrscheinlichkeit P_D für das Streckeninkrement dx_i zwischen zwei Punkten $i - 1$ und i . Der zweite Summand addiert einen streckenabhängigen Anteil zur Funktion hinzu, sodass auch die zurückgelegte Strecke außerhalb des Erfassungsbereichs berücksichtigt wird. Als Wert der Konstante c wird 0,05 verwendet [27, S.49]. Die in MATLAB umgesetzte Optimierung verwendet den Innere-Punkte-Algorithmus mit Standardparametrierung [27, S.69]. Das Ergebnis der Optimierung ist ein systemspezifisch kritischer Annäherungspfad. In Abb. 2.4 ganz rechts ist zusätzlich zu dem lokal optimalen Pfad ein Straßenverlauf eingezeichnet. Wie ersichtlich ist, werden aktuell keine Nebenbedingungen formuliert, um Straßengrenzen zu berücksichtigen [27, S.72]. In einer quasiglobalen Optimierung wird durch Mehrfachausführung der lokalen Optimierung das beste aus n lokalen Optima ausgewählt [27, S.69]. Abb. 2.5 zeigt die iterative Veränderung dieser Pfade. Die entstehenden Pfade sind jedoch nur beschränkt für weitere Anwendungen verwendbar, weil sie aufgrund fehlender Nebenbedingungen kein realistisches Verkehrsszenario auf einer Autobahn ergeben [27, S.78].

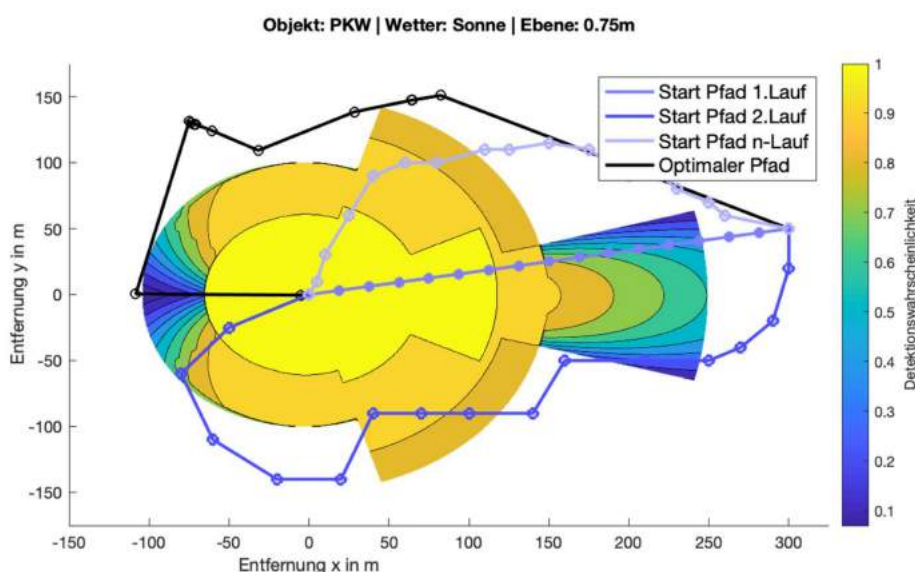


Abbildung 2.5: Optimierte Annäherungspfade der quasiglobalen Optimierung nach MÜLLER [27, S.70]

Bewertung von Kritikalität und Komplexität eines Szenarios

Um relevante, kritische Szenarien (automatisch) identifizieren zu können, ist eine Metrik mit Auswahlbedingungen nötig. Die Suche nach einem validen Risikomaß, Kritikalitätsmetrik genannt, stellt aktuell eine große Herausforderung dar [9, S.459] [29, S.60]. Kritikalitätsmetriken dienen entweder der Trajektorienplanung, der Bestimmung eines Unfallrisikos oder der Identifikation kritischer Situationen in Datenbanken [20, S.149]. Die an die Metrik gestellten Anforderungen sind die Eignung für alle Unfallarten, menschliches und maschinelles Fahren und die Voraussage der Unfallfolgeschwere. Zusätzlich müssen irrelevante Szenarien durch die Metrik ausgeschlossen werden [29, S.61]. Zur Definition der Metriken werden unterschiedliche Parameter

verwendet, z.B. fahrdynamische Größen: Längs- und Querschleunigung [29, S.61], Gierrate, Lenkwinkel und Ruck [20, S.149], deterministische Time-To-X (TTX) Metriken: Dauer bis zu einem definierten Ereignis X [30, S.146] oder Parameter der Verkehrssituation: Fahrzeugabstand und Relativgeschwindigkeit [22, S.5989].

JUNIETZ et al. [29] schlagen vor, einen gemeinsamen Kennwert durch Kombination und Normierung aus den fahrdynamischen Größen zu generieren, in ihrem Ansatz die Kombination aus Längs- und Querschleunigung, Spanne zur Korrektur von Schwimmwinkel und Geschwindigkeit [29, S.61]. Die TTX-Metriken haben ihren Ursprung in der Ermittlung der Zeit zu Beinahe-Unfällen, der Time-To-Collision (TTC) nach HAYWARD [31]. Es existiert eine Vielzahl weiterer verwandter Metriken mit unterschiedlichem Bezugsereignis. SCHREIER [30] nennt: TTC, Time-To-React (TTR), Time-To-Brake (TTB), Time-To-Kickdown (TTK) und Time-To-Steer (TTS) [30, S.146].

Die TTC ist definiert als die Zeit bis zur Kollision zweier Fahrzeug bei ihrer aktuellen Geschwindigkeit auf der Extrapolation ihrer aktuellen Spur [31, S.27] und somit der Abstand der Fahrzeuge geteilt durch die Relativgeschwindigkeit. Je höher der Wert, desto sicherer oder unkritischer ist die Situation [32, S.90]. Befinden sich die Fahrzeuge nicht auf einem Kollisionskurs, ist der TTC-Wert unendlich [31, S.29]. Daraus ergibt sich, dass bei sich nahezu parallel bewegenden Fahrzeugen geringe Lenkwinkeländerungen Sprünge der TTC zwischen unendlich und sehr geringen Werten bewirken können [31, S.29].

Eine detailliertere und globalere Bewertung von Szenarien ist durch die Verwendung komplexerer Kritikalitätsmetriken möglich. Eine Verbesserung dieser Problematik bietet der Time Integrated Time-To-Collision Indicator (TIT) bzw. in einer Vorstufe die Time Exposed Time-To-Collision (TET) nach MINDERHOUD und BOVY [32]. Die TET ist die Zeit, in der die TTC innerhalb eines Manövers unter einen definierten Grenzwert fällt. Dieser Grenzwert soll eine Schranke zwischen kritischen und unkritischen Situationen darstellen. Als Ergebnis liegt ein Wert vor, mit dem die Dauer kritischen Fahrens innerhalb eines Szenarios bewertet werden kann [32, S.92]. In Erweiterung dazu ermöglicht der TIT die Bewertung der Kritikalität eines gesamten Szenarios. Er ist definiert als das Integral über die Differenz aus TTC-Grenzwert und TTC, solange die TTC zwischen 0 s und dem Grenzwert liegt. Kritische Szenarien nehmen dadurch aufgrund der langen Exposition zu geringen TTC-Werten einen hohen Wert an [32, S.93].

2.3 Kritik am Stand der Technik

Dass die Entwicklung und Markteinführung von HAF stattfinden wird, ist kaum noch bezweifelt, die Frage ist nur noch, in welchem Umfang. Das Auflösen der Freigabefalle rückt durch marktreife Systeme wie den „Audi AI Staupilot“ [2] immer mehr in den Vordergrund, weil die bisherige Testmethodik der Automobilindustrie entsprechend den Kriterien von WACHENFELD und WINNER [9, S.448] nicht sinnvoll auf HAF übertragen werden können. Daraus folgt die Forderung nach einer neuen Absicherungsmethodik. Grundsätzlich lassen sich zwei unterschiedliche Ansätze identifizieren, die sich mit dieser Frage beschäftigen. Das sind einerseits die formalen Methoden, andererseits der szenariobasierte Ansatz. Die formalen Methoden bieten potentiell die Möglichkeit, die Sicherheit des Systems durch Verifikation zu beweisen, haben jedoch Limitationen bezüglich der Markteinführung, falls sich dazu ein gesellschaftlicher Konsens ausbildet. Nach aktuellem Stand existiert jedoch noch keine Methode, die derzeit für eine skalierte Anwendung im Realverkehr geeignet ist, und der Hauptfokus deutscher Forschungsaktivitäten richtet sich auf den Bereich des szenariobasierten Ansatzes, wie beispielsweise die Existenz der beiden Projekte PEGASUS [8, S.163] und Ko-HAF [12] zeigt.

Die Kernthematik des szenariobasierten Ansatzes ist das Auffinden der relevanten und kritischen Szenarien. Vorgeschlagene Quellen sind Unfalldatenbanken, Feldstudien, Expertenwissen, Fahr-simulatorstudien und Fahrsimulationen. Für alle Realdatenquellen gilt, dass sich die Kritikalität des Szenarios aus der Kritikalität für den Menschen ergibt. Für simulative Quellen stellt sich nach wie vor die Frage, welche Kriterien nötig sind, um bisher nicht erfasste Szenarien zu identifizieren, und auf welche Art und Weise diese in Szenarien überführt werden. Im Anschluss an das Auffinden möglicher Szenarien und deren Beschreibung ist eine Auswahl nötig, weil ansonsten aus der Kombination der unterschiedlichen Parameter und deren Werten eine unendliche Anzahl Szenarien entsteht. Häufig werden dafür Kritikalitätsmetriken verwendet, die durch Zuweisung einer Kritikalitätsbewertung die Möglichkeit bieten, die kritischsten Szenarien auszuwählen. Der Ansatz von SCHULDT [24] verwendet für diese Aufgabe Kombinatorik, indem zunächst sämtliche mögliche Kombinationen auf eine repräsentative Auswahl eingeschränkt werden. Hier stellt sich, wie bereits bei der Erklärung der Freigabefälle, die Frage, ab wann eine Stichprobe für die Gesamtheit repräsentativ ist. Zusätzlich wird eine definierte Anzahl Parameter verwendet, wobei essentiell wichtig ist, alle für das Gesamtsystem relevanten Parameter abzudecken. Die Vorstellung einiger Kritikalitätsmetriken zeigt, dass diese für spezielle Anwendungsfälle mehr oder weniger gut geeignet sind und vor allem immer auf der Ausprägung definierter Parameter eines spezifischen Szenarios beruhen.

In den Verfahren werden die tatsächliche Ausgestaltung des Systems und die darin enthaltenen Informationen über die Relevanz und Kritikalität einzelner Szenarien für das spezifische System nicht betrachtet. Ein Beispielszenario, das nach Bewertung durch Kritikalitätsmetriken sehr kritisch ist (geringe TTC, hoher TIT), kann für unterschiedliche Systeme mehr oder weniger kritisch sein. Umgekehrt kann ein absolut unkritisches Szenario, beispielsweise das Überqueren einer Straße durch einen Radfahrer auf offener Straße bei nur sehr geringer Verkehrsdichte, zu Unfällen führen, je nachdem zu welcher Tages- und Nachtzeit es stattfindet und mit welcher Sensorik das Fahrzeug ausgestattet ist. Das Auffinden einer Kritikalitätsmetrik, die für die Gesamtheit aller möglichen Fahrzeugsysteme, Szenarien und Bedingungen die Kritikalität spezifisch wiedergeben kann ist, entsprechend dem „No Free Lunch“-Theorem genauso unmöglich, wie einen universell besten Optimierer zu finden.

Aufgrund dessen wird in dieser Arbeit eine Methodik erstellt, die aufbauend auf dem Ansatz zur Modellierung der Sensorabdeckung nach MÜLLER [27] die Sensortopologie in den Auswahlprozess von Szenarien integriert und somit bisher ungenutzte Informationsquellen verwendet. Durch die systemspezifische Modellierung der Sensorabdeckung können aufgrund mangelnder sensorischer Erfassung der Zielobjekte individuelle Schwächen und herausfordernde Szenarien aufgedeckt werden. Der Vorteil dieser Methodik gegenüber den formalen Methoden ist, dass keine Systemfunktionalitäten offengelegt werden müssen, um die Sicherheit des Systems zu beweisen, aber trotzdem die Konfiguration des Systems und nicht nur dessen Verhalten betrachtet wird.

Die explizite Kritikalitätsbewertung von Szenarien ist kein Bestandteil dieser Arbeit, sondern das Auffinden eines worst-case Pfades bezüglich der Sensorabdeckung. Anschließend wird aus diesem Pfad ein Basisszenario generiert, das die Position zweier Fahrzeuge beinhaltet. Der worst-case Pfad ist kritisch in Bezug auf die Güte der sensorischen Erfassung, wobei sinkende Erfassungsgüte steigende Kritikalität bedeutet. Zur Begriffstrennung wird nicht von der Ableitung eines möglichst kritischen Pfades, sondern eines worst-case Pfades gesprochen. Das Resultat dieser Arbeit kann als Eingang für ein durch ROGHARDT [33] erstelltes Modell verwendet werden, das die Kritikalität von Szenarien maximiert. Dafür wird zur Bewertung der Kritikalität von Situationen die TTC und für Szenarien der TIT nach MINDERHOUD und BOVY [32] verwendet.

Das Wissen über die sich anschließende Arbeitskette ermöglicht es, bereits in dieser Arbeit gewisse Annahmen und Methoden zu definieren, die eine Konformität unterstützen. Eine direkte Aussage zur Kritikalität des in dieser Arbeit entstehenden Pfades kann nicht getroffen werden, weil dafür immer eine Bewertung eines konkreten Szenarios und der Bewegung der beteiligten Verkehrsteilnehmer nötig ist. Für die Erstellung des Pfades werden mathematische Optimierungs- und Routenplanungsalgorithmen benötigt. Diese werden im folgenden Kapitel vorgestellt. Zusammenfassend besteht die erarbeitete Methodik aus den Arbeitspaketen aus Abb. 2.6. Aufbauend auf der Sensormodellierung nach MÜLLER [27] und passenden, zu identifizierenden Start- und Nebenbedingungen werden zwei alternative Verfahren zum Innere-Punkte-Verfahren, das MÜLLER [27] verwendet, ausgearbeitet. Daraus resultiert ein worst-case Annäherungspfad, der realitätskonform definiert ist und anschließend in ein realistisches Szenario überführt wird.

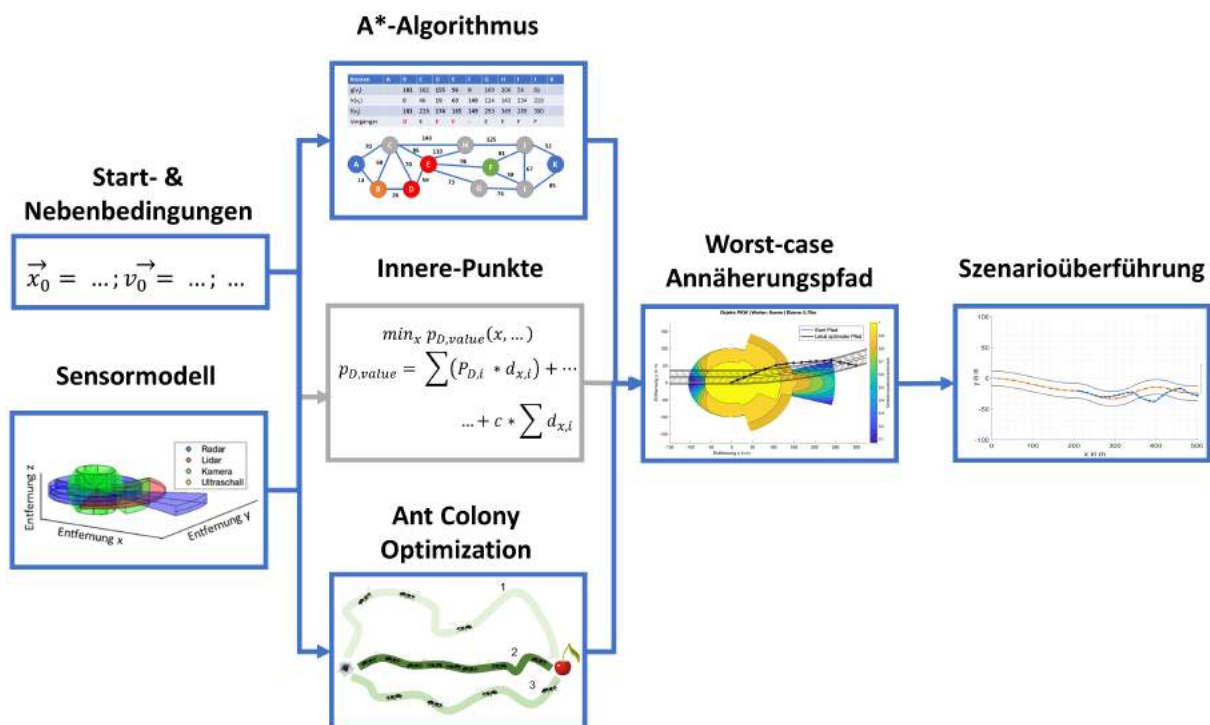


Abbildung 2.6: Ablaufdiagramm der Methodik mit Abbildungen aus MÜLLER [27] und MIRJALILI [34]

3 Mathematische Optimierung

In der Wissenschaft, der Industrie und dem alltäglichen Leben wird man oft mit Fragestellungen und Problemen konfrontiert, die bestmöglich zu lösen oder zu verbessern sind. Diese Aufgabe kann stellvertretend mit Optimierung bezeichnet werden. GRIMME und BOSSEK [35] beschreiben Optimierung als „[...] das Bemühen um Verbesserung von Werkzeugen, Prozessen, Organisationsformen oder auch vollständigen Systemen und [sie] ist ein zentraler Aspekt im menschlichen Dasein.“ [35, S.1]. Optimierung dient somit immer der Verbesserung eines Systems. Im mathematischen Sinne geht sie jedoch über die bloße Verbesserung hinaus und strebt immer das Auffinden des einen Optimums bzw. eines der Optima an [36, S.1]. Anders ausgedrückt ist das Ziel der Optimierung das Identifizieren einer optimalen Entscheidungsfindung durch den Optimierer [37, S.1]. Diese kann definiert werden als diejenige Entscheidung, die die spezifizierte Zielsetzung unter Berücksichtigung der Entscheidungsrestriktionen und -auswirkungen am ehesten erfüllt [37, S.1].

Das Feld der Optimierung umfasst viele Wissenschaftsdisziplinen und weitreichende Forschungs- und Entwicklungsarbeiten. Infolgedessen ist es nicht möglich eine allumfassende Zusammenfassung zu geben. Die folgenden Abschnitte behandeln vielmehr die für diese Arbeit relevanten Teilbereiche und sollen die nötigen Grundlagen zum Verständnis der darauf aufbauenden Arbeit geben. Für diese Arbeit werden deswegen Optimierung im Allgemeinen und Graphen mit den ihnen zugehörigen Optimierungsmethoden im Besonderen betrachtet. Für diese Arbeit ist das Auffinden eines Pfades Kernthema. Deswegen werden benötigte Pfadfindungsalgorithmen aus dem Bereich der Routenplanung detailliert erklärt. Zur ausführlichen Vertiefung wird auf GRIMME und BOSSEK [35], PIEPER [36] und PAPAGEORGIOU et al. [37] verwiesen.

Im ingenieurwissenschaftlichen Kontext wird die Optimierung durch den Computer übernommen, der Optimierer ist derjenige Algorithmus, der verwendet wird, um die optimale Entscheidungsfindung bzw. das Optimum aufzufinden [35, S.2]. Der Erfüllungsgrad der Zielsetzung bzw. die Modellqualität wird durch die sogenannte Zielfunktion f kontrolliert [35, S.2]. Der Prozess der Optimierung selbst lässt sich beschreiben als das Modellieren des Systems sowie das Auffinden der Parameter $x_1, \dots, x_n$ und deren Variation bis bezüglich der verwendeten Zielfunktion ein optimaler Parametersatz des Modells erhalten wird [35, S.1]. Dabei gilt es im Falle „restringierter Optimierung“ [36, S.21] „Entscheidungsrestriktionen“ [37, S.1] bzw. Randbedingungen zu beachten, die bestimmte Parameterwerte und -kombinationen verbieten [35, S.3].

3.1 Optimierung formal

Die obige sprachliche Beschreibung kann in eine entsprechende mathematisch-formale übersetzt werden. Dafür werden zunächst die einzelnen Bestandteile der Optimierung näher betrachtet.

Die zu minimierende Zielfunktion f ist definiert als eine Abbildung aus dem Suchraum $S \subseteq \mathbb{R}^n$ in den Zielraum $Z \subseteq \mathbb{R}$ im reellwertigen Fall bzw. $Z \subseteq \mathbb{R}^m$ im vektorwertigen Fall. Die Dimensionalität des Suchraumes bzw. Zielraumes ist durch n bzw. m gegeben. Die Parameter stammen aus dem Suchraum $x_1, \dots, x_n \in S$ [35, S.2]. Handelt es sich um eine restringierte Optimierung, müssen Restriktionen, die Nebenbedingungen, eingehalten werden, sodass der Suchraum auf Teilgebiete eingeschränkt wird. Diese werden üblicherweise als Gleichungen h und Ungleichungen g definiert und können auch nichtlinear sein [36, S.21]. Existieren für einzelne Variablen spezifische untere (lb_i , engl.: lower bound) und obere (ub_i , engl.: upper bound) Schranken, so werden diese häufig separat angegeben [34, S.5]. In kompakter Darstellung schreibt man:

$$\begin{aligned} \min_{x \in S} f(x) \quad & \text{sodass} & g_i(x) &\leq 0 \quad \text{für } i = 1, \dots, m \\ & & h_i(x) &= 0 \quad \text{für } i = 1, \dots, p \\ & & lb_i &\leq x_i \leq ub_i \quad \text{für } i = 1, \dots, n \end{aligned} \quad (3.1)$$

3.2 Gradientenbasierte Verfahren

Eine ganze Reihe von mathematischen Optimierungsverfahren lassen sich den gradientenbasierten Verfahren zuordnen. Viele numerische Optimierungsalgorithmen benutzen den Gradienten einer Funktion in einem Punkt x , um von dort in Richtung des Gradienten zum Optimum zu gelangen [37, S.41]. PAPAGEORGIOU et al. [37] geben eine anschauliche Erklärung für die Suche nach dem Optimum in einem zweidimensionalen Suchraum [37, S.41]:

Man befinde sich nachts in einer gebirgigen unbekannten Landschaft, ausgerüstet mit einer schwachen Taschenlampe (zur Ermittlung des lokalen Gradienten) und einem Höhenmesser (zur Ermittlung des lokalen Funktionswertes). Wie müssen wir schreiten, um den tiefsten Talpunkt (dessen Höhe und Lage uns unbekannt sind) zu erreichen?

Als Vorgehen, um den tiefsten Punkt zu entdecken, geben sie die folgende Abfolge vor: (1) Abstiegsrichtung wählen, (2) Schrittweite bestimmen und (3) Abbruchbedingung prüfen. Die Bewegung findet an den Berghängen entlang in Richtung Tal statt, wobei die Abstiegsrichtung entsprechend des Gradienten gewählt wird. Für die Überleitung des Gradienten in eine definierte Abstiegsrichtung existieren unterschiedlichste Verfahren. Das Gradientenverfahren richtet sich beispielsweise immer in entgegengesetzte Gradientenrichtung und damit in Richtung des steilsten Abstiegs. Nähere Erklärungen und Beispielverfahren präsentieren PAPAGEORGIOU et al. [37, S.40ff]. Zusätzlich stellen sie gradientenfreie Verfahren vor, die zur Bestimmung der Abstiegsrichtung nicht auf die Ermittlung des Gradienten angewiesen sind [37, S.56ff].

Abstiegsverfahren haben im Allgemeinen Nachteile, die sie für den vorliegenden Anwendungsfall ausschließen. Findet die Auswahl gradientenbasiert statt, muss eine entsprechend starke Strukturierung des Suchraums vorliegen [35, S.191]. Andernfalls wird fälschlicherweise Ähnlichkeit zu Nachbarn vermutet und es werden ähnliche Zielfunktionswertänderungen in der Umgebung der betrachteten Lösung angenommen [35, S.191].

Keines der Verfahren garantiert globale Optimalität der ermittelten Lösung und zusätzlich liegt eine Fixierung auf lokale Optima bei geringer Exploration vor. Durch Mehrfachausführung, Vergleich und Anwendungswissen kann lediglich eine relative Bewertung der Lösung stattfinden [35, S.192]. Das Ausschlusskriterium für die behandelte Problemstellung ist jedoch das Verhalten bei Diskontinuitäten [35, S.191]. Übertragen auf die Bergsteiger-Metapher sind dies auf der Suche nach dem Tal plötzlich auftretende Schluchten oder Steilwände. Auch benachbarte Bereiche inverser Steigung sind denkbar oder plötzliche Veränderungen der Steigung. Diese plötzlichen

Änderungen des Terrains können im Schein der Taschenlampe durch den Bergsteiger nicht erkannt werden. Jegliche Art Diskontinuität verhindert die Annahme, dass der aktuelle oder bisherige Verlauf der Funktionswerte in die Umgebung des aktuell betrachteten Punktes extrapoliert werden kann.

Das vorgestellte Sensormodell weist den Sensoren Erfassungsbereiche zu, in denen sensorspezifisch die Detektionswahrscheinlichkeit für ein Objekt bestimmt wird. Innerhalb eines Sensorbereichs liegt ein kontinuierlicher funktioneller Zusammenhang zwischen Zielfunktionswert (Erfassungswahrscheinlichkeit) und Suchraum (Dreidimensionale Koordinaten) vor. An sämtlichen Übergängen zu nichterfassten, angrenzenden oder überlappenden Sensorbereichen besteht eine Diskontinuität, weil es schlagartig zu einem Wertewechsel inklusive verändertem funktionalem Zusammenhang kommt. Jeder Iterationsschritt, der eine Gebietsgrenze überschreitet, verwendet Informationen des vorherigen Bereichs, um das Verhalten in einem andersartigen Bereich vorherzusagen. Erschwerend kommt hinzu, dass niedrige Erfassungswahrscheinlichkeiten in den Randbereichen der Sensoren bzw. außerhalb der Erfassungsbereiche eine generelle Tendenz der Algorithmen entlang der Bereichsgrenzen bedingen. Im Modell von MÜLLER [27] wird aktuell dennoch der Innere-Punkte-Algorithmus verwendet. Aufgrund der Anwendung im zweidimensionalen Raum ist der Einfluss der Bereichsübergänge vergleichsweise gering. Zusätzlich wird der Algorithmus nur durch sehr wenige Randbedingungen eingeschränkt, sodass sich die Ergebnispfade zu großen Teilen außerhalb der Erfassungsbereiche bewegen und Bereichsübertritte weitestgehend vermieden werden. Aus diesen Gründen wird für die vorliegende Problemstellung ein neuer Algorithmus aus dem Bereich der Graphentheorie zum Auffinden eines bereichsübergreifenden kritischen Pfades implementiert.

3.3 Routenplanung und Graphen

Graphen beschreiben in der Mathematik die Struktur von Optimierungsproblemen der Routenplanung und damit die Basis für Optimierungsverfahren [35, S.27]. Die Routenplanung beschäftigt sich mit dem Auffinden (optimaler) Wege zwischen einem definierten Start- und Zielpunkt [38, S.126]. Diese Art der Suche wird für die Definition des worst-case Pfades benötigt und ermöglicht die Auflösung der Abhängigkeit von der funktionalen Beschreibung des Problems und dessen Kontinuität.

Dafür werden in diesem Abschnitt zuerst Graphen und ihre Bestandteile erklärt (Abschnitt 3.3.1) und anschließend ein Beispielgraph erstellt (Abschnitt 3.3.2), der für die Erklärung der Algorithmen der folgenden Kapitel verwendet wird.

3.3.1 Bestandteile von Graphen

Grundsätzlich besteht ein Graph aus einer nichtleeren Menge V von Knoten $v \in V$ und deren Verbindungen, der nichtleeren Menge E an Kanten $e \in E$ [35, S.28]. Diese können für unterschiedliche Zustände eines Systems mit deren Übergängen oder auch Städte mit ihren verbindenden Straßen stehen. Sowohl der Wechsel zwischen Zuständen als auch das Befahren der Verbindungsstraßen ist mit Aufwand verbunden. Dafür wird jeder Kante ein Kantengewicht zugeordnet, das die Kosten für die Nutzung der Kante angibt und den Grundstein für graphenbasierte Algorithmen legt [35, S.32]. An dieser Stelle werden einige wichtige Begrifflichkeiten im Kontext der Graphentheorie erklärt:

Ungerichtete Graphen

Die Benutzung jeder Kante ist in beide Richtungen erlaubt. Verbindet die Kante e die Knoten v_i und v_j , so kann e von v_i nach v_j oder umgekehrt mit den gleichen Kosten benutzt werden. In gerichteten Graphen ist dies nicht gegeben und Kanten können nur in spezifizierte Richtungen benutzt werden [35, S.29]. Die Straßen zwischen den Städten entsprächen im Falle gerichteter Kanten Einbahnstraßen.

Schlichte Graphen

Es treten weder Schlingen noch Mehrfachkanten im Graph auf. Schlingen sind Kanten, deren Start- und Endknoten identisch sind, Mehrfachkanten sind parallele Kanten zwischen zwei Knoten [35, S.30]. Der verwendete Graph ist ein schlichter Graph.

Adjazenz und Inzidenz

Werden die zwei Knoten v_i und v_j über eine Kante e verbunden, so nennt man die Knoten adjazent oder benachbart. v_i bzw. v_j und e werden inzident genannt [35, S.30]. München (J) und Weilheim (K) besitzen eine Verbindung, dementsprechend sind sie adjazent. Die Straße zwischen Weilheim und München ist inzident zu München und Weilheim.

Nachbarschaft

Alle Knoten v_N , die adjazent zum Knoten v_i sind, bilden dessen Nachbarschaft N . Im Kontext gerichteter Graphen spricht man von Vorgängermenge, die Menge an Knoten von denen man v_i erreichen kann, und Nachfolgermenge, die Menge anschließend erreichbarer Knoten [35, S.30]. Im Beispiel befinden sich in der Nachbarschaft von München (J) vier Städte (Ingolstadt (F), Regensburg (H), Augsburg (I) und Weilheim (K)).

Knotengrad

Der Grad eines Knoten entspricht der Anzahl seiner Nachbarn. Im Kontext gerichteter Graphen spricht man vom Eingrad (Anzahl in Vorgängermenge) und Ausgrad (Anzahl in Nachfolgermenge) [35, S.31].

Zusammenhang

Ein Graph G ist schwach zusammenhängend, wenn im ungerichteten Graph G^* , der aus G konstruiert wurde, ein Weg von jedem Knoten $v_i \in V$ zu jedem anderen Knoten $v_j \in V$ existiert. Ist dies nicht erfüllt, nennt man den Graphen unzusammenhängend. Ist jeder Knoten von jedem anderem Knoten erreichbar, ist der Graph stark zusammenhängend [35, S.33].

Ein Spezialfall eines Graphen ist der Baum, der uns als abstrakte Datenstruktur aus dem Alltag bekannt ist, beispielsweise zur Strukturierung von Vorfahren oder Gliederung von Gruppen [35, S.34]. GRIMME und BOSSEK [35] nennen die drei Bedingungen, die ein Graph erfüllen muss, um ein Baum zu sein. Werden zwei davon erfüllt, ist die dritte implizit gegeben.

- Der Graph ist zusammenhängend.
- Die Anzahl der Knoten im Graph ist um eins größer als die Anzahl an Kanten.
- Der Graph ist azyklisch, es gibt keinen Kreis im Graphen.

Ein allgemeiner Vorteil von Bäumen gegenüber Graphen liegt in der Strukturierung der Daten, sodass das Suchen gewünschter Daten deutlich effizienter gestaltet werden kann [35, S.44]. Es existieren einige Algorithmen, die diesen Vorteil ausnutzen, um effiziente Suchvorgänge durchzuführen.

3.3.2 Erstellung eines Beispielgraphens

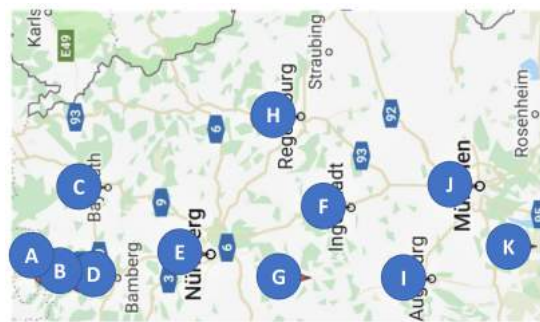
Um die nachfolgenden Erklärungen der unterschiedlichen Algorithmen anschaulich darstellen zu können, wird ein Beispielgraph erstellt. Dafür wird initial eine Auswahl bayrischer Städte getroffen und auf einer Karte markiert (Abb. 3.1(a)). Die Zuordnung der Städte zu den Buchstaben findet sich in Tabelle 3.1. Anschließend werden die benötigten Daten übernommen (Abb. 3.1(b)) und der Zusammenhang zur Karte aufgelöst (Abb. 3.1(c)). Die Erstellung des Graphen wird durch die Auswahl möglicher Kanten und die Berechnung der Pfadkosten finalisiert (Abb. 3.1(d)). In dem vorliegenden Beispiel werden sinnvolle Verbindungen ausgewählt, die keine Kanten durch andere verwendete Städte benötigen. Die Karte wurde mit *batchgeo.com* [39] erstellt und die Entfernungen mit *luftlinie.org* [40] berechnet. Die Städte entsprechen den Knoten des Graphen und deren Verbindungen den Kanten. Die Entfernungen zwischen den Städten, die als Werte auf den Kanten stehen, sind die Kantengewichte.

Tabelle 3.1: Zuteilung der Städte zu Buchstaben

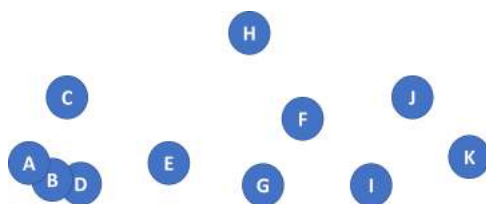
Buchstabe	Stadt	Buchstabe	Stadt	Buchstabe	Stadt
A	Lichtenfels	B	Ebensfeld	C	Bayreuth
D	Bamberg	E	Nürnberg	F	Ingolstadt
G	Treuchtlingen	H	Regensburg	I	Augsburg
J	München	K	Weilheim		



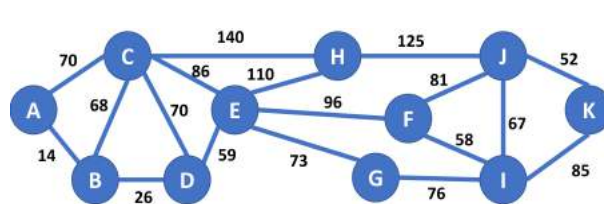
(a) Schritt 1



(b) Schritt 2



(c) Schritt 3



(d) Schritt 4

Abbildung 3.1: Erstellung eines Beispielgraphen

3.4 Optimierung auf Graphen

Die Routenplanung auf Graphen stellt ein gängiges Suchproblem dar [38, S.126]. Gesucht wird eine oder die kürzeste Verbindung zwischen zwei Städten, exemplarisch von Ingolstadt (F) nach Ebensfeld (B) im Graphen aus Abb. 3.1(d).

In diesem Abschnitt werden grundlegende Eigenschaften und Kategorien von Suchalgorithmen definiert (Abschnitt 3.4.1) und anschließend die Algorithmen Tiefensuche (Abschnitt 3.4.2), Breitensuche (Abschnitt 3.4.3), Dijkstra-Suche (Abschnitt 3.4.4) und A* (Abschnitt 3.4.5) anhand des vorgestellten Beispiels erläutert.

3.4.1 Formale Definitionen zu Suchalgorithmen

Suchalgorithmen werden entsprechend den verwendeten Informationen zur Lösungsfindung in die drei folgenden Gruppen eingeteilt [38, S.128]:

- Uninformierte Suchalgorithmen kombinieren blind alle Möglichkeiten durch.
- Informierte Suchalgorithmen verwenden eine problemspezifische Heuristikfunktion, die Metainformationen in den Suchprozess integriert, um schneller zum Ziel zu kommen.
- Lokale Suchalgorithmen suchen minimale oder maximale Funktionswerte im Suchraum.

Tiefen- und Breitensuche sind die beiden Basisalgorithmen, die ausgehend vom Startknoten alle möglichen Kombinationen durchprobieren, um zum Ergebnis zu gelangen [38, S.129]. Sie gehören damit zur Klasse der uninformierten Suchalgorithmen [38, S.125]. Der Dijkstra-Algorithmus verwendet den Gedanken, dass der kürzeste Pfad zwischen zwei Knoten aus kürzesten Teilstrecken aufgebaut sein muss [38, S.136]. Dabei werden keine Metainformationen integriert, sodass er ebenfalls zu den uninformierten Suchalgorithmen gehört. Zur Klasse der informierten Suchalgorithmen gehört der A*-Suchalgorithmus. Informiertes Suchen integriert zusätzliche Informationen in Form einer Bewertungsfunktion in die Auswahl des Pfades durch den Suchbaum [38, S.125]. Die dritte Klasse der lokalen Suchalgorithmen wird verwendet, um minimale oder maximale Funktionswerte zur Lösung von Optimierungsproblemen aufzufinden [38, S.156]. Gängige Vertreter sind beispielsweise das Simulated Annealing und genetische Algorithmen [38, S.125]. Für die vorliegende Arbeit sind uninformierte und vor allem informierte Suchalgorithmen relevant.

Suchalgorithmen werden entsprechend ihrer Eigenschaften klassifiziert, nach Vollständigkeit, Optimalität, Zeit- und Speicherkomplexität [38, S.127]. Auf die beiden Komplexitäts-Kategorien wird nicht näher eingegangen. Eigenschaften, die Suchalgorithmen besitzen können, sind nach DÖRN [38, S.127]:

- *Vollständigkeit*: Fähigkeit, existente Lösungen zu finden
- *Optimalität*: Fähigkeit, Pfad mit niedrigsten Kosten zu finden
- *Korrektheit*: Fähigkeit, tatsächliche Lösungen zu ermitteln
- *Determinismus*: Nachfolgezustände jeder Aktion eindeutig bestimmt

Generell lässt sich sagen, dass informierte Suchalgorithmen deutlich schneller sind als uninformierte, jedoch garantieren sie keine Optimalität [38, S.128]. Für die Praxis sind häufig schnelle, gute Lösungen interessanter als optimale.

3.4.2 Tiefensuche

Die Tiefensuche verwendet ein rekursives Vorgehen, um immer ausgehend vom tiefsten, nicht expandierten Knoten v_i die Suche weiter in die Tiefe fortzusetzen [38, S.129]. Begonnen wird das rekursive Vorgehen beim Startknoten v_s . Entlang der ersten Kante wird der nächste Knoten v_i ausgewählt, getestet, ob v_i bereits betrachtet wurde oder er der Zielknoten v_z ist. Ist beides nicht der Fall, wird für alle weiteren Knoten das Verfahren wiederholt, bis v_z erreicht ist [38, S.132]. Erreicht der Algorithmus einen Knoten mit leerer Nachbarschaft $N \in \emptyset$, kehrt er zum letzten Knoten mit $N \neq \emptyset$ zurück und bildet einen weiteren Ast [38, S.132]. Die formale Beschreibung erfolgt in Algorithmus 1.

Algorithm 1 Tiefensuche (DFS, engl: Depth-First Search) [35, S.44]

```

procedure DFS( $v_i, v_Z$ )
  if  $v_i = v_Z$  then                                ▷ Current Node  $v$  equals Target Node  $v_Z$ ?
    Path found.
  else
     $v_i = \text{NEXTNEIGHBOUR}(v)$                         ▷ Get next neighbour  $v_i$ 
    DFS( $v_i, v_Z$ )                                     ▷ Repeat for next neighbour  $v_i$ 
  end if
end procedure

```

Das Verfahren wird nun anhand des Beispielgraphen (Abb. 3.1) gezeigt, wobei nach alphabetischer Reihenfolge der Knoten vorgegangen wird und nur korrekte Pfade für den ersten Nachbarn (E) dargestellt werden (Abb. 3.2(a)). Die Schrittnummern des Algorithmus zu den unterschiedlichen Knoten sind im Graphen (Abb. 3.2(a)) und im Suchbaum (Abb. 3.2(b)) eingezeichnet. Werden Kanten mehrfach verwendet, stehen sämtliche Schrittnummern an der Kante.

