

# Trajektorienplanung autonomer Fahrzeuge: Diskrete Lösungsmethode für urbane Szenarien

Vom Fachbereich  
Elektrotechnik und Informationstechnik  
der Technischen Universität Darmstadt  
zur Erlangung des akademischen Grades  
eines Doktor-Ingenieurs (Dr.-Ing.)  
genehmigte Dissertation

von

**F. Christoph Ziegler, M. Sc.**

Referent: Prof. Dr.-Ing. J. Adamy  
Korreferent: Prof. Dr.-Ing. S. Peters  
Korreferent: Prof. Dr.-Ing. V. Willert  
Tag der Einreichung: 16. September 2022  
Tag der mündlichen Prüfung: 26. Januar 2023

Darmstadt 2023

Ziegler, Frank Christoph

Trajektorienplanung autonomer Fahrzeuge:

Diskrete Lösungsmethode für urbane Szenarien

Darmstadt, Technische Universität Darmstadt,

Fachgebiet Regelungsmethoden und intelligente Systeme

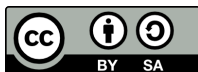
Jahr der Veröffentlichung der Dissertation auf TUprints: 2023

URN: urn:nbn:de:tuda-tuprints-231463

URI: <https://tuprints.ulb.tu-darmstadt.de/id/eprint/23146>

Veröffentlicht unter CC BY-SA 4.0 International

<https://creativecommons.org/licenses/>



Für meine Eltern.

# Erklärungen laut Promotionsordnung

## **§8 Abs. 1 lit. c PromO**

Ich versichere hiermit, dass die elektronische Version meiner Dissertation mit der schriftlichen Version übereinstimmt.

## **§8 Abs. 1 lit. d PromO**

Ich versichere hiermit, dass zu einem vorherigen Zeitpunkt noch keine Promotion versucht wurde. In diesem Fall sind nähere Angaben über Zeitpunkt, Hochschule, Dissertationsthema und Ergebnis dieses Versuchs mitzuteilen.

## **§9 Abs. 1 PromO**

Ich versichere hiermit, dass die vorliegende Dissertation selbstständig und nur unter Verwendung der angegebenen Quellen verfasst wurde.

## **§9 Abs. 2 PromO**

Die Arbeit hat bisher noch nicht zu Prüfungszwecken gedient.

---

Datum und Unterschrift

---

# Inhaltsverzeichnis

<b>Abkürzungen und Symbole</b>	<b>VIII</b>
<b>Kurzfassung</b>	<b>XV</b>
<b>Abstract</b>	<b>XVII</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Beiträge dieser Dissertation . . . . .	4
1.2 Aufbau der Arbeit . . . . .	5
<b>2 Trajektorienplanung autonomer Fahrzeuge</b>	<b>7</b>
2.1 Planungshierarchie autonomer Fahrzeuge . . . . .	7
2.2 Trajektorienplanung . . . . .	11
2.3 Fahrzeugmodelle . . . . .	13
2.4 Zusammenfassung . . . . .	22
<b>3 Idealer Referenzpunkt</b>	<b>23</b>
3.1 Einleitung und Motivation . . . . .	23
3.2 Stand der Technik . . . . .	25
3.3 Bestimmung des Referenzpunktes . . . . .	26
3.3.1 Kurven mit konstanter Krümmung . . . . .	26
3.3.2 Kurven mit variabler Krümmung . . . . .	29
3.3.3 Kreismodell . . . . .	33
3.4 Weiterführende Betrachtungen und Anwendungen . . . . .	35
3.4.1 Kollisionsüberprüfung . . . . .	35
3.4.2 Fahrspurzuordnung . . . . .	36
3.5 Zusammenfassung . . . . .	37
<b>4 Analyse menschlichen Fahrverhaltens</b>	<b>39</b>
4.1 Einleitung und Motivation . . . . .	39
4.2 Stand der Technik . . . . .	41
4.3 Analyse Abtastzeit . . . . .	43
4.4 Verhaltensmodell . . . . .	48

4.4.1	Transformation der Lenkvariablen $\omega_k, \delta_k$ . . . . .	51
4.4.2	Verhaltensmodell menschlicher Fahrer . . . . .	54
4.4.3	Modellanalyse . . . . .	56
4.5	Zusammenfassung . . . . .	58
<b>5</b>	<b>CarPre-Trajektorienplanung</b> . . . . .	<b>60</b>
5.1	Einleitung und Motivation . . . . .	60
5.2	Stand der Technik . . . . .	62
5.3	Planung mittels Monte-Carlo Baumsuche . . . . .	68
5.3.1	Diskretisierung des Suchraumes . . . . .	71
5.3.2	Definition des Spielendes . . . . .	76
5.3.3	Auswahl des Startpunktes . . . . .	77
5.3.4	Standardstrategie . . . . .	77
5.3.5	Initialisierung der Knotenerfolgswerte . . . . .	78
5.3.6	Erfolgswertfunktion . . . . .	80
5.3.7	Bestimmung von Kollisionen . . . . .	82
5.3.8	Parallelisierung . . . . .	85
5.3.9	Extraktion der Trajektorie . . . . .	86
5.4	Evaluierung . . . . .	86
5.4.1	Spurfolgefahrt . . . . .	86
5.4.2	Annäherung an ein statisches Objekt . . . . .	88
5.5	Zusammenfassung . . . . .	91
<b>6</b>	<b>Experimente im Rahmen des Projekts PRORETA 5</b> . . . . .	<b>94</b>
6.1	Einleitung und Motivation . . . . .	94
6.2	Systemaufbau und Softwarearchitektur . . . . .	95
6.3	Ergebnisse für ein prototypisches Fahrzeug . . . . .	98
6.3.1	Analyse der Laufzeit . . . . .	99
6.3.2	Analyse des Fahrkomforts . . . . .	100
6.3.3	Analyse der Abtastzeit . . . . .	105
6.4	Zusammenfassung . . . . .	111
<b>7</b>	<b>Zusammenfassung und Ausblick</b> . . . . .	<b>112</b>
<b>A</b>	<b>Anhang</b> . . . . .	<b>114</b>
A.1	Einfacher Trajektorienplaner . . . . .	114
A.2	Fahrzeugparameter . . . . .	115
A.3	Parameter Verhaltensmodell . . . . .	115
A.4	Parameter CarPre-Trajektorienplaner . . . . .	116
A.5	Experimente . . . . .	117

---

A.6 Referenzierte Abbildungen . . . . .	117
<b>Literaturverzeichnis</b>	<b>118</b>





# Abkürzungen und Symbole

## Abkürzungen

CarPre	Monte-Carlo modellprädiktive Trajektorienplanung (engl.: <i>Monte-Carlo model predictive planner</i> )
DEM	Dynamisches Einspurmodell
EM	Erwartungsmaximierungsalgorithmus (engl.: <i>expectation-maximization algorithm</i> )
GPS	Globales Positionsbestimmungssystem (engl.: <i>global positioning system</i> )
HA	Mitte der Hinterachse
HD	Hohe Auflösung (engl.: <i>high definition</i> )
KDR	Modell konstanter Drehrate
KEM	Kinematisches Einspurmodell
LKW	Lastkraftwagen
MDP	Markow-Entscheidungsprozess (engl.: <i>Markov decision process</i> )
MPC	Modellprädiktive Regelung (engl.: <i>model predictive control</i> )
PDF	Wahrscheinlichkeitsdichtefunktion (engl.: <i>probability density function</i> )
PKW	Personenkraftwagen
PM	Punktmodell
POMDP	Partiell beobachtbares MDP (engl.: <i>partially observable Markov decision process</i> )
PSPACE	Komplexitätsklasse, Lösung mit polynomialen Ressourcen (engl.: <i>polynomial space</i> )
PT1-Glied	Verzögerungsglied erster Ordnung
RRT	engl.: <i>rapidly-exploring random tree</i>
SD	Standardabweichung (engl.: <i>standard deviation</i> )
SP	Schwerpunkt
UCT	Oberes Konfidenzintervall für Bäume (engl.: <i>upper confidence bound applied to trees</i> )
VA	Mitte der Vorderachse
WM	Weitere Modelle

# Symbole

## Notationen

$x$	Skalar
$x'$	Aktualisierte Variable
$\bar{x}$	Transformierte Variable
$x_d$	Wertediskrete Größe
$x_{d,i}$	$i$ -ter wertediskreter Zustand
$x_k$	Größe zu einem zeitdiskreten Zeitpunkt $k$
$x_x$	Größe in Richtung der x-Achse
$x_y$	Größe in Richtung der y-Achse
$\dot{x}$	Zeitliche Ableitung
$dx$	Differential
$\Delta x$	Differenz
$\mathbf{x}$	Spaltenvektor
$\mathbf{x}^\top$	Reihenvektor
$\mathbf{X}$	Matrix
$w_x$	Größe im ortsfesten, kartesischen Weltkoordinatensystem
${}_F x$	Größe im fahrzeugfesten Koordinatensystem
${}_{VR} x$	Größe im Vorderrad-Koordinatensystem
${}_{HR} x$	Größe im Hinterrad-Koordinatensystem
${}_O x$	Größe im objektfesten Koordinatensystem

## Lateinische Großbuchstaben

$B, C, D$	Konstanten des Reifenmodells nach Pacejka [6]
$F$	Kraft
$\mathcal{F}$	Koordinatensystem
$I_z$	Gierträgheitsmoment
$J$	Gütefunktion
$LR(\mathbf{x}, \mathbf{y})$	Likelihood-Quotient zwischen $\mathbf{x}$ und $\mathbf{y}$ (engl.: <i>Likelihood ratio</i> )
$M_N$	Verhaltensmodell auf Basis einer $N(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$
$M_{St}$	Verhaltensmodell auf Basis einer $St(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Lambda}, \nu)$
$N(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$	Multivariate Normal-PDF eines Zufallsvektors $\mathbf{x}$ mit Mittelwertsvektor $\boldsymbol{\mu}$ und Kovarianzmatrix $\boldsymbol{\Sigma}$
$\mathbb{N}$	Menge der natürlicher Zahlen ohne null
$\mathbb{N}_0$	Menge der natürlicher Zahlen mit null
$R$	Erfolg
$R_a$	Erfolgswert bzgl. Beschleunigungsminimierung
$R_{ini}$	Initialer Knotenerfolgswert
$R_{koll}$	Erfolgswert bzgl. Kollisionsvermeidung
$R_{verlust}$	Virtueller Verlustwert
$R_m$	Mittlerer Erfolg des Suchknotens
$R_{m,best}$	Größter Erfolgswert aller Kinderknoten
$R_{m,best,vor}$	Aktueller Erfolgswert für vorherige beste Aktion
$R_{mitte}$	Erfolgswert bzgl. Fahrspurmitte
$R_s$	Summierte Erfolgsabschätzung
$R_{spur}$	Erfolgswert bzgl. Spurhalten
$R_{spur,g}$	Erfolgswert falls Fahrzeug auf Gegenspur
$R_v$	Erfolgswert bzgl. Geschwindigkeitsziel
$R_\psi$	Erfolgswert bzgl. Straßenrichtung
$S$	Schlupf des Reifens
$St(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Lambda}, \nu)$	Multivariate studentsche $t$ -PDF eines Zufallsvektors $\mathbf{x}$ mit Mittelwertsvektor $\boldsymbol{\mu}$ , Skalierungsmatrix $\boldsymbol{\Lambda}$ und Freiheitsgrad $\nu$
$T_{akt}$	Aktualisierungszeit des Planungsalgorithmus
$T_{hor}$	Zeitlicher Planungshorizont
$T_{in}$	Eingangstastzeit
$\mathcal{X}$	Zustandsraum des Fahrzeugs
$\mathcal{X}_{erlaubt}$	Menge aller möglichen Fahrzeugzustände
$\mathbb{Z}$	Menge der ganzen Zahlen

## Lateinische Kleinbuchstaben

$a$	Beschleunigung in Fahrtrichtung
$a_{\text{lat,max}}$	Maximale Querschleunigung
$a_{\text{max}}$	Maximale Beschleunigung in Fahrtrichtung
$a_{\text{min}}$	Minimale Beschleunigung in Fahrtrichtung
$b$	Fahrzeugbreite
$b_{\text{obj}}$	Breite eines Objektes
$b_{\text{obj,min}}$	Minimale Breite eines Objektes
$b_{\text{R}}$	Basis der Größenordnung der Erfolgsfunktion
$d$	Positionsfehler (Abstand $\mathbf{p}_{\text{ref,bm}}$ zu $\mathbf{p}_{\text{ref,inD}}$ )
$d_{\text{a}}$	Benötigter Platzbedarf zur Kurvenaußenseite
$d_{\text{i}}$	Benötigter Platzbedarf zur Kurveninnenseite
$d_{\text{max}}$	Maximaler Abstand
$d_{\text{mitte}}$	Abstand Fahrzeug zur Fahrspurmitte
$d_{\text{opt}}$	Optimaler Platzbedarf, $d_{\text{opt}} = d_{\text{i}} = d_{\text{a}}$
$d_{\text{sicher}}$	Sicherheitsabstand für Kollisionsüberprüfung
$d_{\text{x,off}}$	Abstand Kreismittelpunkt zu Objektaußenkante
$d_{\text{z}}$	Zusatzlänge bei Kreiszerlegung
$e$	Einheitsvektor eines Koordinatensystems
$g$	Fallbeschleunigung der Erde
$h$	Höhe Fahrzeugschwerpunkt über Boden
$h(p(\mathbf{x}))$	Histogramm einer PDF
$k_{\text{acc}}$	Zeitschritte für $a$ -Aktionen in Standardstrategie
$k_{\omega}$	Zeitschritte für $\omega$ -Aktionen in Standardstrategie
$j_{\text{best}}$	Bester Index
$l$	Radstand
$l_{\text{fh}}$	Länge zwischen Fahrzeugfront und Hinterachse
$l_{\text{ges}}$	Länge zwischen Fahrzeugfront und Fahrzeugheck
$l_{\text{obj}}$	Länge eines Objektes
$l_{\text{obj,min}}$	Minimale Länge eines Objektes
$l_{\text{ref}}$	Länge zwischen Hinterachse und Referenzpunkt
$l_{\text{ref,opt},\kappa=0}$	$l_{\text{ref}}$ für idealen Referenzpunkt bei $\kappa = 0$
$l_{\text{ref,opt},\kappa=0,1}$	$l_{\text{ref}}$ für idealen Referenzpunkt bei $\kappa = 0,1 \text{ m}^{-1}$
$l_{\text{ref,opt},\kappa=0,2}$	$l_{\text{ref}}$ für idealen Referenzpunkt bei $\kappa = 0,2 \text{ m}^{-1}$
$l_{\text{ref,VA}}$	$l_{\text{ref}}$ für Referenzpunkt auf der Vorderachse
$m$	Fahrzeugmasse
$n_{\text{a}}$	Anzahl an diskreten Beschleunigungen
$n_{\text{b}}$	Anzahl der Besuche eines Baumknotens
$n_{\text{in,ges}}$	Anzahl aller möglicher Aktionskombinationen

---

$n_{\text{obj}}$	Anzahl an Objektkreisen in Kreiszerlegung
$n_{\text{start}}$	Zeitschritte bis Start der Trajektorienplanung
$n_{\text{threads}}$	Anzahl an parallelen Threads
$n_{\text{tra}_j}$	Anzahl an Punkte innerhalb einer Trajektorie
$n_{\delta}$	Anzahl an diskreten Lenkwinkeln
$n_{\omega}$	Anzahl an diskreten Lenkwinkelgeschwindigkeiten
$p(x)$	PDF einer kontinuierlichen Zufallsvariable
$p(\mathbf{x})$	PDF eines kontinuierlichen Zufallsvektors
$p(\mathbf{x} \mid \mathbf{y})$	Bedingte PDF von $\mathbf{x}$ gegeben $\mathbf{y}$
$p_1, p_2$	Parameter der $\omega$ -Transformation
$p_{\text{min}}$	Mindestverhältnis Kreiszerlegung
$p_{\text{ü}}$	Überlappungsfaktor Kreiszerlegung
$\mathbf{p}$	2D-Position $(x, y)$
$\mathbf{p}_{\text{HA}}$	Punkt in der Mitte der Hinterachse
$\mathbf{p}_{\text{o}}$	Momentaner Drehpunkt
$\mathbf{p}_{\text{ref}}$	2D-Position eines Referenzpunktes
$\mathbf{p}_{\text{ref,bm}}$	2D-Referenzpunktposition des optimierten KEM
$\mathbf{p}_{\text{ref,inD}}$	2D-Referenzpunktposition aus inD-Datensatz
$\mathbf{p}_{\text{SP}}$	Punkt im Schwerpunkt
$\mathbf{p}_{\text{VA}}$	Punkt in der Mitte der Vorderachse
$r_{\text{ego}}$	Radius des Kreismodells
$r_{\text{M}}$	Abstand $\mathbf{p}_{\text{ref}}$ zu $\mathbf{p}_{\text{o}}$
$r_{\text{obj}}$	Radius eines Objektes
$s$	Distanz entlang eines Pfades/Trajektorie
$t$	Zeit
$t_{\text{calc}}$	Verfügbare Rechenzeit
$t_{\text{jetzt}}$	Aktuelle Zeit
$t_{\text{verarb}}$	Berechnungszeit
$u$	Systemeingang
$v$	Geschwindigkeit des Fahrzeugs in Fahrtrichtung
$v_{\text{max}}$	Maximale Geschwindigkeit
$v_{\text{ziel}}$	Zielgeschwindigkeit
$x$	$x$ -Position
$x_{\text{m}}$	$x$ -Position des Mittelpunkts eines Objektkreises
$\mathbf{x}$	Zustandsvektor
$\mathbf{x}_{\text{ende}}$	Endzustand der Planung
$\mathbf{x}_{\text{M}}$	Zustandsvektor des Verhaltensmodells
$\mathbf{x}_{\text{start}}$	Startzustand der Planung
$y$	$y$ -Position; Modellausgang des PT1-Gliedes

## Griechische Buchstaben

$\beta$	Schwimmwinkel
$\delta$	Vorderradlenkwinkel des Fahrzeugs
$\delta_{\max}$	Maximaler Vorderradlenkwinkel
$\epsilon_{\text{R}}$	Erlaubte Erfolgsabweichung
$\theta_{\text{karte}}$	Bewegungsrichtung der Richtungskarte
$\kappa$	Krümmung eines Pfades
$\kappa_{\max}$	Maximal fahrbare Krümmung
$\mu$	Reibungskoeffizient zwischen Reifen und Straße
$\pi$	Kreiszahl
$\Pi$	Trajektorie
$\rho$	Platzhalter für VR, HR
$\sigma$	Platzhalter für $x, y$
$\tau$	Zeitkonstante des PT1-Gliedes
$\psi$	Orientierung des Fahrzeugs (auch: Gierwinkel)
$\psi_{\text{k}}$	Kurswinkel
$\omega$	Vorderradlenkwinkelgeschwindigkeit des Fahrzeugs
$\omega_{\max}$	Maximale Lenkwinkelgeschwindigkeit
$\omega_{\text{best}}$	Beste Lenkwinkelgeschwindigkeit anhand der Richtungskarte
$\omega_{\text{DR}}$	Drehrate des Modells KDR

# Kurzfassung

Autonomes Fahren wird die persönliche Mobilität der Zukunft maßgeblich verändern. Doch bis diese Vision Wirklichkeit wird, ist noch einige Forschungs- und Entwicklungsarbeit nötig. Denn gerade die schwierigsten Verkehrsszenarien, welche insbesondere in städtischen Gebieten auftreten, sind noch oft eine Herausforderung.