Ausgehend vom Startknoten F wird der erste Nachbarknoten E aus $N \in \{E; I; J\}$ verwendet (Schritt 1). Dessen Nachbarschaft ist $N \in \{C; D; H\}$ sodass C ausgewählt wird (Schritt 2). Anschließend wird Knoten A aus $N \in \{A; B\}$ ausgewählt (Schritt 3) und von dort der Zielknoten B (Schritt 4). Der erste Pfad zum Ziel wurde gefunden, sodass der Algorithmus eine Ebene höher bei Knoten A fortsetzt. Dort ist $N \in \emptyset$, sodass eine Ebene höher für Knoten C erneut der Zielknoten B aus $N \in \{A; B; D\}$ ausgewählt wird, weil A bereits verwendet wurde. Auf diese Weise wird das Verfahren fortgesetzt, bis sämtliche Wege von F zu B gefunden sind. Alternativ könnte bei Auffinden der ersten Lösung der Algorithmus beendet werden.

Die Gesamtlänge (oder Gesamtkosten) der gefundenen korrekten Pfade von links nach rechts in Abb. 3.2(b) betragen 264,68 km - 248,73 km - 276,75 km - 180,01 km - 307,74 km - 291,76 km. Führt man den Algorithmus aus, bis alle möglichen Pfade gefunden wurden, erhält man aus dem Vergleich der Pfade ein vollständiges, optimales, korrektes und deterministisches Vorgehen, das allerdings ineffizient ist.

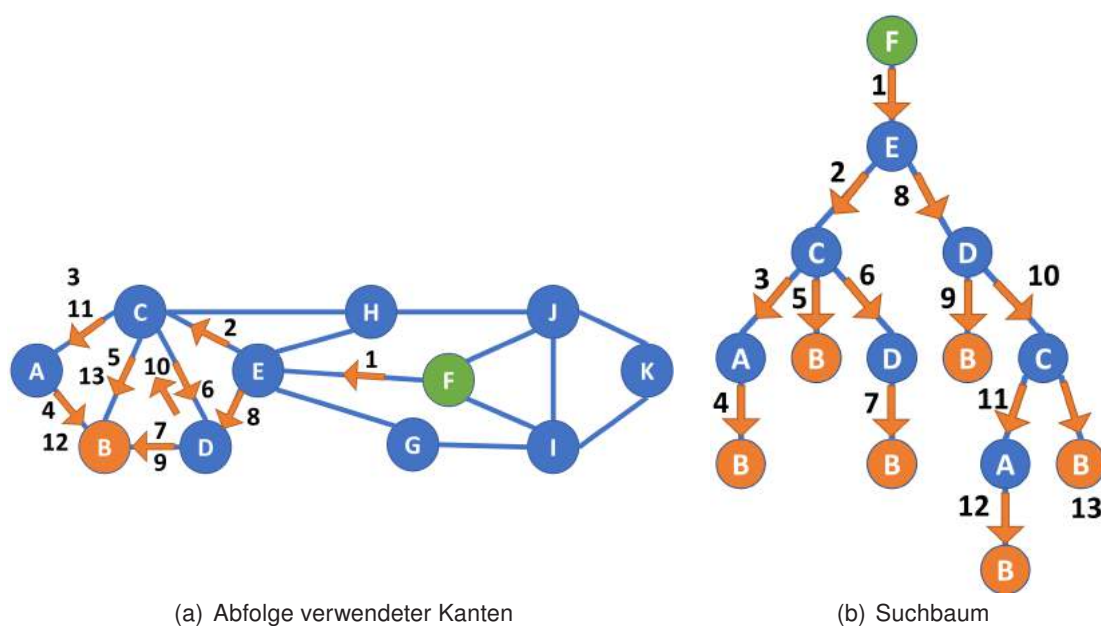


Abbildung 3.2: Ergebnis der Tiefensuche im Beispielgraphen mit Schrittzahl, v_s : grün, v_Z : orange, restliche Knoten: blau

3.4.3 Breitensuche

Die Breitensuche verfolgt grundsätzlich das Prinzip zuerst den flachsten Knoten iterativ zu erweitern [38, S.129], was zu einem „breiten“ Suchbaum führt. Begonnen wird am Startknoten v_s (F: Ingolstadt, grün) und es werden solange alle möglichen Folgeknoten besucht und ans Ende der Liste zu besuchender Knoten angehängen, bis der Zielknoten v_z (B: Ebensfeld, orange) erreicht ist [38, S.129]. Dabei werden zunächst alle adjazenten Knoten N getestet, bevor der Algorithmus zum nächsten Knoten voranschreitet. Algorithmus 2 gibt die formale Definition der Breitensuche.

Algorithm 2 Breitensuche (BFS, engl: Breadth-First Search) [38, S.130]

```

procedure BFS( $v, v_z$ )
  while  $v \neq v_z$  do                                ▷ Current Node  $v$  equals Target Node  $v_z$ ?
     $v = \text{GETFIRSTINLINE}$ 
     $\text{ADDNEIGHBOURLAST}(v)$                                 ▷ Put new nodes at end of queue
    if  $\text{ISEMPTY}(\text{Line})$  then
      No Solution found.
    end if
  end while
end procedure
  
```

Begonnen wird bei v_s . Der Knoten wird dem Suchbaum hinzugefügt. Anschließend werden in alphabetischer Reihenfolge alle adjazenten Knoten $N = \{E, I, J\}$ eingereiht und getestet, wobei deren Nachbarschaften ebenfalls hinzugefügt werden. Weil keiner der drei Knoten der Zielknoten ist, wird zum alphabetisch niedrigsten Knoten der Nachbarschaft, Knoten E, fortgeschritten und dort $N \in \{C; D; G; H\}$ in den Suchbaum aufgenommen. Dieses Verfahren wird angewendet bis in Schritt 15 v_z erreicht wird. Abb. 3.3 zeigt die Ergebnisse der Breitensuche, wobei das Vorgehen der Breitensuche im Beispielsgraphen bis zum Auffinden des ersten Pfades zum Ziel zeigt. In Abb. 3.3(b) werden im Suchbaum des Algorithmus der Reihe nach die Knoten hinzugefügt. Der erste Pfad ist: F - E - C - B bzw. Ingolstadt - Nürnberg - Bayreuth - Ebensfeld, mit einer Gesamtlänge von $95,5 \text{ km} + 85,58 \text{ km} + 67,65 \text{ km} = 248,73 \text{ km}$.

Dieses Vorgehen kann weitergeführt werden, bis sämtliche Pfade durch den Graphen bestimmt wurden. Das Vorgehen liefert erneut ein vollständiges, optimales, korrektes, deterministisches, aber ineffizientes Vorgehen durch den Vergleich der finalen Pfadkosten [38, S.129].

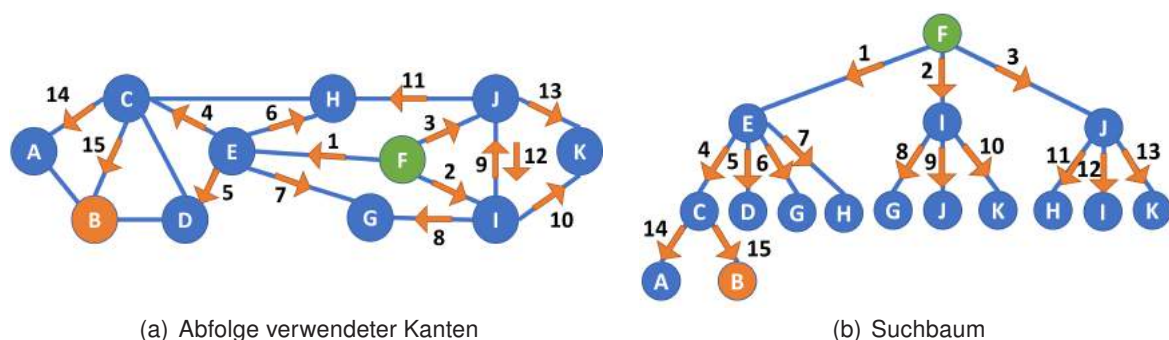


Abbildung 3.3: Ergebnis der Breitensuche im Beispielsgraphen mit Schrittzahl, v_s : grün, v_z : orange, restliche Knoten: blau

3.4.4 Dijkstra-Suchalgorithmus

Die Qualität der Lösung der Breiten- oder Tiefensuche kann erst ermittelt werden, wenn der Algorithmus sämtliche Pfade bestimmt hat. Davor ist keine Aussage über die Optimalität der bisher besten gefundenen Lösung möglich, bzw. nur falls diese bereits a priori bekannt ist. Das führt zu sehr langen Laufzeiten, bis der häufig benötigte kürzeste Pfad zwischen zwei Punkten gefunden wird [38, S.134]. Um diesen effizienter zu finden, macht sich DIJKSTRA [41] eine Beobachtung zu Nutze. Wenn ein Weg der kürzeste Weg zwischen zwei Punkten ist, so müssen sämtliche Teilstrecken zwischen Punkten auf dem Weg ebenfalls kürzeste Wege sein, ansonsten gäbe es eine kürzere Teilstrecke, die somit auch zu einer kürzeren Gesamtstrecke führen würde [38, S.136]. Anders ausgedrückt: Wenn R auf dem kürzesten Pfad von P nach Q liegt, so sind die Pfade von P nach R und R nach Q ebenfalls minimal [41].

Für dieses Verfahren teilt DIJKSTRA die Gesamtheit der Knoten in drei Mengen ein:

- A: Knoten, deren Pfade mit minimalen Kosten vom Startknoten v_s bekannt sind.
- B: Knoten, aus denen der nächste Knoten ausgewählt wird, der zu **A** hinzugefügt wird. Jeder der Knoten in dieser Menge hat mindestens eine Verbindung zu den Knoten aus **A**.
- C: Alle restlichen Knoten

Eine ähnliche Einteilung wird auch für alle Kanten getroffen [41]:

- I: Kanten, aus denen die minimalen Pfade zwischen den Knoten in **A** gebildet werden.
- II: Kanten, aus denen die nächste Kante zu **I** hinzugefügt wird. Jeder Knoten aus **B** ist mit genau einer Kante aus **II** verbunden.
- III: Alle restlichen Kanten

Initial befindet sich nur v_s in **A** und sämtliche Knoten in **C** bzw. Kanten in **III**. Nun wird die Distanz zu sämtlichen Knoten der Nachbarschaft $v_N \in N$ bestimmt und A zu **B** bzw. die inzidenten Kanten zu **II** hinzugefügt. Nun wird aus **B** der Knoten mit kleinster verbindender Kante ausgewählt, zu **A** hinzugefügt und für dessen Nachbarschaft die Distanzen und Pfadkosten ermittelt. Sollte einer der Knoten bereits in **B** enthalten sein, wird getestet, ob der alte Pfad kürzer ist als der neue. Ist dies der Fall, wird der neue verworfen, andernfalls wird der neue abgespeichert. Dies wird so lange ausgeführt, bis das Ziel erreicht ist und ein optimaler Pfad gefunden wurde oder kein Knoten mehr erreichbar ist. Algorithmus 3 erfasst das Vorgehen formal.

Algorithm 3 Dijkstra-Suchalgorithmus nach [41]

```

procedure Dijkstra( $v_i, v_z$ )
  INITIALIZE
  while  $v_i \neq v_z$  do                                ▷ Current Node  $v$  equals Target Node  $v_z$ ?
     $N = \text{GETNEIGHBOURS}(v_z)$ 
     $v = \text{GETNEAREST}$ 
    for all  $v_N \in N$  do                                ▷ Update all Nodes in Neighbourhood
       $d(v_s, v_N) = \text{GETDISTANCE}(v_s, v_N)$ 
      if  $\text{ISNOTKNOWN}(v_N)$  OR  $d(v_s, v_N) < d_{old}$  then ▷ Unvisited Node or shorter Distance?
         $\text{UPDATE}(v_N)$ 
      end if
    end for
  end while
end procedure

```

Die Abbildungen 3.4 und 3.5 zeigen den Algorithmus angewendet auf den Beispielgraphen. Die Werte in den Abbildungen repräsentieren die Kantenkosten bzw. die Fahrtstrecke zwischen den Knoten. Initial befinden sich alle Knoten in **C** und alle Kanten in **III** (Schritt 1). In Schritt 2 wird der Startknoten, im Beispiel Knoten F zu **A** und sämtliche inzidente Kanten (FE , FI , FJ) zu **II** hinzugefügt. Nun wird die Kante FI mit minimalen Kosten 58 und inzidentem Knoten I, ausgesucht.

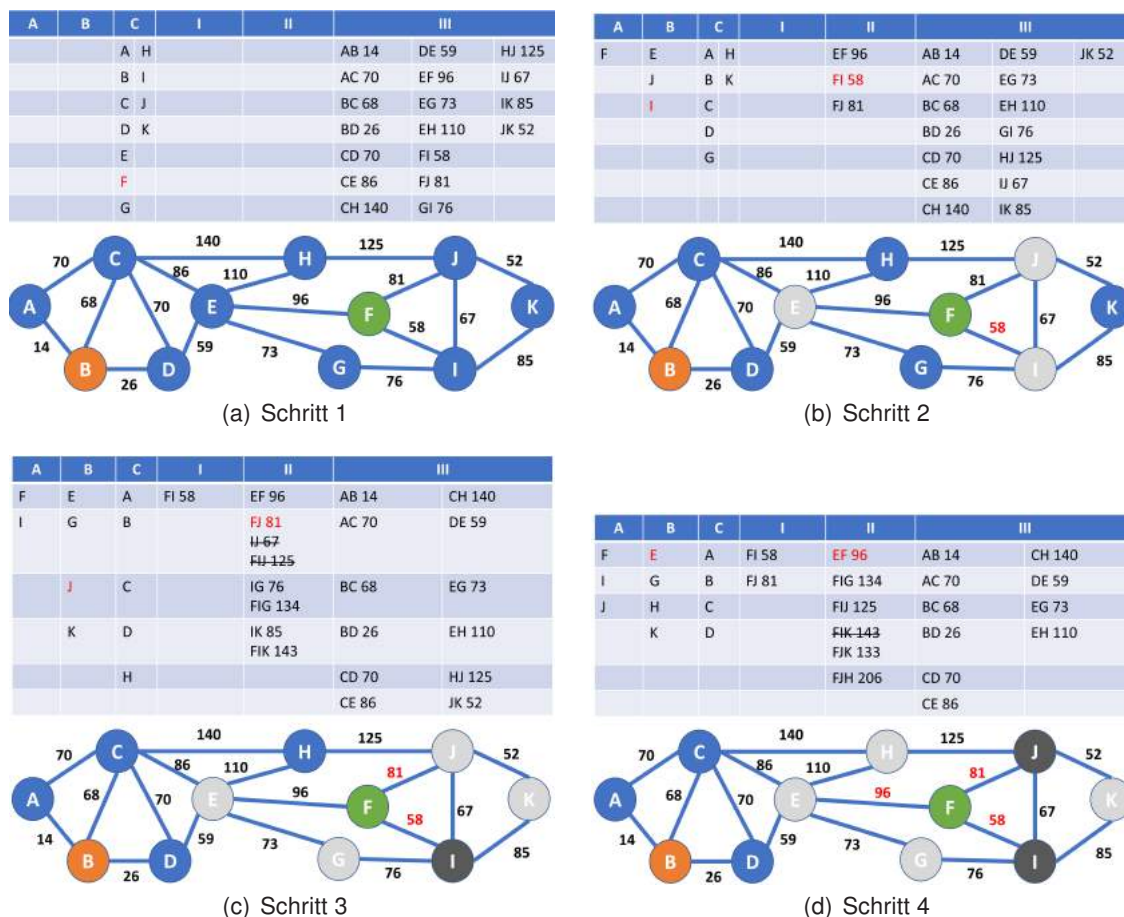


Abbildung 3.4: Ergebnisse des Dijkstra-Suchalgorithmus im Beispielgraphen Schritt 1-4, v_s : grün, v_z : orange, **A**: dunkelgrau, **B**: hellgrau, **C**: blau, Knoten auf Zielpfad: rot, Kantengewichte in I und ausgewählte nächste Kante: rot, Kantengewichte in II: schwarz

Der ausgesuchte Knoten I wird zu **A** hinzugefügt und dessen inzidente Kanten und adjazente Knoten zu **II** und **B** (Schritt 3). Beim Hinzufügen zu **II** werden bestehende Pfade verlängert, z.B. wird Pfad FI - Kosten 58 um IG - 76 erweitert, sodass Pfad FIG - Kosten 134 entsteht. In den Schritten 4 bis 9 wird aufgrund der Übersichtlichkeit nur der resultierende Pfad dargestellt. In Schritt 3 entstehen zwei Pfade zum Knoten J, FJ - Kosten 81 und FIJ - Kosten 125. Entsprechend dem Algorithmus wird der kürzere Pfad abgespeichert, der längere verworfen, im Beispiel durchgestrichen. Auf diese Weise iteriert der Algorithmus wie folgt durch den Graph:

Schritt 4: Hinzufügen von J zu **A**, H zu **B**, FJ zu **I**, FJH & FJK zu **II**,
Auswahl von EF bzw. E

Schritt 5: Hinzufügen von E zu **A**, C & D zu **B**, FE zu **I**, FEG , FEH , FEC & FED zu **II**,
Auswahl von FJK bzw. K

Schritt 6: Hinzufügen von K zu **A**, FJK zu **I**,
Auswahl von FIG bzw. G

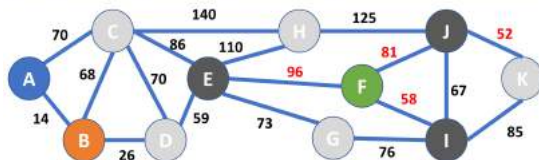
Schritt 7: Hinzufügen von G zu **A**, *FIG* zu **I**,
Auswahl von *FED* bzw. D

Schritt 8: Hinzufügen von D zu **A**, B zu **B**, *FED* zu **I**, *FEDC* zu **II**, *FEDB* zu **II**,
Auswahl von *FEDB* bzw. B, Zielknoten erreicht

Schritt 9: Hinzufügen von B zu **A**, *FEDB* zu **I**

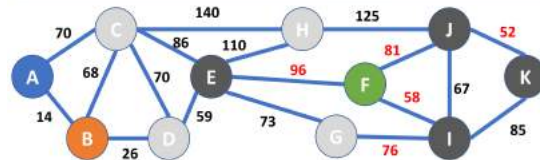
Im finalen Schritt 9 werden die ausgewählten Kanten und Knoten in rot hervorgehoben. Der resultierende optimale Pfad ist *FEDB* mit Kosten 181.

A	B	C	I	II	III
F	C	A	FI 58	FIG 134 FEG 169	AB 14
I	D	B	FJ 81	FJK 143 FIK 133	AC 70
J	G		FE 96	FJH 206 FEH 206	BC 68
E	H			FEC 182	BD 26
	K			FED 155	CD 70
					CH 140



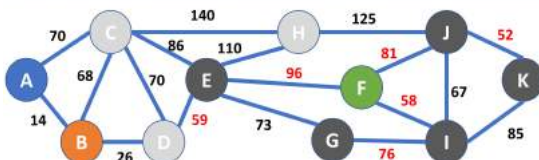
(a) Schritt 5

A	B	C	I	II	III
F	C	A	FI 58	FIG 134 FEG 169	AB 14
I	D	B	FJ 81	FJH 206 FEH 206	AC 70
J	G		FE 96	FEC 182	BC 68
E	H			FED 155	BD 26
K					CD 70
					CH 140



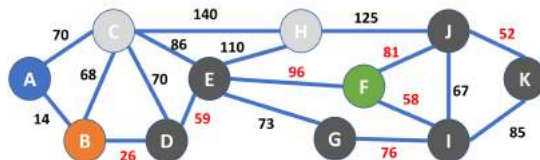
(b) Schritt 6

A	B	C	I	II	III
F	C	A	FI 58	FJH 206 FEH 206	AB 14
I	D	B	FJ 81	FEC 182	AC 70
J	H		FE 96	FED 155	BC 68
E			FJK 133		BD 26
G			FIG 134		CD 70
K					CH 140



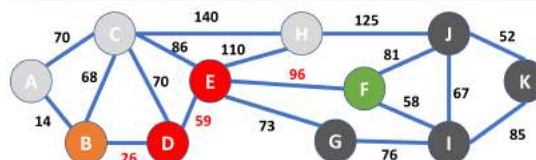
(c) Schritt 7

A	B	C	I	II	III
F	D	B	FI 58	FJH 206 FEH 206	AB 14
I	C		FJ 81	FEC 182 FEDC 225	AC 70
J	H		FE 96	FEDB 181	BC 68
E			FJK 133		CH 140
G			FIG 134		
K			FED 155		



(d) Schritt 8

A	B	C	I	II	III
F	D	A	FI 58	FEDB 181 FIH 206 FEH 206	AC 70
I	B	C	FJ 81	FEC 182 FEDC 225 FEDBC 249	CH 140
J	H		FE 96	FEDBA 195	
E			FJK 133		
G			FIG 134		
K			FED 155		



(e) Schritt 9

Abbildung 3.5: Ergebnisse des Dijkstra-Suchalgorithmus im Beispielgraphen Schritt 5-9, v_s : grün, v_z : orange, **A**: dunkelgrau, **B**: hellgrau, **C**: blau, Knoten auf Zielpfad: rot, Kantengewichte in **I** und ausgewählte nächste Kante: rot, Kantengewichte in **II**: schwarz

3.4.5 Informierte Suche - Der A*-Algorithmus

ERTEL [42, S.112ff] zeigt eine Möglichkeit auf, wie die Rechenzeit von Suchalgorithmen reduziert werden kann und präsentiert die Heuristische Suche als Möglichkeit, den Suchraum einzugrenzen. Heuristiken bezeichnet er als „Problemlösungsstrategien, die in vielen Fällen zu einer schnelleren Lösung führen als die uninformierte Suche“ [42, S.112]. Allerdings kann es auch vorkommen, dass die Rechenzeit im Vergleich zur uninformierten Suche ansteigt oder das Auffinden einer Lösung nicht möglich ist. Es kann also keine allgemeine Optimalität und Vollständigkeit für heuristisches Suchen erteilt werden. Heuristische Entscheidungen zielen auf den Konflikt des Auffindens von Entscheidungen in Realzeit bzw. nicht-unendlicher Zeit ab, weil in der Praxis schnell gefundene gute Lösungen den Nutzen optimaler, aufwändig aufgefundener Lösungen oder Entscheidungen übertreffen.

Die Heuristische Suche als Algorithmus zum beschleunigten Auffinden stellt eine Erweiterung der Breitensuche dar, bei der anstatt von links nach rechts durch den Suchbaum zu explorieren von „gut“ nach „schlecht“ vorgegangen wird [42, S.113]. „Gut“ und „schlecht“ werden durch die Heuristikfunktion bestimmt, die den Wert eines bestimmten Knotens mit möglichst wenig Aufwand ermittelt [42, S.113]. Eine übliche Wahl der Heuristikfunktion h im Kontext des kürzesten Wegeproblems ist die Luftlinienentfernung, die auch im nachfolgenden Beispiel verwendet wird [38, S.144].

In dieser Arbeit wird mit einem weiteren informierten Suchalgorithmus gearbeitet, dem A*-Algorithmus, der den Dijkstra-Algorithmus mit der Heuristischen Suche kombiniert [38, S.148]. Die Heuristische Suche oder Gierige Suche wird infolgedessen nicht näher betrachtet, sondern direkt der A*-Algorithmus vorgestellt. Es wird auf ERTEL [42, S.112ff] und DÖRN [38, S.143ff] verwiesen.

Zusätzlich zur Exploration des aktuell kürzesten Weges (Dijkstra) fokussiert der A*-Algorithmus besser gelegene Knoten in Bezug auf den Zielknoten v_Z . Dafür werden für jeden in Betracht gezogenen Knoten v_i drei Werte in Abhängigkeit des Startknotens v_S und v_Z berechnet [38, S.148]:

- Kostenfunktion [42, S.116] $g(v_i)$: Kosten von v_S nach v_i
- Heuristik [42, S.117] $h(v_i)$: Geschätzte Kosten von v_i nach v_Z
- Heuristische Bewertungsfunktion [42, S.116] $f(v_i) = g(v_i) + h(v_i)$: Geschätzte Kosten von v_S nach v_Z über v_i

$g(v_i)$ wird auch Pfadkosten oder g-Wert genannt, $h(v_i)$ heuristische Schätzfunktion, Heuristikfunktion oder h-Wert und $f(v_i)$ Funktionswert oder f-Wert.

Eine wichtige Bedingung, um den A*-Algorithmus als vollständige und optimale Suchstrategie anzuwenden, ist die sogenannte *Zulässigkeit* der Heuristikfunktion [38, S.148]. Diese ist gegeben, wenn die durch die Heuristik geschätzten Kosten die tatsächlichen Kosten von v_i nach v_Z niemals übersteigt [42, S.116] [38, S.148]. Der A*-Algorithmus ist bei zulässiger Heuristik für jedes lösbare Problem dazu in der Lage, eine optimale Lösung zu finden [42, S.117]. Der Beweis dazu kann ERTEL [42, S.118] und DÖRN [38, S.148] entnommen werden. Das Vorgehen des A*-Algorithmus wird anhand der Pfadfindung von Ingolstadt (F) nach Ebensfeld (B) im Beispielgraphen aus Abb. 3.1(d) erläutert. Als Heuristik wird die Luftlinienentfernung verwendet. Die Werte der Heuristik h der einzelnen Städte sind in Tabelle 3.2 aufgeführt. Die Pfadkosten g werden als Summe aus der Länge des Pfades über alle Knoten auf dem Pfad bis zum aktuellen Knoten berechnet. Die Differenz zweier benachbarter Knoten auf dem Pfad entspricht also genau ihrer Distanz.

Tabelle 3.2: Luftlinienentfernung zwischen Ebensfeld und den Städten des Graphen, berechnet mit *luftlinie.org* [40], Knotennummern nach Tabelle 3.1

Knoten	A	B	C	D	E	F	G	H	I	J	K
Distanz in km	8	0	46	19	69	149	124	143	234	219	249

Die Beschreibung des Vorgehens im Beispielgraph erfolgt nach GRIMME und BOSSEK [35, S.56ff] (Algorithmus 4). Alle Knoten im Suchraum befinden sich in zwei Mengen. Diese sind die Menge **O** (engl.: open set) der noch offenen oder nicht verwendeten Knoten und die Menge **C** (engl.: closed set) der abgeschlossenen oder verwendeten Knoten. Initial befindet sich die Menge aller Knoten **V** weder in **O** noch in **C** und es gilt $\mathbf{O} = \mathbf{C} = \emptyset$. F wird als Startknoten und B als Zielknoten definiert.

Zu Beginn wird in der Funktion INITIALIZE v_s zu **O** hinzugefügt und dessen h -Wert, die Luftlinienentfernung, bestimmt. Er entspricht dem f -Wert, weil der g -Wert als Distanz zu v_s gleich null ist. Anschließend wird die Schleife an Schritten durchlaufen, um v_z zu erreichen. Zuerst wird in CHOOSENODE der Knoten v_i mit minimalem f -Wert aus **O** entnommen, entspricht er v_z , ist das Ziel erreicht. Anschließend wird für alle Knoten v_N der Nachbarschaft N der Wert $g(v_N) = g(v_i) + d(v_i, v_N)$ mit der Distanz $d(v_i, v_N)$ zwischen v_i und v_N bestimmt (CALCULATECOSTS). In UPDATECOSTS werden die Nachbarknoten in die Menge **O** einsortiert. Falls v_N bereits ein Element in **O** war und der aktuell berechnete Wert $g(v_i) + d(v_i, v_N)$ kleiner ist als der bisherige Wert $g(v_N)$, wird $g(v_i) + d(v_i, v_N)$ übernommen und v_i als Vorgänger von v_N gespeichert. Falls v_N noch nicht in **O** enthalten ist, wird v_N mit dem Vorgänger v_i zu **O** hinzugefügt.

Algorithm 4 A*-Algorithmus nach GRIMME und BOSSEK [35, S.58]

```

1: procedure A*(V)
2:   INITIALIZE
3:   while O  $\neq \emptyset$  do                                      $\triangleright$  Open Set is empty?
4:      $v_i = \text{CHOOSENODE}(\mathbf{O})$ 
5:     if  $v_i = v_z$  then
6:       terminate
7:     end if
8:     for all  $v_N \in N$  do                                        $\triangleright$  Update all Nodes in Neighbourhood
9:       CALCULATECOSTS
10:      UPDATEOPENSET
11:    end for
12:  end while
13:  If this step is reached, no solution could be found.
14: end procedure

```

Angewendet auf das Beispiel entsteht die Abfolge aus Abb. 3.6. Als Erstes wird der Startknoten F zu **O** hinzugefügt und $f(F) = h(F) + g(F) = 149$ bestimmt. Anschließend wird die Schleife gestartet:

Abb. 3.6 Schritt 1: Entnehme F aus **O** und füge ihn **C** hinzu.

- Bestimme g , h und f für E, I und J.
- Kein Element war bereits in **O** enthalten.
- Füge E, I und J mit dem Vorgänger F zu **O** hinzu.

Abb. 3.6 Schritt 2: Entnehme E aus **O** und füge ihn **C** hinzu.

- Bestimme g , h und f für C, D, G und H.
- Kein Element war bereits in **O** enthalten.
- Füge C, D, G und H mit dem Vorgänger E zu **O** hinzu.

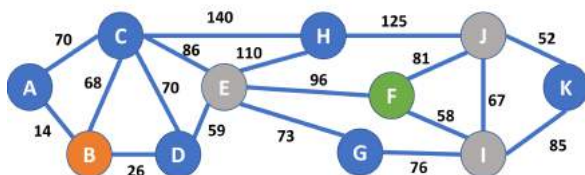
Abb. 3.6 Schritt 3: Entnehme D aus **O** und füge ihn **C** hinzu.

- Bestimme g , h und f für C und B.
- Keine Aktualisierung von $g(C)$ und $f(C)$, weil $g(C) = g(D) + d(D, C) = 225 > g(C) = g(E) + d(E, C) = 182$.
- Füge B mit Vorgänger D **O** hinzu.

Abb. 3.6 Schritt 4: Entnehme B aus **O**, weil **O** = v_Z gilt, terminiert der Algorithmus.

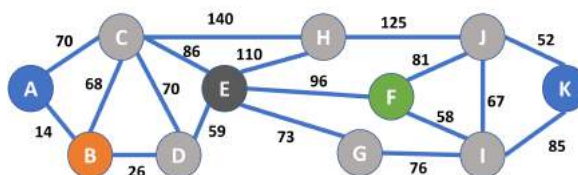
A* bestimmt den optimalen Pfad F-E-D-B mit Kosten 181 bereits nach vier Schritten.

Knoten	A	B	C	D	E	F	G	H	I	J	K
$g(v_i)$					96	0			58	81	
$h(v_i)$					69	149			234	219	
$f(v_i)$					165	149			289	300	
Vorgänger					F	-			F	F	



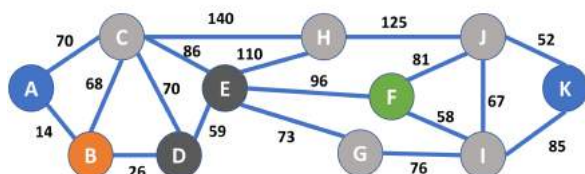
(a) Schritt 1

Knoten	A	B	C	D	E	F	G	H	I	J	K
$g(v_i)$			182	155	96	0	169	206	58	81	
$h(v_i)$			46	19	69	149	124	143	234	219	
$f(v_i)$			228	174	165	149	293	349	289	300	
Vorgänger			E	E	F	-	E	E	F	F	



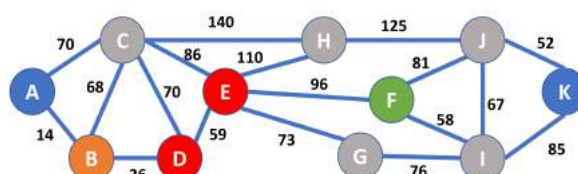
(b) Schritt 2

Knoten	A	B	C	D	E	F	G	H	I	J	K
$g(v_i)$		181	182	155	96	0	169	206	58	81	
$h(v_i)$		0	46	19	69	149	124	143	234	219	
$f(v_i)$		181	228	174	165	149	293	349	289	300	
Vorgänger		D	E	E	F	-	E	E	F	F	



(c) Schritt 3

Knoten	A	B	C	D	E	F	G	H	I	J	K
$g(v_i)$		181	182	155	96	0	169	206	58	81	
$h(v_i)$		0	46	19	69	149	124	143	234	219	
$f(v_i)$		181	228	174	165	149	293	349	289	300	
Vorgänger		D	E	E	F	-	E	E	F	F	



(d) Schritt 4

Abbildung 3.6: Ergebnisse des A*-Suchalgorithmus im Beispielgraphen, v_s : grün, v_Z : orange, **C**: dunkelgrau, **O**: hellgrau, Restliche Knoten: blau, Knoten auf Zielpfad: rot

3.4.6 Vergleich der Algorithmen

Die Qualität und Geschwindigkeit der Breiten- und Tiefensuche ist ausschließlich durch die Struktur des Graphen bzw. Suchbaums definiert [38, S.143]. Der Beispielgraph in der verwendeten Konfiguration eignet sich gut für die Tiefensuche, tatsächlich findet der Algorithmus bereits nach vier Schritten den ersten und nach neun Schritten den optimalen Pfad *FEDB* durch den Graphen. Das entspricht der Anzahl an Schritten des Dijkstra-Algorithmus. Würde man den Algorithmus zuerst spätere Buchstaben im Alphabet auswählen lassen oder das Ziel in Knoten K legen, ergäbe sich ein völlig anderer Suchbaum mit deutlich höherer Schrittzahl bis zum Auffinden des Ergebnisses. Dieser Einfluss ist auf den Dijkstra-Algorithmus nicht gegeben. Die Breitensuche hingegen benötigt 16 Schritte.

Bereits am aktuellen Beispiel mit nur wenigen Städten erkennt man, dass uninformierte Suchalgorithmen eine hohe Laufzeit bzw. hohe Rechenleistung erfordern [42, S.111]. Grundsätzlich liegt das Problem darin begründet, dass der Suchraum vollständig erforscht wird, woraus sich eine hohe Zeitkomplexität ergibt, bzw. darin, dass der Suchraum vollständig erforscht wird, detaillierter in ERTEL [42, S.111]. Dieses Problem kann der A*-Algorithmus effizient lösen. Durch die Heuristik wird die Suche Richtung Ziel gelenkt. Dies liefert für große Netze, wie sie in dieser Arbeit verwendet werden, die Möglichkeit, die Laufzeit gegenüber uninformierten Suchalgorithmen deutlich zu reduzieren. Infolgedessen wird dieser Algorithmus für die spätere Anwendung ausgewählt.

3.5 Evolutionäre Algorithmen

Evolutionäre Algorithmen (EA) sind nach GRIMME und BOSSEK [35, S.191] „von der Natur inspirierte algorithmische Konzepte und Verfahren“, deren Funktionsweise sich an der Evolutionstheorie nach Darwin orientiert. Diese werden mit Darwins Begriffen Variation, Vererbung, Reproduktion und „Kampf ums Überleben“ bzw. natürliche Selektion analysiert, um dessen Theorie der Anpassung der Arten in der Natur als Grundgedanken der EA einzuführen [35, S.196f]. Die Variation des Erbgutes wird als Haupttreiber der Veränderung der Arten hin zu kompetitiveren Individuen angeführt. Einerseits unterliegt das Erbgut zufälligen Veränderungen, den Mutationen, andererseits wird in der (sexuellen) Reproduktion der Individuen das Erbgut der Elterngeneration in der Rekombination neu zusammengefügt. Die natürliche Selektion entscheidet, welche neuen Individuen besser geeignet sind als andere und sich somit durchsetzen, fortpflanzen und ihr Erbgut für die nächste Evolutionsgeneration zur Verfügung stellen können. Die These ist, dass sich auf diese Mechanismen aufbauend inkrementelle Veränderungen der Eigenschaften und Charakteristika der einzelnen Lebewesen ausbilden und somit neue Arten entstehen können. Die metaphorisch benannten EA zielen darauf ab, diesen natürlichen Prozess der „zielgerichteten Optimierung“ auf weitere Problemstellungen zu übertragen.

Häufige Praxisanwendung finden diese Algorithmen im Bereich von Optimierungsproblemen. Ein Grundgedanke sämtlicher EA ist entsprechend der randomisierten Erbgutveränderung die Variation durch Zufall, um die Exploration im Suchraum zu erhöhen [35, S.192]. Zu Beginn der Entwicklung zufallsbasierter Algorithmen wurde vermutet, dass diese eine schlechtere Laufzeit bei ähnlicher Lösungsqualität besitzen, wobei für die Betrachtung eine rein zufällige Auswahl des Nachfolgers herangezogen wurde [35, S.194]. Tatsächlich verschlechtert sich die Laufzeit bei intelligenterer Auswahl der zufällig generierten Lösungen (= Selektion) nicht und insbesondere mit üblichen Verfahren schwer lösbare Probleme können dadurch angegangen werden [35, S.194].

Ein häufig verwendeter Algorithmus aus der Klasse der EA ist die sogenannte genetische Optimierung. Diese ist direkt an der Mutation von Genen orientiert und eignet sich, um den Transfervorgang aus der Natur in einen Algorithmus zu erklären. Ausgehend von einer zufälligen oder voroptimierten Initialpopulation an Lösungskandidaten werden durch eine problemabhängige Bewertungsfunktion die besten selektiert und eine Kindergeneration gleicher Größe durch Rekombination gebildet [38, S.169ff]. Diese wird anschließend durch Mutation variiert. Die Schritte Selektion, Rekombination und Mutation werden eine definierte Anzahl Generationen wiederholt. Aufgrund der Selektion der besten Individuen wird sichergestellt, dass nachfolgende Lösungen in der Nähe der aktuell besten generiert werden und der Algorithmus Richtung (lokalen) Optima konvergiert [38, S.169ff].

Klassische Anwendungsbeispiele sind das Rucksackproblem [38, S.178], die Maschinenkalibrierung [38, S.178] oder das Traveling Salesman Problem (TSP) [35, S.204f]. Üblicherweise wird die genetische Optimierung für diskret beschreibbare Probleme angewendet, bei denen ein Optimum einer Funktion gefunden werden soll. Für Pfadfindungsprobleme ist er nicht geeignet. Das TSP bildet eine Ausnahme, weil dort eine Abfolge zu besuchender Knoten definiert wird, wobei die Knotenanzahl vergleichsweise gering ist zu Pfadfindungsproblemen im offenen Raum.

3.5.1 Ameisen-Kolonie-Optimierung

Der Gedanke, Pfadfindungsstrategien in der Natur zu analysieren und zu übertragen, liegt nahe. Es existiert eine Vielzahl Algorithmen, die von Beobachtungen unterschiedlichster Tierarten abgeleitet sind. Sehr populär sind die Schwarmalgorithmen (PSO, engl.: Particle Swarm Optimization) und die Ant-Colony-Optimization (ACO, dt.: Ameisen-Kolonie-Optimierung) [35, S.218] [34, S.12]. DU und SWAMY [43, S.153ff] erklären die Funktionsweise des PSO ausführlich, an dieser Stelle wird nur der Grundgedanke wiedergegeben. PSO leitet sich aus der Beobachtung von Fisch-, Vogel- und Bienen-Schwärmen ab [43, S.153]. Die Gesamtpopulation wird aus den Individuen, auch Partikeln genannt, aufgebaut. Diese bewegen sich sowohl entsprechend der Richtung der verangenen Bewegung, als auch der Bewegung des Schwarmes. Für den Algorithmus bedeutet das, dass sich die Koordinaten der einzelnen Partikel entsprechend der Überlagerung der Richtung des eigenen bisherigen Optimums und der Richtung zum Optimum des Schwarms verändern, um das Optimum des Suchraumes zu finden [43, S.153].

In dieser Arbeit wird als Vergleichsobjekt für den A*-Algorithmus ein Algorithmus aus der Klasse der ACO erarbeitet. DORIGO et al. [44] haben mit dem Ant System (AS, dt.: Ameisen System) einen naturinspirierten Algorithmus zur Lösung des TSP eingeführt und den Grundstein für sämtliche ACO gelegt. Den Ausgangspunkt bildet die Beobachtung, dass reale, quasi blinde Ameisen in der Lage sind, kürzeste Wege zwischen ihrem Nest und einer Nahrungsquelle zu finden [44, S.29]. Das Auffinden dieses Optimums ist nur der Ameisenkolonie möglich, nicht einer einzelnen Ameise, deren Wahrnehmung auf ihre nahe Umgebung begrenzt ist [35, S.221]. Die Basis der Pfadfindung von Ameisen sind Pheromone, chemische Stoffe. Diese werden von den Ameisen entlang ihrer beschrittenen Pfade abgelagert und verdunsten mit der Zeit [45, S.319ff]. Befinden sich keine Pheromone in Wahrnehmungsreichweite der Ameisen, bewegen sie sich in rein zufälligen Bewegungsmustern. Wird jedoch Pheromon wahrgenommen, steigt die Wahrscheinlichkeit, dass eine Ameise diesen pheromonangereicherten Pfad auswählt, proportional mit der Pheromonkonzentration [45, S.319ff].

Abb. 3.7 zeigt dieses Verhalten von Ameisen auf der Nahrungssuche, wobei die Ameisen auf den drei Pfaden 1-3 zur Kirsche gelangen können. Zu Beginn befinden sich auf keinem der Pfade Pheromonablagerungen, sodass sich die Ameisen anfänglich ohne Präferenz relativ gleichmäßig pro Pfad aufteilen werden. Offensichtlich ist Pfad 2 der kürzeste, gefolgt von Pfad 3

und dann Pfad 1. Entsprechend benötigen die Ameisen auf Pfad 2 am wenigsten Zeit für die Wegstrecke zur Nahrungsquelle und zurück. Nun begeben sich die Ameisen von Pfad 2 erneut auf den Weg zur Nahrungsquelle. Von den drei Pfaden ist die Pheromonzusammensetzung auf Pfad 2 jetzt am höchsten, sodass sich die Ameisen tendenziell für diesen Pfad entscheiden und sich eine selbstverstärkende Zunahme („autokatalytisches Verhalten“ [35, S.223]) des Pheromons durch die steigende Anzahl an Ameisen entwickelt. Eine ausführliche Erklärung findet sich in DORIGO et al. [44, S.29-30] bzw. DORIGO und STÜTZLE [44, S.319-320].

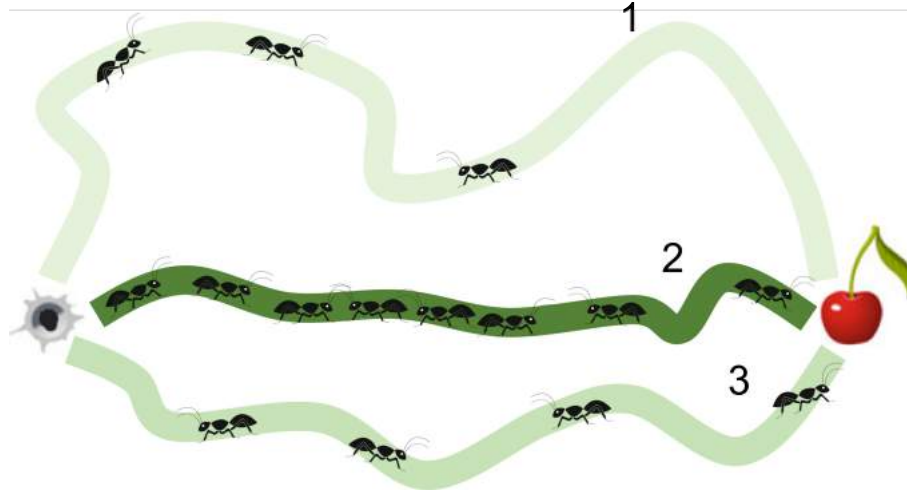


Abbildung 3.7: Verhalten von Ameisen auf der Futtersuche nach MIRJALILI [34, S.34], Farbsättigung der Pfade nimmt mit steigender Pheromonzusammensetzung zu

Aus dieser Strategie zur Ermittlung kürzester Pfade extrahieren DORIGO et al. [44, S.30] das AS inklusive der verwendeten *Ant Algorithms*, um komplexe Optimierungsprobleme zu lösen. Im Unterschied zur ursprünglichen Erklärung des AS als Algorithmus im Kontext des TSP wird es an dieser Stelle leicht abgewandelt anhand einer Pfadfindung zwischen zwei Knoten vorgestellt. Betrachtet wird eine Routenplanung vom Startknoten v_S zum Zielknoten v_Z über die Knoten $v_i \in V$ und deren verbindenden Kanten zu ihren adjazenten Knoten der Nachbarschaft $v_N \in N$. Die euklidische Distanz zwischen zwei Knoten v_i und v_j bzw. die Länge der Kante ij wird mit d_{ij} bezeichnet. Für jede Ameise $a \in A$, der Menge der Ameisen wird initial eine zufällige Lösung in Form eines Pfades P_a mit L_a zwischen v_S und v_Z erstellt. In der Natur legen Ameisen Pheromone während des Laufens ab. Für den Algorithmus wird jedoch zuerst für jede Ameise ein Pfad gesucht und anschließend die Pheromonmenge τ_{ij} auf allen Kanten ij neu berechnet, weil andernfalls jede Ameise die Bedingungen für sämtliche folgenden Ameisen verändert. Initial erhält jede Kante die Pheromonmenge τ_0 zugewiesen. Die zufälligen Lösungen der Ameisen a werden bestimmt durch die Auswahl des Knoten $v_j \in N$ ausgehend vom Knoten v_i entsprechend der Übergangswahrscheinlichkeit p_{ij}^a in einer Rouletterad-Selektion:

$$p_{ij}^a = \begin{cases} \frac{\tau_{ij}^\alpha \eta_{ij}^\beta}{\sum_{j \in N} \tau_{ij}^\alpha \eta_{ij}^\beta} & \text{mit } \eta_{ij} = \frac{1}{d_{ij}} \quad , \text{ wenn } j \text{ zulässig} \\ 0 & , \text{ sonst} \end{cases} \quad (3.2)$$

Die Attraktivität η_{ij} der Kante ij gibt einen Wert an, der für kürzere Kanten ein höheres Interesse durch die Ameisen erzeugt, weil die Länge der Kante d_{ij} invers exponentiell eingeht. Die Wahrscheinlichkeit, dass eine Ameise eine Kante ij auswählt, steigt mit steigender Pheromonmenge und sinkender Kantenlänge. Es gilt $\alpha, \beta \geq 0$, wobei α als der Einfluss des Pheromons und β als der Einfluss der Attraktivität interpretiert werden kann. Jeder Knoten darf nur einmal pro Route besucht werden. So findet jede Ameise einen mehr oder weniger guten Pfad v_S nach v_Z .

Anschließend wird für jede Ameise die Pheromonmenge $\Delta\tau_{ij}^a$ entsprechend Gleichung (3.3) berechnet, die sie auf jede Kante $ij \in P_a$ ablegt. Q ist ein konstanter Pheromonablage-Wert.

$$\Delta\tau_{ij}^a = \begin{cases} \frac{Q}{L_a} & , \text{wenn } ij \in P_a \\ 0 & , \text{sonst} \end{cases} \quad (3.3)$$

Die insgesamt abgelegte Pheromonmenge $\Delta\tau_{ij}$ aller Ameisen pro Kante ergibt sich nach Gleichung (3.4) als die Summe der Pheromonmenge, die jede einzelne Ameise auf der Kante abgelegt hat:

$$\Delta\tau_{ij} = \sum_{a \in A} \Delta\tau_{ij}^a \quad (3.4)$$

Abschließend wird die neue Pheromonmenge τ_{ij} aller Kanten berechnet, die sich aus der hinzugekommenen Gesamtpheromonmenge $\Delta\tau_{ij}$, der bisherigen Pheromonmenge τ_{ij} und dem Pheromonevaporationskoeffizient $\rho \in (0, 1]$ ergibt:

$$\tau_{ij} = \rho\tau_{ij} + \Delta\tau_{ij} \quad (3.5)$$

Der Faktor ρ hat entscheidenden Einfluss auf das Evaporationsverhalten, weil er durch die Verminderung der Pheromonmenge die Wahrscheinlichkeit reduziert, dass beste Pfade mit hohen Kosten aus schlechten Generationen langfristigen Einfluss haben. Der absinkende Pheromonpegel ermöglicht den Ameisen andere Pfade auszuwählen.

Dieser Turnus aus ...

1. Pfadkonstruktion: BUILDPATH - Gleichung (3.2)
2. Pheromonablage: CALCULATEPHEROMONE - Gleichung (3.3)
3. Pheromonzuwachs: CALCULATETOTALPHEROMONE - Gleichung (3.4)
4. Pheromonneuberechnung: UPDATEPHEROMONE - Gleichung (3.5)

... wird eine definierte Anzahl Generationen (oder Iterationen) n_{Gen} wiederholt, womit sich Algorithmus 5 ergibt. Zusätzlich wird abgebrochen, falls alle Ameisen denselben Pfad auswählen. In jeder Iteration wird der kürzeste Pfad abgespeichert.

Algorithm 5 Ant System Algorithmus

```

procedure ACO(V)
  INITIALIZE
  for  $n = 1 : n_{Gen}$  do                                ▷ Number of Generations
    for all  $a \in A$  do                                    ▷ Number of Ants
      while  $v_i \neq v_Z$  do                                ▷ Current Node equals Target Node
        BUILDPATH
      end while
      CALCULATEPHEROMONE
    end for
    for all  $v_i \in v_Z$  do                                ▷ Update all Nodes in Grid
      CALCULATETOTALPHEROMONE
    end for
    UPDATEPHEROMONE
  end for
end procedure

```

Mit den Gleichungen (3.2) bis (3.5) ist das AS vollständig beschrieben. Auf Basis dieses Grundsystem definieren DORIGO et al. [44, S.31] die *Ant Algorithms*: *ant-cycle*, *ant-density* und *ant-quantity*. Diese unterscheiden sich lediglich in der Art und Weise der Pheromonaktualisierung. Der *ant-cycle* aktualisiert nach jeder Iteration auf die vorgestellte Weise, der *ant-density* nach jeder Auswahl einer Kante ij um den Wert $\Delta\tau_{ij}^a = Q$ und der *ant-quantity* nach jeder Auswahl einer Kante ij um den Wert $\Delta\tau_{ij}^a = Q/L_a$. Es hat sich gezeigt, dass *ant-density* und *ant-quantity* dem *ant-cycle* unterlegen sind, sodass sich der *ant-cycle* etabliert hat und häufig, auch in dieser Arbeit, als AS referenziert wird [45, S.321]. Als Hauptgrund wird bereits bei der ursprünglichen Einführung der Systeme angeführt, dass sich der *ant-cycle* auf globale Informationen stützen kann, wohingegen die anderen beiden Varianten auf lokalen Informationen basieren [44, S.33].

In einer Parameterstudie haben DORIGO et al. [44] die Einflüsse der Parameter α, β, ρ, Q untersucht [44, S.33ff]. Für α ergibt sich, dass ein hoher Wert eine starke Fokussierung auf die Pheromonmenge verursacht. Dadurch stagniert der Algorithmus früh, ohne den Suchraum ausreichend zu erkunden, und liefert keine gute Lösung. Zu niedrige Werte bedingen ein ähnliches Verhalten wie beispielsweise ein A*-Algorithmus bei Verwendung der Entfernung als Heuristik. Der Einfluss der Kantenlänge ist dann sehr hoch und es wird sich nicht ausreichend an der Pheromonmenge orientiert. Für β wird ein Wert von 1 – 5 in Abhängigkeit von α vorgeschlagen, um eine akzeptable Gewichtung der Kantenlänge zu erreichen. Ein ausgewogenes Maß aus initial hoher Exploration und Konvergenz zu guten Pfaden durch Ausnutzen der Pheromonspuren wird bei $\rho = 0,5$ beobachtet. Der Parameter Q hat so gut wie keinen Einfluss und ist in der Studie auf 100 gesetzt. Die Studie kann im Detail in DORIGO et al. [44, S.32ff] nachvollzogen werden. Zusammenfassend werden folgende Bereiche der Parameter für gute Ergebnisse empfohlen:

- $0.5 \leq \alpha \leq 1$, $\alpha = 0.5$ nur für $\beta < 2$
- $1 \leq \beta \leq 5$
- $\rho = 0,5$
- $Q \geq 0$

3.5.2 Varianten und Verbesserungen der ACO

Die beschriebene Basisvariante der ACO-Algorithmen lässt sich in zwei Phasen einteilen, die Konstruktionsphase, in der die Pfade nach Gleichung (3.2) berechnet werden und die Pheromonphase, in der nach den Gleichungen (3.3) bis (3.5) die Pheromonmenge für die nächste Generation umgerechnet wird [34, S.31f]. Es existieren viele Ansätze, die den Basisalgorithmus abwandeln, verändern oder verbessern. Häufig geschieht dies durch Hinzufügen einer „Dämon Phase“ (engl.: daemon phase) [34, S.31f] in der die „Dämon Aktionen“ (engl.: daemon actions) ausgeführt werden. Diese sind problemspezifische oder zentralisierte Aktionen, die der Ergebnisverbesserung dienen [45, S.318].

Elitist Strategy

Die erste und bereits von DORIGO et al. [44, S.36] vorgeschlagene Variante ist die „elitist strategy“ (dt.: elitäre Strategie), die dem Gedanken der natürlichen Selektion, dem Überleben der bestangepassten Form, entspricht [44, S.36]. Dafür verstärken sie die abgelegte Pheromonmenge der $m \leq m_{\text{best}}$ besten Ameisen durch zusätzliche Addition von Pheromon $\Delta\tau_{ij}^{\text{best}} = mQ/L_{\text{best}}$ entlang der Knoten des besten Pfades P_{best} , wobei L_{best} die Länge des aktuell besten Pfades ist [44, S.36].

Rank-based Ant System

BULLNHEIMER et al. [46] präsentieren eine rangbasierte Variante des AS, das rank-based Ant System (AS_{rank} , dt.: rangbasiertes Ameisensystem). Sie schlagen vor, die Ergebnispfade der Ameisen mit steigender Länge zu sortieren, um die Pheromonaktualisierung der Knoten entsprechend dem Rang μ der ω besten Pfade durchzuführen. Damit soll verhindert werden, dass sich bei Annäherung der Pfadlängen der einzelnen Ameisen die Pheromonmenge auf guten, aber suboptimalen Pfaden nicht zu stark angleicht. Das hätte zur Folge, dass die Unterschiede der Auswahlwahrscheinlichkeiten p_{ij}^a nach Gleichung (3.2) verschwinden und der Algorithmus stagniert [46, S.7].

Die Pfade der μ -besten Ameisen werden in der Pheromonveränderung nach Gleichung (3.3) mit dem Faktor $\sigma - \mu$ versehen, wobei $\omega = \sigma - 1$ Ameisen bzw. Pfade betrachtet werden. Es wird zusätzlich zu den ω -besten Ameisen die beste Ameise erneut verstärkt. Diese Veränderungen drücken sich in den Gleichungen (3.6) bis (3.8) im Vergleich zu den Gleichungen (3.3) und (3.4) aus. Variablen, die sich auf den besten Pfad P_{best} beziehen, werden mit *best* gekennzeichnet.

$$\Delta\tau_{ij} = \sum_{\mu=1}^{\sigma-1} \Delta\tau_{ij}^{\mu} \quad (3.6)$$

$$\Delta\tau_{ij}^{\mu} = \begin{cases} (\sigma - \mu) \frac{Q}{L_{\mu}} & , \text{wenn } ij \in P_{\mu} \\ 0 & , \text{sonst} \end{cases} \quad (3.7)$$

$$\Delta\tau_{ij}^{\text{best}} = \begin{cases} \sigma \frac{Q}{L_{\text{best}}} & , \text{wenn } ij \in P_{\text{best}} \\ 0 & , \text{sonst} \end{cases} \quad (3.8)$$

Die Parameterwerte α, β, ρ, Q werden von BULLNHEIMER et al. [46, S.9] wie bereits von DORIGO et al. [44] empfohlen. Darüber hinaus konnten BULLNHEIMER et al. [46] zeigen, dass der AS_{rank} -Algorithmus gerade für Problemstellung mit einer großen Anzahl an Knoten und Kanten Vorteile gegenüber anderen Algorithmen hat. In der Studie werden zwei Simulated Annealing Algorithmen, ein genetischer Algorithmus und drei ACO-Algorithmen ($AS, AS_{\text{elite}}, AS_{\text{rank}}$) verglichen [46, S.10-12], wobei der AS_{rank} die besten Ergebnisse erzielt.

Ant Colony System

Das Ant Colony System (ACS, dt.: Ameisen Kolonie System) wird durch DORIGO et al. [47] eingeführt, mit dem ursprünglichen Ziel, die Suche nach dem kürzesten Weg in TSPs effizienter zu gestalten und eine bessere Balance zwischen Exploration und Ausnutzung des Suchraumes zu finden [47, S.53]. Die drei Hauptunterschiede zwischen ACS und AS sind:

1. Die Auswahlregel des nächsten Knotens
2. Die globale Pheromonaktualisierung nach einer elitist strategy mit einer Ameise
3. Das Verwenden einer lokalen Pheromonaktualisierung (vgl. *ant-density/-quantity*)

An dieser Stelle wird nur auf 1. eingegangen, 2. und 3. können DORIGO et al. [47, S.55ff] entnommen werden.

Ist eine Zufallszahl $q \in [0,1]$ größer als ein Grenzwert $q_0 \in [0,1]$, wird nach wie vor die Auswahlstrategie S_{AS} des AS verwendet, auch Random-Proportional Rule (RPR, dt.: zufällig-proportional Regel) genannt. Ansonsten wird nach der Pseudo-Random-Proportional Rule

(PRPR, dt.: pseudo zufällig-proportional Regel) des ACS die Kante mit dem höchsten Wert von $\tau_{ij}^\alpha * \eta_{ij}^\beta$ der Nachbarschaft verwendet [47, S.55f]. Das hat zur Folge, dass die Auswahl der vielversprechendsten Kanten in einem gewissen Prozentsatz der Fälle sichergestellt wird und attraktive Kanten an Einfluss gewinnen. Formal ausgedrückt wird dieser Zusammenhang in Gleichung (3.9) beschrieben.

$$j = \begin{cases} \operatorname{argmax}_{v_j \in N} \{\tau_{ij}^\alpha * \eta_{ij}^\beta\} & , \text{ wenn } q \leq q_0 \\ S_{AS} & , \text{ sonst} \end{cases} \quad (3.9)$$

Als Parameter werden $\alpha = \rho = 0,1$ sowie $\beta = 2$ und $q_0 = 0,9$ verwendet. Ein Aspekt der Studie gilt der Auswahl der elitären Ameisen, wobei zwischen *iteration-best*, dem Verwenden der besten Lösung der aktuellen Iteration, und *global-best*, dem Verwenden der insgesamt besten Lösung, unterschieden wird [47, S.56]. Die besseren Ergebnisse erzielt *global-best*. Es kann eine gute Eignung des ACS für Wegfindungsprobleme bei großer Datenstruktur nachgewiesen werden [47, S.59ff].

AntStar

Als letzte Variante wird der AntStar-Algorithmus nach FAISAL et al. [48] betrachtet. Der Grundgedanke ist die Integration der heuristischen Bewertungsfunktion $f(v_j) = g(v_j) + h(v_j)$ mit den Pfadkosten g und der Heuristikfunktion h des A*-Algorithmus in die Berechnung der Übergangswahrscheinlichkeiten nach Gleichung (3.10) [48, S.4]. Dies hat vor allem bei der Initialisierung Vorteile, weil zu Beginn keine Pheromonspuren vorhanden sind und das AS eine rein stochastische Suche durchführt [48, S.4]. Der neue Ansatz verwendet die A*-Zielorientierung, um die Pfadsuche Richtung Zielknoten zu lenken. Das grundsätzliche System des AS bleibt erhalten, es wird lediglich die Übergangswahrscheinlichkeit der Ameise a von Knoten i nach Knoten j nicht mehr nach Gleichung (3.2) bestimmt, sondern nach:

$$p_{ij}^a = \begin{cases} \frac{\tau_{ij}^\alpha f(v_j)^\beta}{\sum_{v_j \in N} \tau_{ij}^\alpha f(v_j)^\beta} & , \text{ wenn } j \text{ zulässig, mit: } f(v_j) = g(v_j) + h(v_j) \\ 0 & , \text{ sonst} \end{cases} \quad (3.10)$$

Erneut wird $\alpha = 1$ und $\beta = 5$ sowie $\rho = 0,5$ verwendet. Das Verfahren wird Stochastic Best-First-Search [48, S.4] (dt.: Stochastische Besten-Suche) bezeichnet, weil weiterhin nach Wahrscheinlichkeiten ausgewählt wird, jedoch besser bewertete Knoten bevorzugt sind. Das Verfahren wird anhand eines kleinen 10x10 Netzes getestet und kann dort den optimalen Pfad finden [48, S.11]. Der Gedanke der Zielorientierung ist für große Netze ebenfalls erstrebenswert, weil sonst die erste Iteration ein großes Problem darstellen kann. Beispielsweise, wenn in einem dreidimensionalen 1000x1000x1000-Netz mit Hindernissen der kürzeste Pfad gefunden werden soll.

Dieser Ansatz wird auch von ZHONGRUI et al. [49] mit guten Ergebnissen verwendet, wobei zwischen ZHONGRUI et al. [49] und FAISAL et al. [48] eine Unstimmigkeit besteht. Beide arbeiten mit einer Kombination aus ACO- und A*-Algorithmus, indem sie anstatt der Attraktivität η_{ij} eine Abschätzung der Entfernung zum Ziel durch die heuristische Bewertungsfunktion $f(v_j)$ integrieren. Dabei geben FAISAL et al. [48, S.4] Gleichung (3.10) an, wohingegen ZHONGRUI et al. [49, S.1] Gleichung (3.11) verwenden. Diese Gleichung kann als abgewandelter Kehrwert von $f(v_j) = g(v_j) + h(v_j)$ interpretiert werden, wobei die Attraktivität η_{ij} der Bewertungsfunktion f und die Distanz zum Zielknoten d_{jz} der Heuristik h entspricht. Einziger Unterschied ist die

Verwendung der Länge der letzten Kante d_{ij} anstatt der bisherigen Pfadkosten g .

$$\eta_{ij}(v_j) = \frac{1}{d_{ij} + d_{jz}} \quad (3.11)$$

Die Verwendung des Kehrwertes nach ZHONGRUI et al. [49] ist schlüssiger. Die Auswahlwahrscheinlichkeit steigt für steigende Werte von η_{ij} an. Verwendet man den Ansatz von FAISAL et al. [48] steigt η_{ij} direkt proportional mit der Pfadlänge, wodurch längere Pfade bevorzugt werden. Durch den Kehrwert wird die Abhängigkeit umgekehrt proportional und es ergibt sich die gewünschte Bevorzugung kurzer Kanten bzw. Pfade. Allerdings ist die Benutzung der gesamten bisherige Pfadlänge nach FAISAL et al. [48] anstatt der Länge der letzten Kante nach ZHONGRUI et al. [49] zielführender. Dadurch wird sichergestellt, dass der Algorithmus über die gesamte Laufzeit mit ähnlichen Werten für η_{ij} rechnet. Bei Verwendung nur einer Kante in Kombination mit der Luftlinienentfernung zum Ziel nimmt die Summe des Nenners bei Annäherung an den Zielknoten immer weiter ab, sodass die Werte von η_{ij} steigen. Dadurch strebt die Pfadsuche gegen Ende sehr stark dem Ziel entgegen und bereits kleine Unterschiede in der Distanz zum Ziel können enorme Schwankungen ausmachen.

In dieser Arbeit werden die vorgestellten Varianten kombiniert, um einen effizienten und robusten Algorithmus zu erhalten, der sich auch für Probleme mit einer hohen Knotenanzahl mit vertretbarer Laufzeit verwenden lässt. Um Exploration und Konvergenz definiert einstellen zu können, wird einerseits die PRPR des ACS, andererseits die Pheromonaktualisierung des AS_{rank} verwendet. Die Größe der betrachteten Netze wird durch Verwendung der Zielorientierung des AntStar beherrscht.

4 Vorgehen

In diesem Kapitel werden Methoden und Algorithmen entwickelt, um ausgehend von einem Sensorabdeckungsmodell einen relativen Pfad zwischen zwei Fahrzeugen zu generieren. Zunächst wird in Abschnitt 4.1 ein Minimalbeispiel definiert, anhand dessen der verwendete A*-Algorithmus (Abschnitt 4.2) und ACO-Algorithmus (Abschnitt 4.3) erklärt werden. In Abschnitt 4.4 wird das Vorgehen auf den Anwendungsfall angepasst, indem ausgehend von der Sensormodellierung nach MÜLLER [27] Randbedingungen für die Anwendung ausgearbeitet werden. Den Abschluss des Kapitels bildet die Vorstellung des Algorithmus zur Überführung des relativen Annäherungspfades in ein Basisszenario (Abschnitt 4.5).

Das autonome Fahrzeug mit implementiertem Sensormodell wird als Ego-Fahrzeug (Ego-Fzg) bezeichnet und sämtliche damit verbundenen Variablen werden mit *Ego* gekennzeichnet, z.B. die Geschwindigkeit des Ego-Fzg v_{Ego} . Das Fahrzeug bzw. im allgemeinen Fall das Objekt, dass durch das Ego-Fzg erkannt werden soll, ist der Challenger. Entsprechend wird *Ch* als Index verwendet, beispielsweise für die Bezeichnung v_{Ch} der Geschwindigkeit.

4.1 Definition eines Minimalbeispiels

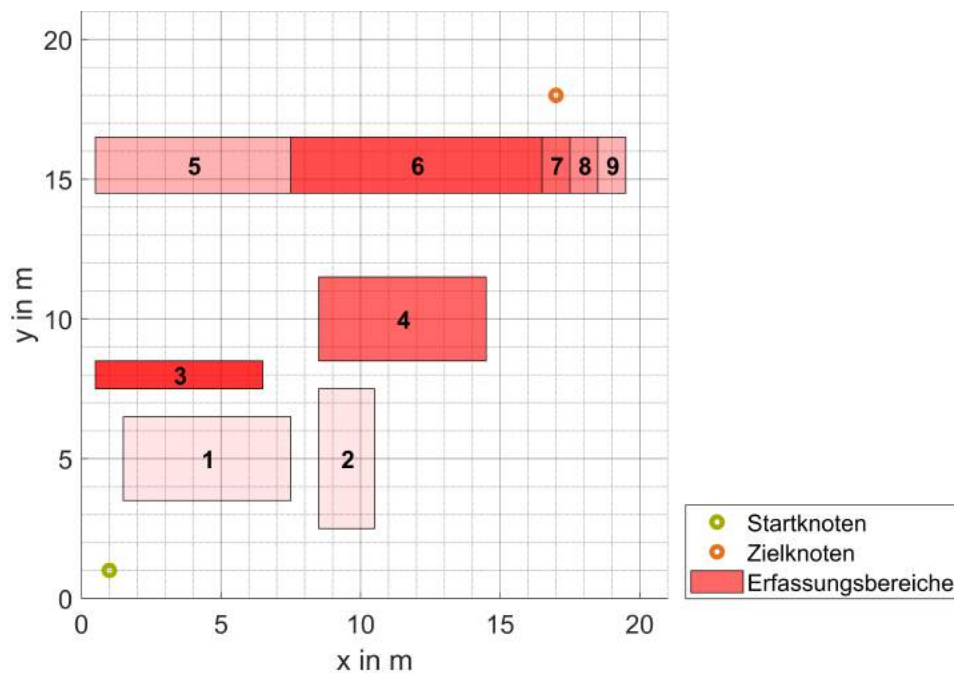
Wie der Name bereits verrät, wird ein vereinfachtes Beispiel für die Definition der Algorithmen aufgebaut. Diese finden im späteren Originalmodell einen worst-case Pfad durch das dreidimensionale Modell der Erfassungswahrscheinlichkeiten. Die Kosten und Güte des Pfades werden durch die Länge des Pfades und die Erfassungswahrscheinlichkeit beeinflusst. Das Minimalbeispiel wird zweidimensional definiert. Für die dreidimensionale Applikation müssen die Algorithmen lediglich um eine Koordinate skaliert werden.

Das Ziel der Algorithmen ist es, den kostengünstigsten relativen Pfad zwischen einem Startknoten v_s und einem Zielknoten v_z zu finden. Es wird ein Feld mit 20 m x 20 m mit Inkrement 1 m in x - und y -Koordinate vorgegeben, sodass für die Pfadsuche ein quadratisches Netz mit 20x20 Knoten entsteht. Der Startknoten v_s wird in $x_s = (1, 1)$ und der Zielknoten v_z in $x_z = (17, 18)$ definiert.

Neben der Distanz der Knoten ist eine dem Knoten v_i zugehörige Erfassungswahrscheinlichkeit $P_{D,i}$ relevant. Um diesen Parameter zu simulieren, werden neun Gebiete definiert, in denen $P_{D,i} > 0 \%$ gilt. Abb. 4.1 zeigt das Minimalbeispiel. Die Transparenz der Erfassungsbereiche nimmt mit steigender Erfassungswahrscheinlichkeit ab und für alle restlichen Punkte außerhalb der Bereiche gilt $P_D = 0 \%$. Die Gebiete sind in Tabelle 4.1 definiert.

Tabelle 4.1: Gebiete mit Erfassungswahrscheinlichkeit $P_D \neq 0$

Gebiet	Bereich in x in m	Bereich in y in m	P_D in %
1	1,5 bis 6,5	3,5 bis 6,5	10
2	4,5 bis 10,5	2,5 bis 7,5	10
3	0,5 bis 6,5	7,5 bis 8,5	80
4	8,5 bis 14,5	8,5 bis 11,5	60
5	0,5 bis 7,5	14,5 bis 16,5	30
6	7,5 bis 16,5	14,5 bis 16,5	70
7	16,5 bis 17,5	14,5 bis 16,5	60
8	17,5 bis 18,5	14,5 bis 16,5	45
9	18,5 bis 19,5	14,5 bis 16,5	30

Abbildung 4.1: Struktur des Minimalbeispiels, v_s : grün, v_z : orange, $P_D \neq 0$: rot

Die Kostenfunktion K (Gleichung (4.1)) bestimmt die Kosten eines Pfades aus dessen Knoten bzw. deren Erfassungswahrscheinlichkeiten und der euklidischen Distanz $d(v_i, v_{i-1})$ zwischen den Knoten v_i und v_{i-1} . Die Konstante k_K wird entsprechend der Parametrisierung des Algorithmus gewählt.

$$K = \sum_{n=2}^N \frac{k_K + P_{D,n}}{k_K + 1} d(v_i, v_{i-1}) \quad (4.1)$$

4.2 Entwicklung des A*-Algorithmus

Abschnitt 3.4 erklärt die Entwicklung der allgemeinen Breitensuche hin zum A*-Algorithmus. Dieser wird nun auf das vorliegende Problem appliziert. Der A*-Algorithmus wird als vollständiger und optimaler Suchalgorithmus verwendet. Dafür muss die Zulässigkeit des Algorithmus gegeben sein, wofür eine zulässige Heuristikfunktion h und eine Kostenfunktion g benötigt werden.

Die Kostenfunktion g wird in Anlehnung an Gleichung (4.1) in Abhängigkeit des Nachbarknotens v_N aus der Nachbarschaft N des Knotens v_i mit der Erfassungswahrscheinlichkeit $P_{D,N}$ von v_N definiert:

$$g(v_N) = g(v_i) + \frac{k_K + P_{D,N}}{k_K + 1} d(v_N, v_i) \quad (4.2)$$

Die Heuristikfunktion h wird gewählt als untere Grenze der Kostenfunktion, sodass das Überschätzen der Kosten ausgeschlossen und die Heuristik zulässig ist. Dafür wird die Luftlinienentfernung $d_L(v_N, v_Z)$ von v_N zu v_Z und $P_D = 0$ für jeden Knoten verwendet. Die Heuristikfunktion entspricht somit einem Anteil der Luftlinienentfernung, z.B. für $k_K = 1$ genau der Hälfte:

$$h(v_N) = \frac{k_K + \overset{0}{P_{D,N}}}{k_K + 1} d_L(v_N, v_Z) = \frac{\overset{k_K=1 \rightarrow \frac{1}{2}}{k_K}}{k_K + 1} d_L(v_N, v_Z) \quad (4.3)$$

Die heuristische Bewertungsfunktion f ergibt sich für A^* üblicherweise als

$$f(v_N) = g(v_N) + h(v_N). \quad (4.4)$$

Im vorliegenden Pfadfindungsproblem gilt es, einen Kompromiss zwischen Weglänge und Erfassungswahrscheinlichkeit zu finden. Dieser Kompromiss wird in Gleichung (4.2) bzw. Gleichung (4.1) durch den Faktor $\frac{k_K + P_{D,N}}{k_K + 1}$ widergespiegelt. Je näher an 1 der Faktor durch Wahl von k_K ist, desto weiter nähert man sich der klassischen Pfadsuche kürzester Wege an. Für den Grenzwert $k_K \rightarrow \infty$ gilt

$$\begin{aligned} \lim_{k_K \rightarrow \infty} g(v_N) &= \lim_{k_K \rightarrow \infty} \left(g(v_i) + \frac{k_K + P_{D,N}}{k_K + 1} d(v_N, v_i) \right) \\ &= g(v_i) + \lim_{k_K \rightarrow \infty} \left(\frac{1 + \frac{P_{D,N}}{k_K}}{1 + \frac{1}{k_K}} d(v_N, v_i) \right) \\ &= g(v_i) + \frac{\overset{1}{1+0}}{\underset{1+0}{1}} d(v_N, v_i) \\ &= g(v_i) + d(v_N, v_i). \end{aligned} \quad (4.5)$$

Die Kostenfunktion entspricht der Distanz zwischen den Knoten. Das Problem entspricht einer Pfadfindung, die unabhängig von Erfassungswahrscheinlichkeiten gelöst wird.

Betrachtet man den Grenzwert für $k_K \rightarrow 0$ gilt

$$\begin{aligned} \lim_{k_K \rightarrow 0} g(v_N) &= \lim_{k_K \rightarrow 0} \left(g(v_i) + \frac{k_K + P_{D,N}}{k_K + 1} d(v_N, v_i) \right) \\ &= g(v_i) + \frac{0 + P_{D,N}}{0 + 1} d(v_N, v_i) \\ &= g(v_i) + P_{D,N} d(v_N, v_i). \end{aligned} \quad (4.6)$$

Die Kostenfunktion entspricht dem Produkt aus Distanz zu v_N und $P_{D,N}$. Ein hoher Wert für k_K führt zum Ignorieren der Erfassungswahrscheinlichkeiten, wohingegen ein geringer Wert die Wegstrecke außerhalb von Gebieten mit $P_D \neq 0$ vernachlässigt. Es sollte ein $k_K \leq 1$ gewählt werden, um einen auf die Erfassungswahrscheinlichkeit ausreichend sensiblen Algorithmus zu erhalten. Wendet man nun das Vorgehen entsprechend Abschnitt 3.4.5 an, ergeben sich die in Abb. 4.2 dargestellten optimalen Pfade für $k_K \in \{1; 0,5; 0,25; 0,1\}$.

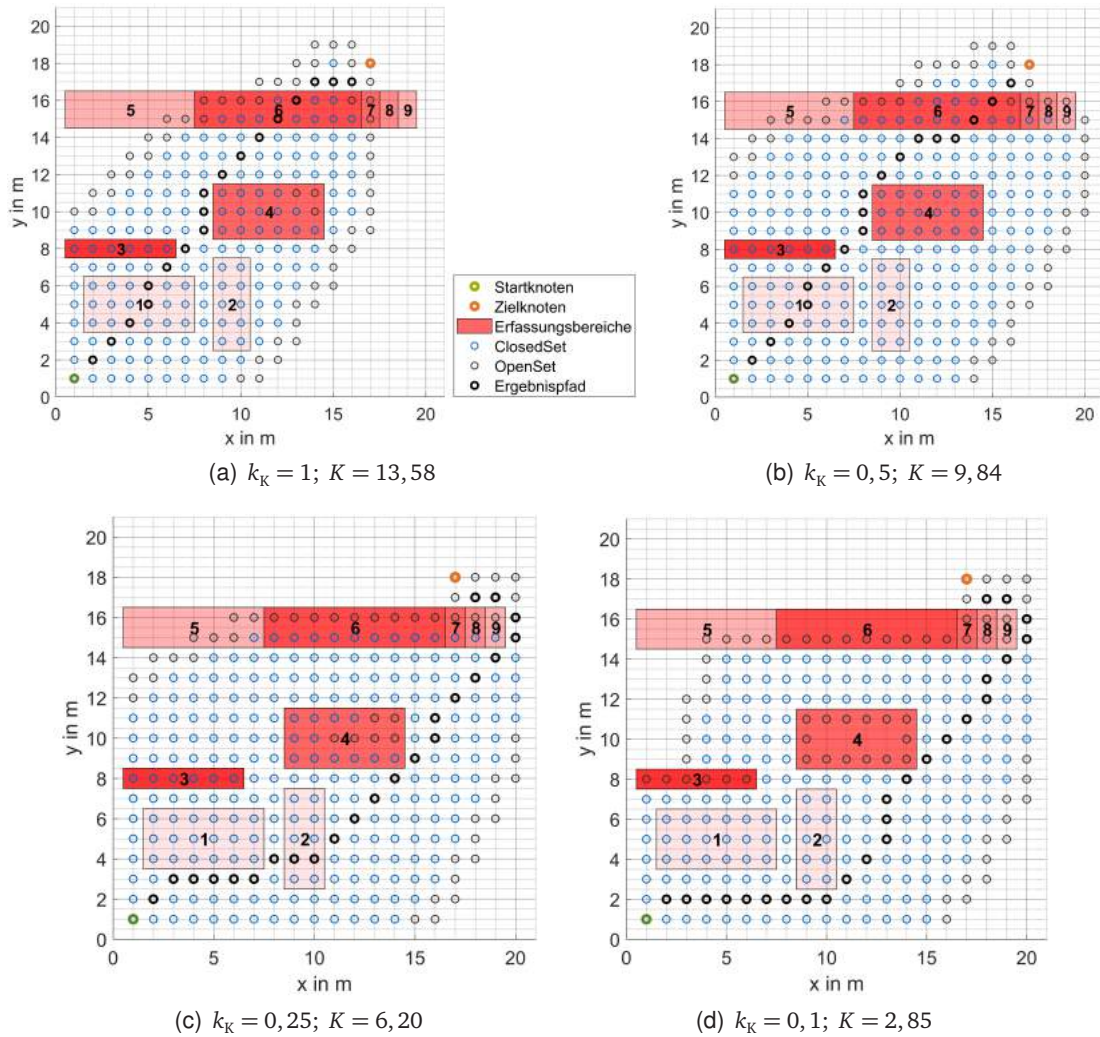


Abbildung 4.2: Ergebnisse des A*-Algorithmus angewendet auf das Minimalbeispiel, v_s : grün, v_z : orange, $P_D \neq 0$: rot, open set **O**: blau, closed set **C**: grau, Knoten auf Ergebnispfad: schwarz

Es ist wichtig zu beachten, dass jeder gefundene Pfad ein globales Optimum darstellt, jedoch nur bezüglich der Parametrierung von k_K . Die starken Differenzen der Kosten zwischen den einzelnen Varianten (Abb. 4.2(a) bis 4.2(d)) ergeben sich aus der Abhängigkeit $K = f(k_K)$.