Der Fokus dieser Arbeit liegt auf der Trajektorienplanung, also der Planung des zukünftigen Fahrtweges in Abhängigkeit der Zeit. Diese Planung kann auch als das Kernelement des autonomen Fahrzeugs angesehen werden, da sie alle bisher berechneten Informationen zusammenfügt und daraus die zukünftige Trajektorie bestimmt. Das Planungsergebnis ist außerdem direkt von den Passagieren wahrnehmbar und kann zur Beurteilung des Gesamtsystems herangezogen werden. Somit hängt die Akzeptanz sowie das Vertrauen in das autonome Fahrzeug maßgeblich vom berechneten Planungsergebnis ab. Die Berechnung eines solchen Planungsergebnisses ist allerdings komplex und muss in Echtzeit gelöst werden. In diesem Kontext behandelt diese Dissertation die folgenden Fragen: 1. wie diese rechenintensive Planungsaufgabe vereinfacht werden kann und 2. wie im gesamten Planungsraum effizient eine Trajektorie berechnet werden kann.

Hierzu wird zuerst die Wahl des Referenzpunktes, also der Bezugspunkt zwischen einem fahrzeugfesten Koordinatensystem zu einem Weltkoordinatensystem, analysiert. Dieser wird als ideal angenommen, wenn der seitliche Platzbedarf zu beiden Fahrzeugseiten gleich groß ist. Für Kurven konstanter Krümmungen wird analytisch die Idealposition hergeleitet. Nach einer Analyse von Kurven mit variablen Krümmungen empfiehlt der Autor den Referenzpunkt auf oder leicht hinter die Vorderachse zu legen. Die bisher theoretischen Überlegungen werden in einer Kollisionsüberprüfung sowie einer Fahrspurzuordnung in die Praxis übertragen.

Für weitere potenzielle Vereinfachungen wird ein Datensatz mit von Menschen gefahrenen Trajektorien analysiert. Dabei wird zuerst untersucht, mit welcher Abtastzeit die Trajektorien durch ein kinematisches Einspurmodell abgebildet werden können, bevor geschwindigkeitsabhängige Transformationen des Lenkwinkels sowie der Lenkwinkelgeschwindigkeit hergeleitet werden. Basierend auf diesen Transformationen

wird ein umgebungsunabhängiges Verhaltensmodell präsentiert, welches ein Standardverhalten menschlicher Fahrenden darstellt.

In einem nächsten Schritt wird ein neuartiges Planungskonzept, die CarPre-Trajektorienplanung, basierend auf der Monte-Carlo Baumsuche vorgestellt. Durch Verwendung der vorher extrahierten Lenkwinkeltransformation wird der Zustandsraum eines kinematischen Einspurmodells diskretisiert. Die wertediskreten Planungsaktionen, bestehend aus Beschleunigungen sowie Lenkwinkelgeschwindigkeiten, werden in einem Suchbaum zur Berechnung der zukünftigen Trajektorie eingesetzt. Zur Kollisionsüberprüfung sowie Fahrspurzuordnung wird hierbei auf die Referenzpunktüberlegungen zurückgegriffen. Das Konzept wird in einem Spurhaltenmanöver sowie einer Annäherung an ein statisches Objekt simulativ verifiziert.

Abschließend wird der Planungsalgorithmus im Rahmen des Projektes PRORETA 5 auf einem prototypischen Fahrzeug analysiert. Durch erste Tests wird ein angenehmer Fahrkomfort festgestellt, welcher auf die zukünftige Akzeptanz des echtzeitfähigen Algorithmus schließen lässt. In einer weiteren Untersuchung werden die praktischen Auswirkungen unterschiedlicher Abtastzeiten innerhalb des Planungsalgorithmus ausgewertet. Durch die Wahl höherer Abtastzeiten wird das exponentielle Wachstum des Planungsproblems verlangsamt. Dies ermöglicht z. B. eine Vergrößerung des Planungshorizonts, sodass vorausschauender geplant werden kann. Somit leistet diese Dissertation einen Beitrag für die Trajektorienplanung autonomer Fahrzeuge und ermöglicht einen Schritt weiter in die Richtung der Vision des autonomen Fahrens; einen Schritt weiter in die Richtung der Mobilität der Zukunft.



# Abstract

Autonomous driving will significantly change the personal mobility of the future. But before this vision becomes reality, some research and development work is still needed. This is because of the most difficult traffic scenarios, which occur especially in urban areas, that still remain a challenge.

The focus of this work is on trajectory planning, i.e. the planning of the future path of movement as a function of time. This planning can also be seen as the core element of the autonomous vehicle, as it combines all the information calculated so far and uses it to determine the future trajectory. The planning result can also be perceived directly by the passengers and can be used to evaluate the overall system. Thus, the acceptance of and trust in the autonomous vehicle depends to a large extent on the calculated planning result. However, the calculation of such a planning result is complex and must be solved in real time. In this context, this dissertation addresses the following research questions: 1. how to simplify this computationally intensive planning task, and 2. how to efficiently compute a trajectory in the entire planning space.

For this purpose, the choice of the reference point, i.e. the reference point between a vehicle-fixed coordinate system to a world coordinate system, is first analyzed. The reference point is assumed to be ideal if the lateral needed space required on both sides of the vehicle is the same. For curves of constant curvatures, the ideal position is derived analytically. After analyzing curves with variable curvatures, the author recommends placing the reference point on or slightly behind the front axle. These theoretical considerations are transferred into practice in a collision check as well as a lane assignment.

For further potential simplifications, a data set of human trajectories is analyzed. First, it is investigated with which sampling time the trajectories can be represented by a kinematic bicycle model, before speed-dependent transformations of the steering angle as well as the steering rate are derived. Based on these transformations, an environment-independent behavior model is presented, which represents a standard behavior of human drivers.

In a next step, a novel planning concept, CarPre trajectory planning, based on Monte-Carlo tree search, is presented. By using the previously extracted steering angle transformation, the state space of a kinematic bicycle model is discretized. The planning actions, consisting of discrete-value accelerations as well as steering rates, are used in a search tree to compute the future trajectory. For collision checking as well as lane assignment, the reference point considerations are used here. The concept is simulatively verified in a lane keeping maneuver as well as in an approach to a static object.

Finally, the planning algorithm is analyzed on a prototypical vehicle within the framework of the project PRORETA 5. Through initial testing, a pleasant driving experience is found, which suggests future acceptance of the real-time capable algorithm. In a further investigation, the practical effects of different sampling times within the planning algorithm are evaluated. By choosing higher sampling times, the exponential growth of the planning problem can be slowed down. This allows e.g. to increase the planning horizon so that planning can be more predictive. Thus, this dissertation contributes to the trajectory planning of autonomous vehicles and brings us one step closer towards the vision of autonomous driving. One step closer towards the mobility of the future.

---

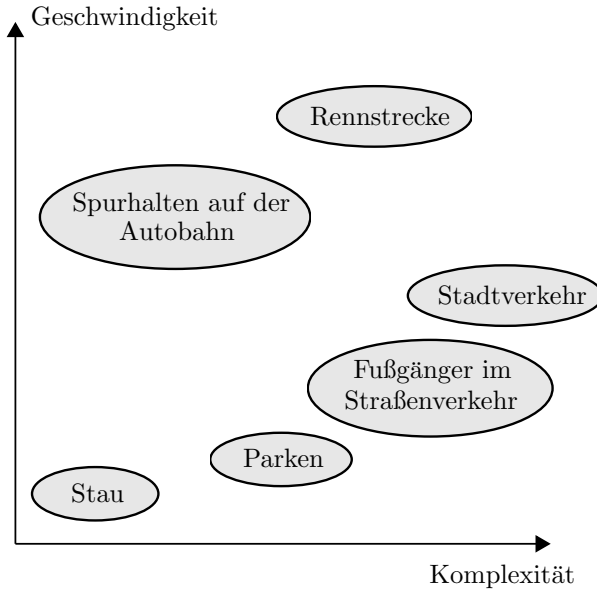
# 1 Einleitung

Autonomes Fahren wird nach der Durchsetzung des privaten PKWs als die nächste Revolution in der Mobilitätsgeschichte angesehen [11, S. VI][85, S. V]. Dieser technische Fortschritt soll eine Vielzahl von Vorteilen mit sich bringen. Beispiele hierfür sind die Reduktion von Verkehrsunfällen, die Bereitstellung von Mobilität für nicht fahrtüchtige Personen sowie das Einsparen von Zeit<sup>1</sup>, benötigtem (Park-)Platz, Energie und Fahrpersonal [52, S. 57ff]. Bis all diese positiven Aspekte zum Tragen kommen, wird allerdings noch einige Zeit vergehen. Derzeit wird für einen vollständigen Umstieg auf das autonome Fahren eine Zeitspanne von bis zu 50 Jahren prognostiziert [70, S. XI]. Diese Entwicklungsverzögerung des autonomen Fahrzeugs ist durch die letzten Prozent an Verkehrssituationen zu begründen, welche die schwierigsten, die insbesondere in städtischen Gebieten auftreten, darstellen [52, S. 6] (vgl. Abb. 1.1). Denn durch die fortschreitende Automatisierung wird zwar ein Großteil der derzeitigen Unfallsituationen vermieden, allerdings ist damit zu rechnen, dass Unfälle, welche ausschließlich auf die Automatisierung zurückzuführen sind, hinzukommen [41, S. 11]. Daraus resultiert für die Freigabe autonomer Fahrzeuge, dass sowohl eine Reduktion der Unfallzahlen, als auch „ein akzeptiertes Verhältnis [...] zwischen vermiedenen [...] und zusätzlich hervorgerufenen Risiken“ zu belegen ist [85, S. 442]. Trotz dieser regulatorischen Risiken wird intensiv an der technischen Umsetzbarkeit des autonomen Fahrens geforscht.

Autonome Fahrzeuge stellen eine Herausforderung an die Software dar [95]. Dabei können die benötigten Kernkompetenzen in drei Kategorien eingeteilt werden: *Perzeption*, *Planung* und *Regelung* [98, S. 2]. Unter *Perzeption* wird die Wahrnehmung der Umgebung durch das autonome System verstanden. Hierbei ist es einerseits wichtig, ein Abbild der Umgebung zu erstellen (*Umfeldwahrnehmung*). Dieses Abbild enthält alle wichtigen Umgebungsinformationen wie z. B. Hindernisse, andere Verkehrsteilnehmer oder Straßenschilder. Andererseits ist die Positionierung relativ zur

---

<sup>1</sup>Durch autonome Fahrzeuge kann die Zeit während der Beförderung anderweitig genutzt werden.



**Abbildung 1.1:** Einordnung verschiedener Verkehrsszenarien nach auftretenden Fahrgeschwindigkeiten und Planungskomplexitäten, übersetzt aus [98, S. 3].

Umgebung (*Lokalisierung*) ebenso wichtig, damit das autonome Fahrzeug sich orientieren kann. In der Planung wird die zukünftige Bewegung des Fahrzeugs berechnet, sodass ein übergeordnetes Ziel<sup>2</sup> erreicht wird. Hierfür wird auf die gesammelten Informationen der Perzeption zurückgegriffen, um wichtige Fahrentscheidungen zu treffen. Abschließend übernimmt die Regelung die Ausführung der geplanten Bewegung, sodass sich das autonome Fahrzeug entlang der geplanten Trajektorie bewegt.

Der Fokus dieser Dissertation liegt auf der Planung, welche durch ihre zentrale Rolle als das Herzstück des autonomen Fahrens angenommen werden kann. Gerade die Planung ist essentiell für die Akzeptanz des autonomen Fahrzeugs, da das Ergebnis direkt mit dem Verhalten menschlicher Fahrer verglichen werden kann. Zudem ist eine geringe Fehlerquote im Planungsergebnis entscheidend, da „Menschen eher bereit sind Fehl-

<sup>2</sup>Ein Beispiel hierfür ist das Erreichen eines gewünschten Zielortes unter minimalem Energieaufwand ohne dabei einen Unfall zu verursachen.

ler anderer Menschen als Fehler von Maschinen zu verzeihen“<sup>3</sup> [52, S. 6]. Darüber hinaus liegt der Fokus dieser Arbeit auf dem städtischen Bereich, in welchem besonders komplexe Verkehrsszenarien zu erwarten sind. Zwar ist hier die Fahrgeschwindigkeit verglichen z. B. mit Autobahnfahrten gering, allerdings liegt eine andere Umgebung mitsamt der unterschiedlichen Verkehrsteilnehmer vor, wodurch die Komplexität deutlich erhöht wird (vgl. Abb. 1.1). Damit die rechenintensive Planung in diesen komplexen Verkehrsszenarien effizient bewältigt werden kann, bedarf es einer intelligenten Lösung des Planungsproblems. Daraus ergibt sich die erste Forschungsfrage dieser Arbeit.

(F1) Wie kann die rechenintensive Aufgabe der Trajektorienplanung *vereinfacht* werden?

Derzeitige Ansätze lösen das Planungsproblem meist anhand von diskreten Verhaltensklassen. Hierbei wird zuerst ein abstraktes Verhalten in einem reduzierten Planungsraum bestimmt. Im Anschluss berechnet der Trajektorienplaner aus diesem abstrakten Verhalten einen konkreten Bewegungsablauf. Dies reduziert die Planungskomplexität, kann aber gerade in städtischen Szenarien mit sehr vielen möglichen Verkehrssituationen problematisch sein (vgl. Abschnitt 5.2). Aus diesen Szenarien resultieren viele mögliche Verhaltensklassen, was eine Herausforderung an die Planungsaufgabe stellt. Denn einerseits muss sichergestellt werden, dass der Trajektorienplaner das abstrakte Verhalten auch umsetzt, und andererseits müssen im Vorfeld alle möglichen Szenarien durch Verhaltensklassen abgebildet werden. Daraus ergibt sich die zweite Forschungsfrage dieser Arbeit.

(F2) Wie kann im gesamten Planungsraum *effizient in Echtzeit* eine Trajektorie geplant werden, *ohne diskrete Verhaltensklassen* zu verwenden?

Auf diese beiden Forschungsfragen wird im Laufe der Arbeit eingegangen und sie werden in den jeweiligen Kapiteln ausführlicher erörtert.

<sup>3</sup>Englischer Originaltext: „When discussing what fault tolerance might be acceptable, it should be borne in mind that people are more likely to forgive mistakes made by other people than mistakes made by machines.“

## 1.1 Beiträge dieser Dissertation

In diesem Abschnitt sind die Beiträge dieser Arbeit zusammengefasst. Sie stehen in direktem Zusammenhang mit den beiden oben genannten Forschungsfragen. Zuerst werden mögliche Vereinfachungen vorgestellt, welche durch theoretische Überlegungen sowie durch Analyse menschlicher Fahrdaten hergeleitet werden. Im Anschluss werden diese Vereinfachungen in einem neuartigen Planungsansatz angewendet. Die Beiträge lassen sich wie folgt zusammenfassen:

- Herleitung einer idealen Referenzpunktposition des Fahrzeugs mit dem Ziel, den seitlichen Platzbedarf zu beiden Seiten entlang der Fahrbahn möglichst gleich zu halten (vgl. Abschnitt 3.1) [100]
  - Herleitung für Kurvenfahrten mit konstanter Krümmung
  - Analyse für Kurvenfahrten mit variabler Krümmung
  - Anwendung dieser theoretischen Überlegung auf eine vereinfachte Kollisionsüberprüfung/Fahrspurzuordnung durch Verwendung eines Punktmodells mit reduziertem Radius (F1)
- Analyse menschlicher Fahrdaten eines urbanen Trajektorien Datensatzes zur Bestimmung des durchschnittlichen menschlichen Fahrverhaltens als mögliche Planungsvereinfachung [141]
  - Analyse der Abtastzeit menschlicher Trajektorien (F1)
  - Einführung einer geschwindigkeitsabhängigen Transformation der Lenkvariablen  $\delta$  und  $\omega$  zur einfachen Berücksichtigung longitudinaler und lateraler Abhängigkeiten
  - Herleitung eines umgebungsunabhängigen Verhaltensmodells zur Abschätzung möglicher Fahraktionen (F1)
- Entwicklung eines neuartigen Ansatzes zur Trajektorienplanung autonomer Fahrzeuge (CarPre-Trajektorienplanung) ohne Verwendung diskreter Verhaltensklassen
  - Diskretisierung des Aktionsraumes anhand der Analyse menschlicher Fahrdaten
  - Anpassung der Monte-Carlo Baumsuche auf das Problem der Trajektorienplanung (F2)

Diese Beiträge werden in dieser Arbeit in einem prototypischen Fahrzeug im Rahmen des Projektes PRORETA5<sup>4</sup> zusammengeführt und zur Anwendung gebracht.

## 1.2 Aufbau der Arbeit

Abbildung 1.2 gibt einen Überblick über den Aufbau dieser Dissertation. Nach der Einleitung in Kapitel 1 werden in Kapitel 2 die Grundlagen der Trajektorienplanung autonomer Fahrzeuge vorgestellt. Zuerst wird eine Einführung in die Planungshierarchie sowie die Trajektorienplanung gegeben, bevor verschiedene Fahrzeugmodelle vorgestellt werden. Darauf basierend wird das kinematische Einspurmodell ausgewählt, welches im weiteren Verlauf dieser Arbeit verwendet wird. In den darauffolgenden Kapiteln werden die neuen Lösungsansätze dieser Arbeit präsentiert. Kapitel 3 diskutiert die Wahl des Referenzpunktes des gewählten Fahrzeugmodells. Dabei ist das Ziel entlang der geplanten Trajektorie einen möglichst zu beiden Fahrzeugseiten einheitlichen Platzbedarf zu benötigen. In Kapitel 4 wird zuerst die Abtastzeit menschlicher Fahrer<sup>5</sup> analysiert, bevor im Anschluss ein zeitdiskretes, umgebungsunabhängiges Verhaltensmodell extrahiert wird.

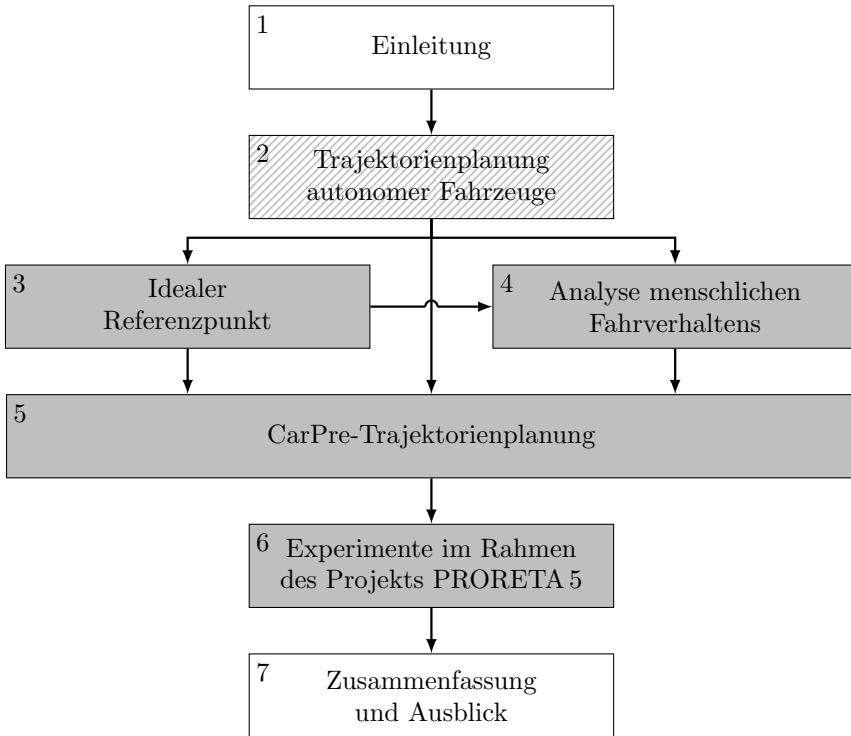
Die Überlegungen des dritten und vierten Kapitels sind bisher einzelne, unabhängige Ideen und Konzepte, welche in einer Vielzahl von Anwendungen und Problemstellungen verwendet werden können. Der Fokus dieser Kapitel liegt dabei hauptsächlich auf Vereinfachungen für die Trajektorienplanung. Diese Erkenntnisse werden in Kapitel 5 im CarPre-Trajektorienplaner verwendet, welcher auf der Monte-Carlo Baumsuche basiert. Daher stellt das Kapitel 5 das Hauptkapitel dieser Dissertation dar. Das neue Planungskonzept wird in Kapitel 6 im Rahmen des Projektes PRORETA 5 anhand eines prototypischen Fahrzeugs analysiert. Abschließend werden in Kapitel 7 die Ergebnisse dieser Arbeit zusammengefasst sowie zukünftige Forschungsthemen vorgestellt.