Die Pfade verändern sich entsprechend der Grenzwertbetrachtung. $k_K = 1$ und $k_K = 0,5$ führen zu einer sehr direkten, ähnlichen Zielannäherung, wobei für $k_K = 0,5$ der erkundete Suchraum deutlich größer ist, erkennbar durch eine höhere Anzahl eingezeichneter Knoten. $k_K = 0,25$ vermeidet die Erfassungsbereiche weitestgehend und passiert nur anfänglich Gebiet 2 mit $P_D = 10\%$. Die Parametrierung mit $k_K = 0,1$ vermeidet sämtliche Erfassungsbereiche. Eine interessante Beobachtung ist die Verkleinerung des Suchraumes bei $k_K = 0,1$ im Vergleich zu $k_K = 0,25$. Dies ist möglich, weil Gebiet 3 die Ausbreitung in diese Richtung aufgrund der hohen Kosten bei Durchschreitung verhindert. Ebenfalls lässt sich in Abb. 4.2(c) erkennen, dass sich einzelne erfasste Pfade in die Gebiete 5-9 erstrecken.

Diese Untersuchung zeigt, dass die Wahl der Parameter eines Algorithmus entscheidenden Einfluss auf seine Performanz und Ergebnisse hat. In der vorliegenden Arbeit wird $k_K = 0,25$ als Ausgangswert für die dreidimensionale Anwendung verwendet, weil der Kompromiss aus Distanz und Erfassungswahrscheinlichkeit gut gelöst wird. Vor allem im dreidimensionalen Raum bei großen Netzen mit komplexen Erfassungsgebieten muss die Wahl der Kostenfunktion und deren Parameter problemabhängig getroffen und kontrolliert werden.

Infolgedessen wird eine Variation der Konstante der Kostenfunktion im Bereich $K \in \{0, 5; 0, 25; 0, 1\}$ durchgeführt (Abschnitt 5.1.2). Die Untersuchungen dieses Abschnittes haben gezeigt, dass sich innerhalb dieses Bereichs der Fokus des Algorithmus von der Pfadlänge ($k_K = 0, 5$) auf die Erfassungswahrscheinlichkeit ($k_K = 0, 1$) verschiebt.

4.3 Entwicklung des ACO-Algorithmus

Als Vergleich für den A*-Algorithmus wird ein ACO-Algorithmus entwickelt. Für die spätere Anwendung sind viele Eigenschaften der ACO-Algorithmen vorteilhaft, beispielsweise die Unabhängigkeit von funktionalen Zusammenhängen und die leichte Problemadaptierbarkeit. Als nachteilig kann sich die Rechenzeit in großen Räumen erweisen.

Aufbauend auf Abschnitt 3.5.1 werden nun die verwendeten Formeln und Funktionen definiert. Die Grundlage bildet, wie bei jeder ACO, das AS. Anschließend wird es durch einige in der Literatur und Abschnitt 3.5.2 diskutierte Varianten und Verbesserungen modifiziert. Verwendet werden dabei die Ansätze der PRPR [47, S.55f], des AntStar-Algorithmus [48, S.4] [49, S.1] und der *elitist strategy* [44, S.36] bzw. der rangbasierten Auswahl des AS_{rank} [46, S.10].

Im Unterschied zur vorgestellten Literatur wird in dieser Arbeit nicht die Pheromonmenge pro Kante verwaltet, sondern pro Knoten. Dieser Ansatz entspringt zwei Überlegungen. Einerseits werden ein Großteil der in der Literatur behandelten ACO-Algorithmen zum Lösen von TSP verwendet. In diesen ist das Ziel, attraktive Kanten zu ermitteln, um sämtliche Knoten mit minimalen Kosten zu verbinden. In dieser Arbeit sollen jedoch diejenigen Positionen (Knoten) in einem Netz ermittelt werden, an denen der Aufenthalt mit möglichst geringen Kosten verbunden ist. Der Weg zu diesen Knoten ist jedoch nicht ausschlaggebend. Der andere Gedanke ist, dass die Anzahl der Kanten n_e in einem zweidimensionalen, quadratischen Graphen der Größe $x \times x$, in dem jeder Knoten mit seinen acht Nachbarknoten verbunden ist, mit der Formel $n_e = 2(x(x-1) + (x-1)^2)$ bestimmt werden kann. Steigt nun die Anzahl der Knoten an, so wächst die Anzahl der Knoten mit x^2 , wohingegen die Anzahl der Kanten näherungsweise mit $4x^2$ steigt. Durch die Verwendung der Knoten als Zielobjekt der Pheromonbestimmung kann die Anzahl der Berechnungselemente deutlich reduziert werden. Das Vorgehen gliedert sich entsprechend Algorithmus 5 in den Pfadfindungs- und Pheromonaktualisierungsteil. Für den Aufbau der einzelnen Pfade der Ameisen wird angelehnt an die PRPR (Gleichung (3.9)) des ACS nach Gleichung (4.7) vorgegangen:

$$v_j = \begin{cases} \operatorname{argmax}_{v_N \in N} \{\tau_j^\alpha \eta_{ij}(v_j)^\beta\} & , \text{ wenn } q \leq q_0 \\ S_{\text{AntStar}} & , \text{ sonst} \end{cases} \quad (4.7)$$

Liegt eine Zufallszahl $q \in [0, 1]$ unterhalb der Auswahlgrenze $q_0 \in [0, 1]$, so wird der beste Knoten der Nachbarschaft $v_j \in N$ als nächster Knoten bestimmt. Die Qualität eines Knotens erfolgt anhand des Bewertungsproduktes $\tau_j^\alpha \eta_{ij}(v_j)^\beta$, wobei τ_j die aktuelle Pheromonmenge und $\eta_{ij}(v_j)$ die Attraktivität des Knotens v_j ist. Je höher die Pheromonmenge und je kürzer die Distanz zwischen den Knoten ist, desto größer ist der Wert des Bewertungsproduktes und der Auswahlwahrscheinlichkeit. Den Einfluss des Pheromons bzw. der Attraktivität gibt α bzw. β an.

Gilt $q > q_0$, so wird nach S_{AntStar} , der Auswahlstrategie des AntStar-Algorithmus entschieden. Diese sieht eine Rouletteradselektion vor, bei der die Übergangswahrscheinlichkeit p_{ij}^a für jede Ameise a von Knoten v_i zu Knoten v_j nach Gleichung (4.8), orientiert an Gleichung (3.10), bestimmt wird. Die Wahrscheinlichkeit wird proportional zum Bewertungsprodukt zugewiesen.

$$p_{ij}^a = \begin{cases} \frac{\tau_j^\alpha \eta_{ij}(v_j)^\beta}{\sum_{j \in v_N} \tau_j^\alpha \eta_{ij}(v_j)^\beta} & , \text{wenn } v_j \text{ zulässig} \\ 0 & , \text{sonst} \end{cases} \quad (4.8)$$

Für $\eta_{ij}(v_j)$ in Gleichung (4.9) wird die Inverse der heuristischen Bewertungsfunktion f aus Gleichung (4.4) verwendet.

$$\eta_{ij}(v_j) = \frac{1}{f_{ij}(v_j)} = \frac{1}{g_{ij} + h_{ij}(v_j)} \quad (4.9)$$

Dabei sind g_{ij} die tatsächlichen Kosten nach Gleichung (4.2) und $h_{ij}(v_j)$ die geschätzten Kosten nach der Heuristikfunktion aus Gleichung (4.3).

Nachdem für sämtliche Ameisen ein Pfad gefunden ist, wird nach dem Ansatz der rangbasierten Pheromonablage für die Pfade P_m der m_{best} besten Ameisen die Pheromonablage berechnet. Dafür wird auf sämtliche Knoten $v_j \in P_m$ die hinzukommende Pheromonmenge $\Delta\tau_j^m$ durch jede der m_{best} besten Ameisen nach Gleichung (4.10) mit Bezug auf Gleichung (3.7) berechnet. Die Pheromonmenge wird für die m beste Ameise mit dem Faktor $\sigma - m$ gewichtet, sodass bessere Ameisen einen höheren Einfluss haben, wobei $\sigma = m_{\text{best}} + 1$ gilt. Die doppelte Verstärkung der besten Ameise nach BULLNHEIMER et al. [46] wird nicht verwendet.

$$\Delta\tau_j^m = \begin{cases} (\sigma - m) \frac{Q}{L_m} & , \text{wenn } v_j \in P_m \\ 0 & , \text{sonst} \end{cases} \quad (4.10)$$

Aus der Pheromonablage $\Delta\tau_j^m$ jeder einzelnen Ameise pro Knoten kann der Pheromonzuwachs $\Delta\tau_j$ jedes einzelnen Knotens nach Gleichung (4.11) mit Bezug auf Gleichung (3.6) berechnet werden.

$$\Delta\tau_j = \sum_{m=1}^{m_{\text{best}}} \Delta\tau_j^m \quad (4.11)$$

Um die neue Pheromonmenge τ_j jedes Knotens zu bestimmen, wird das Verfahren des AS (Gleichung (3.5)) verwendet.

$$\tau_j = \rho \tau_j + \Delta\tau_j \quad (4.12)$$

Entsprechend den Empfehlungen der Literatur wird $\alpha = 1, \beta = 5, Q = 10, \rho = 0,5, m = 5, n_{\text{Gen}} = 20$ und $n_a = 20$ Ameisen verwendet. Initial wird auf jedem Knoten $\tau_0 = 10$ Pheromon platziert. Für die Wahl der Konstante werden erneut die Werte $k_K \in \{1; 0,5; 0,25; 0,1\}$ variiert. Die Ergebnisse zeigt Abb. 4.3. Jede Variante liefert sehr ähnliche Pfade bzw. Pfadkosten verglichen mit den Ergebnissen des A*-Algorithmus (Abb. 4.2): für $k_K = 1$: 13,6 - 13,58; für $k_K = 0,5$: 10,04 - 9,84; für $k_K = 0,25$: 6,82 - 6,2; für $k_K = 0,1$: 2,96 - 2,85. Das spricht für eine gute Performanz des Algorithmus und passende Wahl der Parameter. Die auftretenden Abweichungen sind prinzipbedingt. ACO und EA im Allgemeinen sind für das Auffinden möglichst guter Lösungen entwickelt, wobei diese nicht zwangsläufig bzw. sogar meistens nicht das globale Optimum sind. Dafür sind sie in der Lage, in komplexen, konventionell nicht beherrschbaren Suchräumen und Problemen, Lösungen zu finden.

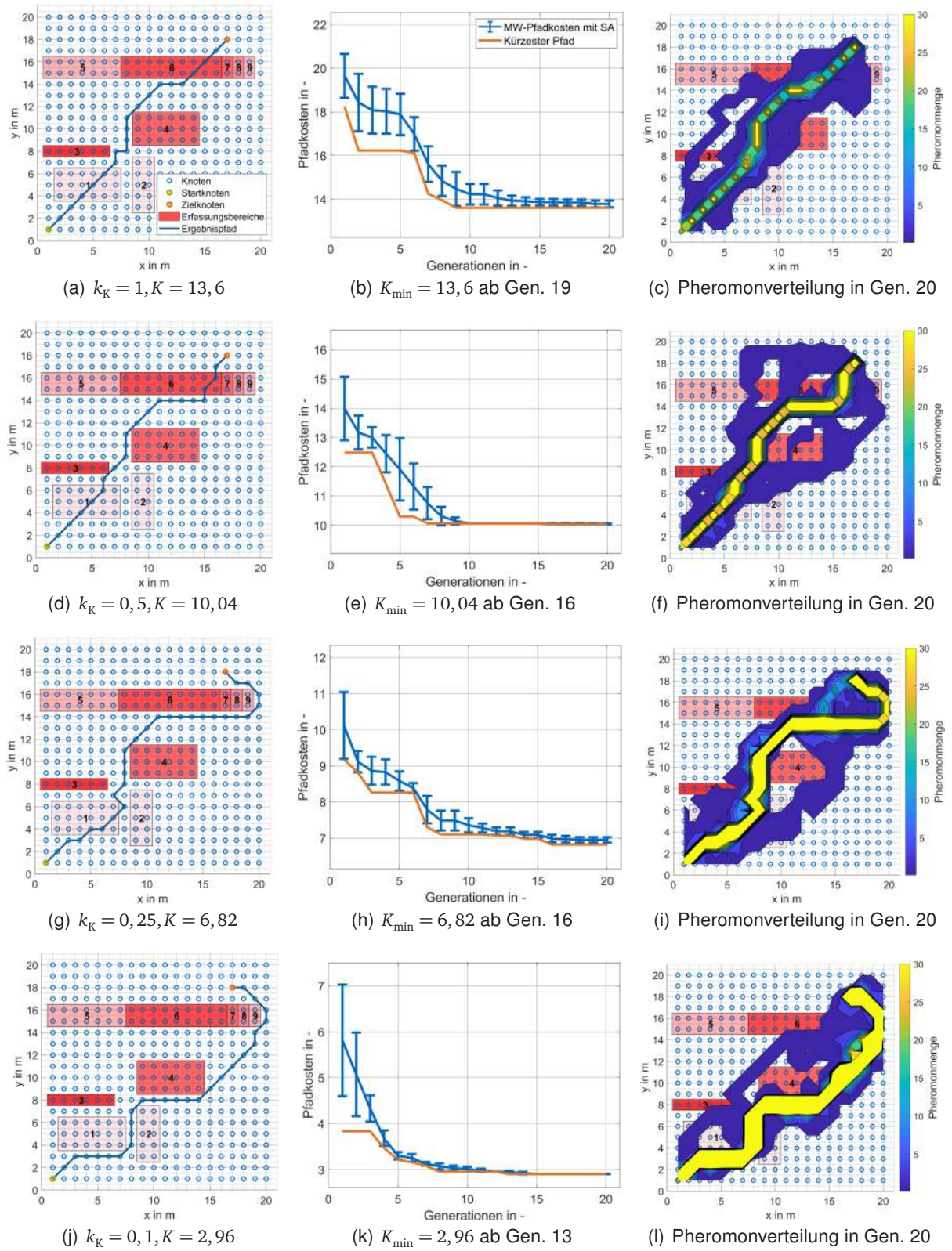


Abbildung 4.3: Ergebnispfad, Pfadkostenverlauf mit Mittelwert (MW) und Standardabweichung (SA) inklusive Angabe des ersten Auftretens des Pfades mit minimalen Kosten sowie finale Phäromonverteilung in Generation 20 des ACO-Algorithmus, angewendet auf das Minimalbeispiel, v_s : grün, v_z : orange, $P_D \neq 0$: rot, Knoten: blau, Ergebnispfad: blau

Erneut lassen sich dieselben Tendenzen, die bereits beim A*-Algorithmus analysiert wurden, erkennen. Die Wahl eines hohen $k_K \in \{1; 0,5\}$ führt zu einem sehr direkten Pfad zum Ziel, wobei die Wahl von $k_K = 0,5$ in der Analyse der Pfade eine höhere Explorationstendenz zeigt. In dem abgebildeten Verlauf kann dadurch jedoch keine Verbesserung des Pfades generiert werden. Dies ist jedoch den bereits sehr guten Pfaden ab Generation fünf geschuldet, die nur gering über dem globalen Optimum liegen. In beiden Fällen werden die Erfassungsbereiche nicht gemieden, selbst in Gebiet 6 mit $P_D = 70\%$, wobei die Strecke innerhalb der Gebiete gering gehalten wird. Wird $k_K \in \{0,25; 0,1\}$ definiert, zeigt sich eine deutliche stärkere Tendenz, die Erfassungsbereiche zu meiden. Beide Parametrierungen zeigen ein gutes Explorationsverhalten, wobei die Wahl $k_K = 0,1$ starken Fokus auf Gebiete außerhalb der Erfassungsbereiche legt. Diese Analyse basiert nicht auf Gebiet 1, durch das beide Varianten ihren besten Pfad legen, sondern auf der Analyse des Komplexes der Bereiche 5-9 (Abb. 4.3(g) und 4.3(j)). Diese werden komplett gemieden, wobei die Alternative dazu nur einer Breite von einem Knoten entspricht. Wie der optimale Pfad des A* für $k_K = 0,25$ zeigt (Abb. 4.2(c)), muss der Algorithmus trotz steigender Kosten die Möglichkeit haben, innerhalb der Erfassungsbereiche nach der optimalen Lösung zu suchen.

Die Konvergenz der Algorithmen ist eine für die Anwendung notwendige Bedingung. Eine der größten Herausforderungen der ACO-Algorithmen ist es, eine adäquate Balance aus Exploration des Suchraums und Konvergenz auf eine (sub)optimale Lösung zu erreichen. Dies kann durch die Analyse des Pfadkostenverlaufs (Abb. 4.3(b), 4.3(e), 4.3(h) und 4.3(k)) in Kombination mit der Pheromonverteilung (Abb. 4.3(c), 4.3(f), 4.3(i) und 4.3(l)) bewertet werden. Die Pheromonmenge wird erst ab einer Mindestmenge von $\tau = 0,1$ dargestellt. Liegt der Wert darunter, ist die Architektur des zu Grunde liegenden Minimalbeispiels zu erkennen.

Der Algorithmus zeigt für alle Parameter ein gutes Konvergenzverhalten auf eine Lösung im Bereich des globalen Optimums auf. Gleichzeitig können alle Parametrierungen im späteren Verlauf des Algorithmus kleine Verbesserungen erzielen. Die terminalen Pheromonverteilungen in Generation 20 für $k_K = 1$ und $k_K = 0,5$ zeigen im Vergleich zu $k_K = 0,25$ und $k_K = 0,1$ geringere Pheromonmengen entlang des Ergebnispfades auf. Gleichzeitig ist der erkundete Suchraum für $k_K = 1$ am kleinsten. Das ist die Folge initial sehr ähnlicher, aber nicht identischer Pfade, sodass grundsätzlich im selben Bereich gesucht wird, jedoch der beste Pfad erst im Verlauf gefunden werden kann. Für die anderen Parametrierungen kann generell eine hohe Explorationstendenz aller Pfade und der besten Pfade festgestellt werden.

Der Algorithmus liefert für $k_K = 0,25$ gute Ergebnisse bei anhaltender Exploration und gutem Konvergenzverhalten. Zusätzlich ermöglicht diese Auswahl, Gebiete mit hohen Erfassungswahrscheinlichkeiten zu meiden und ausreichende Sensibilität gegenüber kleineren Differenzen zu generieren.

4.4 Modellanpassungen und Randbedingungen

Mithilfe der nun definierten Algorithmen werden die relativen Pfade erstellt. Zunächst werden die Veränderungen am Modell aufgezeigt (Abschnitt 4.4.1). Anschließend wird auf die Definition von Randbedingungen für die Optimierung (Abschnitt 4.4.2) eingegangen, die durch Einschränkung des Suchraums entsprechend der Richtlinie zur Anlage von Autobahnen (RAA) [50] die Performanz der Algorithmen deutlich steigern kann. Abschließend werden nötige Anpassungen der Algorithmen für die Anwendung im dreidimensionalen Raum erläutert (Abschnitt 4.4.3).

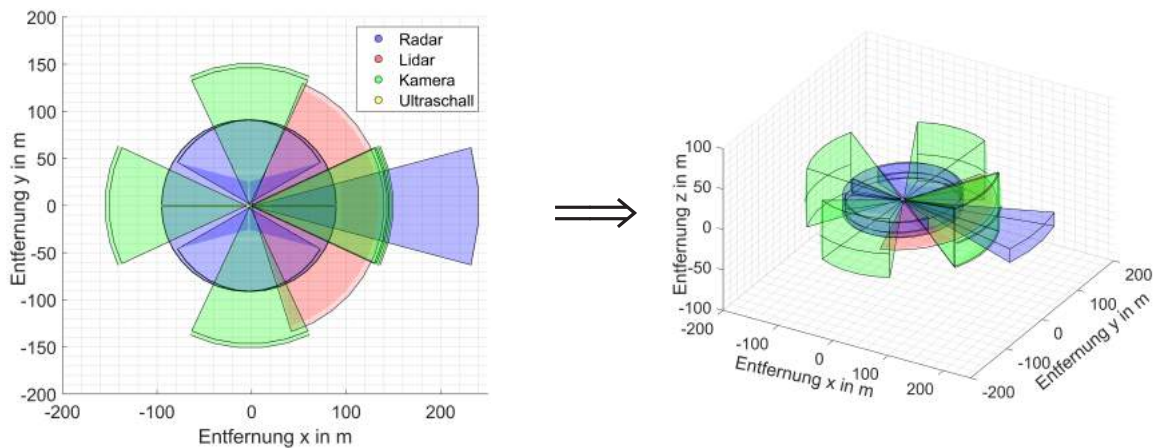


Abbildung 4.4: Erweiterung des Sensormodells um vertikale Komponente

4.4.1 Modellanpassungen

Das Modell von MÜLLER [27] (Abschnitt 2.2.3) dient dieser Arbeit als Absprungbasis zum Auffinden herausfordernder Szenarien. Bevor die Algorithmen verwendet werden, wird das Modell ausgebaut und optimiert. Wesentliche Bestandteile sind die Erweiterung des Suchraums von zwei auf drei Koordinaten und eine verbesserte Schnittstellen-Definition.

In dieser Arbeit wird das Modell zur Pfadfindung in einem dreidimensionalen Raster statt wie bisher in einer zweidimensionalen, zur x - y -Ebene parallelen Schnittebene angepasst (Abb. 4.4). Das ermöglicht die Definition von Szenarien mit Berücksichtigung vertikaler Komponenten. Für reale Verkehrsszenarien ist im Allgemeinen nicht anzunehmen, dass die Beachtung von Ebenen ausreichend ist, um sämtliche Möglichkeiten repräsentativ abzudecken, beispielsweise die Betrachtung von unterschiedlichen Objekthöhen oder Verdeckungen. Die Berechnungsmodelle werden nicht verändert, weil sie bereits im dreidimensionalen Raum definiert sind.

Bisher muss für jede Simulation bzw. Pfadberechnung zunächst die Sensormodellierung durchlaufen werden. Um Rechenzeit einzusparen, ist eine Schnittstelle definiert, die mit Hilfe der MATLAB - *load*-Funktion die Sensormodelle sowie die wichtigsten Parameter direkt in den MATLAB - *Workspace* lädt. Dies bietet außerdem den Vorteil, dass die Variation von Parametern durch den Austausch von Parameterdateien berechnungsschonend möglich ist.

4.4.2 Randbedingungen der Optimierung

Ein erheblicher Kritikpunkt des bisherigen Modells ist der Mangel an Randbedingungen, die den Suchraum auf korrekte Lösungen beschränken. Im konkreten Fall werden bisher weder reale Straßengeometrien noch Fahrphysik beachtet, sondern ein rechteckiger Suchraum von -250 m bis 350 m in Fahrzeuglängsrichtung und -200 m bis 200 m in Fahrzeugquerrichtung bei einer Schrittweite von $dx = dy = 0,5$ m verwendet [27, S.51]. Ein Ziel der Arbeit ist die Ausarbeitung geeigneter Randbedingungen. Dafür wird die relative Position der beiden Fahrzeuge während der Pfadfindung betrachtet. Die fahrphysikalischen Bedingungen werden in der Transformation des Annäherungspfades in ein Szenario berücksichtigt. In der vorliegenden Arbeit wird die Schrittweite auf 1 m angehoben, weil diese eine ausreichende Genauigkeit bei deutlich geringerer Laufzeit ermöglicht.

Tabelle 4.2: Entwurfsklassen für Autobahnen nach [50, S.17, 22f, 27, 29, 99]

Entwurfsklasse	EKA 1 A	EKA 1 B	EKA 2	EKA 3
Bezeichnung	Fernautobahn	Überregional-autobahn	Autobahnähnliche Straße	Stadtautobahn
Zul. Höchstgeschwindigkeit	keine	keine	keine	$\leq 100 \frac{\text{km}}{\text{h}}$
Zul. Höchstgeschwindigkeit bei Nässe in $\frac{\text{km}}{\text{h}}$	130	120	100	80
Fahrstreifenbreite w in m	3,5 bis 3,75		3,5	3,25 bis 3,5
Kurvenmindestradius $R_{\min}$ in m	900	720	470	280
Höchstlängsneigung q in %	4,0	4,5	4,5	6,0
Kuppenmindesthalbmesser H_K in m	13 000	10 000	5000	3000
Wannenmindesthalbmesser H_W in m	8800	5700	4000	2600
Maximaler Versatz $\Delta y_{\max}$ in m	10,875 (RQ 43,5) 7,125 (RQ 36) 3,75 (RQ 31)		3,5 (RQ 28)	10,125 (RQ 38,5) 6,625 (RQ 31,5) 3,375 (RQ 31)

Für die Betrachtung der Hochachse muss ebenfalls eine Reichweite mit Inkrement dz definiert werden, beispielsweise ± 10 m. Im Vorgriff auf die spätere Ableitung eines Szenarios aus dem generierten Pfad (Abschnitt 4.5) wird angemerkt, dass die Inkremente der Hochachse geringer gewählt werden als die der Längs- und Querachse. Ansonsten ergibt sich durch die Veränderung um nur ein Inkrement eine scheinbar hohe Steigung. Vorgeschlagen wird ein Verhältnis von 1 : 5, woraus eine Schrittweite von $dz = 0,2$ m entsteht. Die benötigte Knotenzahl beträgt damit:

- für die x-Koordinate: $\frac{350 \text{ m} - (-250 \text{ m})}{1 \text{ m}} = 600$
- für die y-Koordinate: $\frac{200 \text{ m} - (-200 \text{ m})}{1 \text{ m}} = 400$
- für die z-Koordinate: $\frac{10 \text{ m} - (-10 \text{ m})}{0,2 \text{ m}} = 100$
- und somit für eine resultierende Netzgröße von $600 \times 400 \times 100 = 24\,000\,000$ Knoten.

Neben der physikalischen Sinnhaftigkeit, an die Straßengeometrie adaptierte Randbedingungen einzuführen, wird eine Beschränkung des Suchraumes benötigt, weil 24 Millionen Knoten keine effiziente Berechnung ermöglichen. Ein großer Teil der Punkte liegt zudem in Bereichen, die für reale Verkehrsszenarien nicht relevant sind, z.B. $x < 0$ m oder $|y| > 100$ m. In Deutschland ist die Geometrie von Straßen normiert, Autobahnen durch die RAA [50]. Mit dem Begriff Autobahn bezieht sich die Richtlinie auf „alle anbaufreien, zweibahnig mehrstreifigen und durchgehend planfrei geführten Straßen, die nur für den schnellen Kraftfahrzeugverkehr bestimmt sind. Dazu gehören auch autobahnähnliche Straßen und Stadtautobahnen“ [50, S.7]. Autobahnen werden in der Planung in die Entwurfsklassen für Autobahnen (EKA) entsprechend Tabelle 4.2 unter Berücksichtigung der Verbindungsstrecken und Bebauung eingeteilt [50, S.16]. Die EKA hat Einfluss auf die Charakteristika der Autobahn, z.B. auf Regelquerschnitte (RQ), Entwurfselemente sowie Grundformen und Abstände von Knotenpunkten [50, S.17]. Aus den Begrenzungen der Fahrbahngeometrien und -führung werden anschließend die Randbedingungen der Optimierung bestimmt. In Abb. 4.5 sind die drei RQ der EKA1 beispielhaft dargestellt. Weitere Parameter und Begrenzungen, die für die Berechnung der Randbedingungen von Interesse sind, können Tabelle 4.2 entnommen werden.

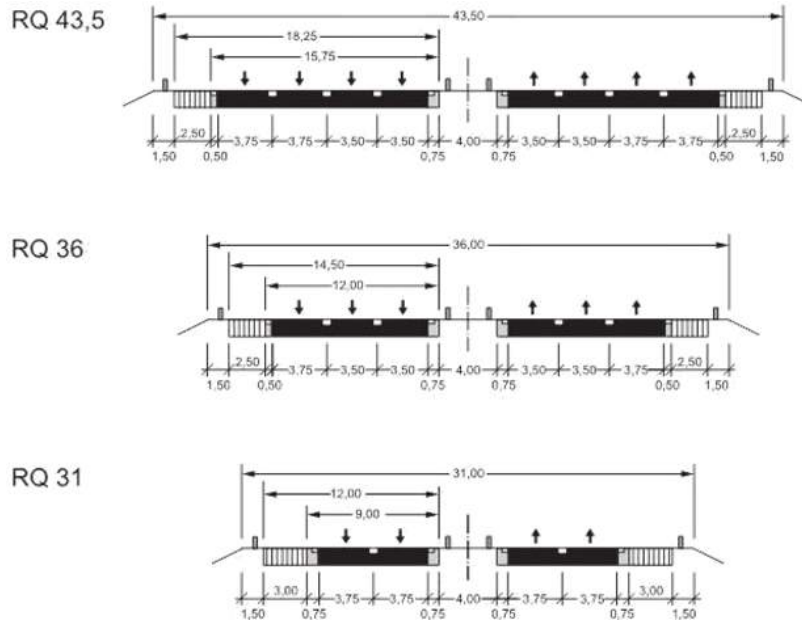


Abbildung 4.5: RQ der EKA1 nach RAA [50, S.22]

Die erste Randbedingung wird in Fahrzeuglängsrichtung definiert. Das berücksichtigte System mit der höchsten Reichweite in der Sensormodellierung nach MÜLLER [27, S.37] ist der Radarsensor mit 250 m. Zur Berücksichtigung von Startpunkten außerhalb der Erfassungsbereiche wird die longitudinale obere Begrenzung auf $x_{\max} = 300$ m gesetzt. In Anlehnung an SHALEV-SHVARTZ et al. [18] wird die untere Begrenzung auf $x_{\min} = 0$ m festgelegt, weil die Sicherheitskontrolle und Verantwortung beim auffahrenden Fahrzeug liegt. Manöver mit rückwärtig annäherndem Fahrzeug, bei denen das Ego-Fzg die Verantwortung trägt, wie beispielsweise ein linker Fahrstreifenwechsel, werden damit ausgeschlossen. Diese Klasse an Szenarien kann bei inverser Beschränkung des Suchraumes adressiert werden. Die gleichzeitige Betrachtung der beiden Klassen ist aufgrund der unterschiedlichen Voraussetzungen nicht sinnvoll kombinierbar.

In Fahrzeugquerrichtung werden die Randbedingungen durch die Fahrstreifenbreite w , bzw. Fahrbahnbreite w_{St} und Kurvenmindestradius $R_{\min}$ vorgegeben. Als worst-case Annahme wird betrachtet, dass sich das Ego-Fzg auf einer vierstreifigen Fahrbahn auf dem ganz linken (rechten) Fahrstreifen und der Challenger auf dem ganz rechten (linken) Fahrstreifen befindet und die Kurve maximal gekrümmt ist, also Kurvenradius $R = R_{\min}$ gilt. Daraus ergibt sich für die laterale Begrenzung eine Kreisbahn mit Mittelpunkt $x_{M,y}$ in $(0, \Delta y_{\max} + R_{\min})$ bzw. $(0, -(\Delta y_{\max} + R_{\min}))$, wobei $\Delta y_{\max}$ der Versatz zwischen den beiden Fahrzeugen ist (Gleichung (4.13)). Die Koordinaten x, y, z geben den Abstand der jeweiligen Koordinatenrichtung zwischen Ego-Fzg und Challenger an.

$$y_{\max} = -y_{\min} = \sqrt{R_{\min}^2 - x^2} - \Delta y_{\max} - R_{\min} \quad (4.13)$$

$\Delta y_{\max}$ beträgt für eine vierstreifige Autobahn der EKA 1 A maximal $\frac{3,75 \text{ m}}{2} + 3,75 \text{ m} + 3,5 \text{ m} + \frac{3,5 \text{ m}}{2} = 10,875 \text{ m}$, bezogen auf die Mitten der beiden Fahrzeuge. Die Werte der restlichen EKA in Tabelle 4.2 ergeben sich analog.

Für die Hochachse wird analog zur Querachse vorgegangen. Betrachtet werden die Mindesthalbmesser für Kuppen H_K und Wannen H_W für Kugeln mit dem Mittelpunkt $x_{M,z}$ in $(0,0,H_K + dz)$ bzw. $(0,0,-H_W - dz)$ mit der Schrittweite dz . Zusätzlich wird die Begrenzung an die Wahl der z -Koordinate z_Z im Zielknoten v_Z angepasst, indem das Maximum bzw. Minimum aus z_Z und der Begrenzung verwendet wird. Man beachte, dass in positiver und negativer z -Koordinate je ein Inkrement als zusätzliche Toleranz gegeben wird, weil andernfalls aufgrund der Interpolation der Koordinaten auf Knotenpunkte die vorgegebenen Punkte außerhalb des Netzes liegen können. Die Begrenzung ist eine Funktion von x und y bzw. der Distanz d zwischen den beiden Fahrzeugen (Gleichungen (4.14) und (4.15)).

$$z_{min} = \operatorname{argmin} \left(z_Z; \sqrt{H_K^2 - \underbrace{(x^2 + y^2)}_{d^2}} - H_K \right) - dz \quad (4.14)$$

$$z_{max} = \operatorname{argmax} \left(z_Z; H_W - \sqrt{H_W^2 - \underbrace{(x^2 + y^2)}_{d^2}} \right) + dz \quad (4.15)$$

Sämtliche Formeln und Begrenzungen werden im Koordinatensystem des Ego-Fahrzeugs berechnet. Dadurch ergibt sich stets eine Orientierung des Fahrzeuges exakt in positive x -Richtung und eine Position in $(0,0,0)$. Die Kreise liegen somit immer tangential zu einer Parallelen der x -Achse und folglich ergeben sich die Kreismittelpunkte $x_{M,y}$ immer in $x = 0$ und die Kugelmittelpunkte $x_{M,z}$ immer in $x = y = 0$.

Abb. 4.6 zeigt die möglichen Positionen des Challengers relativ zum Ego-Fzg für die EKA 1 A. Im Vergleich zum betrachteten Suchraum der Vorgängerarbeit $-250 \text{ m} \leq x \leq 350 \text{ m}$, $200 \text{ m} \leq y \leq -200 \text{ m}$ und $-10 \text{ m} \leq z \leq 10 \text{ m}$ mit $dx = dy = 1 \text{ m}$, $dz = 0,2 \text{ m}$ kann die Anzahl der Knoten für die EKA 1A der RAA von 24 Millionen um 97,2 % auf 676 237 gesenkt werden.

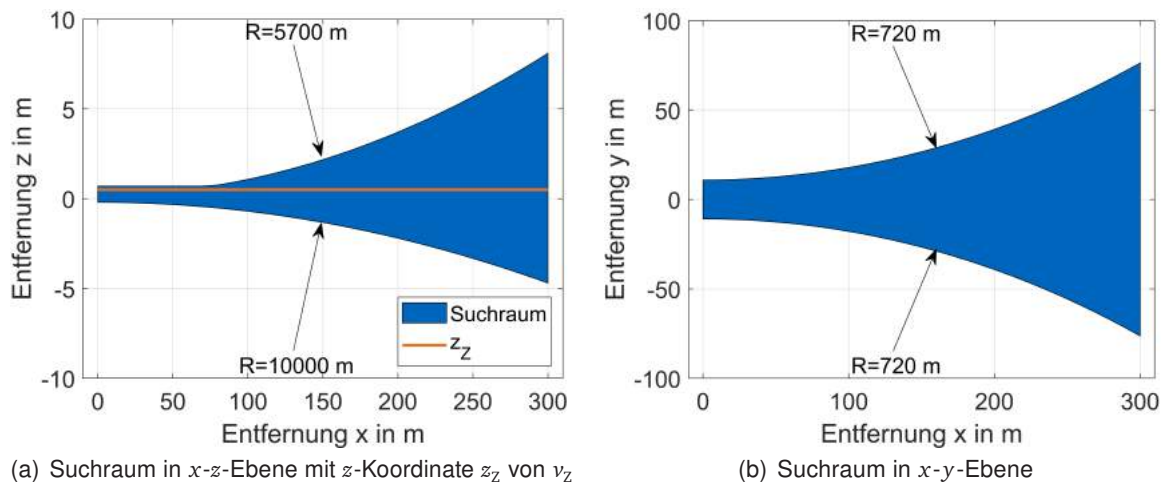


Abbildung 4.6: Zulässiger Bereich der relativen Position des Challengers

4.4.3 Anpassungen der Algorithmen

Die Skalierung der Algorithmen auf drei Koordinatenachsen benötigt Anpassungen im Vergleich zur Definition der Algorithmen im Kontext des Minimalbeispiels (Abschnitt 4.1). Die Generierung eines Straßenverlaufs erfordert eine strengere Einschränkung der Veränderung der z -Koordinate. Kann diese völlig frei gewählt werden, kommt es zu Abschnitten, in denen die Algorithmen ausschließlich vertikal benachbarte Knoten auswählen und somit der Abstandsvektor zwischen Ego-Fzg und Challenger nur in dieser Koordinate verändert wird. Ein Straßenverlauf, der diesen Pfad erfüllen soll, kann nur durch eine aufwändige Berechnung der Relativgeschwindigkeit erstellt werden. Diese wird in dieser Arbeit konstant gesetzt, weil hier die grundsätzliche Methodik ausgearbeitet wird. Eine rein vertikale Progression der Algorithmen wird blockiert.

Die Pfadlängen bzw. -kosten in der Anwendung des ACO sind im Vergleich zu den Pfadlängen des Minimalbeispiels deutlich höher. Infolgedessen ist es nötig, den Pheromonfaktor Q zu erhöhen, weil sonst der Term $(\sigma - m) \frac{Q}{L_m}$ aus Gleichung (4.10) sehr geringe Werte annimmt und der Algorithmus nur sehr langsam konvergiert. Es wird $Q = 300$ verwendet, um eine ausreichende Konvergenz zu gewährleisten.

Aufgrund der deutlich höheren Knotenanzahl muss die Anzahl der Ameisen und Generationen erhöht werden. Es werden $n_a = 200$ Ameisen für $n_{\text{Gen}} = 50$ Generationen verwendet, wobei die besten $m_{\text{best}} = 40$ Ameisen für die Pheromonaktualisierung betrachtet werden.

4.5 Algorithmus zur Szenarioüberführung

Die beiden vorherigen Abschnitte 4.2 und 4.3 stellen zwei Varianten vor, um einen möglichst anspruchsvollen Verlauf der relativen Position des Challengers bezüglich des Ego-Fzg zu erhalten. Im nächsten Abschnitt wird ein Algorithmus entwickelt, der diesen Verlauf in eine realistische Straßengeometrie übersetzt. Gleichzeitig wird die globale Position der Fahrzeuge ermittelt und somit deren Trajektorien festgelegt. Aus diesen Trajektorien resultiert der tatsächliche relative Pfad zwischen den beiden Fahrzeugen, der möglichst geringe Abweichungen zum Eingabe-Pfad aufweisen soll. Für dieses Vorgehen werden in Abschnitt 4.5.1 Randbedingungen und Annahmen definiert, die ein robustes Übersetzen ermöglichen. Anschließend wird in Abschnitt 4.5.2 der Algorithmus zur Szenarioüberführung entwickelt.

4.5.1 Randbedingungen und Annahmen

Die Erstellung der Straßengeometrie aus dem relativen Pfad benötigt erneut Randbedingungen und Annahmen. Grundsätzlich wird das Ego-Fzg als Orientierung verwendet und die Werte des Challengers daran angepasst, sodass in Summe die Abweichung zum vorgegebenen Pfad möglichst gering ist. Die Trajektorie des Ego-Fzg wird zur Erstellung des Straßenverlaufs verwendet, wobei die Trajektorie des Challengers als Orientierung und Vorgabe für das Ego-Fzg dient. Zunächst wird der generierte Pfad geglättet und interpoliert. Durch die Glättung wird zwar eine Abweichung vom Vorgabepfad in das System eingebracht, diese kann jedoch eingestellt werden. Die diskrete Veränderung der Position führt zu diskontinuierlichen Verläufen und kann in Bereichen größerer lateraler und vertikaler Wertänderungen zu Dynamikanforderungen außerhalb des zulässigen Bereichs führen. Dadurch kommt es ebenfalls zu einem Fehler in der Szenarioableitung, der jedoch im Gegensatz zur Glättung nicht einstellbar bzw. vorhersehbar ist. Über den gesamten Pfad ergibt sich durch die Verwendung der Glättung ein geringerer Fehler. Die Glättung des Pfades wird mit der MATLAB-Funktion *smooth* unter Verwendung der Option *lowess* über 20 Knoten erstellt (Abb. 4.9).