Wie in Abb. 1.2 dargestellt, ergeben sich dem interessierten Leser mehrere Lesemöglichkeiten. In Kapitel 3 bis 5 befindet sich der zugehörige Stand der Technik im jeweiligen Kapitel selbst, sodass diese unabhängig voneinander gelesen werden können. Der Autor empfiehlt, Kapitel 5 vor Kapitel 6

---

<sup>4</sup><https://www.proreta.tu-darmstadt.de>

<sup>5</sup>Aus Gründen der Lesbarkeit wird in dieser Arbeit ausschließlich die männliche Form gewählt. Dies entspricht keiner Diskriminierung des weiblichen oder diversen Geschlechts.



**Abbildung 1.2:** Aufbau dieser Dissertation mit durch Pfeile gekennzeichnete Lesemöglichkeiten. Die Kapitel mit größeren Beiträgen des Autors sind grau hinterlegt, während ein Kapitel mit kleinem Beitrag schraffiert dargestellt wird.

zu lesen, da hier der in Kapitel 5 vorgestellte Planungsalgorithmus erneut aufgegriffen wird.



---

## 2 Trajektorienplanung autonomer Fahrzeuge

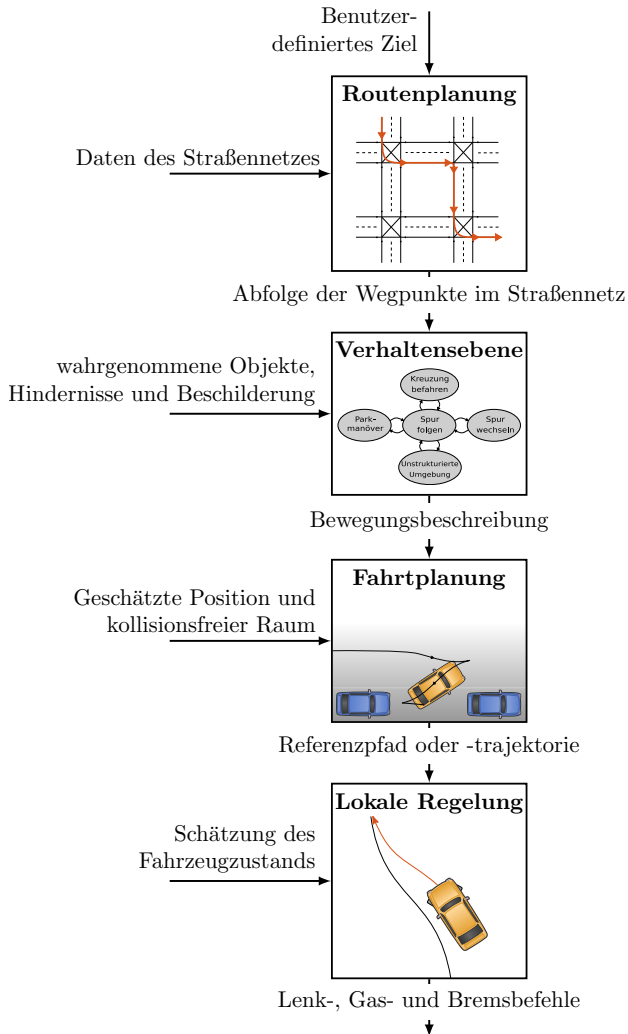
In diesem Kapitel wird zuerst ein kurzer Überblick über die Planungshierarchie autonomer Fahrzeuge gegeben, bevor die Problemstellung der Trajektorienplanung beschrieben und für diese Arbeit definiert wird. Im Anschluss werden unterschiedliche Fahrzeugmodelle, welche die physikalische Basis der Planungsalgorithmen darstellen, vorgestellt. Für einen Überblick über den Aufbau autonomer Fahrzeuge wird auf [97] verwiesen.

### 2.1 Planungshierarchie autonomer Fahrzeuge

Damit ein autonomes Fahrzeug den Passagier zu seinem gewünschten Zielort bringt, wird häufig auf eine Planungshierarchie wie in [5, 97] beschrieben zurückgegriffen. Im Folgenden wird auf den Aufbau dieser eingegangen. Abb. 2.1 zeigt dabei die Zusammenhänge zwischen den einzelnen Planungsebenen auf.

#### Routenplanung

Im ersten Schritt der Planungshierarchie, der Routenplanung, übergibt der Passagier dem autonomen Fahrzeug einen Zielort, an welchen er gefahren werden will. Mit Hilfe von Informationen über das vorliegende Straßennetzwerk kann so eine Route zum Zielort geplant werden. Dieser Schritt ist vergleichbar mit heutigen Autonavigationssystemen. Dabei wird das Straßennetzwerk als Graph dargestellt, bestehend aus Knoten und Kanten. Eine Kante repräsentiert eine Straße, ein Knoten eine Kreuzung/Abzweigung. Jede Kante besitzt einen Kostenterm, welcher z. B. die benötigte Zeit zur Benutzung dieser Straße widerspiegelt. Mit Hilfe von Algorithmen zur Suche von kürzesten Pfaden, wie dem Algorithmus von Dijkstra [31] oder A\*-Algorithmus [48], kann so eine effiziente Route durch das Straßennetzwerk gefunden werden. Um große Straßennetzwerke sowie



**Abbildung 2.1:** Häufige Planungshierarchie autonomer Fahrzeuge. Das Fahrzeug plant zuerst eine Route durch das Straßennetzwerk zum Zielort, wie ein übliches Navigationssystem. Im Anschluss erzeugt ein Verhaltensplaner ein abstraktes Fahrzeugverhalten, welches an den Trajektorienplaner weitergeleitet wird. Dieser berechnet eine kollisionsfreie Trajektorie, welche das abstrakte Fahrzeugverhalten widerspiegelt. Letztendlich wird die Trajektorie durch Einsatz einer Regelung vom Fahrzeug abgefahren. Abbildung nach [97].

Echtzeit-Verkehrsdaten verarbeiten zu können, wurden spezielle Algorithmen entwickelt. Dabei werden in einem Vorverarbeitungsschritt Hilfsinformationen aus dem Straßennetzwerk erzeugt. In einem zweiten Schritt wird mit Hilfe der Hilfsinformationen innerhalb kürzester Zeit die beste Route bestimmt [29]. Für einen Überblick über Algorithmen zur Routenplanung wird auf [8] verwiesen. Nach erfolgreicher Berechnung wird das Ergebnis der Routenplanung an die nächste Ebene der Planungshierarchie, die Verhaltensplanung, weitergegeben.

### Verhaltensplanung

Die in der ersten Ebene geplante Route gibt die Straßenabschnitte vor, auf denen sich das autonome Fahrzeug bewegen soll. In der zweiten Ebene wird ein abstraktes Verhalten<sup>1</sup> geplant, welches das autonome Fahrzeug ausführen soll. Hierzu wird die direkte Umgebung des Fahrzeugs miteinbezogen. Unter Berücksichtigung von z. B. anderen Verkehrsteilnehmern, statischen Hindernissen, Straßenmarkierungen und Verkehrszeichen (Straßenschilder, Lichtzeichenanlagen, etc.) wird ein grobes Verhalten des Fahrzeugs geplant. Dabei können die Definitionen der Verhaltensklassen je nach Design und Anwendungsgebiet des autonomen Systems variieren. Ein Beispielsatz solcher Verhaltensklassen ist z. B. nach [39]:

- Spur halten: Das Fahrzeug folgt der aktuellen Spur und hält den Abstand zum unmittelbar vorausfahrenden Fahrzeug.
- Spurwechsel links/rechts: Jeweils eine Klasse für einen einfachen Spurwechsel zu einer Seite.
- Links-/rechts abbiegen bzw. geradeaus fahren: Jeweils eine Klasse für Verhalten an Kreuzungen.
- Anhalten: Das Fahrzeug bremst in den Stillstand.

Dieses Set an möglichen Verhalten kann je nach Anwendungsgebiet erweitert oder reduziert werden. Soll das autonome System z. B. nur auf der Autobahn fahren, werden die Klassen für Kreuzungen nicht benötigt, dafür kann z. B. eine Klasse zum Bilden einer Rettungsgasse in Stausituationen von Vorteil sein. Da sowohl die Anzahl der Anwendungsgebiete sowie die Anzahl der Verhaltensklassen endlich ist, wird zur Modellierung meist eine endliche Zustandsmaschine wie in [4, 13, 46, 91] gewählt. Durch

---

<sup>1</sup>In anderer Literatur wird ebenfalls der Begriff Manöver verwendet [61].

die Erweiterung dieser zu probabilistischen Modellen wie z. B. Markow-Entscheidungsprozessen (engl.: *Markov decision process* (MDP)) [17] kann die Unsicherheit der Zustandstransitionen berücksichtigt werden. Dies spiegelt die Unsicherheit gegenüber dem Verhalten der anderen Verkehrsteilnehmer wider und ist sinnvoll, da z. B. an einer Kreuzung oft nicht ersichtlich ist, in welche Richtung ein anderer Verkehrsteilnehmer fahren möchte. [7, 16, 39, 135] erweitern den MDP zu einem partiell beobachtbaren Markow-Entscheidungsprozess (engl.: *partially observable Markov decision process* (POMDP)). Beim POMDP wird zusätzlich zur Transitionsunsicherheit die Unsicherheit der Perzeption, also die Unsicherheit darüber, in welchem Zustand sich andere Verkehrsteilnehmer augenblicklich befinden, berücksichtigt. Das berechnete abstrakte Verhalten wird im Anschluss an den Bewegungsplaner weitergegeben.

### Trajektorienplanung

In der dritten Ebene wird das geplante abstrakte Verhalten an die derzeitige Situation und Umgebung angepasst. Aus z. B. dem oben beschriebenen Verhalten „Spur halten“ muss ein konkreter Bewegungsablauf (Trajektorie) berechnet werden, welcher das beschriebene Verhalten umsetzt. In diesem Beispiel soll das Fahrzeug der vorhandenen Spur folgen und auf Änderungen der Geschwindigkeit eines vorausfahrenden Verkehrsteilnehmers eingehen. Weiterhin muss der berechnete Bewegungsablauf physikalisch fahrbar sein, damit das autonome Fahrzeug diesem folgen kann. Eine mathematische Definition des Problems der Trajektorienplanung wird in Abschnitt 2.2 gegeben. In Abschnitt 5.2 wird auf verschiedene in der Literatur verwendete Ansätze eingegangen.

### Trajektorienregelung

In der letzten Ebene wird der geplante Bewegungsablauf umgesetzt. Mittels eines Reglers werden aus der geplanten Trajektorie konkrete Lenk- und Beschleunigungsbefehle für das Fahrzeug berechnet. Diese Befehle werden an die Aktorik des Fahrzeugs weitergeleitet und ausgeführt. Durch die Rückführung des Bewegungsfehlers wird die Abweichung zwischen geplanter und umgesetzter Trajektorie möglichst klein gehalten. Da das System, wie bei jeder Rückführung, instabil werden kann, liegt der Fokus der Forschung nicht nur darauf, den Regelfehler zu minimieren, sondern auch die Stabilität sowie Robustheit des Reglers zu gewährleisten.

Um eine Trajektorienfolgeregelung umzusetzen, gibt es verschiedene

Ansätze. Häufig wird eine modellprädiktive Regelung gewählt [62, 93, 137]. Andere Ansätze unterteilen das Problem in einen Pfadfolgeregler und einen Geschwindigkeitsregler. Der Pfadfolgeregler versucht dabei den lateralen (minimalen) Abstand zwischen Pfad und Fahrzeugposition zu minimieren, während der Geschwindigkeitsregler die longitudinale Bewegung des Fahrzeugs regelt. Für den Pfadfolgeregler sind der Pure Pursuit-Regler [130] und die laterale Regelung der Vorder- [125] oder Hinterachse [107] häufig verwendete Ansätze. Durch Hinzufügen von vorausschauenden Komponenten [136] kann die Genauigkeit der Pfadfolgeregelung weiter verbessert werden.

### Eingliederung dieser Arbeit

Nachdem die einzelnen Ebenen der Planungshierarchie autonomer Fahrzeuge vorgestellt wurden, folgt eine Eingliederung dieser Dissertation. Der Fokus liegt hierbei auf der Ebene drei, der Bewegungsplanung. Kapitel 3 stellt theoretische Überlegungen bzgl. des Referenzpunktes autonomer Fahrzeuge dar, welche für Vereinfachungen bei der Bewegungsplanung benutzt werden können. Die Erkenntnisse aus Kapitel 4, der Analyse menschlichen Fahrverhaltens, kommen ebenso dem Planen von Trajektorien zugute. In Kapitel 5 werden die Überlegungen der vorherigen Kapitel in einem neuartigen Planerkonzept (CarPre-Trajektorienplanung) umgesetzt. Dabei ist das Ziel eine Trajektorie zu generieren (Bewegungsplanung). Die Besonderheit hierbei ist, dass kein direktes Verhalten vorgegeben, sondern stattdessen das kombinierte Problem der Verhaltens- und Bewegungsplanung gelöst wird.

## 2.2 Trajektorienplanung

Dieser Abschnitt orientiert sich an den mathematischen Definitionen nach [97]. Die Trajektorienplanung ist nach Abschnitt 2.1 die dritte Ebene der Planungshierarchie autonomer Fahrzeuge. Die Aufgabe eines Trajektorienplaners ist, aus einem vorgegebenen abstrakten Verhalten einen Verlauf von Zuständen, beginnend bei einem Startzustand<sup>2</sup>  $\mathbf{x}_{\text{start}}$  des Fahrzeugs zu einem Endzustand  $\mathbf{x}_{\text{ende}}$ , zu berechnen. Der Endzustand wird vom Verhaltensplaner vorgegeben und ist dabei nicht zwingend ein bestimmter Fahrzeugzustand, sondern kann z. B. ein Zielbereich sein, den

---

<sup>2</sup>Die Dimension des Fahrzeugzustands kann je nach Problemformulierung und Fahrzeugmodell (vgl. Abschnitt 2.3) variieren.

das Fahrzeug erreichen soll. Eine Trajektorie wird als der zeitliche Verlauf von Zuständen definiert. Die Lösung der Planung ist eine zeitabhängige Trajektorie  $\Pi(t) : [0, T_{\text{hor}}] \rightarrow \mathcal{X}$ , wobei  $T_{\text{hor}}$  der Planungshorizont des Planers und  $\mathcal{X}$  der Zustandsraum des Fahrzeugs ist. Die Menge an erlaubten Zuständen  $\mathcal{X}_{\text{erlaubt}}(t) \subseteq \mathcal{X}$  schränkt die Menge der generell möglichen Zustände des Fahrzeugs ein. Einschränkungen ergeben sich z. B. durch die Berücksichtigung möglicher Fahrbewegungen durch ein Bewegungsmodell (vgl. Abschnitt 2.3) oder durch die Vermeidung von Kollisionen mit statischen oder dynamischen Hindernissen. Mit diesen Einschränkungen ist es möglich, fahrbare Lösungen zu finden, allerdings wird dabei die Qualität der Lösung nicht berücksichtigt. Deshalb wird zusätzlich eine Gütefunktion  $J(\Pi)$  definiert, welche Bedingungen für weitere gewünschte Eigenschaften wie z. B. Fahrzeit, Sicherheit, Komfort oder Energieeffizienz beschreibt. Es wird das folgende Optimierungsproblem definiert:

$$\begin{aligned} & \min_{\Pi} J(\Pi), \\ & \text{unter den Nebenbedingungen} \quad \Pi(0) = \mathbf{x}_{\text{start}}, \\ & \quad \quad \quad \Pi(T_{\text{hor}}) = \mathbf{x}_{\text{ende}}, \\ & \quad \quad \quad \Pi(t) \in \mathcal{X}_{\text{erlaubt}}(t). \end{aligned}$$

Ziel der Trajektorienplanung ist es, genau dieses Optimierungsproblem zu lösen. Das Problem ist nach [103] der Komplexitätsklasse PSPACE<sup>3</sup> zuzuordnen, ebenso wie das Pfadplanungsproblem in statischen Szenarien [104]. Um diese komplexe Aufgabe zu lösen, wird auf numerische Lösungsmethoden zurückgegriffen, da praxisrelevante Algorithmen zur Berechnung exakter Lösungen nicht zur Verfügung stehen [97, S. 8]. Ein Überblick über verschiedene Lösungsansätze wird in Abschnitt 5.2 gegeben.

Im Folgenden wird auf die Einschränkungen des in dieser Arbeit betrachteten Planungsproblems eingegangen. Es werden nur Lösungen für PKW-ähnliche Landfahrzeuge mit lenkbaren Vorderrädern und starren Hinterrädern betrachtet. Durch die Beschränkung auf diese Fahrzeugklasse kann ein einheitliches Bewegungsmodell (vgl. Abschnitt 2.3) verwendet werden. Des Weiteren wird der Planungsraum auf eine 2D-Ebene reduziert, sodass nur eine 2D- anstatt einer 3D-Position des Fahrzeugs verwendet

<sup>3</sup>Ein Problem der Komplexitätsklasse PSPACE kann von einer deterministischen Turingmaschine mit Ressourcen (z. B. Speicherplatz), die polynomial zur Dimension der Eingangsvariablen sind, gelöst werden. Eine zeitliche Begrenzung zur Lösung des Problems ist nicht gegeben.

wird. Die Vernachlässigung von Höheninformation ist für Landfahrzeuge ein gängiger Schritt, da sich das Fahrzeug nur auf einer fest definierten Höhe, seinem Untergrund, bewegen kann.<sup>4</sup> Weitere Annahmen beziehen sich auf die Umgebungssituation. Im Rahmen dieser Arbeit liegt der Fokus auf städtischen Szenarien. Dies bedeutet, dass eine strukturierte Umgebung durch das vorhandene Straßennetzwerk gegeben ist. Darüber hinaus ist die Geschwindigkeit des Fahrzeugs auf 30 km/h limitiert (vgl. Kapitel 5). Im nächsten Abschnitt werden für dieses Planungsproblem übliche Bewegungsmodelle vorgestellt.

## 2.3 Fahrzeugmodelle

Eine Grundvoraussetzung für die Planung von Bewegungsabläufen, wie z. B. Trajektorien für autonome Fahrzeuge, ist ein Bewegungsmodell, welches die Zusammenhänge zwischen dem aktuellen Zustand und möglichen zukünftigen Zuständen beschreibt. Für den Anwendungsfall des autonomen Fahrens sind solche Bewegungsmodelle unabdingbar, da sich nicht-holonome Fahrzeuge durch ihre starre Hinterachse nur beschränkt bewegen können und dies während der Planung berücksichtigt werden muss. Ansonsten könnte es sein, dass der Planer Trajektorien ausgibt, welche das autonome Fahrzeug nicht abfahren kann.

Bei der Auswahl eines Bewegungsmodells müssen gewisse Abwägungen getroffen werden. Je höher das Abstraktionslevel des Modells ist, desto weniger physikalische Zusammenhänge werden dargestellt. Dadurch wird das Modell einerseits einfacher, allerdings werden andererseits die Bewegungen des Fahrzeugs ungenauer abgebildet. Im Gegensatz hierzu wird durch ein niedriges Abstraktionslevel die Genauigkeit des Modells erhöht. Dies geschieht auf Kosten der Modellkomplexität sowie der damit einhergehenden benötigten Rechenzeit.

Es lässt sich schlussfolgern, dass für unterschiedliche Aufgaben unterschiedlich abstrakte Bewegungsmodelle benötigt werden. Beim komplexen Problem der Trajektorienplanung kann ein Modell mit zu niedrigem Abstraktionslevel die ohnehin schon schwierige Lösungsfindung noch weiter erschweren. Daher wird in der Trajektorienplanung ein möglichst einfaches, aber noch ausreichend genaues Modell bevorzugt verwendet. Dagegen wird bei einer Trajektorienfolgeregelung, bei welcher das Fahrzeug

---

<sup>4</sup>Die Planung im 3D-Raum kann unter gewissen Umständen Sinn ergeben, z. B. in mehrstöckigen Parkhäusern, falls das Fahrzeug sich durch mehrere Parkebenen bewegen soll.

möglichst exakt die geplante Trajektorie abfahren soll, eine hohe Modellierungsgenauigkeit benötigt. Wird ein zu ungenaues Modell gewählt, wird die Abweichung vom Sollwert der Regelung zu groß sein.

Im Folgenden werden die wichtigsten Modelle zur Planung nach der Kategorisierung von [84] vorgestellt. Alle Modelle beziehen sich auf einen 2D-Raum.

### Punktmodell (PM)

Bei dem einfachsten Modell wird das Fahrzeug auf einen einzelnen Punkt reduziert. Die Masse des Fahrzeugs wird in diesem Punkt zusammengefasst, welcher sich beliebig im 2D-Raum bewegen kann. Somit werden weder die Ausdehnung noch die Bewegungseinschränkungen durch die Reifen in Kombination mit der starren Hinterachse eines Fahrzeuges berücksichtigt. Damit wird in diesem einfachsten Modell das Fahrzeug nur durch seine Position  ${}_{\mathcal{W}}x_{\text{ref}}, {}_{\mathcal{W}}y_{\text{ref}}$  innerhalb eines Weltkoordinatensystems  ${}_{\mathcal{W}}\mathcal{F}$  beschrieben.

Dieses Modell wird meist in Algorithmen zur Planung eines Pfades, wie z. B. in A\* [48] oder RRT [72], benutzt. Der gefundene Pfad muss nicht stetig sein<sup>5</sup>, schließlich gibt es keine Einschränkungen der möglichen Bewegungen.

### Modell konstanter Drehrate (KDR)

Beim Modell konstanter Drehrate wird das Punktmodell um Bewegungsgleichungen erweitert, sodass eine realitätsnähere Abbildung entsteht. Damit das Fahrzeug sich nicht willkürlich im Raum bewegen kann, wird nun die Fahrzeugpose, also zusätzlich zur Position auch die Orientierung  $\psi$ , betrachtet. Der Winkel  $\psi$  kann sich hierbei nur stetig ändern. Es entsteht das folgende Modell mit den Zuständen  $\mathbf{x} = [{}_{\mathcal{W}}x_{\text{ref}}, {}_{\mathcal{W}}y_{\text{ref}}, \psi]^{\top}$  und den Eingängen  $\mathbf{u} = [v, \omega_{\text{DR}}]^{\top}$ :

$$\begin{aligned} {}_{\mathcal{W}}\dot{x}_{\text{ref}} &= v \cos(\psi), \\ {}_{\mathcal{W}}\dot{y}_{\text{ref}} &= v \sin(\psi), \\ \dot{\psi} &= \omega_{\text{DR}}, \end{aligned}$$

wobei  $v$  der Geschwindigkeit und  $\omega_{\text{DR}}$  der Drehrate des Fahrzeugs entspricht. Dieses Modell wird z. B. in angepassten RRT-Algorithmen [68]

<sup>5</sup>Sowohl beim genannten A\*- als auch beim RRT-Algorithmus ist der geplante Pfad nicht stetig. Erweiterungen [68, 91] benutzen einschränkende Bewegungsmodelle, um diesem Problem entgegenzuwirken.



oder als zeitdiskretes Prädiktionsmodell der konstanten Drehrate und konstanten Geschwindigkeit [110] eingesetzt. Das Modell kann durch Hinzufügen weiterer Ableitungen, wie der Beschleunigung oder dem Ruck, erweitert werden. Dadurch wird die Bewegung des Fahrzeugs geglättet. Der Nachteil hierbei ist der Dimensionszuwachs des Zustandsraums.

### Kinematisches Einspurmodell (KEM)

Das kinematische Einspurmodell (auch: *Fahrradmodell*, *Halbes-Auto-Modell*) berücksichtigt die nicht-holonomen Bewegungen eines Fahrzeugs. Durch die Annahme der schräglaufrfreien<sup>6</sup> Fahrt können die beiden momentanen Drehpunkte eines realen Fahrzeugs zu einem momentanen Drehpunkt vereinfacht werden (vgl. Abb. 2.2a). Daher können die beiden Hinterräder zu einem Rad sowie die beiden Vorderräder ebenfalls zu einem Rad zusammengefasst werden. Das Fahrzeug besitzt somit nur eine Spur (siehe Abb. 2.2b). Der Zustandsvektor ergibt sich aus der Pose des Fahrzeugs, der Eingangsvektor des Systems  $\mathbf{u} = [v, \delta]^\top$  aus der Geschwindigkeit  $v$  und dem Vorderradlenkwinkel  $\delta$  des Fahrzeugs. Weitere Parameter des Modells sind

$l$  : Radstand,

$l_{\text{ref}}$  : Länge zwischen Hinterachse und Referenzpunkt,

$\beta$  : Schwimmwinkel.

Diese sind in Abb. 2.2b veranschaulicht. Die Bewegung des Fahrzeugs wird durch folgende Differentialgleichung beschrieben:

$$w\dot{x}_{\text{ref}} = v \cos(\psi + \beta), \quad (2.1)$$

$$w\dot{y}_{\text{ref}} = v \sin(\psi + \beta), \quad (2.2)$$

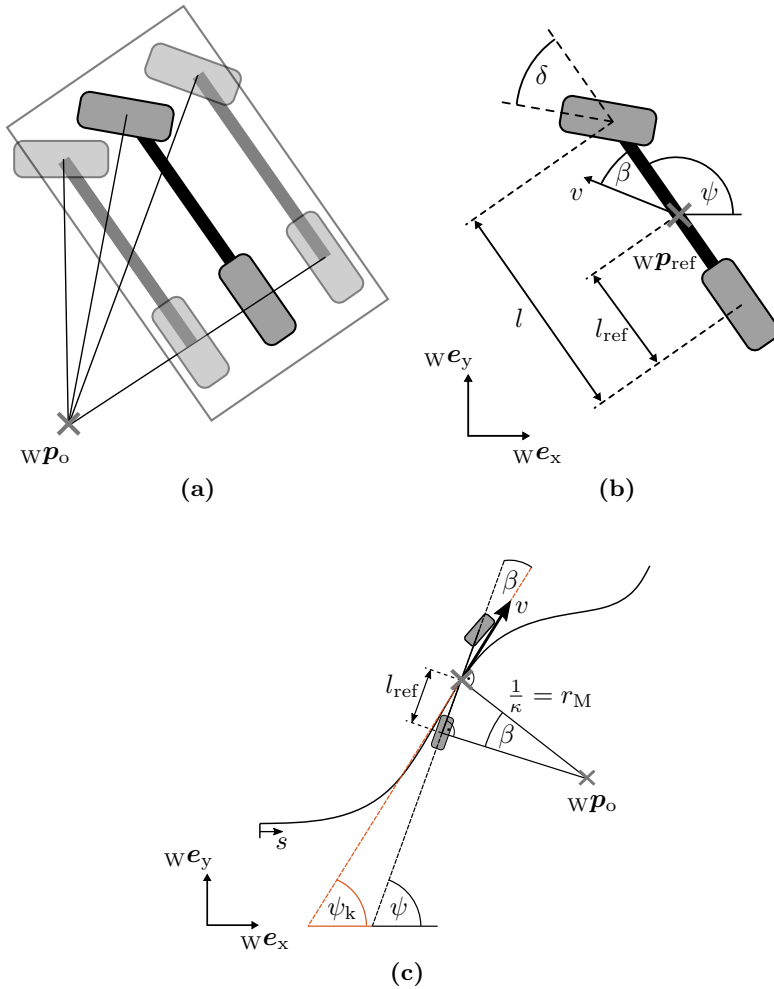
$$\dot{\psi} = \frac{v \cos \beta \tan \delta}{l}, \quad (2.3)$$

mit

$$\beta = \arctan\left(\frac{l_{\text{ref}} \tan \delta}{l}\right). \quad (2.4)$$

---

<sup>6</sup>An der Vorder- sowie der Hinterachse sind jeweils der Radeinschlag und die Bewegungsrichtung des Fahrzeugs gleich. Das Fahrzeug bewegt sich demnach immer in die Richtung seiner Räder. Dabei wird die Wirkung seitlicher Verformung der Reifen vernachlässigt.



**Abbildung 2.2:** Die beiden momentanen Drehpunkte des realen Fahrzeugs werden zu einem einzigen Drehpunkt zusammengefasst (a). Dadurch entsteht das Einspurmodell in (b), welches durch die dargestellten Zustände und Parameter beschrieben wird [141]. Die Parameter zu den Überlegungen des Schwimmwinkels werden in (c) dargestellt.

Die im Weiteren vorgestellten Überlegungen bzgl. des Schwimmwinkels wurden unter Mitarbeit des Autors in [100] veröffentlicht. Die benötigten Parameter sind in Abb. 2.2c dargestellt. Der Schwimmwinkel  $\beta$  beschreibt die Bewegungsrichtung relativ zur Orientierung des Fahrzeugs. Für den Fall, dass  $l_{\text{ref}} = l$  ist, kann der Zusammenhang aus Gleichung (2.4) zu  $\beta = \delta$  vereinfacht werden.

Unter der Annahme des schräglaufreren Fahrens werden entweder  $\beta$  oder  $\delta$  für das Folgen eines Pfades benötigt (Umrechnung nach Gleichung (2.4)). Da bei vielen Planungsansätzen (z. B. A\* [48] oder RRT [72]) Informationen über einen dieser Parameter entlang des Pfades nicht vorhanden sind, wird deshalb  $\beta$  direkt aus einem stetig differenzierbaren Pfad mit den folgenden kinematischen Überlegungen hergeleitet. Die hierzu benötigten Parameter sind in Abb. 2.2c dargestellt. Der Schwimmwinkel  $\beta$  kann durch die Differenz zwischen dem Kurswinkel  $\psi_k$  und der Orientierung  $\psi$  des Fahrzeugs beschrieben werden. Daraus folgt

$$\frac{d\beta}{ds} = \frac{d\psi_k}{ds} - \frac{d\psi}{ds} = \kappa - \frac{d\psi}{ds}, \quad (2.5)$$

wobei  $s$  die zurückgelegte Distanz auf einem Pfad ist. Mit

$$d\psi = \frac{ds}{r_M} = \frac{ds \cdot \sin \beta}{l_{\text{ref}}}, \quad (2.6)$$

wobei  $r_M$  der Abstand zwischen dem Referenzpunkt  ${}_{\text{W}}\mathbf{p}_{\text{ref}}$  und dem momentanen Drehpunkt  ${}_{\text{W}}\mathbf{p}_o$  ist. Unter Verwendung der Kleinwinkelnäherung erhält man

$$\frac{d\beta}{ds} = \kappa - \frac{\sin \beta}{l_{\text{ref}}} \approx \kappa - \frac{\beta}{l_{\text{ref}}} = \frac{1}{l_{\text{ref}}}(\kappa l_{\text{ref}} - \beta). \quad (2.7)$$

Wird dies mit der Dynamik eines Verzögerungsglieds erster Ordnung (PT1-Glied)  $y = \text{PT1}(u, \tau)$  mit Eingang  $u$ , Ausgang  $y$  und Zeitkonstante  $\tau$

$$\frac{dy}{dt} = \frac{1}{\tau}(u - y) \quad (2.8)$$

verglichen, wird der Schwimmwinkel in Abhängigkeit der Pfadkrümmung  $\kappa$  sowie der Fahrzeugkonstante  $l_{\text{ref}}$  beschrieben

$$\beta \approx \text{PT1}(\kappa l_{\text{ref}}, l_{\text{ref}}). \quad (2.9)$$

## Dynamisches Einspurmodell (DEM)

Bei dynamischen Modellen wird im Gegensatz zu kinematischen Fahrzeugmodellen die Annahme der schräglaufrfreien Fahrt fallen gelassen, d.h. es wird die Haftung bzw. Reibung zwischen Fahrbelag und Reifen modelliert. Dies ist wichtig, sobald die Beschleunigung des Fahrzeugs groß genug ist, da dann die Reifen des Fahrzeugs auf dem Fahrbelag anfangen zu rutschen, es ist ein Schräglaufwinkel sowie Schlupf vorhanden. Zur Herleitung eines solchen Modells wird sich an [57, 97, 113, 127] orientiert. Zur Unterstützung dieser sind in Abb. 2.3 alle verwendeten Zustände und Parameter dargestellt. Der Referenzpunkt  ${}_{\text{W}}\mathbf{p}_{\text{ref}}$  wird in den Fahrzeugschwerpunkt  ${}_{\text{W}}\mathbf{p}_{\text{SP}}$ , um welchen sich das Fahrzeug dreht, verschoben. Weiterhin wird das Fahrzeug als Starrkörper mit einer Masse  $m$  und einem Gierträgheitsmoment  $I_z$  angenommen. Die Reifen- und Fahrwerksdynamik sowie aerodynamische Kräfte werden vernachlässigt. Aus dem Kräfte- und Momentengleichgewicht lassen sich folgende Gleichungen herleiten:

$$\begin{aligned} \begin{bmatrix} {}_{\text{W}}\dot{x}_{\text{ref}} \\ {}_{\text{W}}\dot{y}_{\text{ref}} \end{bmatrix} &= \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix} \begin{bmatrix} {}_{\text{F}}v_x \\ {}_{\text{F}}v_y \end{bmatrix}, \\ \text{F}\dot{v}_x &= \frac{1}{m} (\text{VR}F_x \cos \delta - \text{VR}F_y \sin \delta + \text{HR}F_x) + \text{F}v_y \dot{\psi}, \\ \text{F}\dot{v}_y &= \frac{1}{m} (\text{VR}F_x \sin \delta + \text{VR}F_y \cos \delta + \text{HR}F_y) + \text{F}v_x \dot{\psi}, \\ \ddot{\psi} &= \frac{1}{I_z} [(l - l_{\text{ref}}) (\text{VR}F_x \sin \delta + \text{VR}F_y \cos \delta) - l_{\text{ref}} \text{HR}F_y]. \end{aligned}$$

$\text{VR}F_x$ ,  $\text{VR}F_y$  stellen dabei die longitudinalen und lateralen Kräfte auf dem Vorderreifen bzw.  $\text{HR}F_x$  und  $\text{HR}F_y$  die Kräfte auf den Hinterreifen dar. Der Schlupf an einem Reifen setzt sich aus dem longitudinalen und lateralen Schlupf zusammen  $\rho S = \sqrt{\rho S_x^2 + \rho S_y^2}$ ,  $\rho \in \{\text{VR}, \text{HR}\}$ . Der laterale Schlupf an den Reifen wird nach [57] durch

$$\begin{aligned} \text{VR}S_y &= \frac{(\text{F}v_y + (l - l_{\text{ref}})\dot{\psi}) \cos \delta - \text{F}v_x \sin \delta}{\text{F}v_x \cos \delta + (\text{F}v_y + (l - l_{\text{ref}})\dot{\psi}) \sin \delta}, \\ \text{HR}S_y &= \frac{\text{F}v_y - l_{\text{ref}}\dot{\psi}}{\text{F}v_x}, \end{aligned}$$

bestimmt, während der longitudinale Schlupf Teil des Systemeingangs ist. Durch einen linearen Zusammenhang zwischen Reibkraft und der Normal-

kraft für beide Reifen ergeben sich die Reifenkräfte aus

$${}_{\rho}F_{\sigma} = \mu_{\rho\sigma} \cdot {}_{\rho}F_z, \quad \rho \in \{\text{VR}, \text{HR}\}, \sigma \in \{x, y\}.$$

Die Parameter  $\mu_{\rho\sigma}$  stehen dabei für die lateralen sowie longitudinalen Reibwerte an beiden Reifen, welche durch ein beliebiges Reifenmodell bestimmt werden können. Oft wird hier das Reifenmodell von Pacejka [6] gewählt:

$$\begin{aligned} \mu_{\rho\sigma} &= -({}_{\rho}S_{\sigma} / {}_{\rho}S) \mu_{\rho}, \\ \mu_{\rho} &= D_{\rho} \sin(C_{\rho} \tan^{-1}(B_{\rho} s_{\rho})), \end{aligned}$$

wobei  $\rho \in \{\text{VR}, \text{HR}\}, \sigma \in \{x, y\}$  und  $B_{\rho}, C_{\rho}, D_{\rho}$  Konstanten des Reifenmodells sind. Durch das Kräftegleichgewicht in vertikaler Richtung sowie das Momentengleichgewicht am Hinterreifen können für die Normalkräfte nach [127] die folgenden Gleichungen hergeleitet werden:

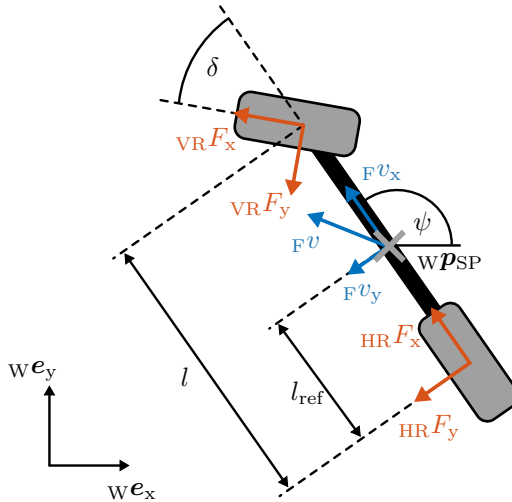
$$\begin{aligned} {}_{\text{VR}}F_z &= \frac{mg(l_{\text{ref}} - h\mu_{\text{HRx}})}{l + h(\mu_{\text{VRx}} \cos \delta - \mu_{\text{VRy}} \sin \delta - \mu_{\text{HRx}})}, \\ {}_{\text{HR}}F_z &= mg - {}_{\text{VR}}F_z, \end{aligned}$$

wobei  $h$  die Höhe des Fahrzeugschwerpunktes über dem Boden ist. Durch das Zusammenfügen aller Gleichungen entsteht das dynamische Fahrzeugmodell mit seinen Zuständen  $\mathbf{x} = [{}_W x_{\text{ref}}, {}_W y_{\text{ref}}, \psi, {}_F v_x, {}_F v_y, \dot{\psi}]^T$ . Als Eingang  $\mathbf{u} = [{}_{\text{VR}} S_x, {}_{\text{HR}} S_x, \delta]^T$  wird der longitudinale Schlupf an beiden Reifen sowie der Lenkwinkel gewählt.

### Weitere Modelle (WM)

In den bisherigen Modellen werden verschiedenste Vereinfachungen getroffen, welche in komplexeren Modellen berücksichtigt werden können. Mögliche komplexere Fahrzeugmodelle können folgende Betrachtungen beinhalten:

- Zweispurmodell: Erweiterung des dynamischen Einspurmodells unter Berücksichtigung von allen vier Fahrzeugreifen.
- Aerodynamische Kräfte: Berücksichtigung des Fahrzeugverhaltens in Abhängigkeit der Umgebungsluft anhand der Relativgeschwindigkeit sowie des Fahrzeugdesigns (Um- und Durchströmung).
- Reifendynamik: Trägheitsmomente der Reifen sowie deren Rollwiderstände.



**Abbildung 2.3:** Dynamisches Einspurmodell, welches durch die dargestellten Zustände und Parameter beschrieben wird.

- Fahrwerksdynamik: Modellierung des Fahrwerks samt Stoßdämpfer.
- Antriebsstrang: Modellierung des Antriebsstrangs sowie des Motors.
- Lenkung: Modellierung der Fahrzeuglenkung.
- Gravitationskraft: Fahrzeug bewegt sich in 3D- statt in 2D-Welt.

Diese Fahrzeugmodelle werden allerdings meist nicht zur Trajektorienplanung sondern zur Trajektorienfolgeregelung oder zur Fahrzeugsimulation verwendet. Da die zur Planung wichtigsten Modelle oben beschrieben wurden, wird für solche komplexeren Modelle auf weiterführende Literatur, wie z. B. [102], verwiesen.