Die Interpolation erfolgt anschließend, wobei der Abstand zwischen zwei Knoten an der Veränderung der relativen Position entsprechend der Relativgeschwindigkeit angepasst wird. Dadurch liegt für den nachfolgenden Algorithmus eine feinere Auflösung der Referenzknoten vor und die Performanz wird verbessert. Weitere Annahmen sind:

- Geschwindigkeit des Ego-Fahrzeugs $v_{\text{Ego}} = 130 \frac{\text{km}}{\text{h}}$
- Geschwindigkeit des Challengers $v_{\text{Ch}} = 80 \frac{\text{km}}{\text{h}}$

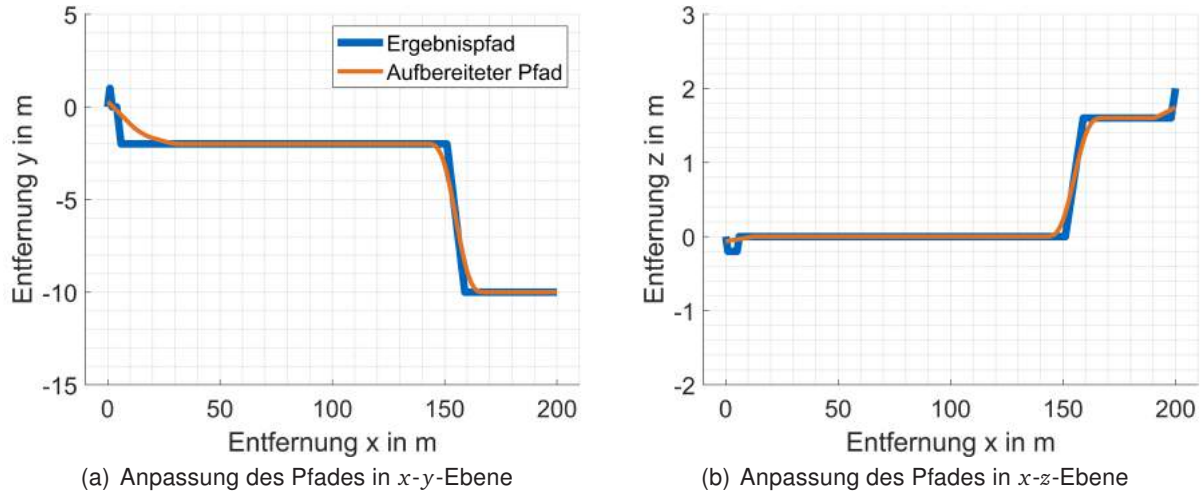


Abbildung 4.7: Glättung des relativen Annäherungspfades

Die Geschwindigkeiten sind autobahntypisch gewählt mit einer ausreichend hohen Relativgeschwindigkeit $\Delta v = 50 \frac{\text{km}}{\text{h}}$. Die Gierrate berechnet sich nach HARRER und PFEFFER [51, S.23] aus folgender Gleichung (4.16):

$$\dot{\psi} = \sqrt{\frac{a_y}{R}} = \frac{a_y}{v} \quad (4.16)$$

Verwendet werden die Querbewegung a_y , der Kurvenradius R und die Geschwindigkeit v . In der Gleichung wird der Zusammenhang $a_y = \frac{v^2}{R}$ angewendet. Durch erneutes Einsetzen dieses Verhältnisses kann Gleichung (4.16) in eine Berechnung überführt werden, die auf den Grenzwerten und bereits getroffenen Annahmen bezüglich des Straßenverlaufs und der Geschwindigkeit der Fahrzeuge beruht. Man erhält für RQ43,5 der RAA [50] nach Gleichung (4.17):

$$\dot{\psi}_{\max,R} = \frac{a_y}{v} = \frac{v_{\text{Ego}}}{R} = \frac{130 \frac{\text{km}}{\text{h}}}{720 \text{ m}} = 0,03 \frac{1}{\text{s}} \quad (4.17)$$

Dieser Wert gilt für konstante Kurvenfahrten. Höhere Gierraten treten üblicherweise beim Fahrsteifenwechsel auf. FENG et al. [52] führen eine Fahrstudie durch, die für Autobahnszenarien maximale Querbewegungen von $a_{y,\max} < 6 \frac{\text{m}}{\text{s}^2}$ bei schnellen Fahrsteifenwechseln liefert. HOFFMEISTER und BECKE [53] stellen in einem Realfahrzeug-Test, der auf die Erprobung des fahrdynamischen Grenzbereichs abzielt, Querbewegungen von bis zu $a_{y,\max} \leq 8 \frac{\text{m}}{\text{s}^2}$ bei einer Geschwindigkeit von $76 \frac{\text{km}}{\text{h}}$ bis $90 \frac{\text{km}}{\text{h}}$ fest. SCHIMMELPFENNIG und NACKENHORST [54] ermitteln für Messungen in Kurvenfahrten maximale Beschleunigungen von $a_{y,\max} \leq 5 \frac{\text{m}}{\text{s}^2}$ und in Fahrsteifenwechseln, die die Probanden „scharf“ durchfahren sollen, Werte $\leq 2,5 \frac{\text{m}}{\text{s}^2}$ [54]. In beiden Untersuchungen sinkt die Querbewegung für Geschwindigkeiten über $100 \frac{\text{km}}{\text{h}}$ unter $2 \frac{\text{m}}{\text{s}^2}$ [54]. SUN und WANG [55] definieren $4 \frac{\text{m}}{\text{s}^2}$ als Grenzwert für sicheres Fahren während eines Fahrsteifenwechsels auf der Autobahn [55, S.951].

Für Fahrsteifenwechsel auf der Autobahn existieren große Unterschiede zwischen den ermittelten Werten der Studien. Ein genereller Konsens herrscht für normale Fahrsteifenwechsel mit einer Querbewegung von maximal $3 \frac{\text{m}}{\text{s}^2}$ bis $4 \frac{\text{m}}{\text{s}^2}$. Die angegebenen Werte der maximalen gemessenen Querbewegung in Grenzmanövern streuen deutlich stärker mit Werten bis zu $8 \frac{\text{m}}{\text{s}^2}$. Als konservative Schätzung mit Sicherheit ist die Annahme eines Grenzwertes von $8 \frac{\text{m}}{\text{s}^2}$ demnach zulässig.

Entsprechend Gleichung (4.18) von SUN und WANG [55, S.954] hat ein charakteristischer Fahrstreifenwechsel auf einer Autobahn mit Fahrstreifenbreite $w = 3,5$ m bei $8 \frac{\text{m}}{\text{s}^2}$ eine Dauer von $\Delta t = 1,6$ s.

$$a_{y,\max} = \frac{10 \sqrt{3} w}{3 \Delta t^2} \rightarrow \Delta t = \sqrt{\frac{10 \sqrt{3} w}{3 a_{y,\max}}} = 1,59 \text{ s} \quad (4.18)$$

Der gewählte Grenzwert der Querschleunigung ergibt unter Verwendung von Gleichung (4.16) eine maximale Gierrate von

$$\dot{\Psi}_{\max} = \frac{a_{y,\max}}{v_{\text{Ego}}} = 0,22 \frac{1}{\text{s}}. \quad (4.19)$$

HARRER und PFEFFER [51, S.140] ermitteln Wertebereiche optimalen Lenkverhaltens für Limousinen bzw. Sportwagen von $0,25 \frac{^\circ}{\text{s}}$ bis $0,28 \frac{^\circ}{\text{s}}$ bzw. $0,30 \frac{^\circ}{\text{s}}$ bis $0,32 \frac{^\circ}{\text{s}}$ ($0,0044 \frac{1}{\text{s}}$ bis $0,0049 \frac{1}{\text{s}}$ bzw. $0,0052 \frac{1}{\text{s}}$ bis $0,0056 \frac{1}{\text{s}}$). Die Grenzwerte aus Gleichungen (4.17) und (4.19) befinden sich weit über diesen Werten und grenzen damit den Suchraum nicht zu stark ein. Eine Berücksichtigung spezifischer Werte der Gierrate kann bei der Generierung des endgültigen Szenarios getätigt werden. Für die Nickrate $\dot{\theta}$ wird ebenfalls ein Maximalwert von $\dot{\theta}_{\max} = 0,22 \frac{1}{\text{s}}$ angenommen und ein maximaler Nickwinkel $\theta_{\max} = q_{\max} = 6 \% = 0,06$ entsprechend der maximalen Straßensteigung $q_{\max}$ des verwendeten RQ (hier: RQ43,5) nach RAA [50].

Um die Stabilität des Algorithmus zu steigern, wird außerdem der Gierwinkel des Challengers $\Psi_{\text{Ch}} \leq \Psi_{\max, \text{Ch}} = \Psi_{\text{Ego}} + \Psi_{\max}$ auf eine maximale Differenz $\Psi_{\max}$ zum Gierwinkel des Ego-Fzg Ψ_{Ego} begrenzt.

4.5.2 Algorithmus zur Überführung in ein Basisszenario

In dieser Arbeit wird eine möglichst optimale Verteilung der Änderung der relativen Position auf die beiden beteiligten Fahrzeuge angestrebt. Dafür wird immer zuerst die Bewegung des Ego-Fzg prädiziert und die Bewegung des Challengers daran angepasst. Dieses Vorgehen eignet sich, weil ab dem Zeitpunkt, an dem das Ego-Fzg die bereits vom Challenger befahrene Strecke passiert, eine genaue Vorgabe des zukünftigen globalen Pfades erfolgen kann. Bis dieser Punkt erreicht ist, wird die Vorgabe anhand einer Interpolation zwischen dem Pfad des Ego-Fzg und des Challengers erzeugt.

Dafür werden zwei Koordinatensysteme verwendet, das globale Koordinatensystem mit Koordinaten $\mathbf{x}_{\text{glob}}$ und das inhärente Koordinatensystem des Ego-Fzg, auch als relatives Koordinatensystem bezeichnet, mit Koordinaten $\mathbf{x}_{\text{rel}}$. Das globale Koordinatensystem enthält die tatsächlichen Positionen der Fahrzeuge $\mathbf{x}_{\text{Ego, glob}}$ bzw. $\mathbf{x}_{\text{Ch, glob}}$ und das relative die Abstände des Challengers zum Ego-Fzg unter Berücksichtigung von dessen Orientierung (Abb. 4.8(a)). Die relative Position des Ego-Fzg $\mathbf{x}_{\text{Ego, rel}}$ ist definitionsgemäß stets im Ursprung und die relative Position des Challengers $\mathbf{x}_{\text{Ch, rel}}$ ergibt sich aus der Differenz der globalen Positionen unter Berücksichtigung der Orientierung

$$\mathbf{x}_{\text{Ch, rel}} = \mathbf{x}_{\text{Ch, glob}} - \mathbf{x}_{\text{Ego, glob}}. \quad (4.20)$$

Das Ziel des Algorithmus ist die Bestimmung der Positionsänderungen $\Delta \mathbf{x}_{\text{Ego}}$ bzw. $\Delta \mathbf{x}_{\text{Ch}}$, sodass $\mathbf{x}_{\text{Ch, rel}}$ möglichst exakt dem relativen Pfad entspricht. Die Änderungen $\Delta \mathbf{x}_{\text{Ego}}$ und $\Delta \mathbf{x}_{\text{Ch}}$ sind für beide Koordinatensysteme identisch. Für den Challenger wird zusätzliche eine Positionsänderung durch die Positions- und Orientierungsänderung des Ego-Fzg berücksichtigt. Diese ist nicht in der Variable $\Delta \mathbf{x}_{\text{Ch}}$ enthalten.

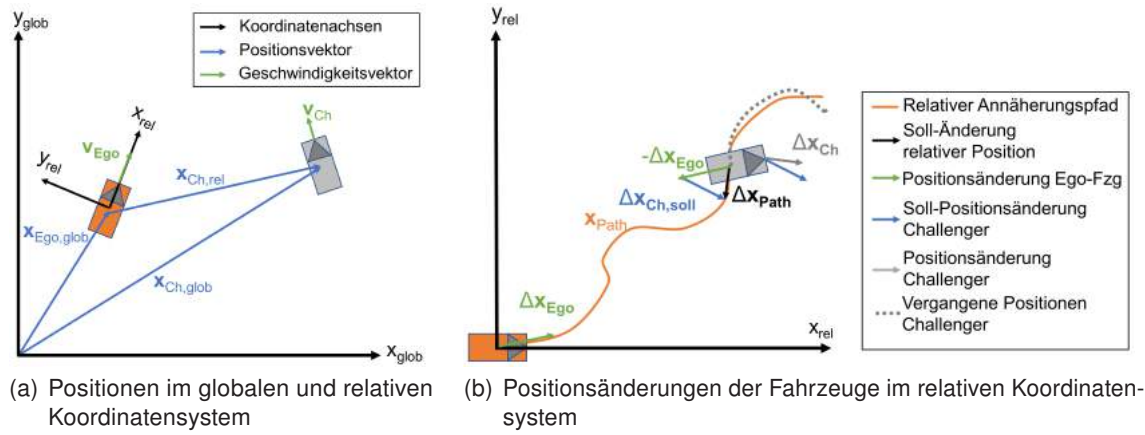


Abbildung 4.8: Positionen der Fahrzeuge in den Koordinatensystemen und Herleitung der Positionsänderung, orange: Ego-Fzg, grau: Challenger

Initialisierung

Zu Beginn befindet sich das Ego-Fzg im definierten Zielknoten v_Z und der Challenger im definierten Startknoten v_S , sodass sich der initiale Abstand des relativen Pfades ergibt. Die Orientierung der Fahrzeuge liegt in Richtung der x -Achse des globalen Koordinatensystems. Die anschließenden Erklärungen zur Positionsänderung der Fahrzeuge sind in Abb. 4.8(b) veranschaulicht.

Für die Initialisierung des Algorithmus wird angenommen, dass sich das Ego-Fahrzeug nur in x -Richtung bewegt, ohne seine Orientierung zu ändern. In Abb. 4.8(b) wird der allgemeine Fall gezeigt. Entsprechend bestimmt sich direkt die Änderung des ersten Zeitschritts des Ego-Fzg

$$\Delta \mathbf{x}_{Ego} = (v_{Ego} dt, 0, 0)^T. \quad (4.21)$$

Aus dem nächsten Referenzknoten $\mathbf{x}_{Ch,ref}$ des Challengers und der aktuellen relativen Position $\mathbf{x}_{Ch,rel}$ kann unter Berücksichtigung der Relativgeschwindigkeit Δv und der Zeitschrittweite dt die Soll-Änderung der relativen Positionen $\Delta \mathbf{x}_{Path}$ bestimmt werden (Gleichung (4.22)).

$$\Delta \mathbf{x}_{Path} = \frac{\mathbf{x}_{Ch,ref} - \mathbf{x}_{Ch,rel}}{\|\mathbf{x}_{Ch,ref} - \mathbf{x}_{Ch,rel}\|} \Delta v dt \quad (4.22)$$

Diese Änderung soll durch den Challenger erreicht werden und wird für die Berechnung der vorgegebenen Position des Challengers verwendet, wobei $\Delta \mathbf{x}_{Ego}$ entsprechend der Vektorsubtraktion auszugleichen ist:

$$\Delta \mathbf{x}_{Ch,Soll} = \Delta \mathbf{x}_{Path} - \Delta \mathbf{x}_{Ego}. \quad (4.23)$$

Anschließend werden die resultierenden Winkeländerungen $\Delta \Psi_{Ch}$ und $\Delta \theta_{Ch}$ bzw. die Winkel Ψ_{Ch} und θ_{Ch} kontrolliert und bei Überschreiten der Grenzwerte $\dot{\Psi}_{max}$ und $\dot{\theta}_{max}$ bzw. $\Psi_{max,Ch}$ und θ_{max} auf diese begrenzt. Aus den nun zulässigen Winkeländerungen lässt sich die tatsächliche, für das globale und relative Koordinatensystem identische Positionsänderung $\Delta \mathbf{x}_{Ch}$ bestimmen.

$$\Delta \mathbf{x}_{Ch} = v_{Ch} dt \begin{pmatrix} \cos \theta_{Ch} \cos \Psi_{Ch} \\ \cos \theta_{Ch} \sin \Psi_{Ch} \\ \sin \theta_{Ch} \end{pmatrix} \quad (4.24)$$

Die neuen globalen Positionen der Fahrzeuge ergeben sich trivial aus der Summe der Positionen mit der Positionsänderung. Für die relative Position des Challengers muss sowohl Δx_{Ch} als auch Δx_{Ego} berücksichtigt werden.

$$\mathbf{x}_{\text{Ch,rel},i+1} = \mathbf{x}_{\text{Ch,rel},i} + \Delta \mathbf{x}_{\text{Ch},i} + \Delta \mathbf{x}_{\text{Ego},i} \quad (4.25)$$

Auf Basis der beiden Änderungen Δx_{Ch} und Δx_{Ego} ist es möglich, einen zunächst ausreichend genauen Pfad zwischen der Startposition des Ego-Fzg und des Challengers zu erzeugen, anhand dessen die Richtungsvorgabe des Challengers erfolgen kann. Aus der aktuellen Positionsänderung Δx_{Ch} werden sieben weitere Knoten mit gleicher Änderung in identischer Richtung bzw. für Δx_{Ego} in entgegengesetzter Richtung bestimmt, sodass 18 Knoten für die Erstellung eines interpolierten Pfades zur Verfügung stehen. Dafür wird die MATLAB-Funktion *fit* mit der Option *smoothingspline* angewendet. Abb. 4.9 zeigt einen Beispiel-Pfad, der als Lösung der Positionsvorgabe für das Ego-Fzg gesetzt wird.

Sobald der Algorithmus innerhalb einer Schleife läuft, wird die Interpolation alle acht Zeitschritte aktualisiert, um einen kontinuierlichen Verlauf zu gewährleisten.

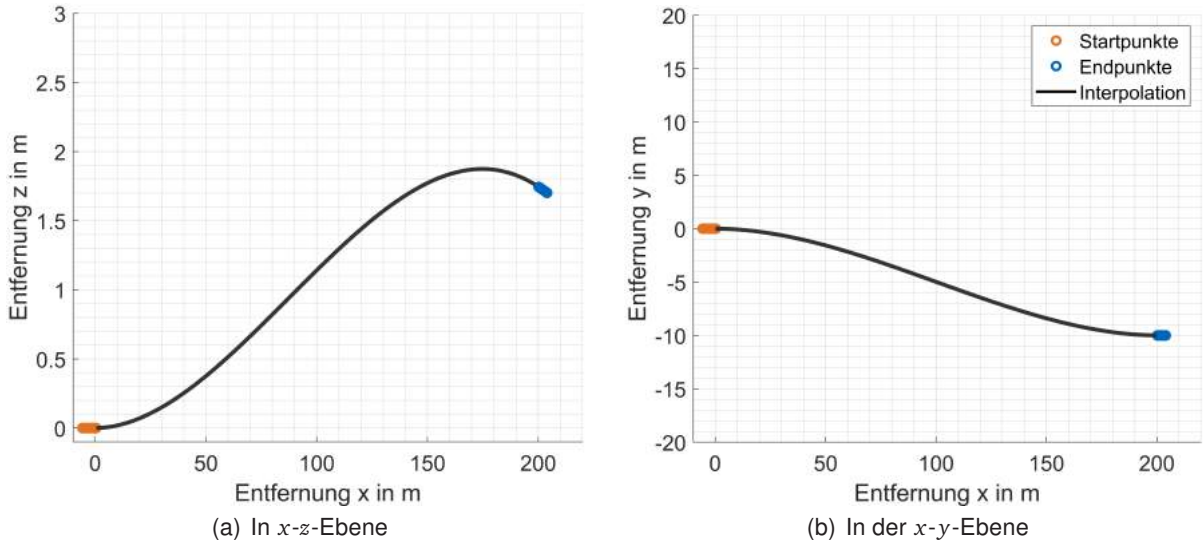


Abbildung 4.9: Generierung des Interpolationspfades für die Richtungsvorgabe des Ego-Fzg

Algorithmus zur Erstellung der Fahrzeugpositionen

Der weitere Verlauf der Szenariogenerierung lässt sich in zwei Abschnitte einteilen. Im ersten Teil wird die Position des Referenzknotens $\mathbf{x}_{\text{Ego,ref}}$ zur Richtungsvorgabe für das Ego-Fzg anhand des Interpolationspfades aus Abb. 4.9 ermittelt. Im zweiten Teil wird diese Referenz aus dem befahrenen Pfad des Challengers bestimmt. Dessen Referenz $\mathbf{x}_{\text{Ch,ref}}$ wird während des gesamten Algorithmus anhand des relativen Pfades bestimmt.

Grundsätzlich bleibt ein Knoten sowohl für das Ego-Fahrzeug als auch den Challenger so lange Referenzknoten, bis seine x-Koordinate x_{Ref} größer ist als die Koordinate des Bezugsobjektes. Anschließend wird die Soll-Änderung $\Delta \mathbf{x}_{\text{Ego,Soll}}$ des Ego-Fzg berechnet als

$$\Delta \mathbf{x}_{\text{Ego,Soll}} = \mathbf{x}_{\text{Ego,ref}} - \mathbf{x}_{\text{Ego,glob}}, \quad (4.26)$$

wobei der Winkel θ_{Ego} und die Winkeländerungen $\Delta \psi_{\text{Ego}}$ und $\Delta \theta_{\text{Ego}}$ auf Überschreiten der Grenzwerte θ_{max} , $\dot{\psi}_{\text{max}}$ und $\dot{\theta}_{\text{max}}$ kontrolliert und im Bedarfsfall darauf limitiert werden.

Es muss keine Kontrolle des Gierwinkels Ψ_{Ego} erfolgen, weil die Orientierung innerhalb der x - y -Ebene anhand der Pfadvorgabe erfolgt und keine absoluten Grenzen nötig sind. Die tatsächliche Änderung ergibt sich unter Verwendung der kontrollierten Winkel:

$$\Delta \mathbf{x}_{\text{Ego}} = v_{\text{Ego}} dt \begin{pmatrix} \cos \theta_{\text{Ego}} \cos \Psi_{\text{Ego}} \\ \cos \theta_{\text{Ego}} \sin \Psi_{\text{Ego}} \\ \sin \theta_{\text{Ego}} \end{pmatrix}. \quad (4.27)$$

Wie bereits bei der Initialisierung wird ausgehend von dieser Positionsänderung und der angestrebten Änderung $\Delta \mathbf{x}_{\text{Path}}$, die Änderung der Position des Challengers berechnet. Bevor dies erfolgen kann, muss jedoch $\mathbf{x}_{\text{Ch,rel}}$ aufgrund der Positions- und Orientierungsänderung des Ego-Fzg angepasst werden. Deswegen wird die verschobene und transformierte Position $\mathbf{x}'_{\text{Ch,rel}}$ berechnet entsprechend

$$\mathbf{x}'_{\text{Ch,rel}} = \begin{pmatrix} \cos \Delta \theta_{\text{Ego}} \cos \Delta \Psi_{\text{Ego}} \\ \cos \Delta \theta_{\text{Ego}} \sin \Delta \Psi_{\text{Ego}} \\ \sin \Delta \theta_{\text{Ego}} \end{pmatrix} \circ (\mathbf{x}_{\text{Ch,rel}} - \Delta \mathbf{x}_{\text{Ego}}). \quad (4.28)$$

Anschließend kann $\Delta \mathbf{x}_{\text{Ch}}$ unter Verwendung der Gleichungen (4.22) und (4.25) mit $\mathbf{x}'_{\text{Ch,rel}}$ statt $\mathbf{x}_{\text{Ch,rel}}$ berechnet werden, wobei erneut die Winkel und deren Änderungen kontrolliert werden. Die beschriebene Abfolge aus Operationen, die durchlaufen wird, um aus der vorausgehenden Position die Positionsänderung und nächste Position zu bestimmen, wird solange wiederholt, bis das Ego-Fzg die Initialposition des Challengers erreicht hat. Ab diesem Punkt können anstelle der interpolierten Positionsvorgabe für das Ego-Fzg die vergangenen Positionen des Challengers verwendet werden, ohne dass sich der Algorithmus anderweitig ändern muss. Erneut werden die Operationen ausgeführt, bis ein Minimalabstand ϵ zwischen den beiden Fahrzeugen unterschritten wird.

Die so ermittelten, aus dem relativen Annäherungspfad generierten Trajektorien der Fahrzeuge bilden direkt ein relevantes Szenario für den Test automatisierter Fahrzeuge.

5 Ergebnisse und Diskussion der Pfadfindungsalgorithmen

Die Ergebnisse der beiden Pfadfindungsalgorithmen A* (Abschnitt 5.1) und ACO (Abschnitt 5.2) werden in diesem Kapitel präsentiert und diskutiert. Anschließend findet ein bewertender Vergleich der entwickelten Algorithmen inklusive der Auswahl des für die Anwendung besser geeigneten statt (Abschnitt 5.3).

5.1 A*-Algorithmus

Der Abschnitt beschäftigt sich mit der Präsentation der Ergebnisse des A*-Algorithmus, der in Abschnitt 4.2 beschrieben ist. Zunächst wird die vorgeschlagene Parametrierung, im Folgenden Ausgangsparametrierung genannt, analysiert (Abschnitt 5.1.1) und anschließend mit den Ergebnissen einer Parametervariation mit Startknoten innerhalb des Sensorfeldes (Abschnitt 5.1.2) verglichen. Daraufhin wird eine Variation bei Verwendung eines Startknotens außerhalb des Sensorfeldes durchgeführt (Abschnitt 5.1.3). Zuletzt wird der A*-Algorithmus im Kontext der erzielten Ergebnisse diskutiert (Abschnitt 5.1.4).

5.1.1 Ausgangsparametrierung

Die Untersuchungen anhand des Minimalbeispiels in Abschnitt 4.2 zeigen, dass die Wahl $k_K = 0,25$ einen guten Kompromiss zwischen Pfadlänge und Erfassungswahrscheinlichkeit darstellt. Verwendet werden für die Pfadfindung die in Abschnitt 4.4.2 vorgeschlagenen Inkremente $dx = dy = 1$ m und $dz = 0,2$ m sowie der RQ43,5 der EKA1A. Die Modellierung der Sensoren erfolgt mit $dr = 1$ m als radialem Inkrement, $d\varphi = 1^\circ$ als Inkrement des Azimutwinkels und $d\theta = 0,8^\circ$ als Inkrement der Elevation. Die Pfadfindung wird zwischen dem Startknoten $v_S = (200 \text{ m}, -20 \text{ m}, 2 \text{ m})^T$ und dem Zielknoten $v_Z = (0 \text{ m}, 0 \text{ m}, 0,6 \text{ m})^T$ durchgeführt.

Der Algorithmus liefert den Ergebnispfad aus Abb. 5.1. In größerer Entfernung zum Zielknoten v_Z wird zunächst die Erfassungswahrscheinlichkeit P_D gesenkt, indem der Pfad in den Randbereich des Suchraumes verläuft. Dabei wird die Distanz zu v_Z konstant gesenkt und keine Streckenverlängerung für niedrigere P_D in Kauf genommen. Je näher am Ego-Fzg sich die betrachteten Knoten befinden, desto größer wird P_D , auch in Folge der Fusion unterschiedlicher Sensoren. Dadurch sinken die Unterschiede zwischen den Knoten und die Distanz gewinnt einen höheren Einfluss, wodurch eine relativ zentrale, leicht in z und y versetzte Annäherung erfolgt. Der ermittelte Pfad besitzt die Kosten $K = 198,8$.

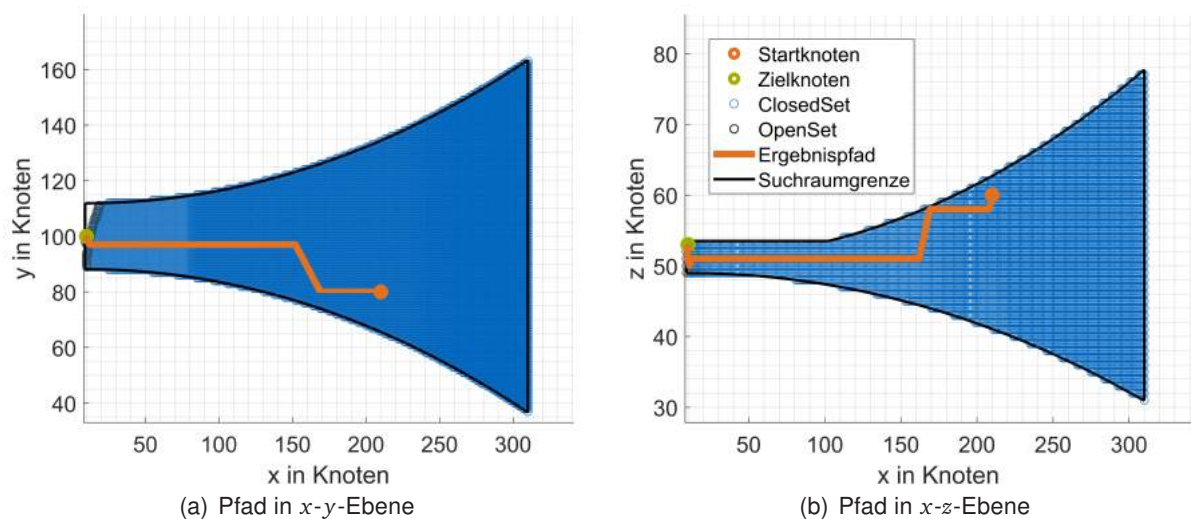


Abbildung 5.1: Ergebnis der Pfadfindung des A*-Algorithmus in Ausgangsparametrierung dargestellt im Suchraum

In Tabelle 5.1 ist die Anzahl der Knoten des Ergebnispfades angegeben, aufgeteilt nach longitudinaler Entfernung und Erfassungswahrscheinlichkeit. 133 bzw. knapp 66 % der Knoten liegen bei einer sehr hohen Erfassungswahrscheinlichkeit von $P_D \geq 99,9$ %, wobei diese vollständig im Bereich $x < 150$ m liegen. Erst ab einer Entfernung von $x \geq 150$ m werden Knoten mit $P_D \leq 90$ % verwendet, die insgesamt mit 52 Knoten ca. 26 % des Pfades ausmachen.

Auffällig ist, dass nur drei Knoten bei weniger als $P_D = 70$ % liegen, jedoch im Nahbereich mit $x < 50$ m. Unter einer Entfernung in longitudinaler Richtung von 147 m existieren insgesamt nur zwölf Knoten, die $P_D = 99,5$ % unterschreiten. Diese zwölf Knoten mit $P_D = 0$ % liegen mit einer Entfernung von maximal 2,5 m zum Ursprung in dem von MÜLLER [27] aufgezeigten, nicht erfassten Nahfeldbereich oder blinden Bereich des Fahrzeugs, bzw. teilweise innerhalb des Fahrzeuges. Das ergibt sich aus der Sensormodellierung. Diese Knoten besitzen im Vergleich zu ihrer Umgebung deutlich geringere Kosten, sodass der Algorithmus diese ausnutzt und der Pfad lateral leicht versetzt bei $y = -3$ m verläuft. Eine angepasste Betrachtung im Nahfeld, die unzulässige Knoten im Fahrzeug ausschließt bzw. die Annahme einer Erfassungswahrscheinlichkeit von $P_D = 100$ % für diese Knoten kann die Güte bzw. Realitätskonformität des Ergebnispfades steigern.

Tabelle 5.1: Anzahl Knoten des Ergebnispfades nach Erfassungswahrscheinlichkeit P_D in % und x -Koordinate in m

P_D in %	$0 \leq x < 50$	$50 \leq x < 100$	$100 \leq x < 150$	$150 \leq x < 200$	$200 \leq x$	Gesamt
$0 \leq P_D < 70$	3	0	0	0	0	3
$70 \leq P_D < 80$	0	0	0	23	1	24
$80 \leq P_D < 90$	0	0	0	25	0	25
$90 \leq P_D < 99$	0	0	2	2	0	4
$99 \leq P_D < 99,9$	0	0	14	0	0	14
$99,9 \leq P_D$	49	50	34	0	0	133
Σ	52	50	50	50	1	203

Abb. 5.2 zeigt den Pfad im kartesischen Koordinatensystem mit den tatsächlichen Entfernungen x, y, z und eingezeichneten Erfassungsbereichen aller Sensoren. Die laterale und vertikale Veränderung im Pfad erfolgt kurz vor Beginn des Erfassungsbereiches von Kamera und Lidar (Abb. 5.2(a) und 5.2(b)), die durch die Sensormodellierung von MÜLLER [27] auf 150 m begrenzt sind.

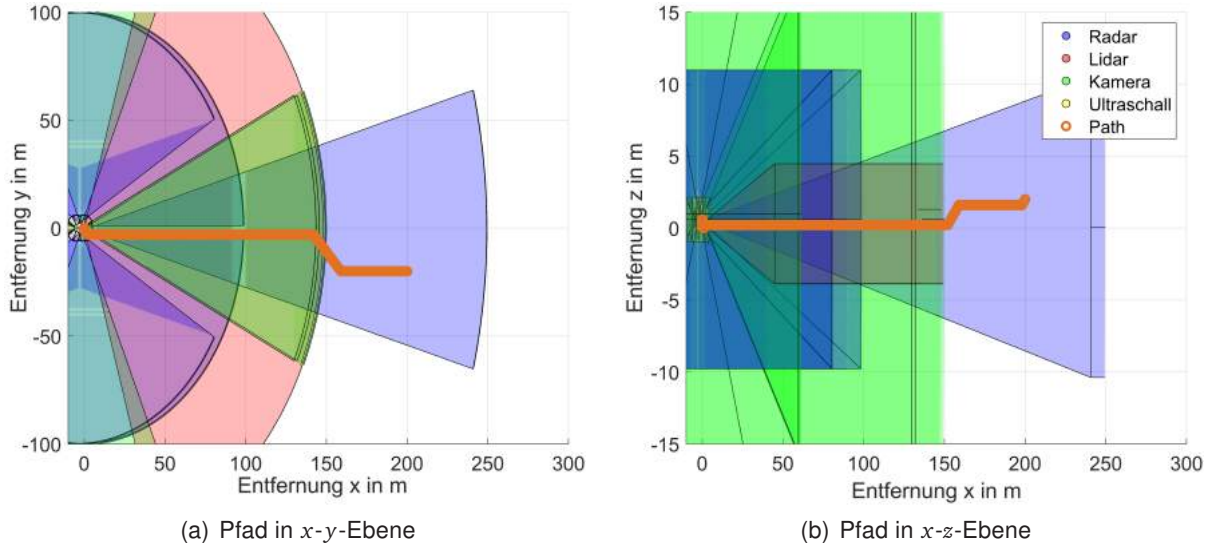


Abbildung 5.2: Ergebnis der Pfadfindung des A*-Algorithmus in Ausgangsparametrierung dargestellt mit Sensorabdeckung

Zur genaueren Erklärung dieses Verlaufs wird die Erfassungswahrscheinlichkeit über der x -Koordinate der Knoten des Ergebnispfades in Abb. 5.3 betrachtet. Für den Bereich $x \leq 150$ m bzw. fast 75 % des Pfades liegt die Erfassungswahrscheinlichkeit bei $P_D \approx 100$ %, wohingegen $P_D \leq 91$ % für $x > 150$ m gilt. Dieser Unterschied ergibt sich, weil für $x > 150$ m der Radar das alleinige Sensorsystem ist, bei Unterschreiten dieser Grenze die drei Systeme Radar, Kamera und Lidar jedoch eine deutlich höhere, fusionierte Erfassungswahrscheinlichkeit bilden. Die Maximierung des Radar-Bereichs bzw. Minimierung des fusionierten Bereichs ergibt die geringsten Kosten für den Algorithmus. Daraus resultiert ein möglichst langes Verweilen im Randbereich des Radars mit anschließender Annäherung auf eine y - und z -Koordinate, die im weiteren Verlauf quasi konstant bleiben.

Die Linearität der Annäherung ab $x < 150$ m resultiert ebenfalls aus der hohen Erfassungsgüte. Eine Erfassungswahrscheinlichkeit von $P_D \approx 100$ % bedeutet für die Kostenfunktion, dass nur noch die Pfadlänge betrachtet wird, unabhängig von k_K und P_D (Gleichung (5.1)). In Bereichen hoher Erfassungsgüte wird das Pfadfindungsproblem zum klassischen Kürzeste-Wege-Problem.

$$K(P_D = 100 \%) = \frac{k_K + \overset{1}{P_D}}{k_K + 1} L_P = \frac{k_K + \overset{1}{1}}{k_K + 1} L_P = L_P \quad (5.1)$$

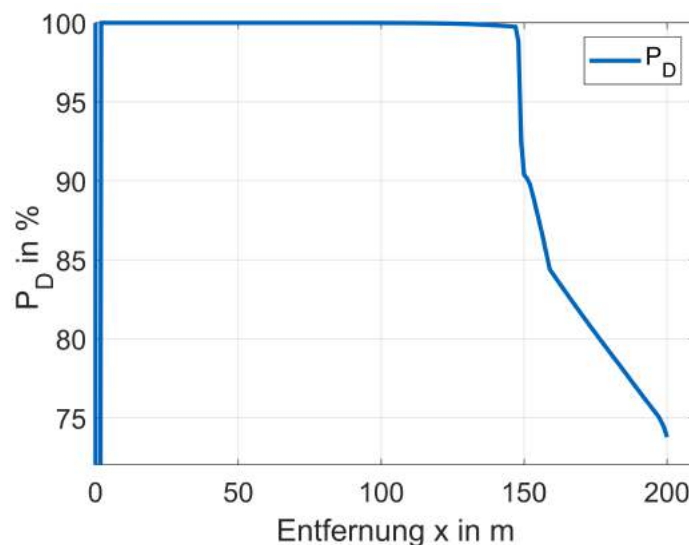


Abbildung 5.3: Erfassungswahrscheinlichkeit P_D des Ergebnispfades der Ausgangsparametrierung, aufgetragen über der Projektion des Pfades auf die x -Achse

5.1.2 Variation innerhalb Sensorabdeckung

Bereits am Minimalbeispiel in Abschnitt 4.2 wird die Parametrierung des Algorithmus untersucht. Die Korrektheit und Auswirkung der gewählten Parameter wird durch eine erneute Parametervariation geprüft. Es werden Start- und Zielknoten aus Abschnitt 5.1.1 verwendet. Neben $k_K = 0,25$ der Ausgangsparametrierung werden für die Variation die Streckengewichtungen $k_K = 0,5$ und $k_K = 0,1$ getestet. In Abschnitt 4.2 stellt sich der gewählte Bereich der Gewichtung als geeignet für die Bewertung der Sensibilität auf die vorliegenden Erfassungswahrscheinlichkeiten dar. Zusätzlich wird der Einfluss der Inkremente überprüft, indem die drei Stufen aus Tabelle 5.2 abgeprüft werden. Die Inkremente der Kugelkoordinaten (dr radial, $d\varphi$ azimuthal, $d\theta$ in Elevation) werden für die Erstellung des Sensormodells nach MÜLLER [27] benötigt. Alle weiteren Parameter und Bedingungen werden beibehalten. Jede Inkrementvariation wird mit jedem Parameter berechnet, sodass neun Ergebnispfade entstehen (Abb. 5.4). Die Variation der Konstante wird durch eine Farbveränderung und die Inkrementvariation durch eine Symbolveränderung abgebildet. Liegen mindestens zwei der drei Pfade einer Inkrementstufe übereinander, wird die Markierung in schwarz dargestellt. Ist ein Pfad farbig markiert, verdeckt er keinen weiteren Pfad. Die Variationen haben nur geringen Einfluss auf den Ergebnispfad (Abb. 5.4). Die hohe Übereinstimmung zwischen den Pfaden führt dazu, dass nur die Markierungen von vier Pfaden zu sehen sind. Für grobe und mittlere Inkremente liegen alle Markierungen übereinander, sodass zwei Pfade mit schwarzen Markierungen zu erkennen sind. Für feine Inkremente und $k_K = 0,5$ bestimmt der Algorithmus einen unterschiedlichen Pfad. Die Markierungen der anderen beiden Pfade liegen erneut deckungsgleich. Tabelle 5.3 führt die Kosten und Längen der Variationen auf. Die übereinstimmenden Längen der Pfade innerhalb einer Inkrementstufe zeigen erneut die Ähnlichkeit der Ergebnisse und die Unabhängigkeit von der Wahl von k_K , die bereits in Abb. 5.4 zu erkennen ist.