## Modellauswahl

Nach der Vorstellung mehrerer Bewegungsmodelle, wird in diesem Abschnitt ein Modell für den weiteren Verlauf dieser Arbeit ausgewählt. In Tabelle 2.1 ist ein Überblick über die Komplexität der einzelnen Modelle sowie häufige Anwendungsfälle dargestellt.

In der Literatur existieren einige Untersuchungen, inwieweit das Fahrzeugmodell der Planung Einfluss auf das Fahrverhalten hat. Mit Hilfe von

**Tabelle 2.1:** Vergleich der vorgestellten Fahrzeugmodelle:

Punktmodell (PM), Modell konstanter Drehrate (KDR), kinematisches Einspurmodell (KEM), dynamisches Einspurmodell (DEM), weitere Modelle (WM).  
 Modellkomplexität: -- sehr niedrig, - niedrig, 0 neutral, + hoch, ++ sehr hoch.

Modell	Komplexität	Anwendungen
PM	--	Pfadplanung ohne Bewegungseinschränkungen
KDR	-	Trajektorienplanung & -prädiktion
KEM	0	Trajektorienplanung mit geringen lateralen Beschleunigungen
DEM	+	Trajektorienplanung im Grenzbereich des Fahrzeugs
WM	++	Regelung und Simulation

Fahrzeugsimulationen untersuchen Polack et al. [99] inwieweit das kinematische Einspurmodell zur Trajektorienplanung verwendet werden kann. Das Ziel der Planung mittels eines Bewegungsmodells ist, eine konsistente Lösung zu berechnen, d.h. eine Trajektorie zu bestimmen, welche der Fahrzeugregler innerhalb eines gegebenen Fehlermaßes abfahren kann. Polack et al. finden in ihrer Arbeit heraus, dass eine konsistente Planung mit dem kinematischen Einspurmodell sichergestellt ist, falls die Querbeschleunigung des Fahrzeugs auf  $0,54\mu g$  beschränkt wird. Dabei stellt  $g$  die Fallbeschleunigung der Erde und  $\mu$  den Reibungskoeffizienten zwischen den Reifen des Fahrzeugs und der Straße dar.

Subosits et al. [121] haben die Auswirkungen unterschiedlicher dynamischer Bewegungsmodelle auf die Fahrzeit sowie die Konsistenz der Planung an einem Rennfahrzeug untersucht. Dabei wurden optimale Trajektorien mit unterschiedlichen Modellen, beginnend bei einem dynamischen Punktmodell bis hin zu einem dynamischen Zweispurmodell mit Lastwechsel, geplant und auf einem realen Versuchsträger bei unterschiedlichen Straßenbedingungen getestet. Die Haupte Erkenntnis ist dabei, dass „der Vorteil komplexerer Modelle weniger darin besteht, dass sie das Verhalten des Fahrzeugs besser vorhersagen, sondern dass die Fahrzeuggrenzen in Extremsituationen besser abgeschätzt werden können“<sup>7</sup> [121, S. 1]. Hier-

<sup>7</sup>Englischer Originaltext: „The tests reveal that the advantage of more complex models is less that they better predict the vehicle’s behavior, but that they provide a more nuanced view of the vehicle’s limits“.

durch ermöglicht ein komplexeres Modell die bestmögliche Nutzung der limitierten Reifenreibung.

Um sich für ein Modell entscheiden zu können, muss der spezifische Anwendungsfall betrachtet werden. Die Trajektorienplanung eines Fahrzeugs, welches Driften oder auf einer Rennstrecke Bestzeiten fahren soll, benötigt ein anderes Fahrzeugmodell als z. B. ein autonomer Taxiservice. Da der Fokus dieser Arbeit auf Trajektorienplanung in innerstädtische Szenarien liegt, kann auf ein dynamisches Fahrzeugmodell verzichtet werden. Stattdessen ist die einfache Anwendbarkeit wichtiger als das Ausreizen von fahrdynamischen Grenzen. Es soll ein System eines komfortablen, autonomen, innerstädtischen Taxis entstehen, sodass ein hoher Fahrkomfort ohne zu große Querschleunigungen das Ziel ist.

Das kinematische Einspurmodell bietet gegenüber dem Modell konstanter Drehrate eine Berücksichtigung der nicht-holonomen Bewegungen des Fahrzeugs. So kann sich das Fahrzeug beim Modell konstanter Drehrate auf der Stelle drehen, ohne sich dabei vorwärts zu bewegen. Dies ist keine realistische Abbildung der Fahrzeugbewegungen. Da bei innerstädtischen Szenarien, z. B. beim Vorfahrt gewähren an einer Engstelle zwischen zwei parkenden PKWs, eine konsistente Lösung auf begrenztem Raum gefunden werden muss, sollte mindestens ein Modell von der Komplexität des kinematischen Einspurmodells gewählt werden. Somit wird für diese Arbeit das einfachste Modell, welches die grundlegenden Fahrzeugbewegungen abbildet, gewählt, das kinematische Einspurmodell. Die Limitierung der lateralen Beschleunigung für eine konsistente Planung nach [99] auf  $0,54\mu g$  stellt hierbei kein Problem dar, da große laterale Beschleunigungen auf Grund des hohen Fahrkomforts und der innerstädtischen Szenarien ausgeschlossen werden können.

## 2.4 Zusammenfassung

In diesem Kapitel wird ein Überblick über die Thematik der Trajektorienplanung gegeben. Nach der Vorstellung der Planungshierarchie autonomer Fahrzeuge, werden die weiteren Kapitel dieser Arbeit in die einzelnen Ebenen der Hierarchie eingeteilt. Im Anschluss wird detaillierter auf die Aufgabe der Trajektorienplanung eingegangen. Hier werden Einschränkungen bzgl. der in dieser Arbeit verwendeten Planungsaufgabe getroffen. Abschließend werden unterschiedliche Bewegungsmodelle für die Trajektorienplanung vorgestellt. Durch einen Vergleich wird ein Modell für die weitere Arbeit ausgewählt.



---

## 3 Idealer Referenzpunkt

In diesem Kapitel wird eine Analyse zur Wahl des Referenzpunktes für die Trajektorienplanung präsentiert. Der Referenzpunkt wird als der Bezugspunkt eines fahrzeugfesten Koordinatensystems  $_{\mathcal{F}}\mathcal{F}$  zu einem Weltkoordinatensystem  $_{\mathcal{W}}\mathcal{F}$  definiert. Nach der thematischen Einleitung sowie einem Überblick zur Wahl von Referenzpunkten in der Literatur wird die Wahl des Referenzpunktes untersucht. Hierzu werden Kurvenfahrten konstanter und variabler Krümmung anhand von geometrischen Überlegungen analysiert. Das Ziel dabei ist die Minimierung des seitlichen Fahrzeugplatzbedarfs. Diese theoretischen Überlegungen werden im Anschluss in einem Kreismodell mit nur einem Kreis verwendet. Durch die vorherige Platzbedarfsanalyse kann der Radius des Modells deutlich kleiner gewählt werden als ein Kreis, welcher durch die vier Fahrzeugecken verläuft. Die Fahrzeugorientierung wird bei der Verwendung des Kreismodells durch die Rotationsinvarianz nicht benötigt. Nach der Vorstellung möglicher Anwendungen des Kreismodells schließt das Kapitel mit einer Zusammenfassung ab. Die hier vorgestellten Aspekte wurden unter Mitarbeit des Autors in [100] veröffentlicht und werden hier erneut dargestellt.

### 3.1 Einleitung und Motivation

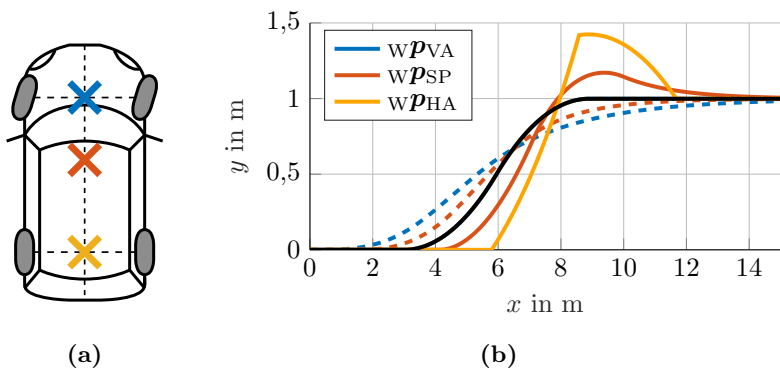
Mit dem Übergang des autonomen Fahrens von der Autobahn in innerstädtische Bereiche werden die Situationen, welche das autonome Fahrzeug absolvieren muss, komplexer (vgl. Abb. 1.1). Da die Trajektorienplanung in Echtzeit und mit limitierter Rechenkapazität ausgeführt werden muss, ist es hilfreich, möglichst viele Vereinfachungen im Planungsalgorithmus zu integrieren. Potenzielle Vereinfachungen sind z. B. die Wahl simplerer Fahrzeugmodelle (vgl. Abschnitt 2.3). So können die physikalischen Bewegungen eines Fahrzeugs z. B. durch ein dynamisches Fahrzeugmodell [122] oder durch ein kinematisches Fahrzeugmodell [125] beschrieben werden. Beide Modelle sind Näherungen für das exakte Fahrverhalten des Fahrzeugs. Während das dynamische Modell die Eigenschaften der Reifen mitberücksichtigt, wird im kinematischen Modell die Bewegung nur

anhand geometrischer Gleichungen beschrieben. Eine weitere Möglichkeit Rechenzeit zu reduzieren ist es, mathematische Vereinfachungen zu nutzen.

Mit dem Ziel, die Rechenzeit während der Planung zu minimieren, wird die Wahl des Referenzpunktes analysiert. Das Planungsproblem wird, wie in Abschnitt 2.2 beschrieben, auf eine 2D-Umgebung vereinfacht, in der  ${}_W\mathbf{p}$  als ein 2D-Punkt im Weltkoordinatensystem  ${}_W\mathcal{F}$  definiert wird. Außerdem wird mit dem Referenzpunkt  ${}_W\mathbf{p}_{\text{ref}}$  der Ursprung des fahrzeugfesten Koordinatensystems  ${}_F\mathcal{F}$  beschrieben. Übliche Bezugspunkte sind die Mitte der Vorderachse (VA), die Mitte der Hinterachse (HA) oder der Schwerpunkt (SP), diese sind in Abb. 3.1a dargestellt.

In [33] wird die Hinterachse des Fahrzeugs als Referenzpunkt für die Lokalisierung verwendet. Indem eine inertielle Messeinheit auf der Hinterachse positioniert wird, werden die kinematischen Eigenschaften des Fahrzeugs in diesem Punkt genutzt. Denn hier gilt für eine schräglaufrfreie Fahrt, dass die Querschleunigung des Fahrzeugs nur von der Zentripetalkraft abhängt. Bei einer anderen Position der inertialen Messeinheit wird auch die Rotationsbeschleunigung des Fahrzeugs gemessen, die dann im Lokalisierungsalgorithmus berücksichtigt werden muss.

Durch die Wahl eines anderen Referenzpunktes werden die physikalischen Eigenschaften des zu betrachtenden Fahrzeugs nicht verändert. Aber wie im Beispiel der Lokalisierung kann eine intelligente Wahl zu Vereinfachungen führen. In Abb. 3.1b folgt ein Fahrzeug einem vorgegebenen Referenzpfad unter Verwendung verschiedener Referenzpunkte  ${}_W\mathbf{p}_{\text{ref}}$ . Für jeden Referenzpunkt wird der ideale Lenkwinkel des Fahrzeugs berechnet und anschließend die Positionen der Mitte seiner Vorderachse (durchgezogene Linien) und seiner Hinterachse (gestrichelte Linien) aufgezeichnet. Aufgrund der unterschiedlichen Referenzpunkte werden unterschiedliche Trajektorien gefahren. Die Wahl eines Referenzpunktes hinter der Vorderachse führt zu einem Überschwingen der Vorderachse, während die Wahl eines Referenzpunktes vor der Vorderachse den gefahrenen Weg glättet. Dieser Effekt wird bei der Planung oft vernachlässigt, sollte aber berücksichtigt werden, insbesondere bei der Verwendung von Punktplanungsalgorithmen wie Hybrid A\* [91] oder RRT\* [59], bei denen das Fahrzeug als Punkt approximiert und seine räumliche Ausdehnung vernachlässigt wird. Daher wird in den folgenden Abschnitten die Position des Referenzpunktes analysiert, sodass der Platzbedarf der gefahrenen Fahrspur auf beiden Seiten der geplanten Trajektorie gleich ist.



**Abbildung 3.1:** Der Fahrzeugreferenzpunkt  $w\mathbf{p}_{\text{ref}}$  (Ursprung des fahrzeugfesten Koordinatensystems  $\mathcal{F}$ ) wird häufig zur Mitte der Vorderachse  $w\mathbf{p}_{\text{VA}}$  (blau), Mitte der Hinterachse  $w\mathbf{p}_{\text{HA}}$  (gelb) oder zum Schwerpunkt  $w\mathbf{p}_{\text{SP}}$  (rot) gewählt, welche in (a) dargestellt sind. Auch wenn die Wahl des Referenzpunktes die physikalischen Eigenschaften des Fahrzeugs nicht verändert, beeinflusst sie dennoch in einem Pfadfolgeszenario den gefahrenen Pfad des Fahrzeugs. In (b) stellt die schwarze Linie den Referenzpfad dar, eine S-Kurve, die unter Verwendung der oben genannten Referenzpunkte nachgefahren wird. Anschließend wird der gefahrene Pfad der Mitte der Vorderachse (durchgezogene Linien) und der Mitte der Hinterachse (gestrichelte Linien) eingezeichnet. [100]

## 3.2 Stand der Technik

Nach Wissen des Autors gibt es keine Veröffentlichung, die sich mit der Wahl des Referenzpunktes für die Planungsaufgabe beschäftigt. In den meisten Veröffentlichungen wird die Lage des Referenzpunktes nur kurz, entweder im Text oder in Grafiken, dargestellt. Daher werden im Folgenden die in anderen Publikationen gewählten Positionen des Referenzpunktes diskutiert.

Zunächst befinden sich alle Referenzpunkte in der lateralen Mitte des Fahrzeugs, also genau in einer der drei in Abb. 3.1a vorgestellten Positionen. Dabei ist zu beachten, dass die Position des Schwerpunktes nur eine Annäherung ist, da sie im realen Fahrzeug z. B. von Passagieren, Beladung und Kraftstoffstand abhängt.

Planungsalgorithmen sind meist unabhängig vom gewählten Referenzpunkt und Veröffentlichungen konzentrieren sich auf die Lösung des PSPACE-schweren Planungsproblems [104]. Daher wird der Einfluss des

Referenzpunktes auf die gefahrene Trajektorie vernachlässigt. Werling et al. [133] zum Beispiel verwenden  $\mathbf{wp}_{\text{SP}}$  als Referenzpunkt für ihren Frenet-Koordinatenplaner. Seccamonte et al. [115] verwendet  $\mathbf{wp}_{\text{HA}}$  für ihren modellprädiktiven Regelungsplaner, um den Seitenabstand des Fahrzeugs zu maximieren.

### 3.3 Bestimmung des Referenzpunktes

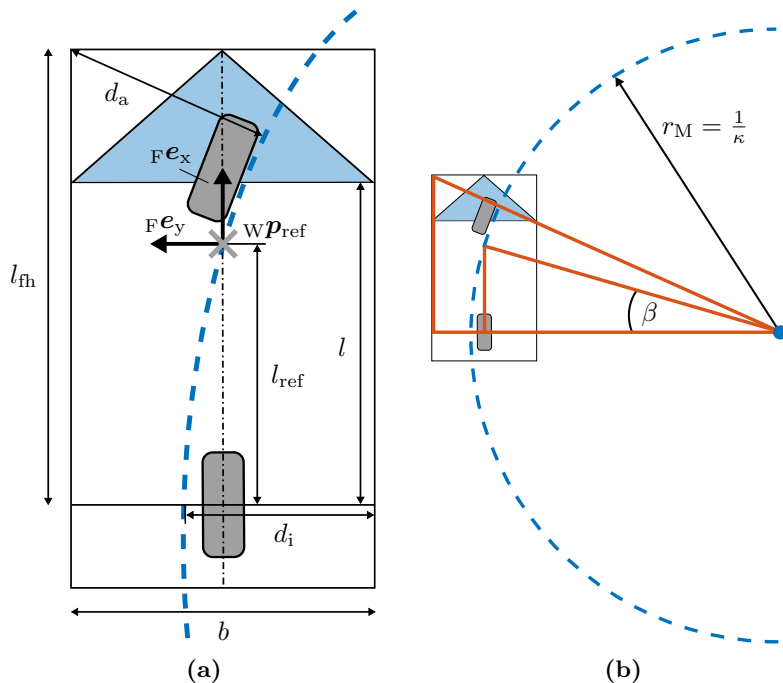
Zur Analyse des idealen Referenzpunktes wird das kinematische Einspurmodell aus Abschnitt 2.3 benutzt. Die Ausmaße des Fahrzeugs werden dabei als ein Rechteck in der  $xy$ -Ebene approximiert. Im Vergleich zu realen Fahrzeugen führt diese Vereinfachung zu einer Überschätzung der Spurbreite, da die Fahrzeugecken in der Regel abgerundet sind. Die im Folgenden durchgeführte Betrachtung beruht auf dem verwendeten Versuchsträger, einem Volkswagen Passat (B8). Die Fahrzeugparameter sind in Tabelle A.1 aufgeführt. Für die Trajektorienplanung wird die ideale Position des Referenzpunktes so definiert, dass die benötigte Spurbreite des Fahrzeugs rechts und links des geplanten Trajektorienpfades gleich groß ist. Dies wird in Abschnitt 3.3.1 zuerst für konstante Pfadkrümmungen hergeleitet, bevor Untersuchungen zu variablen Krümmungen in Abschnitt 3.3.2 präsentiert werden. Im Anschluss wird in Abschnitt 3.3.3 ein vereinfachtes Kreismodell zur Berücksichtigung der Fahrzeugausmaße aus den Überlegungen hergeleitet.

#### 3.3.1 Kurven mit konstanter Krümmung

Für Kurven mit konstanter Krümmung kann über geometrische Überlegungen der ideale Referenzpunkt hergeleitet werden. In Abb. 3.2 wird ein Überblick über die hier benötigten Parameter gegeben. Gesucht wird die Position des Referenzpunktes, welche durch die Distanz  $l_{\text{ref}}$  zwischen der Mitte der Hinterachse und dem Referenzpunkt definiert wird. Dabei soll der benötigte Platzbedarf zur Innenseite des Pfades  $d_i$  und zur Außenseite  $d_a$  gleich groß sein, es gilt

$$d_{\text{opt}} = d_i = d_a,$$

wobei sich der benötigte Platzbedarf des Fahrzeugs aus  $2d_{\text{opt}}$  ergibt. In dem in Abb. 3.2 gegebenen Beispiel einer Rechtskurve mit konstanter Krümmung  $\kappa$ , wird in der Innenseite des Fahrzeugs am meisten Platz an der Hinterachse benötigt.  $d_i$  ergibt sich aus der Distanz zwischen dem



**Abbildung 3.2:** In (a) sind die geometrischen Parameter des Fahrzeuges abgebildet: die Länge von der Hinterachse zur Fahrzeugfront  $l_{\text{fh}}$ , der Radstand  $l$  und die Fahrzeugbreite  $b$ . Für den idealen Referenzpunkt gilt  $d_{\text{opt}} = d_i = d_a$ , wobei  $d_i$  der Platzbedarf zur Innenseite des Fahrzeuges und  $d_a$  den zur Außenseite darstellt. In (b) sind die beiden Dreiecke (rote Linien) für die Herleitung des idealen Referenzpunktes sowie der Schwimmwinkel  $\beta$  dargestellt. Durch Anwendung des Satzes des Pythagoras ergeben sich Gleichungen (3.1) und (3.2) aus welchen  $d_{\text{opt}}$  und  $l_{\text{ref}}$  berechnet werden können. [100]

inneren Ende der Hinterachse und dem geplanten Pfad in lateraler Fahrzeugrichtung bzw. in radialer Richtung der Kreisbahn. Auf der Außenseite der Krümmung benötigt die vordere, äußere Fahrzeugecke den meisten Platz. Daher ist  $d_a$  durch die kürzeste Distanz zwischen der Fahrzeugecke und der Kreisbahn gegeben, welche ebenso senkrecht auf der Kreisbahn steht.

In Abb. 3.2b sind zwei Dreiecke eingezeichnet (rote Linien). Die Länge der Hypotenuse des kleineren Dreiecks entspricht dem Kurvenradius  $r_M$

und die zugehörigen Katheten werden durch die Länge  $l_{\text{ref}}$  bzw. dem Abstand vom Drehpunkt zur Mitte der Hinterachse beschrieben. Die Hypotenuse des zweiten Dreiecks entspricht dem Abstand vom Drehpunkt zur vorderen linken Ecke des Fahrzeugs und die Katheten werden durch die Distanz zwischen Hinterachse und Fahrzeugfront  $l_{\text{fh}}$  sowie durch den Abstand zwischen dem Drehpunkt und dem linken Ende der Hinterachse beschrieben. Mit Hilfe des Satzes des Pythagoras werden die beiden folgenden Gleichungen formuliert:

$$\left(\frac{1}{\kappa}\right)^2 = l_{\text{ref}}^2 + \left(\frac{1}{\kappa} - d_{\text{opt}} + \frac{b}{2}\right)^2, \quad (3.1)$$

$$\left(\frac{1}{\kappa} + d_{\text{opt}}\right)^2 = l_{\text{fh}}^2 + \left(\frac{1}{\kappa} - d_{\text{opt}} + b\right)^2, \quad (3.2)$$

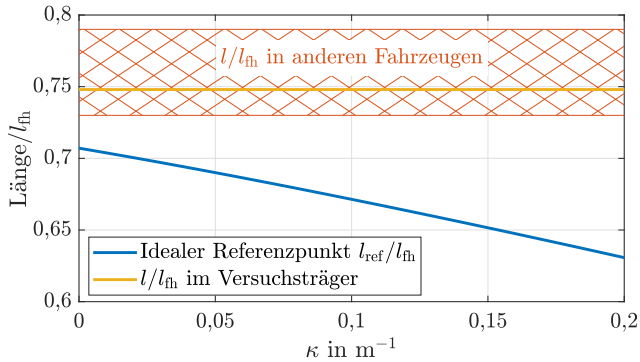
wobei  $b$  der Fahrzeugbreite entspricht. Das Lösen dieser Gleichungen führt zu:

$$l_{\text{ref}}^2 = \left(\frac{l_{\text{fh}}^2}{2 + \kappa b}\right) \left(1 - \frac{\kappa^2 l_{\text{fh}}^2}{4(2 + \kappa b)}\right),$$

$$2d_{\text{opt}} = \frac{2b + \kappa(b^2 + l_{\text{fh}}^2)}{2 + \kappa b}, \quad (3.3)$$

sodass die Position des Referenzpunktes und der benötigte Platzbedarf von der Pfadkrümmung  $\kappa$  sowie den Fahrzeugparametern abhängt. In Abb. 3.3 wird der ideale Referenzpunktabstand  $l_{\text{ref}}$  ins Verhältnis zu  $l_{\text{fh}}$  gesetzt und über  $\kappa$  aufgetragen, wobei  $\kappa = 0,2 \text{ m}^{-1}$  annähernd die kleinste fahrbare Kreiskurve darstellt. Bei dem verwendeten Versuchsfahrzeug liegt die Vorderachse bei  $l/l_{\text{fh}} = 0,75$ , sodass die ermittelten Idealpositionen des Referenzpunktes etwas hinter der Vorderachse liegen. Die Position der Vorderachse wird auch für 37 weitere Fahrzeuge verschiedener Marken und Fahrzeugklassen analysiert. Dort reichen die Vorderachspositionen von  $l/l_{\text{fh}} = 0,73$  bis  $0,79$ , so dass die bisherige Aussage über die idealen Referenzpunkte auch für diese Fahrzeuge gilt.

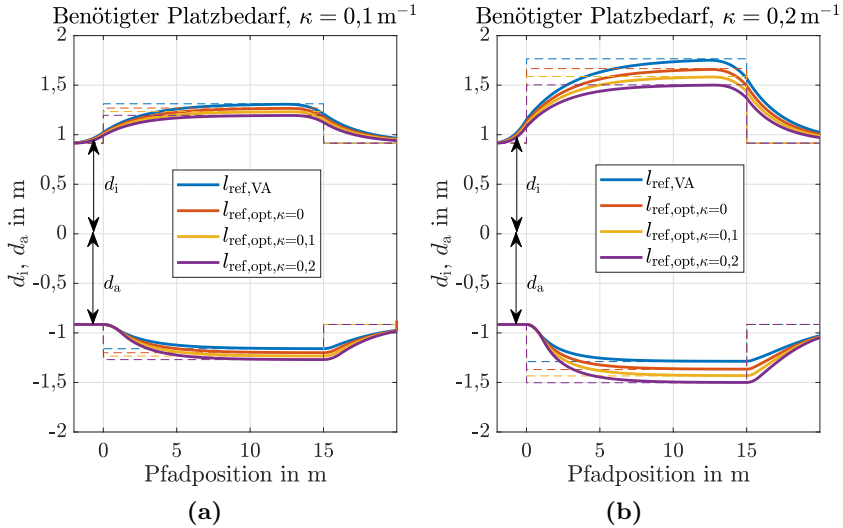
Wäre der Referenzpunkt in Abhängigkeit von  $\kappa$  definiert, würde er während der Fahrt auf Pfaden variabler Krümmung seine Längsposition ändern. Dies würde sowohl die Planung als auch die Regelung des Fahrzeugs verkomplizieren, sodass diese Idee nicht weiterverfolgt wird. Ziel ist es vielmehr, einen Referenzpunkt festzulegen, der für verschiedene Szenarien zu den geringsten Abweichungen im Platzbedarf des Fahrzeugs führt.



**Abbildung 3.3:** Die ideale Position des Referenzpunktes  $l_{\text{ref}}/l_{\text{fh}}$  für  $\kappa = \text{const.}$  liegt zwischen 0,63 und 0,71, abhängig von  $\kappa$ . In dem vorhandenen Versuchsträger liegt der ideale Referenzpunkt zwischen 0,15 m (für  $\kappa = 0$ ) und 0,44 m (für  $\kappa = 0,2 \text{ m}^{-1}$ ) hinter der Vorderachse, welche bei  $l/l_{\text{fh}} = 0,75$  liegt. Bei 37 anderen untersuchten Fahrzeugen verschiedener Marken und Fahrzeugkategorien liegt die Frontachse zwischen  $l/l_{\text{fh}} = 0,73$  und 0,79. [100]

### 3.3.2 Kurven mit variabler Krümmung

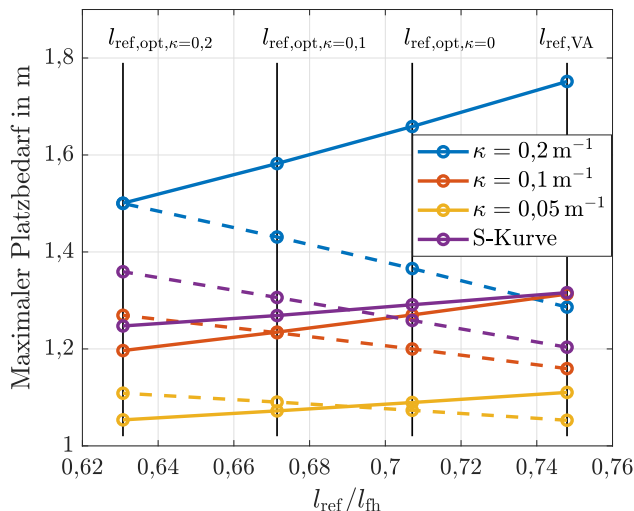
Nachdem im vorherigen Abschnitt nur Szenarien mit konstanter Krümmung berücksichtigt wurden, werden nun Szenarien mit variabler Krümmung untersucht. Abb. 3.4 zeigt die Abweichungen des benötigten Platzbedarfs, wenn der Referenzpunkt nicht ideal gewählt ist. Um die Abweichungen bei unterschiedlichen Krümmungen zu vergleichen, werden  $\kappa = 0,1 \text{ m}^{-1}$  und  $\kappa = 0,2 \text{ m}^{-1}$  näherungsweise bis zur stationären Kurvenfahrt simuliert. In Abb. 3.4a folgt das Fahrzeug einer Kreisbahn mit einer Krümmung von  $\kappa = 0,1 \text{ m}^{-1}$ . Nach 15 m konstanter Krümmung verläuft die Bahn gerade, sodass  $\kappa$  wieder auf null springt. Abb. 3.4b zeigt dasselbe Manöver mit  $\kappa = 0,2 \text{ m}^{-1}$ . Beide Szenarien werden mit vier verschiedenen Referenzpunkten simuliert: den idealen Referenzpunkten für  $\kappa = \{0,2; 0,1; 0\} \text{ m}^{-1}$  und dem Referenzpunkt auf der Vorderachse. Der Vergleich beider Diagramme zeigt, dass die Abweichungen vom gleichverteilten Platzbedarf zwischen dem Kurveninneren und -äußeren, die durch die suboptimale Wahl von Referenzpunkten verursacht werden, mit steigender Krümmung zunehmen. Der Einfluss der Wahl des Referenzpunktes auf die Abweichung nimmt zu kleineren Krümmungen hin ab, wie am maximalen benötigten Platzbedarfs in Abb. 3.5 zu erkennen ist. Hier zeigen die



**Abbildung 3.4:** Benötigter Platzbedarf ( $d_i + d_a$ ) für einen Sprung in  $\kappa$  von null auf  $0,1 \text{ m}^{-1}$  in (a) und auf  $0,2 \text{ m}^{-1}$  in (b).  $\kappa$  springt zurück auf null nach einer Pfadlänge von 15 m. Die gestrichelten Linien zeigen den benötigten Platzbedarf für stationäre Kurvenfahrten mit dem gegebenen  $\kappa$  an. Die Wahl eines nicht idealen Referenzpunktes hat für kleinere  $\kappa$ , wie in (a), weniger Einfluss auf die ideale Platzverteilung zwischen Kurveninneren und -äußeren als für größere  $\kappa$ , wie in (b). Die Annahme einer stationären Kurvenfahrt überschätzt den tatsächlichen Platzbedarf nach dem ersten Sprung von  $\kappa$  bis zum Erreichen des stationären Zustands, unterschätzt aber auch den Platzbedarf, z. B. nach der Rückkehr von  $\kappa$  auf null. [100]

durchgezogenen Linien den benötigten Platzbedarf auf der Innenseite der Kurve und die gestrichelten Linien den Platzbedarf auf der Außenseite der Kurve. Für  $\kappa = 0,05 \text{ m}^{-1}$  beträgt der größte Unterschied im Platzbedarf zwischen Kurveninnerem und -äußerem  $0,06 \text{ m}$  und bei Geradeausfahrt ist der Einfluss des Referenzpunktes auf den Platzbedarf überhaupt nicht mehr vorhanden. Im Vergleich dazu führen nicht ideale Referenzpunkte bei hohen Krümmungen wie  $\kappa = 0,2 \text{ m}^{-1}$  zu sehr ungleich verteiltem Platzbedarf mit Differenzen zwischen Kurveninnerem und -äußerem von bis zu  $0,48 \text{ m}$ . Einerseits ist dies ein Grund, den Referenzpunkt auszuwählen, der ideal für die höchste fahrbare Krümmung ist. Andererseits würden in diesem Fall in jeder Situation, in der die Krümmung kleiner als das Maximum





**Abbildung 3.5:** Maximaler Platzbedarf für unterschiedliche Referenzpunkte. Die blauen, roten und violetten Linien entsprechen Manövern mit konstanter Krümmung und stationärer Kurvenfahrt. Die durch die violetten Linien dargestellte S-Kurve besteht aus zwei Abschnitten von je 3 m Länge, wobei die Krümmung im ersten Abschnitt  $\kappa = 0,2 \text{ m}^{-1}$  beträgt und dann sofort auf  $\kappa = -0,2 \text{ m}^{-1}$  für den zweiten Abschnitt springt. Der Platzbedarf auf der Innenseite der Kurve (des ersten Abschnitts) wird durch die durchgezogenen Linien dargestellt, der Platzbedarf auf der Außenseite der Kurve durch die gestrichelten Linien. [100]

ist, Abweichungen auftreten.

Da sich die Krümmung im realen Fahrbetrieb häufig ändert, wird im Folgenden der tatsächliche Platzbedarf in nicht-stationären Situationen genauer betrachtet. Abb. 3.4 zeigt den berechneten Platzbedarf nach dem kinematischen Einspurmodell (durchgezogene Linien) und den Platzbedarf unter der bisherigen Annahme einer konstanten Krümmung (gestrichelte Linien). Es ist zu erkennen, dass die Annahme von  $\kappa = \text{const.}$  den Platzbedarf nach  $\kappa$ -Sprüngen von null überschätzt, aber auch manchmal unterschätzt. Letzteres ist der kritischere Fall und tritt in Abb. 3.4 auf, bevor die Krümmung von null abweicht und nachdem die Krümmung auf null zurückgeht. Der Grund für die Unterschätzung des Platzbedarfs vor

der Position null liegt darin, dass die eingezeichneten Linien die innere und äußere benötigte Fahrbahnbreite entlang des Referenzpfades darstellen. Beginnt der Referenzpunkt die Kurvenfahrt an der Bahnposition null, wird die seitliche Position der Hinterachse bereits vor dem Passieren dieser Bahnposition beeinflusst, was zu einem höheren Platzbedarf an dieser Stelle führt.

Im Folgenden werden drei verschiedene Manöver untersucht, bei welchen es sich nach [100] um die kritischsten oder kurvenreichsten handelt:

- Wendemanöver:  $180^\circ$  Kurve mit  $\kappa = 0,2 \text{ m}^{-1}$
- Rechtskurve:  $90^\circ$  Kurve mit  $\kappa = -0,1 \text{ m}^{-1}$
- S-Kurve:  $\kappa$  springt von  $0,2$  auf  $-0,2 \text{ m}^{-1}$ ,  $34,4^\circ$  jeweils pro Abschnitt (Länge pro Abschnitt:  $3 \text{ m}$ ), z. B. wenn das Ego-Fahrzeug hinter einem geparkten Fahrzeug steht und vorwärts ausparkt.

In der bereits vorgestellten Abb. 3.5 entsprechen die blauen Kurven dem Wendemanöver. Bei diesem Manöver fährt das Fahrzeug eine annähernd stationäre Kurve mit maximaler Krümmung und der kleinste maximale Platzbedarf für beide Seiten tritt auf, wenn der Referenzpunkt ideal für eben dieses  $\kappa = 0,2 \text{ m}^{-1}$  bei  $l/l_{\text{fh}} = 0,63$  positioniert ist. Je näher der Referenzpunkt an der Fahrzeugfront liegt, desto mehr Platz wird auf der Innenseite des Wendemanövers benötigt und desto weniger Platz auf der Außenseite.

Die Rechtskurve kann auch als stationäre Kurvenfahrt am Ende des Manövers betrachtet werden. Die Weglänge dieses Manövers beträgt  $15,7 \text{ m}$  und kommt damit dem Szenario in Abb. 3.4a nahe, bei dem der Platzbedarf sein Maximum bereits bei etwa  $10 \text{ m}$  erreicht. Daher entsprechen die roten Kurven in Abb. 3.5 dem Rechtsabbiegemanöver. Da die Krümmung bei der Rechtskurve kleiner ist als beim Wendemanöver, ist auch der Einfluss des Referenzpunktes geringer.

Bei der S-Kurve ist der Einfluss des Referenzpunktes deutlich geringer als beim Wendemanöver. Der größte Unterschied zwischen dem maximalen Platzbedarf auf beiden Fahrzeugseiten beträgt bei der S-Kurve  $0,11 \text{ m}$ . Die S-Kurve benötigt auch weniger seitlichen Platz als das Wendemanöver, obwohl  $|\kappa|$  gleich groß ist. Der Grund dafür ist, dass die S-Kurve diese großen Krümmungen nur auf einer kurzen Strecke von  $3 \text{ m}$  aufweist und somit nicht zu einer stationären Kurvenfahrt führt, bei der für ein bestimmtes  $|\kappa|$  der maximale Platzbedarf benötigt wird. Dies zeigt sich auch in Abb. 3.4b. Je länger das Fahrzeug auf der konstant gekrümmten Strecke

fährt, desto größer wird der Platzbedarf, bis er in die Sättigung läuft, was dann einer stationären Kurvenfahrt mit  $\kappa = \text{const}$  entspricht.

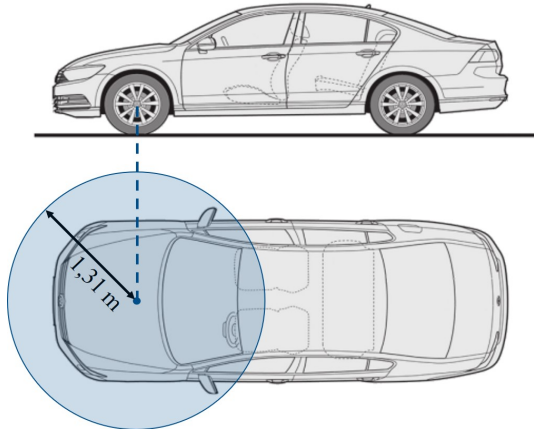
### 3.3.3 Kreismodell

Im vorherigen Abschnitt wird das Einspurmodell benutzt, um den Platzbedarf des Fahrzeugs zu berechnen. Der Nachteil hierbei ist, dass das Modell immer Informationen über die Fahrtrichtung des Fahrzeugs benötigt und somit den Kontext der umliegenden Wegpunkte voraussetzt. Zwar kann der Schwimmwinkel  $\beta$  des Einspurmodells durch das Verzögerungsglied  $\text{PT1}(\kappa l_{\text{ref}}, l_{\text{ref}})$  modelliert werden (vgl. Gleichungen (2.5) bis (2.9) in Abschnitt 2.3), allerdings wäre es einfacher, ein Kreismodell zu verwenden. Bei diesem wird das Fahrzeug als Kreis modelliert, welcher den benötigten Platzbedarf des Fahrzeugs approximiert. Jeder Wegpunkt wird separat und unabhängig von anderen Wegpunkten betrachtet, da die Fahrtrichtung durch die Rotationsinvarianz des Kreises nicht relevant ist. Dabei liegt der Mittelpunkt des Kreises auf dem Referenzpunkt. Nur für diesen Fall wird ausschließlich die Position des Fahrzeugs benötigt, die Orientierung kann vernachlässigt werden. Dies ist besonders nützlich bei der Verwendung von suchbasierten Planungsalgorithmen (z. B.  $A^*$  [48], RRT [72]), bei denen weder die Orientierung noch ein kinematisches Modell des Fahrzeugs berücksichtigt wird. Im Folgenden wird die Position des Referenzpunktes und die Größe des Kreises diskutiert.