Tabelle 5.2: Auflistung der verwendeten Inkremente: dx longitudinal, dy lateral, dz vertikal und für die Sensormodellierung nach MÜLLER [27]: dr radial, $d\varphi$ azimuthal, $d\theta$ in Elevation

Inkrement	dx in m	dy in m	dz in m	dr in m	$d\varphi$ in °	$d\theta$ in °
fein	1	1	0,2	1	1	0,4
mittel	2	2	0,4	2	2	0,8
grob	4	4	0,8	2	2	0,8

Tabelle 5.3: Kosten K und Länge L_p der ermittelten Pfade des A*-Algorithmus bei Startpunkt innerhalb des Sensorfeldes nach Inkrementen und Parametrierung von k_K

Inkrementale Parameter	fein		mittel		grob	
	K	L_p in m	K	L_p in m	K	L_p in m
$k_K = 0,5$	200,5	208,4	198,2	208,9	200,7	208,4
$k_K = 0,25$	198,8	209,3	196,0	208,9	199,1	208,4
$k_K = 0,1$	197,3	209,3	194,3	208,9	197,9	208,4

Die Erklärung für die hohe Ähnlichkeit liefert die bereits angeführte Tatsache, dass sich die Padsuche fast durchgängig im Bereich hoher Erfassungswahrscheinlichkeiten befindet. Aufgrund der hohen Ähnlichkeit der Pfade kann der Verlauf von P_D (Abb. 5.3) als repräsentativ für alle Ergebnisfade betrachtet werden. Entsprechend Gleichung (5.1) liegt damit ab $x = 150$ m eine Suche nach dem kürzesten Weg ohne Einfluss der Sensorbereiche vor, wodurch die unterschiedlichen Parametrierungen nur auf den ersten 50 m Einfluss haben und die restlichen entstehenden Unterschiede inkrementbedingt sind.

Beim Vergleich der Pfadkosten unterschiedlicher Inkremente zeigt sich, dass die mittleren Inkremente die geringsten Kosten produzieren. Daraus ließe sich ableiten, dass diese am besten geeignet sind. Allerdings ergibt sich dieser Umstand aus den bereits erwähnten blinden Bereichen im Nahfeld des Ego-Fzg. Für beide Schrittweiten liegen der viertletzte bis vorletzte Knoten innerhalb dieses Bereichs. Das ergibt für mittlere Inkremente eine nicht erfasste Pfadlänge von 7,7 m bzw. ein Segment mit $K(k_K = 0,25) = 1,5$, wohingegen für feine nur eine Länge von 3,9 m bzw. ein Segment mit $K(k_K = 0,25) = 0,7$ vorliegt. Die direkte Umgebung der nicht erfassten Knoten besitzt $P_D = 1$, sodass für diese Bereiche die Kostenfunktion der Pfadstrecke entspricht (Gleichung (5.1)). Der Kostenunterschied entspricht der durch die Inkrementwahl unterschiedlichen Distanz innerhalb des blinden Bereichs und beträgt somit $7,7 - 3,9 = 3,8$. Die mittleren Inkremente liefern damit Ergebnisse mit geringeren Kosten, weil die nicht erfassten Bereiche aufgrund der Interpolation der Erfassungswahrscheinlichkeiten ein größeres Areal um das Ego-Fzg einnehmen, als es bei der genaueren Erfassung durch die feineren Inkremente der Fall ist. Ansonsten entsprechen sich die Ergebnisfade beinahe. Der einzige Unterschied liegt in der Wahl der z -Koordinate im Bereich $x \leq 150$ m. Allerdings liegt die Höhe der feinen Inkremente zwischen zwei der mittleren Inkremente, sodass diese nicht vom Algorithmus gewählt werden kann. Die groben Inkremente können die blinden Bereiche nicht auflösen. Deswegen ermitteln diese einen anderen Pfad, der innerhalb des Sensorfeldes streckenoptimal ist.

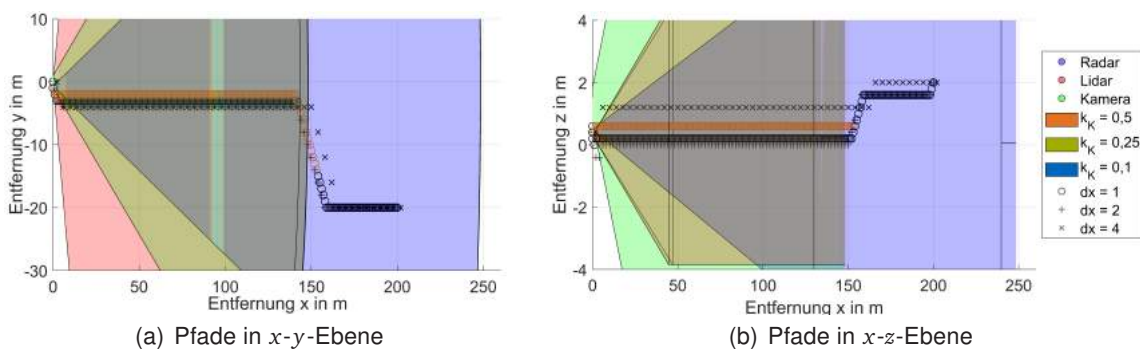


Abbildung 5.4: Ergebnisfade der Variation bei Startknoten im Sensorfeld, Variation der Inkremente durch Symbolveränderung, Variation von k_K durch Farbveränderung, schwarze Symbole für verdeckende Pfade gleichen Inkrements

Das zeigt, dass eine getrennte Betrachtung für Fern- und Nahfeld einen Zugewinn bietet. Im Fernfeld ist eine mittlere oder sogar grobe Schrittweite ausreichend genau. Im Nahfeld werden optimalerweise feine Inkremente verwendet. Zusätzlich kann eine besondere Handhabung der blinden Stellen erfolgen.

Die Wahl der feinen Inkrementierung in Kombination mit $k_K = 0,5$ produziert einen kürzeren Pfad aufgrund der höheren Gewichtung der Pfadlänge. Anstatt wie die anderen beiden Parametrierungen die nicht erfassten Knoten im Bereich $0 \text{ m} \leq z \leq 0,4 \text{ m}$ auszunutzen, verweilt der Algorithmus bei $z = 0,6 \text{ m}$ und nutzt nur einen nicht erfassten Knoten bei $z = 0,4 \text{ m}$. Die nötige Streckenverlängerung für die Ausnutzung weiterer nicht erfasster Punkte entsprechend dem optimalen Pfad für $k_K = 0,25$ und $k_K = 0,1$ verursacht einen Kostenzuwachs von 0,02. Das zeigt, dass die Wahl der Kostenfunktion sensitiv genug ist, um einen Kompromiss aus Pfadlänge und Erfassungswahrscheinlichkeit entsprechend der gewählten Parametrierung abzubilden.

5.1.3 Variation außerhalb Sensorabdeckung

Für die ersten Berechnungen liegt der Startknoten v_s innerhalb des Sensorfeldes. Daraus folgt eine kontinuierlich hohe Erfassungswahrscheinlichkeit, die nur zu geringen Veränderungen der Pfade unterschiedlicher Parameter führt. Vergleichend wird nun der Startknoten v_s außerhalb des Sensorfeldes gelegt, sodass zu Beginn die Erfassungswahrscheinlichkeit $P_D = 0 \%$ ist. Dadurch können sich die Algorithmen den parametrierungsspezifisch optimalen Eintrittspunkt in das Sensorfeld bestimmen. Verwendet wird $v_s = (300 \text{ m}, -20 \text{ m}, 2 \text{ m})^T$ und $v_z = (0 \text{ m}, 0 \text{ m}, 0,5 \text{ m})^T$ bei groben und mittleren Inkrementen nach Tabelle 5.2.

Zunächst werden grobe Inkremente betrachtet. In Abb. 5.5 werden zur besseren Darstellbarkeit der Pfade die Erfassungsbereiche der drei maßgeblichen Sensorsysteme dargestellt. Das sind der Front-Radar, der Front-Lidar und die Front-Kamera. Alle anderen Sensorfelder sind ausgeblendet, aber für die Berechnung berücksichtigt.

In Tabelle 5.4 werden die Pfadlängen und Kosten der verschiedenen Parametrierungen angegeben. Die berechneten Pfade werden in allen drei Parametrierungen bewertet. Entspricht die zur Bewertung verwendete Konstante derjenigen, mit der der Pfad berechnet wird, wird der Wert fett gedruckt.

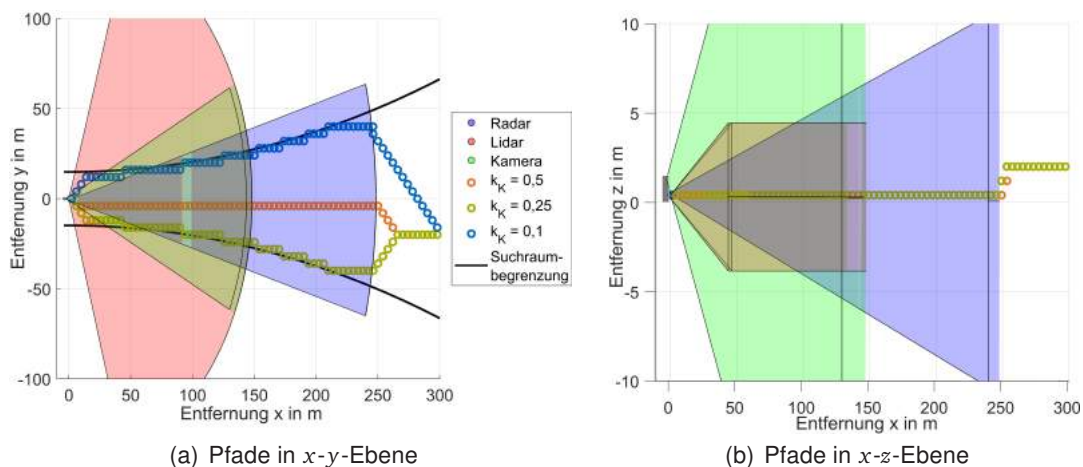


Abbildung 5.5: Ergebnispfade des A*-Algorithmus mit groben Inkrementen und Startpunkt außerhalb des Sensorfeldes

Tabelle 5.4: Kosten K und Länge L_p der ermittelten Pfade des A*-Algorithmus bei Startpunkt außerhalb des Sensorfeldes und groben Inkrementen

Berechnungskonstante	$K(k_K = 0,5)$	$K(k_K = 0,25)$	$K(k_K = 0,1)$	L_p in m
$k_K = 0,5$	251,3	239,9	230,6	308,4
$k_K = 0,25$	252,6	238,1	226,3	325,0
$k_K = 0,1$	257,5	240,0	226,1	343,9

Abb. 5.5(b) zeigt, dass alle drei Varianten den Pfad außerhalb des Sensorfeldes auf $z = 0$ m legen. Dadurch wird in dieser Ebene die Distanz innerhalb des Sensorfeldes minimal. Die Variante $k_K = 0,5$ tritt bei $y = -4$ m in das Sensorfeld ein im Vergleich zu $y = -40$ m ($k_K = 0,25$) und $y = 40$ m ($k_K = 0,1$). Aufgrund der kugelförmigen Begrenzung sinkt die z -Koordinate für $k_K = 0,5$ bereits ein Inkrement früher ab. Ansonsten verlaufen die Pfade identisch.

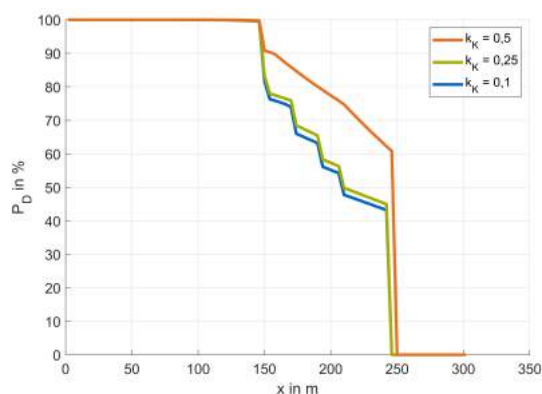
In der x - y -Ebene zeigen sich deutlich größere Unterschiede bezüglich der Wahl des Eintrittspunktes und des Ergebnispfades. Für $k_K = 0,5$ optimiert der Algorithmus die Distanz und verläuft um ein Inkrement zur Mitte des Ego-Fzg versetzt. Der Einfluss der Erfassungswahrscheinlichkeit ist zu gering, sodass keine Streckenverlängerungen in Kauf genommen werden. Dies ist für die beiden anderen Varianten nicht der Fall. Der Pfad für $k_K = 0,25$ verläuft an der negativen Suchraumgrenze in y und für $k_K = 0,1$ an der positiven. Der Wechsel der Suchraumgrenze lässt sich anhand der Positionen der vier dominanten Sensoren (ein Radar, ein Lidar, zwei Kameras) erklären, die maßgebenden Einfluss auf die Erfassungswahrscheinlichkeit haben. Die beiden Kameras befinden sich bei $(0 \text{ m}, 0 \text{ m}, 0,6 \text{ m})^T$ und $(-2 \text{ m}, 0 \text{ m}, 1,3 \text{ m})^T$, der Lidar bei $(0 \text{ m}, 0 \text{ m}, 0,3 \text{ m})^T$ und der Radar bei $(0 \text{ m}, -0,7 \text{ m}, 0,5 \text{ m})^T$. Der laterale Versatz von $-0,7$ m des Radars hat zur Folge, dass auf dieser Seite des Fahrzeuges P_D leicht erhöht liegt. Diesen Unterschied nutzt der Algorithmus für $k_K = 0,1$ aus, wofür er eine Streckenverlängerung von 18,9 m in Kauf nimmt, um die Pfadkosten um 0,2 zu senken (Tabelle 5.4). Für $k_K = 0,25$ überwiegt die kürzere Strecke den Einfluss der höheren Erfassungswahrscheinlichkeit, sodass die Pfadkosten um 1,9 niedriger liegen als die des Pfades für $k_K = 0,1$ in der Bewertung mit $k_K = 0,25$ bewertet (Tabelle 5.4). Beide Algorithmen verlaufen am Rande des Suchraums. Die Verwendung des zentral verlaufenden Pfades für $k_K = 0,5$ steigert die Kosten in den anderen beiden Bewertungen um 1,8 ($k_K = 0,25$) bzw. 4,5 ($k_K = 0,1$) (Tabelle 5.4).

Den Unterschied der Erfassungswahrscheinlichkeiten entlang der Pfade zeigt Abb. 5.6(a), wobei die Werte über der Projektion des Pfades auf die x -Achse aufgetragen werden. Beachtet man den gespiegelten Verlauf der Pfade für $k_K = 0,1$ und $k_K = 0,25$ bei betragsmäßig gleicher y -Koordinate, ist der durch die Radarposition bedingte Unterschied der Erfassungswahrscheinlichkeit im Bereich $150 \text{ m} \leq x \leq 250 \text{ m}$ eindeutig erkennbar. Ab dem Beginn des fusionierten Erfassungsbereichs ist dieser nur noch marginal vorhanden.

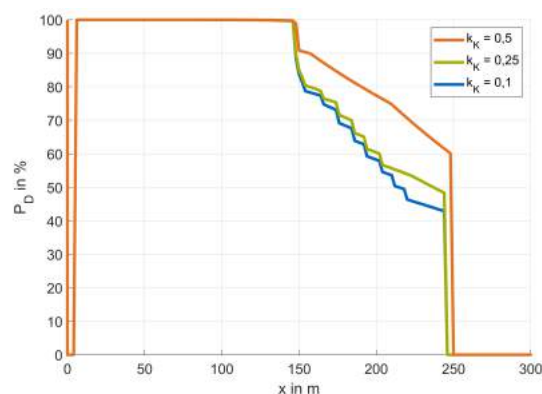
Unter Verwendung mittlerer Inkremente erhält man makroskopisch identische Ergebnisse bzw. Verläufe der Ergebnispfade (Abb. 5.7). Bei detaillierter Betrachtung ergeben sich Unterschiede. Wie bereits bei der Ausgangsparametrierung diskutiert (Abschnitt 5.1.2), ist der Algorithmus infolge der besseren lokalen Auflösung der mittleren Inkremente in der Lage, die blinden Bereiche im Nahfeld des Fahrzeuges auszunutzen. Das geschieht in allen drei Parametrierungen (Abb. 5.6(b)). Für die Wahl von $k_K = 0,5$ verlängert sich der Pfad um 0,5 m, die Kosten sinken jedoch um 0,6 (Tabelle 5.4 und Tabelle 5.5).

Tabelle 5.5: Kosten K und Länge L_P der ermittelten Pfade des A*-Algorithmus bei Startpunkt außerhalb des Sensorfeldes und mittleren Inkrementen

Berechnungskonstante	$K(k_K = 0,5)$	$K(k_K = 0,25)$	$K(k_K = 0,1)$	L_P in m
$k_K = 0,5$	250,7	239,1	229,6	308,9
$k_K = 0,25$	252,1	238,1	226,6	322,1
$k_K = 0,1$	258,1	240,6	226,3	345,5



(a) Grobe Inkremente



(b) Mittlere Inkremente

Abbildung 5.6: Erfassungswahrscheinlichkeit P_D des Ergebnispfades mit Startknoten außerhalb Sensorfeld, aufgetragen über der Projektion des Pfades auf die x -Achse

Die mittleren Inkremente zeigen den Unterschied zwischen der Verwendung von $k_K = 0,25$ und $k_K = 0,1$. Für $k_K = 0,25$ verschiebt sich der Eintrittspunkt in das Sensorfeld von $y = -40$ m auf $y = -35$ m, sodass ein um 2,9 m kürzerer Pfad identischer Kosten trotz einer Streckenverlängerung im Nahfeld verwendet wird. Im Gegensatz dazu bleibt der Eintrittspunkt für $k_K = 0,1$ gleich, aber die Kosten steigen um 0,2. Durch die genauere Auflösung des Suchraumes werden die Erfassungswahrscheinlichkeiten des Radars genauer berechnet. Aufgrund der Interpolation steigt deswegen die Erfassungswahrscheinlichkeit der mittleren Inkremente gegenüber groben derselben Knoten an. Dadurch kommt es zur Kostensteigerung. Zusätzlich ermöglicht die feinere Auflösung dem Algorithmus eine genauere Approximation der Suchraumgrenze, wodurch die Pfadlänge steigt. In Abb. 5.6(b) im Vergleich zu Abb. 5.6(a) ist dies anhand der höheren Anzahl an Sprüngen der Erfassungswahrscheinlichkeit zu erkennen. Dies kommt zu Stande, weil der Algorithmus die geringere Erfassungswahrscheinlichkeit an der Suchraumgrenze bevorzugt.

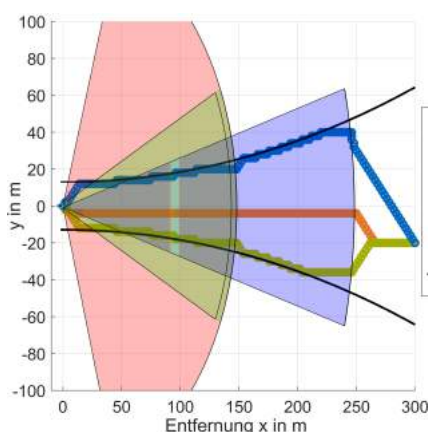
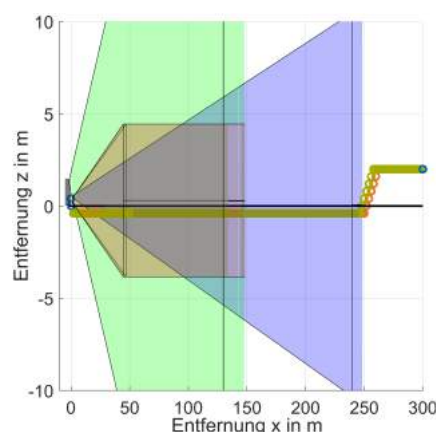

(a) Pfade in x - y -Ebene

(b) Pfade in x - z -Ebene

Abbildung 5.7: Ergebnispfade des A*-Algorithmus mit mittleren Inkrementen und Startpunkt außerhalb des Sensorfeldes

5.1.4 Diskussion des A*-Suchalgorithmus

Die Ergebnisse zeigen, dass die Anforderung eines sensorsensitiven Suchalgorithmus erfüllt wird. Die Kombination aus Distanz und Erfassungswahrscheinlichkeit in der Heuristik- und Kostenfunktion des A* ermöglicht die Generierung herausfordernder Szenarien unter Berücksichtigung der Sensorabdeckung. Durch die Wahl des Parameters k_K kann ein einstellbarer Fokus auf die Pfadlänge oder Erfassungsgüte gelegt werden. Das zeigen vor allem die Ergebnisse der Variation unter Verwendung eines Startknotens außerhalb der Sensorabdeckung. Die Verwendung von $k_K = 0,5$ führt zu einem starken Fokus auf die Pfadlänge. $k_K = 0,1$ richtet sich an der Erfassungswahrscheinlichkeit aus und nimmt dafür auch deutlich längere Pfade in Kauf. $k_K = 0,25$ erzielt einen Kompromiss aus beiden Charakteristika.

Daraus kann eine Parameterempfehlung für die Anwendung abgeleitet werden. Im allgemeinen Fall wird aufgrund des ausgewogenen Kompromisses zwischen Pfadlänge und Erfassungswahrscheinlichkeit $k_K = 0,25$ empfohlen. Liegt ein besonderer Fokus auf der Minimierung der Erfassungsgüte, ist $k_K = 0,1$ geeignet. Liegt der Startpunkt innerhalb des Sensorfeldes kann sogar aufgrund der Annäherung der Kostenfunktion an die Distanz (Gleichung (5.1)) die Wahl noch geringerer Werte in Betracht gezogen werden. Diese Annäherung ist auch sinnvoll, weil bei sicherer Erfassung ($P_D = 100\%$) der kürzeste Weg der kritischste ist.

Die Sensibilität auf Bereiche geringer Erfassungswahrscheinlichkeit zeigt das konsequente Ausnutzen der blinden Bereiche in allen Parametrierungen. Allerdings muss diskutiert werden, ob diese Bereiche tatsächlich sinnvoll sind. MÜLLER [27] zeigt, dass vor allem im Nahfeld die z -Koordinate hohen Einfluss auf die Erfassungsgüte hat. Im aktuellen Fall werden Punktmassen betrachtet, die sich auf einem Knotengitter bewegen. Aufgrund der Interpolation ist ein Knoten repräsentativ für einen Quader der Größe $dx * dy * dz$. Entsprechend des Inkrementvorschlags aus Tabelle 5.2 nimmt dz zwischen 0,2 m und 0,8 m ein. Die Höhe üblicher Detektionsobjekte (Mensch, PKW, LKW, Motorrad) überschreitet diese Werte. Ein tatsächlicher blinder Bereich liegt erst dann vor, wenn kein einziger Punkt des Objektes erfasst werden kann. Diese Betrachtungsweise ist jedoch sehr rechenaufwändig, weil die Lage sämtlicher Punkte kontrolliert und deren Wahrscheinlichkeiten kombiniert werden müssen. In größeren Entfernungen zum Ego-Fzg spielt das außerdem nur eine untergeordnete Rolle. Die Aufteilung des Algorithmus in einen Nah- und einen Fernfeldbereich adressiert diese Problematik. Zusätzlich bietet das die Möglichkeit, in der näheren Umgebung des Fahrzeuges feinere Inkremente zu verwenden und so lokale Besonderheiten genauer aufzulösen, gleichzeitig aber die Anzahl an Knoten beherrschbar gering zu halten.

Die Wahl der Inkremente nimmt durch die Anzahl an Knoten und die Interpolation der Werte großen Einfluss auf die Performanz des Algorithmus. Für die resultierenden Pfade gilt: je feiner, desto besser. Die Wahl kleinerer Inkremente führt zu besserer Repräsentation des Suchraums durch die Knoten und kann lokale Unterschiede besser abbilden. Das ermöglicht dem Algorithmus, diese Gebiete gezielt auszunutzen. Erkennbar ist dieses Verhalten anhand der Diskussion der nicht erfassten Bereiche im Nahfeld (Abschnitt 5.1.2) und der verbesserten Annäherung an die Suchraumbegrenzung (Abschnitt 5.1.3).

Die Feinheit der Inkremente bedingt die Anzahl der Knoten. In Abschnitt 5.1.3 werden 9069 Knoten durch die grobe Variante, 58825 durch die mittlere und 419155 durch die feine bewertet. Zur Berechnung wird ein Dell Precision T3600 mit 64 GB und mit einem Intel Xeon Prozessor mit 3,6 GHz und vier Kernen verwendet. Dieser benötigt durchschnittliche Laufzeiten von: grob - 4 min 8 s, mittel - 32 min 53 s, fein - 8 h 10 min 18 s. Die Variation mit Startknoten innerhalb und außerhalb der Sensorabdeckung zeigen beide eine Beeinflussung durch die Verwendung grober statt mittlerer Inkremente. Der Unterschied zwischen mittel und fein fällt im vorliegenden Fall geringer aus. Die Laufzeit eines Algorithmus ist wichtiger Bewertungsgegenstand, auch wenn

er für die angestrebte Anwendung nur von untergeordneter Bedeutung ist, weil keine besonderen Anforderungen wie Echtzeitfähigkeit gestellt werden. Dennoch ermöglicht eine geringere Laufzeit eine schnellere Berechnung von Einzelergebnissen und Variationen und ermöglicht die Komplexitätssteigerung des Algorithmus, beispielsweise durch zusätzliche Bedingungen.

Aufgrund der vorliegenden Arbeit können Empfehlungen bezüglich der Parametrierung getroffen werden. Die Verwendung mittlerer Inkremente kann für die grundlegende Analyse einer Sensorkonfiguration erfolgen. Beispielsweise kann dadurch die Lage eines geeigneten Startpunktes oder Asymmetrie in der Erfassungsgüte erarbeitet werden. Theoretisch ist auch die Wahl grober Inkremente dafür möglich, allerdings steigt damit das Risiko, lokale Besonderheiten nicht zu erfassen. Die angestrebte Laufzeitverbesserung gegenüber mittleren Inkrementen ist nicht zwingend erforderlich, da die Laufzeit dort bereits gut genug ist. Zusätzlich eignen sich mittlere Inkremente für die Analyse unterschiedlicher Parameter-Variationen, z.B. von k_K , v_S , v_Z . Im Vergleich zu feinen Inkrementen wird ein Fünfzehntel der Rechenzeit festgestellt, sodass diese Analysen ressourceneffizient erfolgen können.

Die Erstellung einzelner spezifischer Szenarien, die für die Erprobung und Absicherung entwickelt werden, erfolgt optimalerweise in feinen Inkrementen. Dadurch kann eine möglichst gute Auflösung erreicht werden und bei nur einmaliger Berechnung ist die Laufzeit des Algorithmus im Bereich von unter zehn Stunden akzeptabel.

Zusätzlich kann die Verwendung unterschiedlicher Inkremente in x - und y -Richtung umgesetzt werden. Wie die Struktur des Suchraumes zeigt (Abb. 5.2), werden ab einer Entfernung in Längsrichtung des Ego-Fzg von 100 m weniger als 40 Knoten in lateraler Richtung betrachtet. Gerade im nahen Umfeld haben kleine Unterschiede der Position jedoch großen Einfluss. Außerdem kann durch kleine Distanzen eine kontinuierlichere Querdynamik für die anschließende Szenarioüberführung abgebildet werden.

Die Analysen der Arbeit zeigen außerdem, dass eine Aufteilung der Pfaderstellung in Nah- und Fernfeld sinnvoll sein kann. Im Fernfeld sind Inkrementunterschiede aufgrund der geringen relativen Änderung der Werte eher zweitrangig. Die Erfassungswahrscheinlichkeit befindet sich jedoch teilweise in Bereichen unter 50 %. Entsprechend kann die Parametrierung mit $k_K = 0,25$ diese Unterschiede gezielt ausnutzen (Abschnitt 5.1.3). Im Nahfeld verändert sich der Verlauf des Pfades in Abhängigkeit der Inkrementwahl. Im Gegenzug verschwindet jedoch die Differenz unterschiedlicher k_K aufgrund der sehr hohen Erfassungswahrscheinlichkeiten. Entsprechend kann dieser Parameter im Nahfeld deutlich niedriger gewählt werden, um feine Unterschiede im Bereich 99 % bis 100 % zu adressieren. Als Umschaltpunkt zwischen Nah- und Fernfeld wird 20 m vorgeschlagen. Das ermöglicht die Wahl feiner Inkremente bei überschaubarer Knotenanzahl.

Je nach Anwendungsfall ist die Betrachtung der dritten Koordinate nicht nötig. Die Ergebnispfade sämtlicher Berechnungen (Abb. 5.2, 5.4, 5.5 und 5.7) streben einmalig gegen eine bevorzugte z -Koordinate (meistens 0 m) und zeigen anschließend keinerlei Veränderung bis zum Zielknoten. Im Gegensatz zur y -Koordinate wählt der Algorithmus selbst bei $k_K = 0,1$ nicht den Bereich entlang der Suchraumgrenze. Dadurch kann die Anzahl Knoten auf einen Bruchteil reduziert werden, sodass die Inkremente deutlich feiner gewählt werden können.

Werden allerdings Verdeckungen durch andere Verkehrsteilnehmer oder topologische Elemente in den Algorithmus integriert, ist die Betrachtung der dritten Koordinate zwingend erforderlich. Selbiges gilt für Auswirkungen der Fahrdynamik auf die Orientierung des Ego-Fzg.

5.2 ACO-Algorithmus

Dieser Abschnitt befasst sich mit der Präsentation der Ergebnisse des ACO-Algorithmus. Begonnen wird in Abschnitt 5.2.1 mit der Parametrierung entsprechend Abschnitt 4.3. Diese wird anschließend in Abschnitt 5.2.2 optimiert. Abschließend wird der ACO-Algorithmus und dessen Applizierbarkeit auf die vorliegende Problemstellung diskutiert (Abschnitt 5.2.3).

5.2.1 Ergebnisse der Ausgangsparametrierung

Für die erste Berechnung werden nach Abschnitt 4.3 und Abschnitt 4.4.3 die Parameter $\alpha = 1$, $\beta = 5$, $\rho = 0,5$, $\tau_0 = 10$, $Q = 300$, $n_{\text{Gen}} = 50$, $n_a = 200$, $m_{\text{best}} = 40$ und $q_0 = 0,5$ verwendet. Die Inkremente werden zunächst nach Tabelle 5.2 grob gewählt mit $dx = dy = 4$ m und $dz = 0,8$ m. Die Berechnung der Sensorabdeckung erfolgt durch das Modell von MÜLLER [27] in dessen Parametrierung. Es wird für die Modellierung der einzelnen Sensoren $dr = 2$ m als radiales Inkrement, $d\varphi = 2^\circ$ als Inkrement des Azimutwinkels und $d\vartheta = 0,8^\circ$ als Inkrement der Elevation verwendet. Für die erste Variante wird $k_K = 0,25$ für die Konstante der Kostenfunktion gewählt. Die Ergebnisse des Algorithmus werden in Abb. 5.8 dargestellt. Abb. 5.8(b) zeigt, dass keine Verbesserung der besten Pfade der ersten Generation stattfindet und somit keine Konvergenz vorliegt. Der beste Pfad weist Kosten von $K = 281,8$ bzw. eine Abweichung von fast 50 % im Vergleich zum optimalen Pfad dieser Parametrierung auf. Dieser wird durch den A*-Algorithmus (Abschnitt 5.1.2) gefunden. Die Betrachtung der besten fünf gefundenen Pfade zeigt diskontinuierliche Verläufe mit vielen Richtungswechseln und Oszillationsbewegungen in lateraler und vertikaler Richtung. Longitudinal streben die Pfade quasi-kontinuierlich dem Zielknoten entgegen.

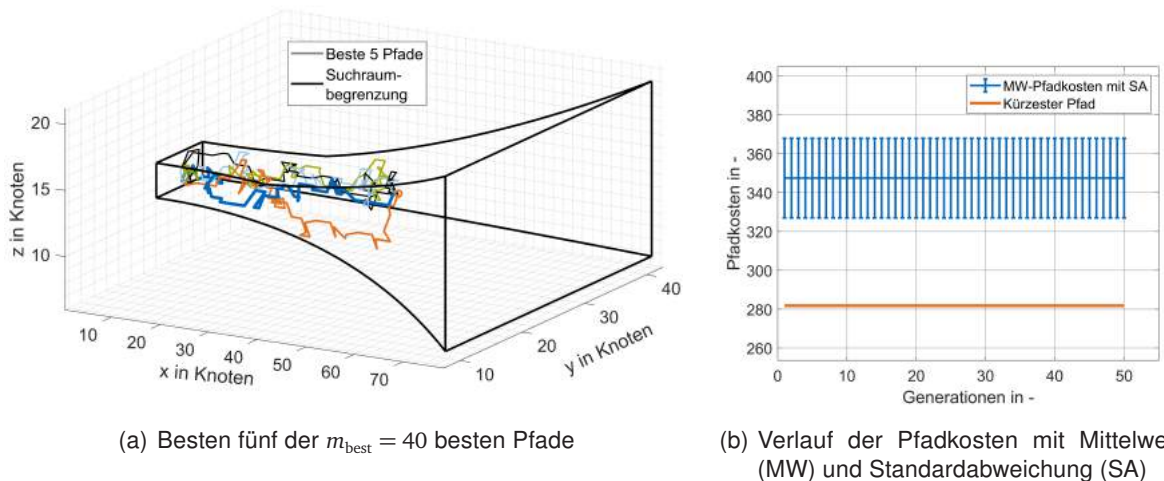


Abbildung 5.8: Ergebnis der Pfadfindung des ACO-Algorithmus in Ausgangsparametrierung

Die initialen Pfade zeigen großes Verbesserungspotential, nicht nur im globalen, sondern auch im lokalen Verlauf. Geringe lokale Abweichungen der ACO zu optimalen Lösungen sind bekannt, sodass teilweise lokale Optimierer verwendet werden. Eine grundlegende Eigenschaft der ACO ist jedoch das Ausnutzen globaler Verbesserungspotentiale, die aus der Kombination der zufälligen Pfadvariation und Selektion der besten Lösungen aus den neu generierten und vorherigen entstehen. Dass keine Verbesserung stattfindet, zeigt, dass die gewählte Parametrierung nicht für den Anwendungsfall geeignet ist. Die durchschnittlich sehr langen Pfade mit einer hohen Anzahl an Knoten benötigen außerdem lange Rechenzeiten, sodass keine effiziente Parameter-variation trotz grober Inkrementen erfolgen kann.

Infolgedessen wird die PRPR angepasst, sodass ein gezielteres Ausnutzen der geschätzten zukünftigen Pfadkosten durch die Erhöhung des Grenzwertes q_0 von 50 % auf 80 % erfolgt. Dadurch wird in vier von fünf Fällen der beste nächste Knoten ausgewählt und eine stärkere Orientierung der Ameisen an der Entfernung zum Zielknoten und den Pheromonspuren umgesetzt. Zusätzlich werden die Anzahl der Ameisen auf $n_a = 80$, die Anzahl bester Ameisen auf $m_{\text{best}} = 20$ und die Anzahl Generation auf $n_{\text{Gen}} = 30$ gesenkt, weil infolge der Reduktion der zufälligen Auswahl eine schnellere Konvergenz und Rechenzeiten im Bereich von 3 h anstatt 30 h vorliegen, wie im nächsten Abschnitt gezeigt wird.

5.2.2 Ergebnisse der optimierten Parameter

Für die zweite Parametrierung werden Ergebnisse erzielt. Es werden erneut grobe Inkremente nach Tabelle 5.2 verwendet, um die Ergebnisse für $k_K \in \{0,5; 0,25; 0,1\}$ in Abb. 5.9 zu generieren. Bewertet man die Pfade qualitativ, verlaufen alle drei ähnlich und in den Randbereichen des Suchraumes. Zusätzlich wird für die drei identisch verlaufenden, optimalen Pfade des A*-Algorithmus ein Vergleichspfad eingezeichnet. In den ersten 20 m der x -Koordinate treten keine Unterschiede zwischen den Pfaden auf. Anschließend variieren die Pfade stark.

In der x - y -Ebene ist keine Ameise in der Lage, den Optimalpfad bei $y = -4$ m zu verwenden. Alle gefundenen Pfade befinden sich an der negativen y -Begrenzung, wobei für $k_K = 0,25$ ein sehr kontinuierlicher Verlauf entsteht, der insgesamt sehr nahe an den optimalen Kosten und der Länge des A* liegt (Tabelle 5.6). Für $k_K = 0,5$ kommt es zu einer Oszillation bei $50 \text{ m} \leq x \leq 65 \text{ m}$, obwohl diese Parametrierung Verlängerungen stärker bestraft. Entsprechend fällt der Unterschied zum A*-Pfad deutlicher aus. Das schlechteste Verhalten zeigt die Parametrierung mit $k_K = 0,1$. Hier kommt es zu den stärksten Oszillationen und der Pfad nähert sich auf den letzten Inkrementen frontal bei $y = z = 0$ m an das Fahrzeug an, obwohl die Parametrierung besonders sensibel für die Erfassungswahrscheinlichkeit sein soll.

In der x - z -Ebene zeigen alle drei Parametrierungen sprunghafte Verläufe. Lediglich der Pfad für $k_K = 0,5$ zeigt einen etwas kontinuierlicheren Verlauf.

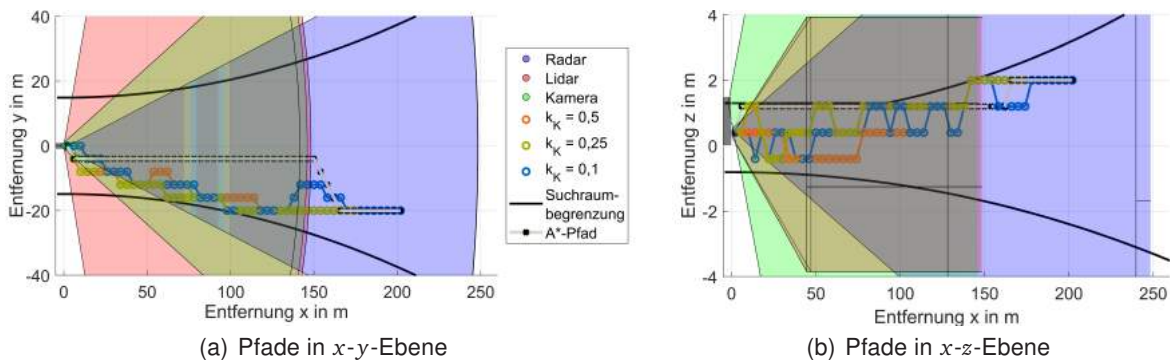


Abbildung 5.9: Ergebnis der Pfadfindung des ACO-Algorithmus mit Vergleich zum Pfad des A*-Algorithmus bei groben Inkrementen

Quantitativ betrachtet bedeuten die qualitativen Beobachtungen eine unterschiedlich stark ausgeprägte Verschlechterung gegenüber den A*-Ergebnissen. Die Auflistung in Tabelle 5.6 zeigt die Pfadkosten und -längen der gefundenen Pfade der drei Parametrierungen und bewertet diese vergleichend für alle drei Kostenfunktionen. Im Falle fett geschriebener Werte stimmt die Konstante zur Berechnung des Pfades mit der Konstante zur Bewertung des Pfades überein. Die Kosten und Längen der Ergebnisse des A*-Algorithmus werden zu Vergleichszwecken ebenfalls dargestellt.

Tabelle 5.6: Kosten K und Länge L_p der ermittelten Pfade des ACO-Algorithmus bei groben Inkrementen

Konstante	$K(k_K = 0, 5)$	$K(k_K = 0, 25)$	$K(k_K = 0, 1)$	Kosten A^*	Differenz ΔK	L_p in m	$L_p A^*$ in m
$k_K = 0, 5$	204,4	202,9	201,6	200,7	3,7 (1,8 %)	211,9	208,4
$k_K = 0, 25$	201,4	199,9	198,6	199,1	0,8 (0,4 %)	208,9	208,4
$k_K = 0, 1$	208,5	207,0	205,8	197,9	7,9 (3,9 %)	216,0	208,4

Die quantitativen Bewertungen bestätigen die qualitativen Analysen der Pfade. Der gewählte Randbereich des Suchraumes liefert zwar höhere Kosten als die zentrale Annäherung des A^* , jedoch nur geringfügig, wie der Pfad für $k_K = 0, 25$ zeigt. Der Unterschied fällt in allen Parameterstufen mit $\Delta K = 0, 7 - 0, 8$ (0,4 %) sehr gering aus. Bei der Wahl $k_K = 0, 5$ fällt der Vergleich mit $\Delta K = 3, 7$ (1,8 %) aufgrund der stärkeren Schwankungen im Pfad schlechter aus. Die größten Differenzen zeigt der Pfad für $k_K = 0, 1$ mit $\Delta K = 7, 9$ (3,9 %).