Wenn der ideale Referenzpunkt für  $\kappa = 0,2 \text{ m}^{-1}$  als Referenzpunkt im vorgestellten Kreismodell gewählt wird, muss der Radius der Scheibe mindestens  $r_{\text{ego}} = 1,5 \text{ m}$  betragen, um alle vorgestellten, kritischsten Manöver abzudecken (vgl. Abb. 3.5). Setzt man den Referenzpunkt auf die Vorderachse, sollte dieser Radius mindestens  $r_{\text{ego}} = 1,76 \text{ m}$  betragen. Für dieses vereinfachende Kreismodell wird jedoch angenommen, dass das Wendemanöver irrelevant ist, weil hohe Krümmungen nur dann auftreten, wenn nicht genügend Platz für einen Pfad mit geringerer Krümmung vorhanden ist. Wenn der Platz nicht ausreicht, ist eine hohe Präzision und damit auch eine komplexere Modellierung erforderlich (z. B. für Parkmanöver). Dies stellt kein Problem dar, da ein geringer Platzbedarf auch zu einer niedrigen Geschwindigkeit des Fahrzeugs führt, sodass genügend Rechenzeit für komplexere Ansätze zur Verfügung stehen sollte. Demnach werden die blauen Linien in Abb. 3.5 vernachlässigt, sodass der Referenzpunkt mit dem kleinsten maximalen Platzbedarfs etwas hinter der Vorderachse bei etwa  $l/l_{\text{fh}} = 0,735$  liegt. Der Einfachheit halber wird deshalb der Referenzpunkt genau auf der Vorderachse platziert, was erfordert, dass der

Radius des Kreises mindestens  $r_{\text{ego}} = 1,31 \text{ m}$  beträgt (siehe maximalen Platzbedarf für  $l_{\text{ref,VA}}$  in Abb. 3.5 unter Vernachlässigung der blauen Linien). Das bedeutet, dass eine Fahrspurbreite von  $1,31 \text{ m}$  auf jeder Seite der Trajektorie immer ausreicht, um ein Hindernis oder eine Verengung zu passieren.

Die Vereinfachungen zum Kreismodell gehen mit Einschränkungen einher, die im Folgenden dargestellt werden. Da der Radius des Kreises größer ist als die tatsächliche Fahrzeugbreite (vgl. Abb. 3.6), ist der angenäherte seitliche Platzbedarf größer als der tatsächliche. Daher bewirkt die Verwendung dieser Platzbedarfsabschätzung ein konservatives Verhalten bei der Platznutzung, was dazu führen kann, dass z. B. eine schmale Passage nicht passiert werden kann, obwohl es in der Realität möglich wäre. Darüber hinaus wird von einem normal großen PKW ausgegangen, wie in Abb. 3.6 gezeigt. Ist die Länge  $l_{\text{th}}$  (vgl. Abb. 3.2) zu groß, z. B. bei der Betrachtung eines Busses, wird der Radius  $r_{\text{ego}}$  im Vergleich zur Fahrzeugbreite  $b$  zu groß und damit ist dieser Ansatz nicht mehr sinnvoll. Außerdem begrenzt der gewählte Kreisradius die zulässige Bewegung des Fahrzeugs, die mit dieser Platzbedarfsabschätzung sicher gefahren werden kann. Die maximale sichere Krümmung ist direkt vom Radius  $r_{\text{ego}}$  abhängig (vgl. Gleichung (3.3)). Die oben getroffene Wahl von  $r_{\text{ego}} = 1,31 \text{ m}$  führt zu



**Abbildung 3.6:** Kreis mit Radius  $r_{\text{ego}} = 1,31 \text{ m}$  mit Mittelpunkt auf der Vorderachse des VW Passat VIII (B8) (Bild basiert auf [128]). Der Kreis umfasst den benötigten Platzbedarf in Kurven mit  $\kappa \leq 0,1 \text{ m}^{-1}$  und beinhaltet die Fahrzeugfront. [100]

einer maximal fahrbaren, konstanten Krümmung von  $\kappa = 0,1 \text{ m}^{-1}$  (vgl. Abb. 3.5). Unter Berücksichtigung nicht-stationärer Kurven wäre es auch möglich, höhere Krümmungen zu fahren, wie bei der S-Kurve in Abb. 3.5 gezeigt. Da es schwierig ist, festzustellen, ob das betrachtete Manöver außerhalb der spezifizierten Voraussetzungen liegt, empfiehlt es sich, für diese Situationen einen komplexeren Ansatz zu verwenden. Dies ist auch der Fall, wenn eine hohe Platzausnutzung Voraussetzung ist, wie beim automatisierten Parken. Für diese Fälle wird auf Ansätze mit höherer Genauigkeit wie Heinrich et al. [50] verwiesen, bei dem der Platzbedarf anhand eines zweiten Kreises geschätzt wird.

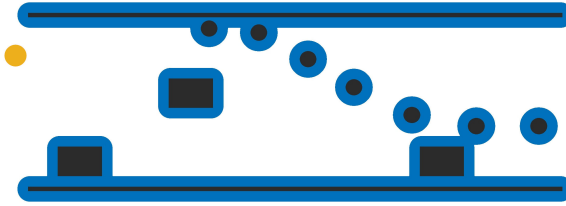
## 3.4 Weiterführende Betrachtungen und Anwendungen

Nachdem die theoretischen Betrachtungen zur Wahl des Referenzpunktes in der Planung vorgestellt wurden, werden in diesem Abschnitt mögliche Anwendungsfälle vorgestellt.

### 3.4.1 Kollisionsüberprüfung

Das in Abschnitt 3.3.3 vorgestellte Kreismodell zur Abschätzung des seitlichen Platzbedarfs lässt sich ebenso für eine konservative, aber schnelle Kollisionsüberprüfung verwenden. Hier ist gerade die Schnelligkeit des Ansatzes wichtig, da die Kollisionsüberprüfung für jeden Planungsschritt durchgeführt werden muss und dabei einen erheblichen Anteil der Rechenzeit in Anspruch nimmt [118, 123]. Während in der Literatur optimierte Ansätze wie [50, 143] auf die Überprüfung durch mehrere Kreise setzen, ist die Überprüfung bei dem hier vorgestellten Ansatz durch nur einen Kreis ohne Berücksichtigung der Orientierung des Fahrzeugs möglich.

Durch die Wahl des Referenzpunktes auf der Vorderachse wird die Fahrzeugfront vollständig durch den Kreis abgedeckt (siehe Abb. 3.6). Somit kann dieser Kreis nicht nur für konservative seitliche Platzbedarfsabschätzungen, sondern auch für Kollisionsprüfungen im vorderen Bereich des Fahrzeugs, z. B. beim Anhalten, verwendet werden. Hierbei müssen die in Abschnitt 3.3.3 getroffenen Einschränkungen berücksichtigt und durch die folgende ergänzt werden. Da sich die Kollisionsscheibe im vorderen Teil des Fahrzeugs befindet, ist diese Prüfung nur bei Vorwärtsfahrten gültig, nicht jedoch bei Rückwärtsfahrten.



**Abbildung 3.7:** Einfache Berechnung von Kollisionspunkten mittels einer 2D-Faltung. Die Belegungskarte der Umgebung (schwarz) wird mit dem kreisförmigen Kernel (gelb, an der linken Bildseite dargestellt) gefaltet, sodass der Kollisionsbereich (blau) entsteht. [100]

In Abb. 3.7 wird die schnelle Kollisionsüberprüfung kurz präsentiert. Liegt das Umgebungsmodell (schwarz) in einer Grid-basierten Darstellung vor, kann die 2D-Faltung<sup>1</sup> (blau) dieser berechnet werden. Der Kernel der Faltung ist dabei der Kreis des Kreismodells (gelb). Die Kollisionsüberprüfung kann anhand des Ergebnisses der 2D-Faltung direkt durchgeführt werden. Liegt die Position des Fahrzeugs im blauen Bereich wird von einer Kollision ausgegangen. Dies ist durch die getroffenen Einschränkungen eine korrekte, aber konservative Abschätzung.

### 3.4.2 Fahrspurzuordnung

Ein weiterer Anwendungsfall ist die Zuordnung der Fahrzeugposition auf eine Fahrspur. Das Ziel ist eine schnelle Aussage darüber, auf welcher Fahrspur (z. B. Gegenfahrspur, eigene Fahrspur, außerhalb des fahrbaren Bereiches, etc.) sich das Fahrzeug befindet. Für die schnelle Zuordnung wird ein vorliegendes Straßenmodell in eine Grid-basierte Darstellung umgewandelt. Ähnlich zur Fahrbarkeitskarte aus [68] sowie der Kollisionsüberprüfung kann durch eine einzelne Speicherabfrage die aktuelle Fahrspur des Fahrzeugs bestimmen werden.

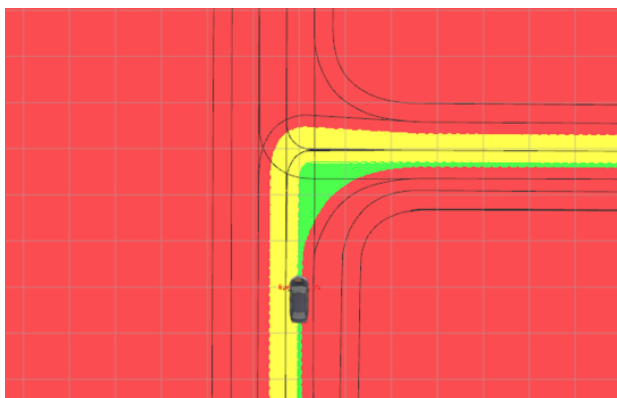
Dabei werden, im Gegensatz zu [68], die vorangegangenen Überlegungen bzgl. des seitlichen Platzbedarfs miteinbezogen. Bei der Umwandlung des Straßenmodells in die Grid-basierte Darstellung wird der auf der Straße fahrbare Bereich an den Straßenrändern um den Radius des Kreismodells

<sup>1</sup>Alternativ erreicht die morphologische Operation der Dilatation dasselbe Ergebnis für eine Kollisionsüberprüfung.

verringert. Gleiches gilt für die eigene Fahrspur am Übergang zwischen der eigenen und der Gegenseite. Die Gegenseite wird dann bis zur eigenen Fahrspur aufgefüllt. Dies gleicht der Annahme, dass das Fahrzeug sich auf der Gegenseite befindet, sobald sich ein kleiner Teil des Fahrzeugs in dieser Spur befindet. Somit kann durch eine einzelne Überprüfung festgestellt werden, ob sich das komplette Fahrzeug in der eigenen Fahrspur befindet oder nicht. Im Gegensatz dazu wird durch ein Vorgehen wie in [68] der Platzbedarf des Fahrzeugs vernachlässigt und nur überprüft, wo sich der Referenzpunkt (z. B. Mittelpunkt der Vorderachse) auf der Straße befindet. Ein Beispiel des beschriebenen Vorgehens ist in Abb. 3.8 dargestellt.

### 3.5 Zusammenfassung

In diesem Kapitel wird der Einfluss des Referenzpunktes für vorderachsengelenkte Fahrzeuge auf den benötigten seitlichen Platzbedarf untersucht. Dabei wird die ideale Position des Referenzpunktes so definiert, dass der seitliche Platzbedarf auf beiden Fahrzeugseiten gleich groß ist. Anhand von geometrischen Überlegungen wird für stationäre Kurvenfahrten die ideale Position hergeleitet, welche von der Krümmung der gefahrenen Kurve



**Abbildung 3.8:** Einfache Zuordnung der Fahrspur mittels des Kreismodells. Die eigene Fahrspur (grün) wird um den Radius des Kreismodells zu beiden Seiten verkleinert, während die Gegenfahrspur (gelb) am Straßenrand verkleinert und bis zur eigenen Fahrspur aufgefüllt wird. In Rot wird der nicht befahrbare Bereich dargestellt.

abhängt. Die Analyse nicht stationärer Kurvenfahrten zeigt einerseits, dass sie hinsichtlich der erforderlichen Fahrbahnbreite weniger anspruchsvoll ist als die stationäre Kurvenfahrt. Andererseits zeigt diese, dass es in dynamischen Szenarien keinen stationären idealen Referenzpunkt gibt. Nach der Analyse verschiedener kritischer Manöver kommt der Autor jedoch zu dem Schluss, dass ein Referenzpunkt auf oder leicht hinter der Vorderachse PKW-ähnlicher Fahrzeuge eine geeignete Wahl ist. Diese Position des Referenzpunktes führt zu einem vereinfachten Kreismodell, welches z. B. zu Kollisions- oder Fahrspurüberprüfungen benutzt werden kann. Die Einfachheit ergibt sich aus der Annäherung der Ausdehnung des Fahrzeugs durch nur einen Kreis. Somit ist die Orientierung des Fahrzeugs entlang der geplanten Trajektorie nicht erforderlich. Zwar ist die Idee eines solchen Kreismodells nicht neu, allerdings wird durch die vorher vorgestellte Platzbedarfsanalyse der Radius sowie die Position des Kreises begründet. Hierdurch kann der Radius deutlich kleiner gewählt werden als ein Kreis, welcher durch die vier Fahrzeugecken verläuft.



---

## 4 Analyse menschlichen Fahrverhaltens

Die Analyse des Fahrverhaltens menschlicher Autofahrer kann für die Entwicklung verschiedener Anwendungen von Nutzen sein. Beispiele solcher Anwendungen sind unter anderem die Prädiktion oder Planung von Trajektorien, die Erkennung von Fahrfehlern, die Klassifizierung des Fahrverhaltens oder die Modellierung von Verkehrssimulationen. In diesem Kapitel wird menschliches Fahrverhalten modelliert, um Vereinfachungen für die Trajektorienplanung zu finden. Hierzu wird zuerst eine thematische Einleitung sowie der Stand der Technik präsentiert. Im Anschluss wird für ein zeitdiskretes Fahrzeugmodell die Abtastzeit analysiert, mit welcher menschliche Trajektorien abgebildet werden können. Durch eine Erhöhung der Abtastzeit kann der Aktionsraum eines Trajektorienplanungsproblems sowie die Planungskomplexität reduziert werden. Auf Basis des zeitdiskreten Fahrzeugmodells wird ein probabilistisches Verhaltensmodell vorgestellt, welches mögliche Fahraktionen abbildet, bevor das Kapitel mit einer Zusammenfassung abschließt. Um ein einfacheres Verhaltensmodell zu ermöglichen, wird eine Transformation der Lenkvariablen vorgestellt, welche auch zur Diskretisierung des Planungsraumes verwendet werden kann (vgl. Abschnitt 5.3.1). Die hier vorgestellten Aspekte wurden vom Autor in [141] veröffentlicht und werden hier erneut dargestellt.

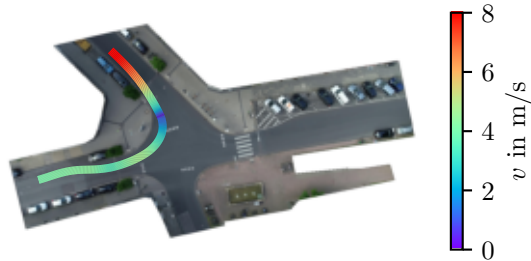
### 4.1 Einleitung und Motivation

Intelligente Fahrzeuge sind eine Schlüsselkomponente für den Verkehr der Zukunft. Das Verständnis und die Modellierung des menschlichen Fahrverhaltens können der Forschung und der Automobilindustrie dabei helfen, solche Fahrzeuge in verschiedenen Aspekten zu verbessern. Für Anwendungen des autonomen Fahrens oder fortgeschrittene Fahrerassistenzsysteme hilft das Wissen über das menschliche Fahrverhalten bei der Entwicklung intelligenter Systeme oder entsprechender Werkzeuge (z. B. Verkehrssimulatoren).

Abb. 4.1 zeigt eine beispielhafte Trajektorie eines menschlichen Fahrers, der an einer innerstädtischen Kreuzung rechts abbiegt. Ziel der Analyse menschlichen Fahrverhaltens ist es, aus dieser sowie Trajektorien anderer Verkehrsteilnehmer Informationen über das allgemeine Fahrverhalten zu gewinnen. Hieraus entsteht ein Modell, welches das Verhalten unterschiedlicher Fahrer abbildet, ein Modell eines durchschnittlichen Fahrers. Da das Anwendungsgebiet dieser Arbeit die Trajektorienplanung ist, liegt der Fokus des Modells darauf, eine einfachere Planung zu ermöglichen. Trotzdem können die hier vorgestellten Ergebnisse auch für andere Anwendungen genutzt werden.

Die Berechnung einer sicheren, komfortablen, ökonomischen und fahrbaren Trajektorie für ein autonomes Fahrzeug ist eine Schlüsselkomponente, um autonomes Fahren in der Zukunft zu etablieren. Insbesondere im urbanen Raum ist es für die Aufgabe der Trajektorienplanung wichtig, vorausschauend zu fahren, andere Verkehrsteilnehmer zu antizipieren und menschenähnlich zu planen [27]. Um menschenähnliche Trajektorien zu planen, können die Kostenfunktionen des Planungsalgorithmus aus menschlichen Fahrdaten gelernt werden [49, 106]. Da die komplexe Aufgabe der Trajektorienplanung, insbesondere in Echtzeit, nicht einfach zu lösen ist, besteht die Zielsetzung dieses Kapitels darin, den Aktionsraum des autonomen Fahrzeugs während der Planungsphase einzuschränken, bevor die Kostenfunktion optimiert wird. Dazu können sowohl die physikalischen Grenzen des Fahrzeugs als auch das durchschnittliche menschliche Fahrverhalten genutzt werden.

In diesem Kapitel werden die physikalischen und menschlichen Aspekte für die Planungsbeschränkungen kombiniert. Für den physikalischen Teil wird das kinematische Einspurmodell verwendet (vgl. Abschnitt 2.3), um die nicht-holonomen longitudinalen und lateralen Bewegungen eines PKW-ähnlichen Fahrzeugs darzustellen. Durch Hinzufügen eines probabilistischen Verhaltensmodells können die Korrelationen zwischen menschlichen Handlungen und den Zuständen des Fahrzeugs berücksichtigt werden. Auf diese Weise kann das menschliche Fahrverhalten durch Begrenzung der Eingänge des kinematischen Modells berücksichtigt werden. Um ein umgebungsunabhängiges Modell zu erhalten, werden nur die Trajektorien jedes Fahrzeugs separat betrachtet, ohne die umgebende Straßenführung oder Verkehrsteilnehmer zu berücksichtigen. Da in städtischen Umgebungen die Anzahl der möglichen Situationen und Manöver so hoch ist, dass es nicht möglich ist, sie alle in Bezug auf die Umgebung zu definieren und zu modellieren, entscheidet sich der Autor für die Extraktion eines umgebungsunabhängigen Modells. Außerdem sollte das Modell nicht re-



**Abbildung 4.1:** Beispielhafte Trajektorie eines menschlichen Fahrers an einer innerstädtischen Kreuzung. Die Trajektorie stammt aus dem inD-Datensatz [12] (Aufzeichnung 28, Trajektorie 64). [141]

chenintensiv sein, da es häufig ausgeführt werden muss, um die komplexe Aufgabe der Planung zu erleichtern. Das extrahierte Modell kann dann z. B. in einem stochastischen prädiktiven Modellregler verwendet werden.

## 4.2 Stand der Technik

Das menschliche Fahrverhalten variiert und hängt von zahlreichen Faktoren ab, wie dem Können und der Stimmung des Fahrers, dem gefahrenen Fahrzeug, der Kultur, dem aktuellen Verkehrsfluss und der Situation, den Wetterbedingungen usw. [81]. Um Sicherheits- sowie Fahrerassistenzsysteme von Fahrzeugen zu verbessern, konzentrieren sich viele Forschungsarbeiten auf die Klassifizierung von diskreten Bewegungsmodellen [63, 109] und Fahrstilen (z. B. ruhig/aggressiv) [18, 79, 92, 132]. Dieses Wissen kann z. B. genutzt werden, um riskantes Verhalten des aktuellen Fahrers zu erkennen [37, 138]. Da solche diskreten Klassifizierungen für die Trajektorienplanung weniger interessant sind, liegt der Schwerpunkt im Folgenden auf kontinuierlichen Verhaltensmodellen.