Zentraler Bewertungsgegenstand des Algorithmus ist neben der Betrachtung der Pfadlänge die Höhe der Erfassungswahrscheinlichkeit entlang des Pfades (Abb. 5.10). Die Verläufe zeigen, dass alle drei Varianten in der Lage sind, geringe Erfassungswahrscheinlichkeiten der Sensorfelder auszunutzen. Im Vergleich zum A^* -Ergebnispfad zeigt sich sogar eine Reduzierung der Erfassungswahrscheinlichkeit, die allerdings nicht in einer Kostenreduktion resultiert (Tabelle 5.6). Wie die Pfadverläufe (Abb. 5.9) zeigen, werden die Randbereiche des Radars verwendet, in denen entsprechend die geringeren Erfassungswahrscheinlichkeiten auftreten. Ab einer Entfernung vom 150 m wird der Bereich mit fusionierter Erfassungswahrscheinlichkeit betreten, sodass erneut $P_D \approx 100$ % gilt und sich die Pfadsuche in die Suche nach dem kürzesten Weg transformiert. Daraus erklärt sich auch der schwankende Verlauf der z -Koordinate in Abb. 5.9(b), weil die Veränderung um ein Inkrement dz in Kombination mit der reduzierten Gewichtung der Strecke durch k_K nur geringen Einfluss auf die Kostenfunktion hat. Entsprechend fallen die Unterschiede der Auswahlwahrscheinlichkeit geringer aus und die Ameisen können mit ähnlicher Wahrscheinlichkeit den geringfügig schlechter bewerteten Pfad wählen. Diese Erklärung ist konsistent mit der Steigerung der Variabilität der z -Koordinate bei Absinken von k_K .

Im vorherigen Abschnitt wird die Parametrierung aufgrund der fehlenden Konvergenz des Algorithmus verworfen. Entsprechend wird an dieser Stelle das Konvergenzverhalten der vorliegenden Varianten betrachtet, wobei alle drei Varianten konvergieren. Die mittlere Gewichtung der Pfadlänge konvergiert nach zehn Generationen, die hohe bereits nach neun. Das zeigt eine starke Konvergenzneigung bei verminderter Exploration. Für die mittlere Gewichtung wird der nahe am Optimum liegende Ergebnispfad bereits nach fünf Generationen gefunden. Eine weitere Verbesserung in der lokalen Umgebung des Pfades kann nur in der x - z -Ebene erfolgen (Abb. 5.9(b)). Die Erfassungswahrscheinlichkeit liegt beinahe bei 100 %, sodass ein weiteres Absenken der Kosten durch Optimierung der Strecke erfolgen muss. Die dafür nötigen Verbesserungen erschließen nur geringe Potentiale, sodass die Bewertung der Pfade ebenfalls nur minimal besser ausfällt. Aufgrund des akkumulierten Pheromons auf dem besten Pfad ist das Auswählen eines abweichenden Pfades unwahrscheinlich. Für $k_K = 0, 5$ konvergiert der Algorithmus frühzeitig auf eine Lösung mit ausreichend vorhandenem Verbesserungspotential, das jedoch nicht erschlossen wird.

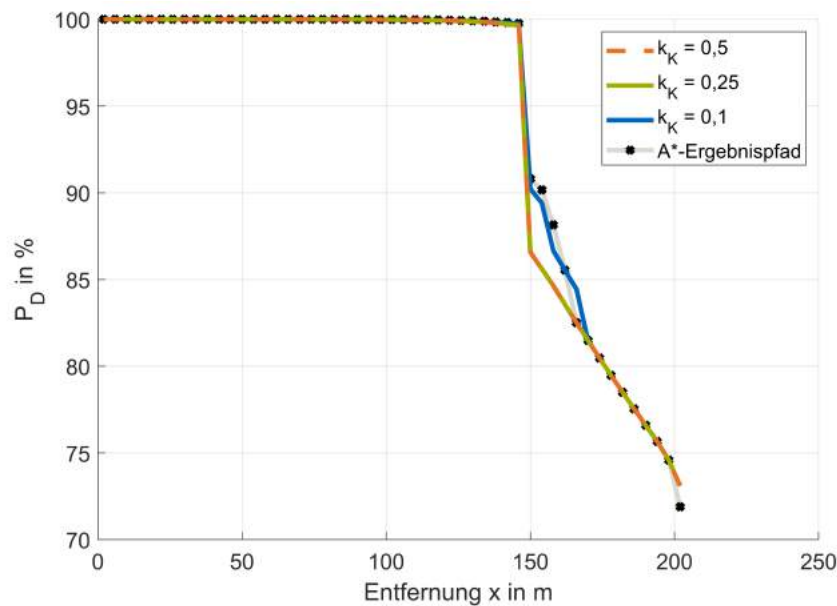


Abbildung 5.10: Erfassungswahrscheinlichkeiten über der x-Koordinate der Knoten der Ergebnispfade des ACO-Algorithmus mit Vergleich zum Ergebnispfad des A*-Algorithmus

Die geringe Gewichtung der Pfadlänge für $k_K = 0,1$ resultiert im Auffinden des Ergebnispfades nach 24 Generationen. Anschließend konvergieren alle anderen 19 beste Pfade auf diesen Pfad. Allerdings hat der Algorithmus sowohl lokal als auch global Verbesserungspotentiale. Bereits zu Beginn zeigt der Algorithmus Stagnation bei einer schlechten Lösung auf und erst nach 14 Generationen kann eine bessere Lösung gefunden werden. Einerseits ist das ein Indiz für hohe Exploration. Andererseits birgt das die Gefahr, dass aufgrund der hohen akkumulierten Pheromonmengen der Knoten dieser Pfade keine weitere Verbesserung stattfindet. Infolgedessen kann die Berechnung weiterer Generationen zwar eine bessere Lösung produzieren, dies ist allerdings aufgrund der hohen Pheromonmengen entlang des Ergebnispfades und der weitestgehend evaporiierten Pheromone der umgebenden Knoten unwahrscheinlich.

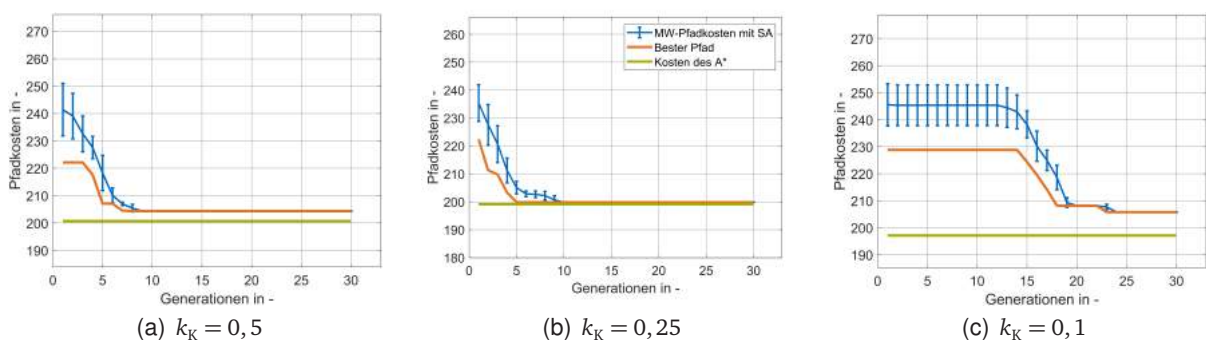


Abbildung 5.11: Verlauf der Pfadkosten der drei Varianten und Vergleich des A*-Ergebnispfades mit Mittelwert (MW) und Standardabweichung (SA)

5.2.3 Diskussion des ACO-Algorithmus

Die präsentierten Ergebnisse zeigen, dass auch der ACO-Algorithmus geeignet ist, um herausfordernde Annäherungspfade unter Berücksichtigung der Sensorabdeckung zu generieren. Die Einbeziehung der Erfassungswahrscheinlichkeit P_D in die klassischerweise distanzbasierte Kostenfunktion ermöglicht durch die gewichtete Kombination der beiden Einflussgrößen eine einstellbare Fokussierung auf P_D . Das zeigen die Verläufe unterschiedlicher k_K (Abb. 5.9). Aufgrund der zufallsbasierten Suche kommt es zu Schwankungen und Oszillationen im Verlauf der Pfade, sodass die Ergebnispfade nicht das globale Optimum der jeweiligen Parametrierung erreichen. Durch die Integration weiterer, in der Literatur erforschter Ansätze kann die Qualität der Ergebnisse gesteigert werden, beispielsweise durch lokale Pfadoptimierungen oder intelligente adaptive Pheromonverwaltung.

Die initial gewählten Parameter des Algorithmus sind nicht geeignet, um eine Lösung zu generieren, weil der Algorithmus keine Konvergenz erreichen kann (Abschnitt 5.2.1). Die Erhöhung der Auswahlgrenze q_0 von 0,5 auf 0,8 für die Selektion des besten Nachbarknotens durch die PRPR ermöglicht dem Algorithmus in Abschnitt 5.2.2 zu konvergieren und gute Lösungen zu produzieren. In den vorliegenden Ergebnissen hat die Wahl von $k_K = 0,25$ die besten Ergebnisse erzielt, die fast kostenoptimal sind. Die Wahl von $k_K = 0,5$ liefert ähnliche Ergebnisse bei quasi identischer Charakteristik des Pfades. In der x - z -Ebene treten weniger Veränderungen auf. Für $k_K = 0,1$ werden die schlechtesten Ergebnisse produziert. Die Verläufe sind in y und z stark schwankend und zusätzlich zeigt der Algorithmus schlechteres Konvergenzverhalten aufgrund sinkender Unterschiede der Kostenfunktion und damit der Auswahlwahrscheinlichkeiten.

Die Unterschiede der Ergebnisse zwischen Abschnitt 5.2.1 und Abschnitt 5.2.2 zeigen die Sensibilität des ACO-Algorithmus bezüglich der Parametrierung. Zusätzlich gilt es, mit den Parametern α , β , ρ , Q , q_0 , m_{best} , n_{Gen} , n_a und k_K eine hohe Anzahl an Parametern zu kontrollieren. Es wird die verwendete Parametrierung mit $\alpha = 1$, $\beta = 5$, $\rho = 0,5$, $Q = 300$, $q_0 = 0,8$, $m_{\text{best}} = 20$, $n_{\text{Gen}} = 30$, $n_a = 80$ und $k_K = 0,25$ zur Verwendung bei vergleichbaren Problemstellungen vorgeschlagen. Ein wichtiger Bestandteil der Bewertung ist die Laufzeit des Algorithmus. Diese fällt bereits bei groben Inkrementen mit durchschnittlich 2 h 45 min 41 s hoch aus ($k_K = 0,5$: 2 h 33 min 12 s, $k_K = 0,25$: 2 h 16 min 12 s, $k_K = 0,1$: 3 h 27 min 39 s).

Die hohe Laufzeit des Algorithmus ergibt sich aus der benötigten Anzahl an Generationen und Ameisen. In gewisser Weise müssen nicht eine, sondern $n_{\text{Gen}} * n_a$ ($30 * 80 = 2400$) Pfadfindungen durchgeführt werden, weil für jede Ameise in jeder Generation ein Pfad berechnet wird. Durch die nötige repräsentative Abdeckung des Suchraumes entsteht eine hohe Anzahl Ameisen und aus der Sicherstellung der Konvergenz eine hohe Anzahl an Generationen. Insgesamt kann die geringe Rechenzeit der Einzelpfaderstellung die hohe Anzahl nicht ausgleichen und die Rechenzeit steigt im Vergleich zum A*-Algorithmus stark an (Abb. 5.12).

Durch die dreidimensionale Anwendung des Algorithmus steigt die Anzahl an Knoten stark an. Das verstärkt die Problematik der repräsentativen Abdeckung. In der lokalen Auswahl des nächsten Knotens hat der Algorithmus außerdem 26 statt acht, bzw. bei Einschränkung der alleinigen Ausbreitung in der z -Koordinate 24 statt acht zur Verfügung. Das führt dazu, dass die erste Parametrierungsvariante mit $q_0 = 0,5$ nicht konvergieren kann und sehr lange Pfade produziert. Das steigert die Rechenzeit ebenfalls, daher wird eine Verfeinerung der Inkremente nicht durchgeführt. Bereits der Vergleich der Ergebnisse der groben Inkremente mit denen des A*-Algorithmus zeigen dessen bessere Eignung für die Methodik. Zusätzlich werden bereits bei der Berechnung mit mittleren Inkrementen extreme Laufzeiterhöhungen festgestellt. Infolgedessen wird auf einen Vergleich bei mittleren oder feinen Inkrementen verzichtet. Im anschließenden Abschnitt wird der Vergleich der Algorithmen ausführlich dargestellt.

5.3 Vergleich der Algorithmen

Ziel dieser Arbeit ist das Auffinden einer alternativen Testmethodik. Deswegen werden nicht die Ergebnisse für die verwendete Sensormodellierung detaillierter verglichen, sondern die Eignung der Algorithmen zur Integration in ein Testkonzept. Dennoch lässt sich anhand der präsentierten Ergebnisse zeigen, dass der A*-Algorithmus bessere Ergebnisse liefert. Ein wichtiger prinzipbedingter Vorteil des A* ist die Erstellung optimaler Ergebnisse bezüglich der gewählten Parametrierung. Diese stellen gleichzeitig die im besten Fall durch den ACO-Algorithmus generierbaren Ergebnisse dar. Der ACO-Algorithmus ist durchaus in der Lage, ähnliche Ergebnisse zu erzielen, wie beispielsweise der Vergleich für $k_K = 0,25$ bei groben Inkrementen zeigt. Allerdings kann er das bei gleicher Inkrementwahl nur auf Kosten einer vielfach höheren Laufzeit (Abb. 5.12).

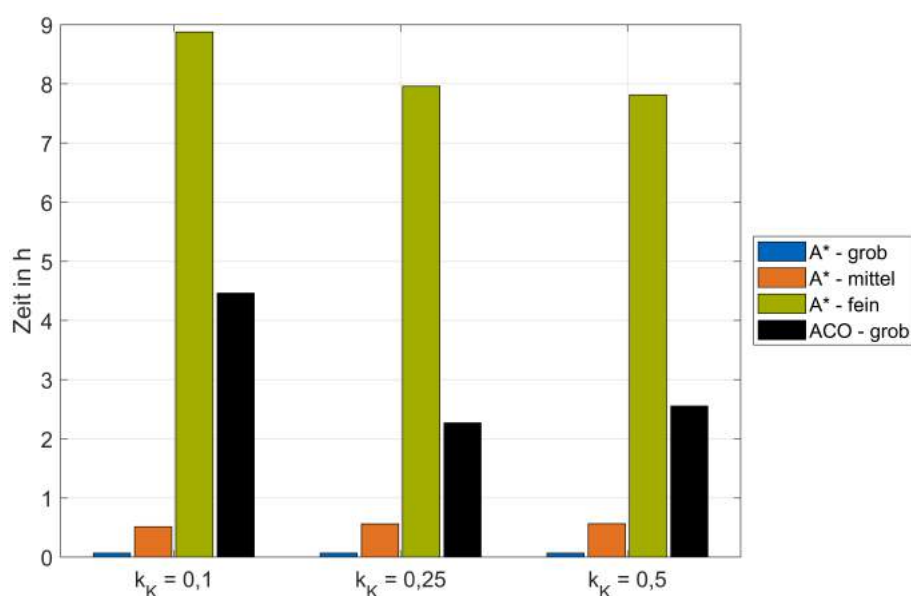


Abbildung 5.12: Vergleich der Laufzeit der Algorithmen in Abhängigkeit der Konstante k_K und der Wahl der Inkremente

Zum Vergleich der beiden Algorithmen im Kontext der Anwendung empfiehlt es sich, die Anforderungen an ein Testkonzept von WACHENFELD und WINNER [9, S.448] (Abschnitt 2.2.1) zu berücksichtigen. Es gilt, die Kategorien der Effizienz- und Effektivitätskriterien zu erfüllen. Im Bereich der Effizienzkriterien sind die Anforderungen *ökonomisch* und *reproduzierbar* bedeutend. Bei vergleichbaren Ergebnissen zweier Algorithmen wird die Entscheidung stets zugunsten des effizienteren ausfallen. Selbst bei Effektivitätsvorteilen setzt sich meist eine effizientere Lösung durch. Weil jedoch der A*-Algorithmus bei nur einem Bruchteil der Laufzeit des ACO-Algorithmus stets das bestmögliche Ergebnis liefert, kann eine Verwendung des ACO-Algorithmus nicht empfohlen werden. Die Laufzeit der ACO liegt selbst bei groben Inkrementen weit jenseits der tolerierbaren Grenze (Abb. 5.12). Wie in Abschnitt 5.2.3 dargestellt, liegen dafür mehrere Gründe vor. Die Suche durch Zufall in einem sehr großen Suchraum benötigt eine hohe Anzahl an Pfaden (bzw. Ameisen), um diesen repräsentativ abzudecken. Zusätzlich wird eine entsprechend hohe Anzahl an Generationen benötigt, um Konvergenz zu erreichen. Im Vergleich dazu steht das methodische Analysieren des A*, bei dem jeder Knoten genau einmal in den Pfad aufgenommen wird, mit dem Resultat einer vielfach geringeren Laufzeit.

Ein weiterer Aspekt ist, dass das Verfahren die methodische Generierung von Testfällen ermöglichen und somit reproduzierbar sein soll. Die Erstellung zufallbasierter Lösungen der ACO erfüllt diese Anforderung im Zweifelsfall nicht. Durch eine entsprechende Wahl der Parameter kann die Wahrscheinlichkeit der Konvergenz auf identische oder zumindest ähnliche Lösungen gesteigert, aber nicht gewährleistet werden. Im Kontrast hierzu garantiert der A*-Algorithmus für gleiche Parameter stets reproduzierbare Ergebnisse. Zusätzlich ist er durch die deutlich geringe Zahl zu wählender Parameter besser ausrichtbar. Ein Vorteil der ACO besteht in der Möglichkeit, dass aus einer Parametrierung mehrere unterschiedliche Szenarien resultieren können, beispielsweise unter Berücksichtigung mehrerer Ergebnispfade oder durch geringe Unterschiede bei Mehrfachausführung.

Im Bereich der Effektivitätskriterien ist die Beobachtbarkeit bzw. die Bewertbarkeit der Ergebnisse das ausschlaggebende Kriterium. Im besten Fall generiert der ACO-Algorithmus ein global optimales Ergebnis. Allerdings kann dieses ohne Referenz nicht bewertet, sondern lediglich die Aussage getroffen werden, dass der gefundene Pfad kritischer ist als alle sonstigen gefundenen Pfade. Das erschwert zusätzlich den Vergleich unterschiedlicher Parametrierungen. Es ist nicht möglich zu entscheiden, ob die Lösung „gut“ oder „schlecht“ in Bezug auf die Parameter ist, ohne Kenntnis über bestmögliche Pfade zu haben. Damit kann eine Parametrierung scheinbar bessere Ergebnisse liefern, obwohl eine alternative Parametrierung in der Lage wäre, eine noch bessere Lösung zu finden. Für den A* ist das Gegenteil der Fall. Durch die Wahl einer zulässigen Heuristik ist gewährleistet, dass die gefundene Lösung optimal bezüglich der gewählten Parameter ist. Entsprechend kann der Vergleich dieser Lösungen stattfinden, weil einem Parametersatz genau eine global optimale Lösung zugeordnet ist.

Zusammengefasst ist das strukturierte und effizientere Vorgehen des A*-Algorithmus mit gleichzeitig garantierter Optimalität und Korrektheit der Lösungen besser für die Integration in eine methodische Testfallgenerierung geeignet. Entsprechend wird dieser Algorithmus zur Überführung der Sensormodellierung von MÜLLER [27] in einen relativen Pfad verwendet.

6 Ergebnisse und Diskussion der Szenarioüberführung

Dieses Kapitel befasst sich mit den Ergebnissen der Szenarioüberführung, die in Abschnitt 6.1 präsentiert und anschließend in Abschnitt 6.2 diskutiert werden.

6.1 Ergebnisse der Szenarioüberführung

Aufbauend auf den durch die Pfadfindungsalgorithmen (A^* , ACO) generierten, relativen Annäherungspfad wird der in Abschnitt 4.5 entwickelte Algorithmus verwendet, um ein Basisszenario zu definieren. In diesem Abschnitt werden die Ergebnisse des A^* -Algorithmus in Ausgangsparametrierung verwendet ($k_K = 0,25$, feine Inkremente, $v_S = (200 \text{ m}, -20 \text{ m}, 2 \text{ m})^T$, $v_Z = (0 \text{ m}, 0 \text{ m}, 0,5 \text{ m})^T$).

Bei einer konstanten Differenzgeschwindigkeit von $\Delta v = 50 \frac{\text{km}}{\text{h}}$ dauert das resultierende Szenario 14,06 s. In dieser Zeit legt das Ego-Fzg eine Strecke von 507,7 m zurück und der Challenger 312,4 m. In den entfernungsbasierten Abbildungen dieses Abschnittes werden zur besseren Orientierung alle 50 Zeitschritte bzw. 0,5 s Markierungen eingezeichnet.

Abb. 6.1 zeigt den Vergleich zwischen Ausgangspfad bzw. aufbereitetem relativen Pfad der Pfadfindung und tatsächlich entstandenen relativen Ergebnispfad zwischen den beiden Fahrzeugen. Für die z -Koordinate werden sowohl relative als auch globale Abweichungen von $\Delta z = 0,5 \text{ m}$ als akzeptabel angenommen. In der x - z -Ebene kommt es zu einer Abweichung von 0,2 m zwischen der geforderten und der tatsächlichen Position (Abb. 6.1(b)), damit liegen die Ergebnisse im zulässigen Bereich. Die x - y -Ebene zeigt größere Abweichungen, die bis zu 10 m betragen (Abb. 6.1(a)).

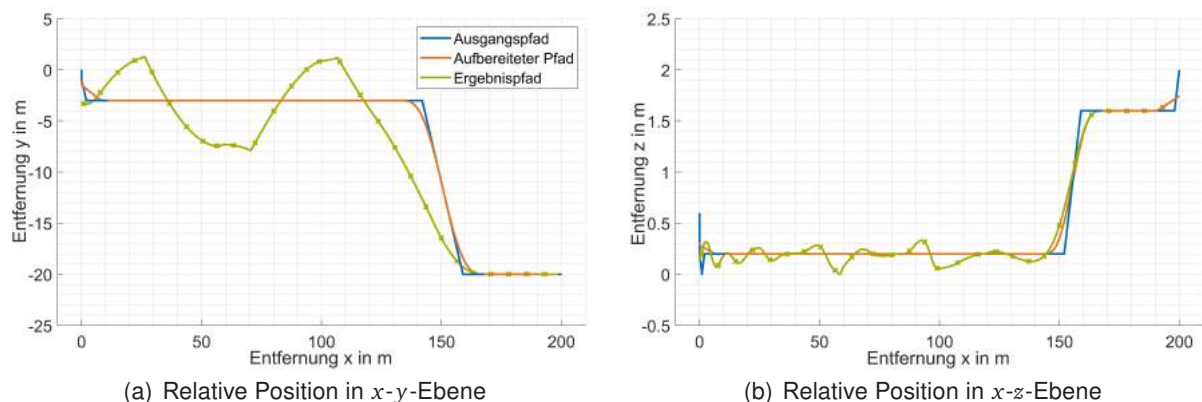


Abbildung 6.1: Relative Position des Challengers im Basisszenario in Bezug auf das Ego-Fzg

Für die Betrachtung der globalen Position der beiden Fahrzeuge wird Abb. 6.2 verwendet. In Abb. 6.2(a) sind zusätzlich die lateralen Fahrbahnbegrenzungen in Bezug auf die Position des Ego-Fzg eingezeichnet. Diese entsprechen in beiden Richtungen dem maximal zulässigen Versatz $\Delta y_{\max}$ aus Tabelle 4.2. Für die Begrenzungen muss die doppelte Fahrbahnbreite verwendet werden, weil sich innerhalb einer Fahrbahn das Ego-Fzg ganz links und der Challenger ganz rechts und umgekehrt befinden kann.

Ziel des Algorithmus ist es, ein Szenario mit Straßenverlauf und zugehörigen validen Fahrzeugbewegungen und -positionen zu generieren. Dafür müssen die globalen Fahrzeugpositionen zu jedem Zeitpunkt innerhalb der lateralen Begrenzungen und vertikalen Schwankungen der Fahrbahn liegen. Die maximalen Abweichungen von 0,4 m der x - z -Ebene sind ausreichend genau. In der x - y -Ebene treten stärkere Schwankungen auf, die im Bereich $360 \text{ m} \leq x \leq 420 \text{ m}$ unzulässiger Weise außerhalb der Fahrbahnbegrenzung liegen.

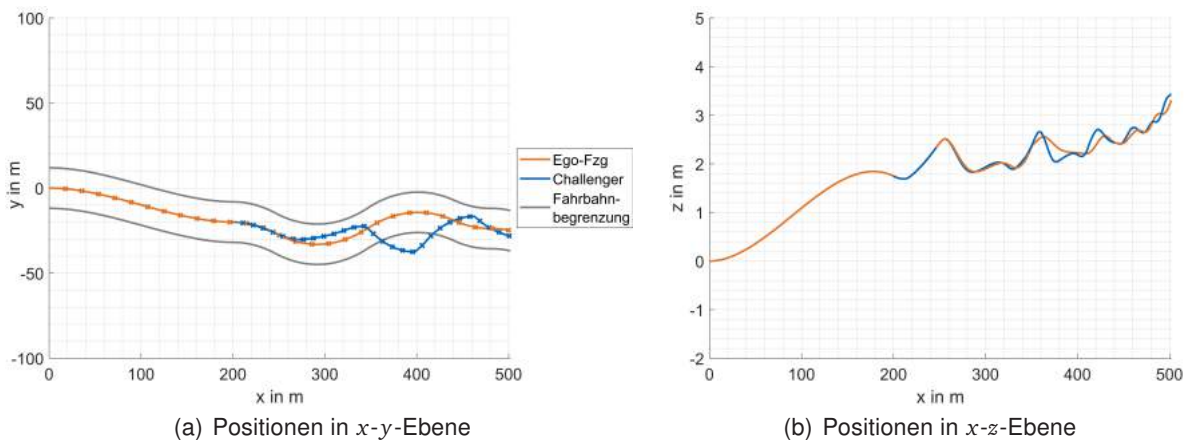


Abbildung 6.2: Globale Positionen der Fahrzeuge unter Berücksichtigung der Fahrbahnbegrenzung

Das Ergebnis des Algorithmus ist im vorliegenden Fall nicht valide, dennoch eignet sich das Verfahren grundsätzlich, um Szenarien aus Pfaden abzuleiten. Zur Analyse des Problems werden die Fahrzeugtrajektorien näher betrachtet. Dafür sind in Abb. 6.3 die Gierraten der beiden Fahrzeuge aufgetragen. Beide Fahrzeuge zeigen eine starke Neigung zu scharfen Kurvenfahrten. Der Challenger verzeichnet innerhalb der ersten 2 s und das Ego-Fzg innerhalb der ersten 5 s stark wechselnde Gierraten, die ein reales Fahrzeug nicht in so kurzen Abständen erreichen kann. Deshalb wird ein gleitender Mittelwert über 20 Datenpunkte bzw. 0,2 s betrachtet, um eine realistische Bewertung durchführen zu können. Dieser zeigt für das Ego-Fzg eine kontinuierliche Zunahme der Gierrate im Bereich $|\dot{\psi}| \leq 0,1 \frac{1}{s}$. Das entspricht moderaten Querschleunigungen von $|a_y| \leq 3,6 \frac{m}{s^2}$.

Die Gierrate des Challengers weist auch bei Betrachtung des gleitenden Mittelwertes hohe und stark schwankende Werte im Bereich der Grenzwerte auf. Anschließend nehmen beide Fahrzeuge abwechselnd den Maximal- und Minimalwert der Gierrate $\dot{\psi}_{\max} = -\dot{\psi}_{\min} = 0,22 \frac{1}{s}$ an. Der Unterschied zwischen den beiden Fahrzeugen entsteht, weil das Ego-Fzg zu Beginn einer kontinuierlichen, interpolierten Positionsvorgabe folgt, der Challenger jedoch neben dem Verlauf des relativen Annäherungspfades auch die Änderungen der Position und Orientierung des Ego-Fzg kompensieren muss.

Nach ca. 2 s erreicht der Challenger gleichzeitig die erste Veränderung der y - und z -Koordinate des relativen Pfades und den Beginn des Bereichs der alternierenden maximalen Gierraten. Das Ego-Fzg bewegt sich zu diesem Zeitpunkt nach der Vorgabe des interpolierten Pfades. Die wachsenden Abweichungen zwischen dem tatsächlich entstehenden relativen Annäherungspfad

und der Vorgabe resultiert aus der Begrenzung der Gierrate, sodass der Challenger die Differenz nicht kompensieren kann. Damit erklärt sich die Differenz zum relativen Pfad als die mangelnde Fähigkeit des Challengers, die Positions- und Orientierungsänderung des Ego-Fzg innerhalb der zulässigen Dynamik auf den Vorgabepfad anzugleichen.

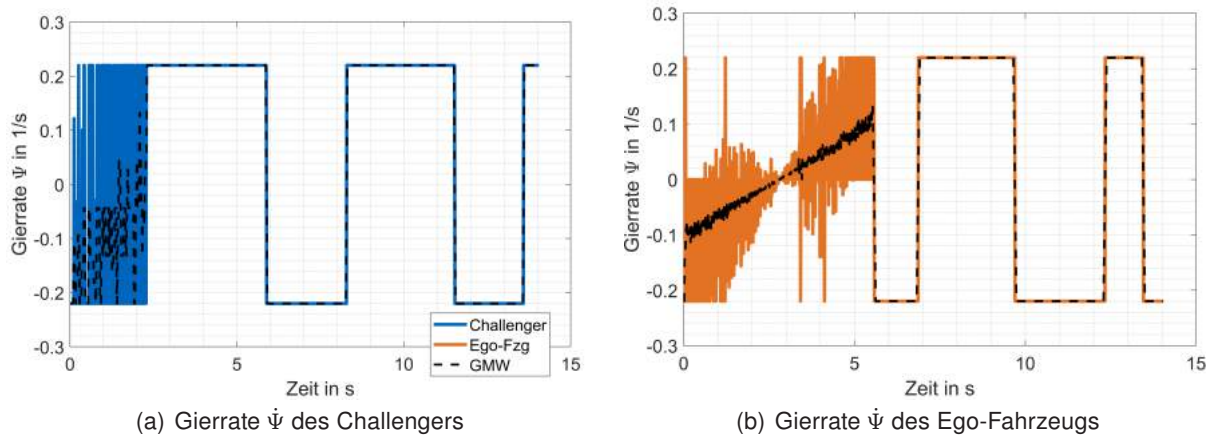


Abbildung 6.3: Gierraten des Ego-Fahrzeugs und des Challengers mit gleitendem Mittelwert (GMW)

Bei ca. 5,6 s erreicht das Ego-Fzg den Übergang von der Positionsvorgabe des relativen Pfades zur Vorgabe durch den zurückgelegten Pfad des Challengers. Gleichzeitig wird das erste Plateau minimaler Gierrate erreicht. Für den restlichen Verlauf des Pfades muss die Dynamik des Ego-Fzg ausreichen, um den Positionsänderungen und Bahnradien des Challengers zu folgen. Das schneller fahrende Ego-Fzg muss nach Gleichung (4.17) eine höhere Gierrate erreichen, um denselben Kurvenradius zu befahren. Allerdings benötigt bereits der Challenger die maximale Gierrate, um die geforderte relative Position einzuhalten. Für das Ego-Fzg ist es nicht möglich, die nötige höhere Gierrate zu erreichen, sodass der Kurvenradius steigt und die Vorgabe nicht eingehalten werden kann. Die Abweichungen des globalen Pfades mit der Konsequenz des Verlassens des zulässigen Bereiches resultieren aus der für das Einhalten der Positionsvorgabe entsprechend der Trajektorie des Challengers mangelnden Querdynamik des Ego-Fzg.

Allerdings ist es je nach Ausprägung des relativen Pfades nicht möglich oder sinnvoll, dass das Ego-Fzg exakt dem Pfad des Challengers folgt. Grundsätzlich ist es ausreichend, wenn sich beide Fahrzeuge innerhalb einer Fahrbahnbreite bewegen.

Die Abweichungen der relativen und globalen Positionen werden verstärkt durch das Fehlen einer Prädiktion zukünftiger Zustände. Der Algorithmus verwendet stets die Koordinaten des in x -Richtung nächstgrößeren Referenzpunktes und richtet die Positionsänderungen der Fahrzeuge an diesen aus. Durch das Fehlen einer Regelung kommt es im relativen Annäherungspfad im Bereich $0 \text{ m} \leq x \leq 120 \text{ m}$ zu Überschwingen und anhaltenden Fehlern im relativen Pfad, die durch eine Regelung ausgeglichen werden können (Abb. 6.1(a)). Dadurch kommt es weiterhin zum Verlassen des relativen und globalen Sollpfades. Integriert man die zukünftige Vorgabe der relativen Position in die aktuelle Vorgabe, kann ein kontinuierlicher Übergang erfolgen. Im vorliegenden Beispiel entspricht das einem kontinuierlichen Annähern an den relativen Pfad anstatt des Überschwingens bei ca. 100 m (Abb. 6.1(a)), sodass der relative Pfad präziser eingehalten wird.

6.2 Diskussion der Szenarioüberführung

Grundsätzlich können drei Problemfelder identifiziert werden:

1. Fehlende Berücksichtigung der Querdynamik während der Pfadfindung
2. Fehlende Prädiktion bei der Vorgabe der Referenzposition
3. Fehlende Toleranz der Fahrbahnbreite in der Positionsvorgabe des Ego-Fzg aus der Challenger-Trajektorie

Das erste Problem resultiert direkt aus den Randbedingungen während der Suche des worst-case Pfades im Sensorfeld. Diese beachten lediglich mögliche Positionen des Challengers, jedoch nicht die zulässige Fahrphysik der beteiligten Fahrzeuge. Hier wäre bereits eine übersichtliche Berücksichtigung ein deutlicher Zugewinn, weil der Challenger die Dynamikanforderungen nicht befriedigen kann.

Die Integration einer Prädiktion der zukünftigen Vorgaben in einen intelligenten Regelungsmechanismus kann ein Überspringen der Relativposition verhindern und somit einen kontinuierlicheren Verlauf mit geringerem Fehler ermöglichen. Wird zusätzlich noch berücksichtigt, dass das Ego-Fahrzeug nicht die exakte Position des Challengers nachstellen muss, kann eine robuste Überführung in ein erstes Basisszenario erfolgen. Die Vorgabe einer Position, die innerhalb der Fahrbahn liegt, ist dafür weitaus besser geeignet.

Darüber hinaus existieren weitere Möglichkeiten, den Algorithmus robuster zu machen. Beispielsweise kann die Geschwindigkeit des Challengers dynamisch angepasst werden. Vorstellbar ist eine Kopplung der Geschwindigkeitsvorgabe an die aktuelle Querbewegung oder Gierrate, die das Ego-Fzg zum Befahren der Trajektorie benötigt. Dadurch kann sichergestellt werden, dass das Ego-Fzg dazu in der Lage ist, seine Position innerhalb der Fahrbahn zu bestimmen. Ein alternativer Ansatz, um die Position der Fahrzeuge zu bestimmen, kann die Rückwärtsimulation des Szenarios bieten. Durch den relativen Annäherungspfad sind die Endpunkte der beiden Fahrzeuge definiert. Die weiteren Positionen können iterativ mit einer ähnlichen Methodik zur hier präsentierten vorgestellt werden. Die initiale Interpolation der Positionsvorgabe des Ego-Fzg entfällt in diesem Ansatz.

Die erzielten Ergebnisse zeigen, dass die Überführung eines relativen Pfades in ein Szenario möglich ist. Der präsentierte Algorithmus liefert die Funktionalitäten, um einen Pfad in Fahrzeugpositionen zu übersetzen. Allerdings ist die aktuelle Ausführung noch nicht für die Anwendung geeignet, weil weitere Randbedingungen und Anpassungen benötigt werden, um das resultierende Basisszenario als realistisches und abprüfbares Szenario verwendbar zu machen.

Damit ist bewiesen, dass es möglich ist, ausgehend von einer spezifischen Modellierung der Sensorabdeckung automatisierter Fahrzeuge ein herausforderndes Szenario abzuleiten.

7 Zusammenfassung und Ausblick

In diesem Kapitel werden die erzielten Ergebnisse der Arbeit zusammengefasst (Abschnitt 7.1) und anschließend wird ein Ausblick auf Entwicklungspotentiale der präsentierten Methodik und allgemeine Handlungsfelder gegeben (Abschnitt 7.2).

7.1 Zusammenfassung

Eingangs wird die von WINNER [11] formulierte Forderung einer neuen Freigabemethodik für HAF aufgegriffen. Diese beinhaltet die Abkehr von der konventionellen Methodik, um nicht in der Freigabefalle ausufernden Testaufwandes zu enden. Konsequenterweise werden alternative Verfahren benötigt, um weiterhin Freigaben durchführen zu können. Diese sind die vom Konzept mathematischer Beweisbarkeit inspirierten formalen Methoden sowie der szenariobasierte Ansatz. Auf absehbare Zeit und aus Gründen der Realisierbarkeit wird nicht mit einem Durchbruch formaler Methoden gerechnet, sodass der Fokus auf dem szenariobasierten Ansatz liegt. Dieser sieht vor, dass die Absicherung eines Fahrzeuges durch die Abdeckung des möglichen Testraumes anhand spezifischer Szenarien erfolgt. Der Testraum besteht jedoch aus einer unendlichen Anzahl möglicher abzuprüfender Szenarien durch Kombination aller möglichen Einflussparameter. Die Lösung dieses Problems liegt in der Auswahl repräsentativer und kritischer Szenarien. Existierende Methodiken sehen für diese Auswahl entweder die Verwendung einer Kritikalitätsmetrik oder kombinatorischer Verfahren vor. Als Alternative wurde in dieser Arbeit eine neue Methodik entwickelt, welche die Sensorik des spezifischen Fahrzeugs berücksichtigt, um kritische relative Annäherungspfade zu bestimmen und diese anschließend in ein Szenario zu übersetzen.

Der erste Baustein der Methodik ist MÜLLERS [27] Modellierung der Sensorabdeckung, die jedem Raumsegment eine Erfassungswahrscheinlichkeit zuordnet in Abhängigkeit von definierten Randbedingungen, z.B. das zu erfassende Objekt und Umweltbedingungen. Auf Basis der Erfassungswahrscheinlichkeit wird ein dreidimensionaler Pfad erstellt, der die relative Position des Ego-Fahrzeuges und des Challengers widerspiegelt. Für die Erstellung dieses Pfades werden zwei Algorithmen verglichen. Der A*-Algorithmus stammt aus der Klasse der informierten Suchalgorithmen und kann durch Verwendung einer Heuristik eine zielgerichtete und optimale Pfadfindung durchführen. Als Alternative wird ein evolutionärer Algorithmus aus der Klasse der Ant Colony Optimization auf die Problemstellung appliziert.

Einen Kernaspekt stellen die berücksichtigten Randbedingungen dar, die beide Algorithmen gleichermaßen erfüllen müssen. Ein Ergebnispfad der Optimierung kann nur verwendet werden, wenn es möglich ist, daraus ein valides Szenario zu generieren. Als Bezug wird die Richtlinie zur Anlage von Autobahnen [50] verwendet, um den Suchraum auf Positionen einzuschränken, für die ein zulässiger verbindender Straßenverlauf existieren kann. Daraus resultiert eine drastische Verringerung der betrachteten Knotenzahl der Algorithmen mit der damit verbundenen Reduk-

tion der Berechnungsdauer in Abhängigkeit der gewählten Inkremente und des betrachteten Feldes. Im Vergleich zur Auswahl der Vorgängerarbeit kann die Knotenzahl für die EKA 1A der RAA bei einer Berücksichtigung von bis zu 350 m in der x -Koordinate von 24 Millionen um 97,2 % auf 676 237 gesenkt werden. Daraus entsteht eine vielfach geringere Laufzeit und die Möglichkeit, weitere Bedingungen in den Algorithmus zu integrieren, ohne extreme Laufzeiten zu produzieren.

Für beide Algorithmen wird eine identische Kostenfunktion definiert, um die resultierenden Pfadkosten zu bewerten. Diese berücksichtigt ein parametrierbares Verhältnis aus Pfadlänge und Erfassungswahrscheinlichkeit P_D an den Knoten des Pfades. Der A*-Algorithmus verwendet diese Kostenfunktion für die Bestimmung der Pfadkosten und deren untere Grenze bei $P_D = 0$ als Heuristik. Diese kann damit als zulässig bezeichnet werden, sodass garantiert ist, dass die Ergebnispfade bezüglich der Parametrierung optimal sind.

Als Alternative und Vergleichsobjekt dient ein knotenbasierter ACO-Algorithmus. Dieser verwendet die Kostenfunktion und Heuristik des A*-Algorithmus weiter als Eingang für die Bestimmung von Auswahlwahrscheinlichkeiten benachbarter Knoten, sodass die Zielführungscharakteristik übernommen wird. Aufgrund der teilweise zufallsbasierten Auswahl des nächsten Knotens während der Pfaderstellung kann der Algorithmus bestenfalls die Ergebnisse des A* reproduzieren. Gleichzeitig weist er deutlich längere Laufzeiten bei ähnlicher Parametrierung und Schrittweite und eine deutlich höhere Anzahl Parameter auf. In der erstellten Arbeit lag die Laufzeit des ACO-Algorithmus mit bis zu viereinhalb Stunden zwischen 30 und 60 mal höher (Abb. 5.12). Selbst bei Wahl feinerer Inkremente ist der A*-Algorithmus noch um ein Vielfaches effizienter. Theoretisch wäre es möglich, durch weitere Anpassungen des Algorithmus weitere Verbesserungen der Ergebnislänge zu erzielen. Jedoch würde das erneut zu Lasten der Laufzeit geschehen. Insgesamt präsentiert sich der A*-Algorithmus überlegen und er wird zur weiteren Verwendung empfohlen.

Der A*-Algorithmus ist sensitiv auf Veränderungen der Erfassungswahrscheinlichkeit. In welchem Maß er Pfadverlängerungen in Kauf nimmt, um Knoten mit geringerer Wahrscheinlichkeit auszunutzen, kann durch die Wahl des Parameters k_K bestimmt werden. In den Ergebnissen der Arbeit (Abschnitt 5.1.3) zeigt sich für $k_K = 0,5$ eine starke Fokussierung auf die Pfadlänge und für $k_K = 0,1$ auf die Erfassungswahrscheinlichkeit. Die Wahl von $k_K = 0,25$ zeigt einen ausgewogener Kompromiss, sodass sie empfohlen wird. Je nach Anwendungsfall kann dementsprechend eine Auswahl der gewünschten Charakteristik des Ergebnispfades erfolgen.

Zusätzlich wurde der Einfluss unterschiedlicher Inkrementstufen (Tabelle 5.2) betrachtet. Ist das Ziel der Anwendung die grundsätzliche Analyse einer Sensorkonfiguration, sind mittlere Inkremente geeignet. Mögliche Anwendungsfälle sind die Analyse globaler Auswirkungen einer spezifischen Sensorposition oder das Auffinden eines geeigneten Startpunkts der Optimierung. Theoretisch ist auch die Wahl grober Inkremente dafür möglich, allerdings steigt damit das Risiko, lokale Besonderheiten nicht zu erfassen, und die Laufzeit ist auch bei mittleren ausreichend gering. Die Verwendung grober Inkremente wird daher für keinen Anwendungsfall empfohlen. Die Berechnungen der Variationen in der erstellten Arbeit zeigen, dass für eine hohe Anzahl unterschiedlicher Parametrierungen, z.B. von k_K , v_S , v_Z , die Laufzeit feiner Inkremente mit bis zu 9 h fünfzehnmal höher liegt als die mittlerer. Die dabei festgestellten Unterschiede der resultierenden Pfade sind jedoch nur gering, sodass zugunsten schnellerer Berechnungen mittlere Inkremente verwendet werden können. Die Erstellung einzelner spezifischer Szenarien, die für die Erprobung und Absicherung entwickelt werden, erfolgt optimalerweise in feinen Inkrementen. Dadurch kann eine möglichst gute Auflösung erreicht werden und bei nur einmaliger Berechnung ist die Laufzeit des Algorithmus im Bereich unter zehn Stunden akzeptabel.

Die Verwendung unterschiedlicher Inkremente in x - und y -Richtung kann die Auflösung feiner Unterschiede in der Erfassung bei gleichzeitig relativ geringer Laufzeit ermöglichen. Gleichzeitig kann die Qualität der Ergebnisse gesteigert werden, weil vor allem laterale Unterschiede der Erfassung Einfluss auf die resultierenden Pfade haben. Außerdem zeigen die Verläufe der Ergebnispfade, dass die Verwendung der vertikalen Komponente keine maßgeblichen Unterschiede liefert. Nahezu sämtliche Pfade wählen initial eine z -Koordinate, die bis zu einem der letzten Knoten beibehalten wird. Entsprechend ist für die meisten Anwendungsfälle eine Betrachtung des vertikalen Verlaufs nicht zwingend erforderlich. Alternativ kann zur Verringerung der Suchraumgröße entweder eine Ebene in einer definierten Schnitthöhe verwendet werden, wie es bereits das Modell von MÜLLER vorsieht, oder eine kombinierte Erfassungswahrscheinlichkeit aus sämtlichen Knoten bei identischem x - und y -Wert erstellt werden. Dafür spricht, dass die Höhe üblicher Detektionsobjekte (Auto, Mensch, Fahrrad) die Schrittweite von 0,2 m bis 0,8 m übersteigt. Eine erhebliche Reduktion der Knotenzahl und Laufzeit ist die Folge der Verringerung der Dimension, wodurch die Wahl der Inkremente deutlich feiner erfolgen kann.

Im Anschluss an die beiden Pfadfindungsalgorithmen wurde ein Algorithmus zur Generierung eines Szenarios aus dem relativen Annäherungspfad entwickelt. Die Ergebnisse zeigen, dass der vorgestellte Algorithmus dazu grundsätzlich in der Lage ist. Unter Beachtung der Vorgabe des relativen Pfades wird die Positionsvorgabe des Challengers an die berechnete Positionsänderung des Ego-Fzg angepasst. Dafür werden für beide Fahrzeuge physikalische Randbedingungen bezüglich Gier- und Nickraten und -winkel beachtet. Die Anwendung auf den Beispielpfad des A*-Algorithmus zeigt jedoch, dass das Ergebnis aktuell kein realistisches Szenario darstellt, weil die Positionsvorgaben nicht innerhalb einer Fahrbahnbreite erfolgen. Als Grund dafür wird die zu starke Vereinfachung des Modells in Kombination mit zu geringen Einschränkungen gesehen. Nach aktuellem Stand wird während der Pfadfindung nur die mögliche Position des Challengers beschränkt, jedoch nicht die nötige Dynamik betrachtet, um den definierten Pfad zu gewährleisten. Der Algorithmus zur Szenarioüberführung verwendet für die Positionsvorgaben der beiden Fahrzeuge stets die Position je eines Knotens. Dabei wird jedoch nicht berücksichtigt, dass jede Position, die innerhalb einer Fahrbahnbreite liegt, valide ist. Stattdessen strebt der Algorithmus zur Vereinfachung die exakte Position der Vorgabe an. Im Falle des schneller fahrenden Ego-Fzg werden die früheren Positionen des Challengers verwendet. Weil von beiden Fahrzeugen zur Erfüllung der Trajektorien quasi dauerhaft maximale Gierraten gefordert werden, kann das schneller fahrende Ego-Fzg dem Verlauf des Challengers nicht folgen, sodass es zu wachsenden Abweichungen kommt. Alternativ kann eine Regelung stattfinden, die den makroskopischen Verlauf des Challengers verwendet, um die Positionen der Fahrzeuge innerhalb einer Fahrbahn vorzugeben. Dennoch wurde ein Algorithmus umgesetzt, der in der Lage ist, einen Annäherungspfad in ein Szenario zu übersetzen. Insgesamt ist damit der Weg von der Modellierung der Sensorabdeckung bis zur Definition eines spezifischen Testfalls abgedeckt.

Zusammenfassend legt diese Arbeit eine Methodik vor, die ausgehend von einer system-spezifischen Sensormodellierung Szenarien erstellt, die für das jeweilige System besonders herausfordernd sind. Damit ist ein neuartiges Verfahren definiert, das nicht auf Realdaten des menschgeführten Verkehrs basiert und die Anforderungen der Absicherung automatisierter Systeme gezielt anspricht.

7.2 Ausblick

Innerhalb der vorliegenden Arbeit werden drei Algorithmen verwendet. Zu Beginn erfolgt die Sensormodellierung nach MÜLLER [27] zur Erstellung der räumlichen Erfassungswahrscheinlichkeiten. Darauf aufbauend wird ein Pfadfindungsalgorithmus zur automatischen Generierung herausfordernder Annäherungspfade verwendet, wobei der A*-Algorithmus empfohlen wird. Abschließend wird aus dem relativen Annäherungspfad durch die Szenarioüberführung ein Szenario gebildet.

In den beiden Kern-Algorithmen der vorliegenden Arbeit, dem A* Algorithmus und der Szenarioüberführung, werden weitere Verbesserungspotentiale identifiziert. Der A*-Algorithmus erstellt innerhalb von Bereichen mit hoher Erfassungswahrscheinlichkeit einen streckenoptimalen Pfad, weil keine erfassungsbedingten Unterschiede vorhanden sind. Die Einbeziehung bisher nicht betrachteter Charakteristika der Sensorabdeckung kann eine weitere Diversifizierung des Suchraumes ermöglichen. Denkbar ist die Berücksichtigung der Anzahl erfassender Sensoren oder die Anzahl unterschiedlicher erfassender Sensorsysteme.

Ein weiterer Ansatzpunkt sind die verwendeten Inkremente des Algorithmus. Im Vergleich der unterschiedlichen Stufen zeigt sich, dass mittlere und grobe Inkremente lokale Besonderheiten nur schlecht bzw. gar nicht auflösen können. Dieses Phänomen ist hauptsächlich in der y -Koordinate aufgetreten. Eine Möglichkeit, dieses Problem zu lösen, ist die Verwendung feinerer Inkremente in der y -Koordinate als in der x -Koordinate.

In der nahen Umgebung des Fahrzeugs ist es zusätzlich sinnvoll, feinere Inkremente zu wählen, als in großer Entfernung. Infolgedessen kann die Pfaderstellung um eine Unterscheidung zwischen Nah- und Fernfeld erweitert werden. Im Fernfeld sind Inkrementunterschiede aufgrund der geringen relativen Änderung der Werte an der Position des nächsten Inkrements eher zweitrangig. Im Nahfeld fällt die Anzahl an Knoten in lateraler Richtung ab. Eine verbesserte Auflösung des Bereichs durch eine feinere Inkrementwahl steigert die Güte der Ergebnisse. Als Umschaltpunkt zwischen Nah- und Fernfeld wird 20 m vorgeschlagen. Das ermöglicht die Wahl feiner Inkremente bei überschaubarer Knotenanzahl.

Diese Aufteilung hätte weiterhin den Vorteil, dass die Wahl von k_K in den beiden Bereichen unterschiedlich ausfallen kann. Im Fernfeld liegt die Erfassungswahrscheinlichkeit teilweise in Bereichen unter 50 %, wohingegen im Nahfeld fast durchgängig 100 % vorliegen. Entsprechend kann die Wahl $k_K = 0,25$ im Fernfeld erfolgen und im Nahfeld deutlich niedriger gewählt werden, um feine Unterschiede im Bereich 99 % bis 100 % zu adressieren.

Die abgeleiteten Pfade offenbaren neben möglichen Testszenarien konzeptbedingte Besonderheiten und Schwächen. Zum Beispiel können nicht erfasste Bereiche oder die Auswirkungen des lateralen Versatzes des Radars auf die Erfassungsgüte im Sensorfeld aufgewiesen werden. Damit kann der A*-Algorithmus auch zur entwicklungsbegleitenden Absicherung der Sensorabdeckung verwendet werden.

Die Ergebnisse der Szenarioüberführung zeigen, dass eine Verstärkung der Randbedingungen der Pfadfindung erforderlich ist. Die Berücksichtigung der nötigen Dynamik, um den geforderten Pfad zu erstellen, kann eine weitere Randbedingung oder einen Parameter der Kostenfunktion darstellen. Beispielsweise kann die Änderung einer Koordinatenrichtung nur nach Ablauf von n Inkrementen in x -Koordinate erfolgen und andernfalls ein Bestrafungsterm auf die Kostenfunktion addiert werden. Das vermindert Bereiche mit kontinuierlichen Wertveränderungen der y - oder z -Koordinate. Die Wahl geringerer Inkremente in y - als in x -Koordinate hat ebenfalls geringere Anforderungen an die Querdynamik zur Folge, ähnlich wie die Begrenzung der auftretenden Steigungen durch die Wahl $dz = 0,2 dx$.

Für die Erstellung valider Szenarien ist die Berücksichtigung der zukünftigen Pfadvorgabe sinnvoll. Ein präzifizierender Regler kann beispielsweise aktuelle Abweichungen vom Vorgabepfad erlauben, um zukünftige Fehler und Überspringen zu vermeiden. Zusätzlich sollte die Positionsvorgabe des Ego-Fzg innerhalb einer Fahrbahnbreite zum vergangenen Pfad des Challengers erfolgen, anstatt diese exakt nachzustellen. Das kann durch eine adaptive Geschwindigkeitsvorgabe unterstützt werden.

Alternativ zur Erstellung der Szenarien in einer Vorwärtssimulation, wie sie in der vorliegenden Arbeit verwendet wird, kann eine Rückwärtssimulation erfolgen. Für die Positionsvorgabe bedeutet das, dass nicht mehr die Trajektorie des Ego-Fzg an den langsamer fahrenden Challenger angepasst wird, sondern umgekehrt. Dadurch kann der Problematik der mangelnden Gierrate mit der Konsequenz des Verlassens zulässiger Bereiche entgegengewirkt werden.

Auf Basis des aktuellen Modells der Sensoren und dem Algorithmus zur Definition relativer Annäherungspfade im dreidimensionalen Raum ist es möglich, Verdeckungen durch Kuppen und andere Fahrzeuge in Abhängigkeit der Lage des Sensors und der Dimensionen des verdeckenden Objektes zu bestimmen. Dadurch kann die Erstellung der Szenarien um einen zusätzlichen Parameter erweitert werden. Entsprechend kann die Auswahl des passenden Modells für einen Anwendungsfall erfolgen. Einerseits kann die dreidimensionale Betrachtung fahrdynamische Effekte und Verdeckungen berücksichtigen, andererseits kann die zweidimensionale Variante deutlich geringere Rechenzeiten erzielen bzw. feinere Inkremente verwenden.

Durch die Integration der Erstellung herausfordernder Szenarien auf Basis eines Ausgangsszenarios nach ROGHARDT [33] kann die vorliegende Methodik weitergeführt werden und neben der räumlichen Komponente der Erfassungswahrscheinlichkeiten durch die Anpassung der Geschwindigkeiten eine zeitliche hinzugefügt werden. Das Ergebnis sind dann einerseits kritische Pfade in Bezug auf die Erfassungswahrscheinlichkeiten, andererseits herausfordernde in Bezug auf Kontrollierbarkeit und Reaktionszeit.

Die vorliegende Arbeit erweitert das aktuelle Portfolio an Möglichkeiten zur Identifikation und Generierung von Szenarien um einen weiteren relevanten Aspekt, die sensorische Erfassungswahrscheinlichkeit definierter Objekte. Jedoch wird wohl keine dieser Möglichkeiten die alleinige, optimale Quelle für Testszenarien sein. Letztendlich birgt analog zur Koexistenz einer Vielzahl unterschiedlichster Optimierungsalgorithmen ein verteilter Ansatz zur Definition von Szenarien das größte Potential, den Testraum repräsentativ abzudecken und somit die Freigabefälle erfolgreich zu überwinden.

Abbildungsverzeichnis

Abbildung 1.1:	Absicherungsansatz Stochastische Felderprobung nach MAZZEGA et al. [8, S.172].....	2
Abbildung 1.2:	Inhalt und Aufbau der Arbeit	4
Abbildung 2.1:	Darstellung einer Szenariorepräsentation nach ULBRICH et al. [13, S.115]	6
Abbildung 2.2:	Abstraktionsebenen von Szenarien nach BAGSCHIK [14, S.132]	7
Abbildung 2.3:	Sechs-Ebenen-Modell nach SAUERBIER [26, S.43].....	12
Abbildung 2.4:	Vorgehen und Aufbau des Modells von MÜLLER [27]	13
Abbildung 2.5:	Optimierte Annäherungspfade der quasiglobalen Optimierung nach MÜLLER [27, S.70].....	14
Abbildung 2.6:	Ablaufdiagramm der Methodik	17
Abbildung 3.1:	Erstellung eines Beispielgraphen.....	23
Abbildung 3.2:	Ergebnis der Tiefensuche im Beispielgraphen	25
Abbildung 3.3:	Ergebnis der Breitensuche im Beispielgraphen	26
Abbildung 3.4:	Ergebnisse des Dijkstra-Suchalgorithmus im Beispielgraphen Schritt 1-4 .	28
Abbildung 3.5:	Ergebnisse des Dijkstra-Suchalgorithmus im Beispielgraphen Schritt 5-9 .	29
Abbildung 3.6:	Ergebnisse des A*-Suchalgorithmus im Beispielgraphen	32
Abbildung 3.7:	Verhalten von Ameisen auf der Futtersuche nach MIRJALILI [34, S.34], Farbsättigung der Pfade nimmt mit steigender Pheromonzkonzentration zu	35
Abbildung 4.1:	Struktur des Minimalbeispiels	42
Abbildung 4.2:	Ergebnisse des A*-Algorithmus angewendet auf das Minimalbeispiel	44
Abbildung 4.3:	Ergebnispfad, Pfadkostenverlauf (mit Mittelwert (MW), Standardabweichung (SA) und Angabe des ersten Auftretens des Pfades mit minimalen Kosten) und finale Pheromonverteilung in Generation 20 des ACO-Algorithmus, angewendet auf das Minimalbeispiel	47
Abbildung 4.4:	Erweiterung des Sensormodells um vertikale Komponente	49
Abbildung 4.5:	RQ der EKA1 nach RAA [50, S.22]	51
Abbildung 4.6:	Zulässiger Bereich der relativen Position des Challengers	52
Abbildung 4.7:	Glättung des relativen Annäherungspfades	54
Abbildung 4.8:	Positionen der Fahrzeuge in den Koordinatensystemen und Herleitung der Positionsänderung	56

Abbildung 4.9:	Generierung des Interpolationspfades für die Richtungsvorgabe des Ego-Fzg	57
Abbildung 5.1:	Ergebnis der Pfadfindung des A*-Algorithmus in Ausgangsparametrierung dargestellt im Suchraum	60
Abbildung 5.2:	Ergebnis der Pfadfindung des A*-Algorithmus in Ausgangsparametrierung dargestellt mit Sensorabdeckung	61
Abbildung 5.3:	Erfassungswahrscheinlichkeit P_D des Ergebnispfades der Ausgangsparametrierung, aufgetragen über der Projektion des Pfades auf die x -Achse	62
Abbildung 5.4:	Ergebnispfade der Variation bei Startknoten im Sensorfeld	63
Abbildung 5.5:	Ergebnispfade des A*-Algorithmus mit groben Inkrementen und Startpunkt außerhalb des Sensorfeldes	64
Abbildung 5.6:	Erfassungswahrscheinlichkeit P_D des Ergebnispfades mit Startknoten außerhalb Sensorfeld, aufgetragen über der Projektion des Pfades auf die x -Achse	66
Abbildung 5.7:	Ergebnispfade des A*-Algorithmus mit mittleren Inkrementen und Startpunkt außerhalb des Sensorfeldes	66
Abbildung 5.8:	Ergebnis der Pfadfindung des ACO-Algorithmus in Ausgangsparametrierung	69
Abbildung 5.9:	Ergebnis der Pfadfindung des ACO-Algorithmus mit Vergleich zum Pfad des A*-Algorithmus bei groben Inkrementen	70
Abbildung 5.10:	Erfassungswahrscheinlichkeiten über der x -Koordinate der Knoten der Ergebnispfade des ACO-Algorithmus mit Vergleich zum Ergebnispfad des A*-Algorithmus	72
Abbildung 5.11:	Verlauf der Pfadkosten der drei Varianten und Vergleich mit dem A*-Ergebnispfad	72
Abbildung 5.12:	Vergleich der Laufzeit der Algorithmen in Abhängigkeit der Konstante k_K und der Wahl der Inkremente	74
Abbildung 6.1:	Relative Position des Challengers im Basisszenario in Bezug auf das Ego-Fzg	77
Abbildung 6.2:	Globale Positionen der Fahrzeuge unter Berücksichtigung der Fahrbahnbegrenzung	78
Abbildung 6.3:	Gierraten des Ego-Fahrzeugs und des Challengers mit gleitendem Mittelwert (GMW)	79

Tabellenverzeichnis

Tabelle 3.1:	Zuteilung der Städte zu Buchstaben	23
Tabelle 3.2:	Luftlinienentfernung zwischen Ebenfeld und den Städten des Graphen	31
Tabelle 4.1:	Gebiete mit Erfassungswahrscheinlichkeit $P_D \neq 0$	42
Tabelle 4.2:	Entwurfsklassen für Autobahnen nach [50, S.17, 22f, 27, 29, 99]	50
Tabelle 5.1:	Anzahl Knoten des Ergebnispfades nach Erfassungswahrscheinlichkeit P_D in % und x -Koordinate in m	60
Tabelle 5.2:	Auflistung der verwendeten Inkremente	62
Tabelle 5.3:	Kosten K und Länge L_p der ermittelten Pfade des A*-Algorithmus bei Start- punkt innerhalb des Sensorfeldes nach Inkrementen und Parametrierung von k_K	63
Tabelle 5.4:	Kosten K und Länge L_p der ermittelten Pfade des A*-Algorithmus bei Startpunkt außerhalb des Sensorfeldes und groben Inkrementen	65
Tabelle 5.5:	Kosten K und Länge L_p der ermittelten Pfade des A*-Algorithmus bei Startpunkt außerhalb des Sensorfeldes und mittleren Inkrementen	66
Tabelle 5.6:	Kosten K und Länge L_p der ermittelten Pfade des ACO-Algorithmus bei groben Inkrementen	71

Literaturverzeichnis

- [1] E. Rotter. „*Automobilindustrie in Deutschland - Rahmenbedingungen und Herausforderungen*,“ München, 6.07.2015. Verfügbar: <https://www.vda.de/de/presse/Presse-meldungen/20150706-Automobilindustrie-in-Deutschland-Rahmenbedingungen-und-Herausforderungen.html> [aufgerufen 09. 01. 2019].
- [2] AUDI AG, Hrsg. „*Audi A8 - Audi AI Staupilot: Weltpremiere für das hochautomatisierte Fahren: der Audi AI Staupilot*,“ 2017. Verfügbar: <https://www.audi-technology-portal.de/de/elektrik-elektronik/fahrerassistenzsysteme/audi-a8-audi-ai-staupilot> [aufgerufen 09.01.2019].
- [3] T. Gasser et al. „*Rechtsfolgen zunehmender Fahrzeugautomatisierung*,“ Gladbach, 2012. Verfügbar: https://www.bast.de/BAST_2017/DE/Publikationen/Foko/Downloads/2012-11.html;jsessionid=417A5AD63F99B2D88236B8766942E2C0.live11292?nn=1836812 [aufgerufen 09. 01. 2019].
- [4] M. Heerwagen, „Rechtlich ausgebremsst Hochautomatisiertes Fahren auf Stand-by,“ *ATZ - Automobiltechnische Zeitschrift*, Bd. 120, Rn. 11, S. 10–15, 2018, doi: 10.1007/s35148-018-0150-0.
- [5] Wirtschaftskommission der Vereinten Nationen für Europa. „*Regelung Nr. 79 der Wirtschaftskommission der Vereinten Nationen für Europa (UN/ECE) - Einheitliche Bedingungen für die Genehmigung der Fahrzeuge hinsichtlich der Lenkanlage: UNECE Nr. 79*,“ 9.10.2007. Verfügbar: <https://eur-lex.europa.eu/legal-content/DE/TXT/?uri=CELEX%3A42008X0527%2801%29> [aufgerufen 21. 01. 2019].
- [6] K. Reif, *Automobilelektronik*, Wiesbaden, Springer Fachmedien Wiesbaden, 2014, isbn: 978-3-658-05047-4. doi: 10.1007/978-3-658-05048-1.
- [7] D. Angermeier et al. „*V-Modell XT: Das deutsche Referenzmodell für Systementwicklungsprojekte*,“ hrsg. von Verein zur Weiterentwicklung des V-Modell XT e.V. 2006. Verfügbar: https://www.cio.bund.de/Web/DE/Architekturen-und-Standards/V-Modell-XT/vmodell_xt_node.html [aufgerufen 08. 01. 2019].
- [8] J. Mazzega und H.-P. Schöner, „Wie PEGASUS die Lücke im Bereich Testen und Freigabe von automatisierten Fahrzeugen schließt,“ in *Methodenentwicklung für aktive Sicherheit und automatisiertes Fahren* (Haus der Technik Fachbuch), W. Klaffke, Hrsg. Renningen: expert Verlag, 2017, S. 163–176, isbn: 978 3 8169 3365 6. Verfügbar: <https://elib.dlr.de/108124/> [aufgerufen 09. 01. 2019].
- [9] W. Wachenfeld und H. Winner, „Die Freigabe des autonomen Fahrens,“ in *Autonomes Fahren*. Bd. 116, M. Maurer et al., Hrsg. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, S. 439–464, isbn: 978-3-662-45853-2. doi: 10.1007/978-3-662-45854-9_21.
- [10] „*Mercedes-Benz präsentiert in Genf Limousine und Coupé der neuen E-Klasse*,“ Stuttgart, 3.03.2009. Verfügbar: <https://www.presseportal.de/pm/68912/1362873> [aufgerufen 08. 01. 2019].

- [11] H. Winner und W. Wachenfeld. „*Absicherung automatischen Fahrens*,“ München, 29. Nov. 2013. Verfügbar: https://www.fzd.tu-darmstadt.de/media/fachgebiet_fzd/publikationen_3/2013/2013_Winner_FAS_Muenchen_Absicherung_automatischen_Fahrens_3_0_Praesentation.pdf [aufgerufen 12. 01. 2019].
- [12] A. Hohm. „*Ko-HAF - Safety by Cooperation*,“ Frankfurt, 19. Nov. 2018. Verfügbar: <https://www.ko-haf.de/index.php?id=34> [aufgerufen 21. 01. 2019].
- [13] S. Ulbrich et al., „Definition der Begriffe Szene, Situation und Szenario für das automatisierte Fahren,“ in *10. Workshop Fahrerassistenzsysteme* Darmstadt: Uni-DAS e.V, 2015, S. 105–117, isbn: 978-3-00-050746-5.
- [14] G. Bagschik et al., „Szenarien für Entwicklung, Absicherung und Test von automatisierten Fahrzeugen,“ in *11. Workshop Fahrerassistenzsysteme und automatisiertes Fahren* Darmstadt: Uni-DAS e.V, 2017, S. 125–135, isbn: 978-3-00-055656-2.
- [15] W. A. Halang und R. M. Konakovsky, „Einige formale Methoden zur Programmverifikation,“ in *Sicherheitsgerichtete Echtzeitsysteme*. Bd. 12, W. A. Halang und R. M. Konakovsky, Hrsg. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018, S. 293–329, isbn: 978-3-662-56368-7. doi: 10.1007/978-3-662-56369-4_10.
- [16] S. Mitsch et al., „On Provably Safe Obstacle Avoidance for Autonomous Robotic Ground Vehicles,“ in *Robotics: Science and Systems IX*, June 24–28, 2013, isbn: 9789810739379. doi: 10.15607/RSS.2013.IX.014.
- [17] K. Macek et al., „Safe Vehicle Navigation in Dynamic Urban Scenarios,“ in *2008 11th International IEEE Conference on Intelligent Transportation Systems*, 12.10.2008 - 15.10.2008, S. 482–489, isbn: 978-1-4244-2111-4. doi: 10.1109/ITSC.2008.4732685.
- [18] S. Shalev-Shwartz et al. „*On a Formal Model of Safe and Scalable Self-driving Cars*,“ 21. Aug. 2017. Verfügbar: <http://arxiv.org/pdf/1708.06374v6>.
- [19] J. Mazzega et al., „Absicherung hochautomatisierter Fahrfunktionen,“ *ATZ - Automobiltechnische Zeitschrift*, Bd. 118, Rn. 10, S. 48–53, 2016, doi: 10.1007/s35148-016-0101-6.
- [20] P. Junietz et al., „Metrik zur Bewertung der Krtikalität von Verkehrssituationen und -szenarien,“ in *11. Workshop Fahrerassistenzsysteme und automatisiertes Fahren* Darmstadt: Uni-DAS e.V, 2017, S. 149–160, isbn: 978-3-00-055656-2.
- [21] A. Pütz et al., „Absicherung hochautomatisierter Fahrfunktionen mithilfe einer Datenbank relevanter Szenarien,“ in *11. Workshop Fahrerassistenzsysteme und automatisiertes Fahren* Darmstadt: Uni-DAS e.V, 2017, S. 161–168, isbn: 978-3-00-055656-2.
- [22] J. Zhou und L. d. Re, „Reduced Complexity Safety Testing for ADAS & ADF,“ *IFAC-PapersOnLine*, Bd. 50, Rn. 1, S. 5985–5990, 2017, doi: 10.1016/j.ifacol.2017.08.1261.
- [23] F. Schuldt, „Ein Beitrag für den methodischen Test von automatisierten Fahrfunktionen mit Hilfe von virtuellen Umgebungen,“ Dissertation, Fakultät für Elektrotechnik, Informationstechnik, Physik, Technischen Universität Carolo-Wilhelmina zu Braunschweig, Braunschweig, 30.10.2016. Verfügbar: https://publikationsserver.tu-braunschweig.de/receive/dbbs_mods_00064747 [aufgerufen 24. 01. 2019].
- [24] F. Schuldt et al., „Effiziente systematische Testgenerierung für Fahrerassistenzsysteme in virtuellen Umgebungen,“ 2013. Verfügbar: https://publikationsserver.tu-braunschweig.de/receive/dbbs_mods_00052570 [aufgerufen 12. 01. 2019].
- [25] G. Bagschik et al., „Ontology based Scene Creation for the Development of Automated Vehicles,“ in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 26.06.2018 - 30.06.2018, S. 1813–1820, isbn: 978-1-5386-4452-2. doi: 10.1109/IVS.2018.8500632.

-
- [26] J. Sauerbier et al., „Definition von Szenarien zur Absicherung automatisierter Fahrfunktionen,“ *ATZ - Automobiltechnische Zeitschrift*, Bd. 121, Rn. 1, S. 42–45, 2019, doi: 10.1007/s35148-018-0205-2.
- [27] F. Müller, „Modellierung der Sensorabdeckung autonomer Fahrzeuge zur Berechnung optimaler Annäherungspfade,“ Masterarbeit, Lehrstuhl für Fahrzeugtechnik (FTM), Technische Universität München, München, 15.11.2018.
- [28] AUDI AG. „*Sensorfelder der Umfeldüberwachung*,“ Verfügbar: <https://www.audi-media-center.com/de/fotos/detail/sensorfelder-der-umfelddueberwachung-52926> [aufgerufen 25.01.2019].
- [29] P. Junietz et al., „Criticality Metric for the Safety Validation of Automated Driving using Model Predictive Trajectory Optimization,“ in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 04.11.2018 - 07.11.2018, S. 60–65, isbn: 978-1-7281-0321-1. doi: 10.1109/ITSC.2018.8569326.
- [30] M. Schreier, „Bayesian Environment Representation, Prediction and Criticality Assessment for Driver Assistance Systems,“ Dissertation, Institut für Automatisierungstechnik und Mechatronik, Technische Universität Darmstadt, Darmstadt, 21.09.2015. Verfügbar: <http://tuprints.ulb.tu-darmstadt.de/5353/> [aufgerufen 22.01.2019].
- [31] J. Hayward, „Near-Miss Determination through use of a Scale of Danger,“ in *Traffic records* (Highway research record) Washington, DC: 1972, S. 24, isbn: 0-309-02054-9. Verfügbar: 34 [aufgerufen 22.01.2019].
- [32] M. M. Minderhoud und P. H. Bovy, „Extended time-to-collision measures for road traffic safety assessment,“ *Accident Analysis & Prevention*, Bd. 33, Rn. 1, S. 89–97, 2001, doi: 10.1016/S0001-4575(00)00019-1.
- [33] C. Roghardt, „Entwicklung und Programmierung herausfordernder Szenarien zur Absicherung der Trajektorienplanung autonomer Fahrzeuge,“ Masterarbeit, Lehrstuhl für Fahrzeugtechnik (FTM), Technische Universität München, München, 23.10.2018.
- [34] S. Mirjalili, *Evolutionary Algorithms and Neural Networks*. Bd. 780, Cham, Springer International Publishing, 2019, isbn: 978-3-319-93024-4. doi: 10.1007/978-3-319-93025-1.
- [35] C. Grimme und J. Bossek, *Einführung in die Optimierung: Konzepte, Methoden und Anwendungen*, Wiesbaden, Springer Vieweg, 2018, isbn: 978-3-658-21150-9. doi: 10.1007/978-3-658-21151-6. Verfügbar: <http://dx.doi.org/10.1007/978-3-658-21151-6>.
- [36] M. Pieper, *Mathematische Optimierung*, Wiesbaden, Springer Fachmedien Wiesbaden, 2017, isbn: 978-3-658-16974-9. doi: 10.1007/978-3-658-16975-6.
- [37] M. Papageorgiou et al., *Optimierung*, Berlin, Heidelberg, Springer Berlin Heidelberg, 2015, isbn: 978-3-662-46935-4. doi: 10.1007/978-3-662-46936-1.
- [38] S. Dörn, *Programmieren für Ingenieure und Naturwissenschaftler*, Berlin, Heidelberg, Springer Berlin Heidelberg, 2017, isbn: 978-3-662-54175-3. doi: 10.1007/978-3-662-54176-0.
- [39] BatchGeo LLC. „*BatchGeo: Create maps from your data*,“ Verfügbar: <https://www.batchgeo.com/> [aufgerufen 23.04.2019].
- [40] Stephan Georg. „*Luftlinie.org*,“ Verfügbar: <https://www.luftlinie.org/> [aufgerufen 23.04.2019].
- [41] E. W. Dijkstra, „A note on two problems in connexion with graphs,“ *Numerische Mathematik*, Bd. 1, Rn. 1, S. 269–271, 1959, doi: 10.1007/BF01386390.

- [42] W. Ertel, *Grundkurs Künstliche Intelligenz*, Wiesbaden, Springer Fachmedien Wiesbaden, 2016, isbn: 978-3-658-13548-5. doi: 10.1007/978-3-658-13549-2.
- [43] K.-L. Du und M. N. S. Swamy, *Search and Optimization by Metaheuristics*, Cham, Springer International Publishing, 2016, isbn: 978-3-319-41191-0. doi: 10.1007/978-3-319-41192-7.
- [44] M. Dorigo et al., „Ant system: optimization by a colony of cooperating agents,“ *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, Bd. 26, Rn. 1, S. 29–41, 1996, doi: 10.1109/3477.484436.
- [45] M. Dorigo und T. Stützle, „Ant Colony Optimization: Overview and Recent Advances,“ in *Handbook of Metaheuristics* (International Series in Operations Research & Management Science). Bd. 272, M. Gendreau und J.-Y. Potvin, Hrsg. Cham: Springer International Publishing, 2019, S. 311–351, isbn: 978-3-319-91085-7. doi: 10.1007/978-3-319-91086-4_10.
- [46] B. Bullnheimer et al. „A new rank based version of the Ant System. A computational study.“ Wien, 1997. Verfügbar: <http://epub.wu.ac.at/616/> [aufgerufen 27. 04. 2019].
- [47] M. Dorigo und L. M. Gambardella, „Ant colony system: a cooperative learning approach to the traveling salesman problem,“ *IEEE Transactions on Evolutionary Computation*, Bd. 1, Rn. 1, S. 53–66, 1997, doi: 10.1109/4235.585892.
- [48] M. Faisal et al., „AntStar: Enhancing Optimization Problems by Integrating an Ant System and A* Algorithm,“ *Scientific Programming*, Bd. 2016, S. 1–12, 2016, doi: 10.1155/2016/5136327.
- [49] Y. Zhongrui et al., „Improved Ant Colony Optimization Algorithm for Intelligent Vehicle Path Planning,“ in *2017 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII)*, 02.12.2017 - 03.12.2017, S. 1–4, isbn: 978-1-5386-2434-0. doi: 10.1109/ICIICII.2017.55.
- [50] M. Rohloff, *Richtlinien für die Anlage von Autobahnen: RAA*, (FGSV R1 - Regelwerke). Bd. 202, Ausg. 2008, Köln, FGSV-Verl., 2008, isbn: 978-3-939715-51-1.
- [51] M. Harrer und P. Pfeffer, „Grundlegendes zum Lenkvorgang,“ in *Lenkungsbandbuch*, P. Pfeffer und M. Harrer, Hrsg. Wiesbaden: Springer Fachmedien Wiesbaden, 2013, S. 21–42, isbn: 978-3-658-00976-2. doi: 10.1007/978-3-658-00977-9_2.
- [52] J. Feng et al., „Study on Intelligent Vehicle Lane Change Path Planning and Control Simulation,“ in *2006 IEEE International Conference on Information Acquisition*, 20.08.2006 - 23.08.2006, S. 683–688, isbn: 1-4244-0528-9. doi: 10.1109/ICIA.2006.305809.
- [53] L. Hoffmeister und M. Becke, „Extreme Spurwechsel in Notsituationen,“ in *Verkehrs-RechtsReport*. Bd. 05 2007, S. 179–181. Verfügbar: <http://ureko.de/forschung-und-publikationen/veroeffentlichungen/> [aufgerufen 20. 05. 2019].
- [54] K.-H. Schimmelpfennig und U. Nackenhorst, „Bedeutung der Querbesehleunigung in der Verkehrsunfallrekonstruktion/Sicherheitsgrenze des Normalfahrers,“ in *Verkehrsunfall und Fahrzeugtechnik*. Bd. 23 1985, S. 94–96. Verfügbar: <http://www.ureko.de/forschung-und-publikationen/veroeffentlichungen/> [aufgerufen 20. 05. 2019].
- [55] W. Sun und S. Wang, „Research on Lateral Acceleration of Lane Changing,“ in *Frontier Computing* (Lecture Notes in Electrical Engineering). Bd. 542, J. C. Hung et al., Hrsg. Singapore: Springer Singapore, 2019, S. 950–960, isbn: 978-981-13-3647-8. doi: 10.1007/978-981-13-3648-5_120.

Anhang

A	Anhang: Datenträger.....	xi
----------	---------------------------------	-----------

A Anhang: Datenträger

Anbei befindet sich der Datenträger mit allen Unterlagen in digitaler Form. Im Ordner „03_Matlab“ befinden sich neben den Skripten und Funktionen dieser Arbeit auch die erzielten Ergebnisse. Der Ordner „04_ Abbildungen“ enthält alle verwendeten Grafiken. Im Falle selbst erstellter Abbildungen ist die Ausgangsdatei ebenfalls vorhanden.

Inhalt des Datenträgers

Ausarbeitung.pdf	Diese Arbeit (als PDF-Dokument)
01_ Latex	Die Dateien zur Erstellung der Arbeit
02_ Literatur	Verwendete Literatur (als PDF-Dokumente)
03_ Matlab	Verwendete Matlab-Skripte (als .m- oder .mat-Datei)
04_ Abbildungen	Verwendete und erstellte Abbildungen (als .png- und .fig-Datei)

- Datenträger -