Im Allgemeinen wird das Beschleunigungs-/Verzögerungsverhalten (Längsbewegung) von Menschen gefahrenen Fahrzeugen häufig analysiert. Bokare und Maurya [14] modellieren das Beschleunigungs-/Verzögerungsverhalten mehrerer Fahrzeugklassen in Indien, indem sie GPS-Sensoren an den Fahrzeugen anbringen. Durch Experimente auf geraden Straßen finden sie heraus, dass die Beschleunigung von LKW, Diesel- und Benzinfahrzeugen exponentiell von der Geschwindigkeit abhängig

ist, während es bei der Verzögerung keine statistische Signifikanz zwischen einem exponentiellen und einem polynomialen Modell gibt. Andere Arbeiten wie Miyajima et al. [88] modellieren das Beschleunigungs-/Verzögerungsverhalten mit einem Gaußschen Mischmodell für Gas- und Bremspedalbefehle, um den Fahrer des Fahrzeugs zu identifizieren. Wagner et al. [129] analysiert die Längsbewegung von PKW-Folgeszenarien. Unter Verwendung des statistischen Maßes des maximalen Informationsgehalts [105] finden sie die dominante Korrelation zwischen der Geschwindigkeit und dem Abstand zum vorausfahrenden Fahrzeug.

In den oben genannten Arbeiten wird nur das Längsverhalten von Fahrzeugen analysiert. Allerdings können ebenfalls die Querbewegungen durch die Fahrzeuglenkung berücksichtigt werden. Miyajima et al. [89] teilen die Handlungen des Fahrers anhand der Beschleunigung, der Verzögerung und der Lenkung des Fahrzeugs in Risikogruppen ein. Durch das Hinzuziehen von Expertenwissen werden drei Modelle (Beschleunigung, Verzögerung und Lenkung) abgeleitet, um riskantes Fahrverhalten zu kategorisieren. Yokoyama und Toyoda [138] analysieren in einer Langzeitstudie die Korrelation zwischen Unfällen und Fahrverhalten. In ihrer Arbeit werden erneut Beschleunigung, Verzögerung und Lenkung berücksichtigt, um das Ergebnis ihres Modells zu verbessern. Bei allen oben genannten Arbeiten ist ein anwendungsspezifischer Klassifikator (d.h. Erkennung von riskantem Verhalten) der Ausgang des abgeleiteten Modells.

Im Bereich der Trajektorienplanung sind datengetriebene Modelle ebenfalls eine gängige Wahl. Hier kann der Modellausgang eine Kostenfunktion [106] oder eine auf Unsicherheit basierende Trajektorie [120] sein. Schulz et al. [111] leitet mit Hilfe von Deep-Learning-Methoden ein interaktionsfähiges, probabilistisches Markov-Verhaltensmodell ab, um Gaußverteilungen der nächsten Beschleunigung und des Lenkwinkels des Ego-Fahrzeugs in Abhängigkeit des aktuellen Fahrzeugzustands sowie der derzeitigen Verkehrssituation zu berechnen. Die Idee hinter diesem Modell ähnelt dem hier präsentierten Ansatz, bezieht aber Umweltdaten wie die Straßenführung und den Zustand anderer Verkehrsteilnehmer mit ein. Im Gegensatz zu Schulz et al. konzentriert sich das hier vorgestellte, umgebungsfreie Modell eines durchschnittlichen Fahrers nur auf die internen Zustände des Fahrzeugs, verwendet die Beschleunigung und die Lenkgeschwindigkeit als Systemeingang und kann aufgrund seiner dadurch erlangten Einfachheit als Basis für weitere, komplexere Ansätze verwendet werden. Da keine Deep-Learning-Methoden zugrunde liegen, werden nur reine Messdaten zur Modellbestimmung der longitudinalen sowie lateralen Bewegung verwendet. Das Einbinden künstlich erzeugter Daten wie in

[111] entfällt. Letztendlich kann das umgebungsfreie Modell direkt verwendet werden. Für spezifische Anwendungen wie eine Sicherheitsüberprüfung geplanter Trajektorien bietet sich ebenso die Extraktion der Aktionsgrenzen an.

## 4.3 Analyse Abtastzeit

Die meisten Planungsalgorithmen berechnen Trajektorien mit zeitdiskreten Eingängen. Das bedeutet, dass sich der Eingang der kontinuierlichen Bewegungsmodelle aus Abschnitt 2.3 erst nach einer bestimmten Zeit, der Eingangsabtastzeit<sup>1</sup>  $T_{\text{in}}$ , ändern kann und ansonsten konstant gehalten wird. Es stellt sich nun die Frage, welche Eingangsabtastzeit verwendet werden kann, um menschlich gefahrene Trajektorien wie die in Abb. 4.1 abzubilden.

Durch die Verwendung einer größeren Eingangsabtastzeit kann das Planungsproblem vereinfacht werden, da die nächste Aktion weniger oft berechnet werden muss. So können bei gleichem Rechenaufwand längere Trajektorien berechnet werden. Dies kann zu stetigeren Trajektorien mit weniger hochfrequenten Schwingungen führen, insbesondere wenn Zustandsableitungen als Eingangsgröße verwendet werden. Wenn jedoch die Abtastzeit der Eingänge zu groß ist, wird die resultierende Trajektorie zu unflexibel, um alle erforderlichen Verkehrsszenarien zu bewältigen. Daher muss ein Kompromiss zwischen der Einfachheit des Planungsproblems und der Flexibilität der Ergebnismenge gefunden werden.

An dieser Stelle wird zwischen der Eingangsabtastzeit  $T_{\text{in}}$  und der Aktualisierungszeit  $T_{\text{akt}}$  eines Planungsalgorithmus unterschieden. Die Aktualisierungszeit ist definiert als die Zeit zwischen der letzten und der nächsten geplanten Trajektorie. Im Gegensatz zur Abtastzeit, die den Zustandsraum begrenzt, beeinflusst die Aktualisierungszeit vor allem die Fähigkeit, auf andere dynamische Objekte in der Umgebung zu reagieren. Beide können unabhängig voneinander gewählt werden, auch wenn einige Algorithmen Zwischenberechnungen aus vorherigen Iterationen wiederverwenden können, wenn  $T_{\text{akt}} = T_{\text{in}}$  (z. B. in Suchbäumen). Wenn die Ak-

---

<sup>1</sup>An dieser Stelle wird explizit von der Eingangsabtastzeit gesprochen, da nur die Abtastzeit des Systemeingangs gemeint ist. Der Systemausgang kann durch Lösen der kontinuierlichen Differentialgleichungen auch zu anderen Zeitpunkten ausgewertet werden. Für längere Eingangsabtastzeiten bietet es sich beispielsweise an, die Ausgangsabtastzeit auf ein  $n$ -tel der Eingangsabtastzeit zu verkleinern, um so zusätzliche Fahrzeugzustände zu erhalten. Diese können z. B. zur Regelung oder in einer Kollisionsüberprüfung verwendet werden.

tualisierungszeit entsprechend gewählt wird, führt eine große Eingangsabtastzeit  $T_{\text{in}}$  nicht dazu, dass das Fahrzeug auf zeitkritische Manöver nicht mehr reagieren kann. Im Folgenden werden zunächst die in der Literatur gewählten Abtastzeiten vorgestellt, bevor die Abtastzeit für von Menschen gefahrenen Trajektorien in städtischen Szenarien analysiert wird.

### Eingangsabtastzeiten in der Literatur

Nach bestem Wissen des Autors existiert keine Veröffentlichung, in der die Wahl der Abtastzeit für von Menschen gefahrenen Trajektorien diskutiert wird. In der Trajektorienplanung konzentrieren sich die meisten Veröffentlichungen auf das jeweilige Konzept zur Lösung des Planungsproblems und erwähnen die gewählte Abtastzeit bei der Beschreibung des Experiments. Im Folgenden werden Pfadplanungsansätze vernachlässigt, die in einem zweiten Schritt die Geschwindigkeit über den geplanten Pfad optimieren. In den meisten dieser Ansätze wird lediglich eine Abtastdistanz für die Pfadplanung und keine Abtastzeit erwähnt. Tabelle 4.1 zeigt die Eingangsabtastzeit von veröffentlichten Trajektorienplanern. Hier beträgt die längste Abtastzeit 0,333 s.

### Fahrzeugmodell

Wie in Abschnitt 2.3 begründet, wird, ebenso wie in Kapitel 3, das kinematische Einspurmodell als Bewegungsmodell verwendet. Bei der Planung zeitdiskreter Trajektorien mit dem Bewegungsmodell aus Gleichungen (2.1) bis (2.4) entstehen Pfade mit konstanter Krümmung zwischen zwei Trajektorienpunkten. Es wird für größere Eingangsabtastzeiten  $T_{\text{in}}$  immer schwerer diesen unstetigen Trajektorien zu folgen, da die unstetigen Sprünge größer ausfallen können. Eine Lösung zu diesem Problem ist die Erweiterung des Modells, sodass die Modelleingänge durch Ableitungen der bisherigen Eingänge ersetzt werden. So verwendet z. B. Matute et al. [83] ein kinematisches Einspurmodell mit Ruck und Lenkwinkelgeschwindigkeit als Eingang für ihren Trajektorienfolgeregler. Der Nachteil dieses Lösungsansatzes ist allerdings die Dimensionsvergrößerung des Modells, welche einen Planungsalgorithmus möglicherweise komplexer werden lässt [97]. Da  $T_{\text{in}}$  möglichst maximiert werden soll, wird dies in Kauf genommen. Als Eingänge werden die Beschleunigung  $a$  sowie die Lenkwinkelgeschwindigkeit  $\omega$  gewählt, sodass folgendes Bewegungsmodell mit den Zuständen

**Tabelle 4.1:** Gängige Werte der Eingangsabtastzeit  $T_{in}$  für Trajektorienplanungsalgorithmen in der Literatur. Die meisten Ansätze verwenden eine Modellprädiktive Regelung (engl.: *model predictive control*, MPC).

Literatur	Modell	Abtastzeit
Seccamonte et al. [115]	MPC, Einspurmodell abhängig von $\kappa$	0,05 s
Dixit et al. [34]	MPC, Einspurmodell mit Beschleunigung	0,1 s
Gutjahr et al. [47]	MPC, linearisiertes Ein- spurmodell	0,2 s
Kong et al. [65]	MPC, dynamisches Ein- spurmodell	0,2 s
Ziegler et al. [145]	Ruckfreie, kontinuierliche Optimierung	0,333 s

$\mathbf{x} = [w x_{\text{ref}}, w y_{\text{ref}}, \psi, v, \delta]^\top$  entsteht:

$$w \dot{x}_{\text{ref}} = v \cos(\psi + \beta), \quad (4.1)$$

$$w \dot{y}_{\text{ref}} = v \sin(\psi + \beta), \quad (4.2)$$

$$\dot{\psi} = \frac{v \cos \beta \tan \delta}{l}, \quad (4.3)$$

$$\dot{v} = a, \quad (4.4)$$

$$\dot{\delta} = \omega, \quad (4.5)$$

mit

$$\beta = \arctan\left(\frac{l_{\text{ref}} \tan \delta}{l}\right). \quad (4.6)$$

Der Schwimmwinkel  $\beta$  ändert sich nun während eines konstanten Eingangs  $\mathbf{u} = [a, \omega]^\top$ , was zu variablen momentanen Krümmungen führt. Dies ermöglicht die Verwendung längerer Abtastzeiten, wie im Weiteren analysiert wird.

### Analyse der Eingangsabtastzeit von Menschen gefahrenen Trajektorien

Um die Abtastzeit von Menschen gefahrenen Trajektorien in städtischen Szenarien zu analysieren, wird im Folgenden der durch eine Drohne aufge-

nommene Datensatz inD [12] verwendet. In diesem werden die Trajektorien von LKWs, PKWs, Fahrrädern und Fußgängern an vier verschiedenen Kreuzungen in Aachen, Deutschland, erfasst. Für jedes Fahrzeug und jeden Zeitschritt enthält der Datensatz extrahierte Informationen wie die Position des Fahrzeugs in einem erdfesten lokalen Koordinatensystem  ${}_W\mathcal{F}$ , die Geschwindigkeit, die Beschleunigung und die Ausmaße des Fahrzeugs. Da die Trajektorien aus Drohnenbildern berechnet werden, enthalten die Daten keine Informationen über den Lenkwinkel des Fahrzeugs. Wie in jedem realen Datensatz ist Messrauschen enthalten, aber nach [12] ist der dadurch entstehende Positionierungsfehler typischerweise kleiner als 0,1 m. Der Datensatz ist in 33 verschiedene Aufzeichnungen unterteilt, die jeweils mit einer Frequenz von 25 Hz und einer Länge von 20–22 Minuten aufgezeichnet wurden. Insgesamt werden in allen Aufnahmen die Trajektorien von 7326 Fahrzeugen betrachtet, was zu ca. 2,5 Mio. Datenpunkten führt. Im Weiteren wird dieser Datensatz als Ground-Truth-Daten bezeichnet.

Zunächst wird das erweiterte Einspurmodell in die Messdaten für verschiedene Eingangsabtastraten eingesetzt. Um die Modelleingänge zu berechnen, wird das folgende beschränkte Optimierungsproblem formuliert. Die kontinuierlichen Gleichungen (4.1) bis (4.6) werden mit der Runge-Kutta-Methode vierter Ordnung mit der Abtastrate des Datensatzes (25 Hz) gelöst. Daraus entstehen  $m$  Fahrzeugzustände für jeden Eingang  $\mathbf{u}_k$  mit  $m = T_{\text{in}} \cdot 25 \text{ Hz}$ . Für jede zeitdiskrete Aktion  $\mathbf{u}_k$  wird der Mittelwert der  $m$  quadrierten euklidischen Abstände zwischen der Position des Einspurmodells  ${}_W\mathbf{p}_{\text{ref,bm}}$  und des inD-Datensatzes  ${}_W\mathbf{p}_{\text{ref,inD}}$  minimiert:

$$\min_{\mathbf{u}_k} J(\mathbf{u}_k) = \sum_{k=0}^f \frac{1}{m} \sum_{n=0}^m \left\| \begin{bmatrix} {}_Wx_{\text{ref,bm}}(\mathbf{u}_k, n) \\ {}_Wy_{\text{ref,bm}}(\mathbf{u}_k, n) \end{bmatrix} - \begin{bmatrix} {}_Wx_{\text{ref,inD}}(n + mk) \\ {}_Wy_{\text{ref,inD}}(n + mk) \end{bmatrix} \right\|_2^2 \quad (4.7)$$

unter den Nebenbedingungen  $v \geq 0$ ,

$$\begin{aligned} & -6 \text{ m/s}^2 < a_k \leq 6 \text{ m/s}^2, \\ & -\arcsin(\kappa_{\text{max}} l) \leq \delta_k \leq \arcsin(\kappa_{\text{max}} l), \\ & -\pi \text{ rad/s} \leq \omega_k \leq \pi \text{ rad/s}, \end{aligned} \quad (4.1) - (4.6).$$

Die maximale fahrbare Krümmung ist zu  $\kappa_{\text{max}} = 0,2 \text{ m}^{-1}$  gesetzt, welches einem typischen Wert eines Kleinwagens entspricht. Da  $\kappa_{\text{max}}$  bei größeren Fahrzeugen abnimmt, wird dies als gültige Obergrenze angenommen. Die



Optimierung beginnt mit dem ersten Wert der Ground-Truth-Daten und wird so lange gelöst, bis das Ende der Trajektorie  $T_f$  erreicht ist.

Da der Datensatz nur die Länge  $l_{\text{ges}}$  von der Fahrzeugfront bis zum Heck der extrahierten Fahrzeuge, nicht aber den Radstand  $l$ , enthält, werden 37 Pkws verschiedener Marken und Fahrzeugklassen analysiert. Daraus wird für den Radstand 60% der extrahierten Fahrzeuglänge  $l_{\text{ges}}$  und für die Position des Referenzpunktes  $l_{\text{ref}}$  28,9% des Radstandes angenommen. Im Anschluss wird das Optimierungsproblem mit Hilfe der Software *SciPy* [124], unter Verwendung des in [15] vorgestellten Algorithmus, gelöst. Indem die Eingänge sowie die Zustände des Einspurmodells während der Optimierung begrenzt werden, wird sichergestellt, dass die entsprechende Bewegung des Fahrzeugs physikalisch möglich ist. Außerdem wird in Gleichung (4.7) nur der Positionsfehler der Fahrzeuge für jeden Zeitschritt  $n$  minimiert, da das Hinzufügen weiterer Kostenterme (z. B. Geschwindigkeit, Orientierung) die Konvergenz verringert. Aufgrund der Zeitabhängigkeit ist in der gewählten Kostenfunktion die Geschwindigkeit des Fahrzeugs indirekt enthalten.

Um das Ergebnis des Optimierungsproblems in Bezug auf verschiedene Abtastzeiten zu analysieren, werden die Optimierungsergebnisse in der Frequenz der Ground-Truth-Daten (25 Hz) herangezogen. Eine Optimierung wird als gescheitert erklärt, wenn der Abstand zwischen einem beliebigen Punkt des Optimierungsergebnisses und dem entsprechenden Ground-Truth-Punkt größer als 0,3 m ist. Mit diesem Grenzwert wird das Messrauschen des Datensatzes sowie das Ziel, die ursprüngliche Trajektorie möglichst gut abzubilden, berücksichtigt.

Für eine erste Analyse wird erneut die in Abb. 4.1 dargestellte Trajektorie betrachtet. Hier hält der Fahrer kurz vor dem Einfahren in die Kreuzung an und biegt anschließend nach rechts ab. Abb. 4.2 zeigt die zugehörigen Ergebnisse des Optimierungsproblems. Für verschiedene Abtastzeiten  $T_{\text{in}}$  zwischen 0,2 s und 1,0 s sind die zurückgelegte Strecke  $s$ , die Geschwindigkeit  $v$  sowie der euklidische Abstand  $d$  zu den Ground-Truth-Daten aufgetragen. Wird die Abtastzeit  $T_{\text{in}}$  zu groß gewählt, beginnt die optimierte Trajektorie um die Ground-Truth-Trajektorie zu oszillieren und damit den Fehler  $d$  periodisch zu vergrößern und zu verkleinern. Bei größeren Abtastzeiten nimmt der Effekt der Oszillation zu und führt schließlich zu fehlgeschlagenen Optimierungen.

In Tabelle 4.2 sind die Ergebnisse aller optimierten Trajektorien für verschiedene Eingangsabtastzeiten zwischen 0,2 s und 1,0 s dargestellt. Mit zunehmender Abtastzeit  $T_{\text{in}}$  steigen sowohl die Anzahl der fehlgeschlagenen Trajektorien als auch der Mittelwert und die Standardabweichung

**Tabelle 4.2:** Prozentsatz der fehlgeschlagenen Trajektorienoptimierungen aller 7326 Trajektorien sowie Mittelwert und Standardabweichung (SD) des Abstands  $d$  des erweiterten Einspurmodells zu den Ground-Truth-Daten in Abhängigkeit von verschiedenen Eingangsabtastzeiten  $T_{\text{in}}$ . Eine Trajektorie gilt als fehlgeschlagen, wenn der euklidische Abstand  $d$  eines optimierten Trajektorienpunktes zum entsprechenden Ground-Truth-Punkt größer als 0,3 m beträgt.

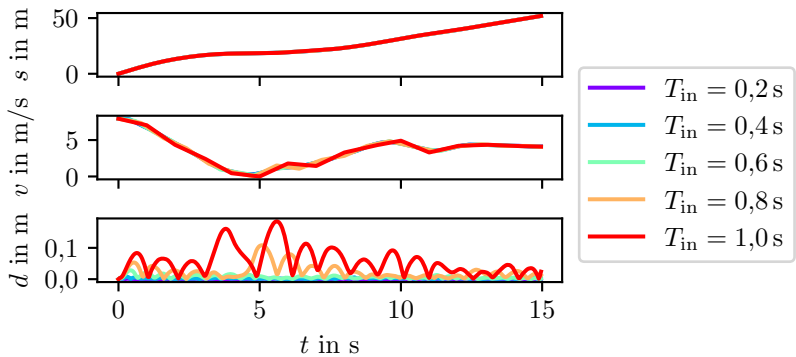
$T_{\text{in}}$	0,2 s	0,4 s	0,6 s	0,8 s	1,0 s
Fehlgeschlagen	0,3 %	0,1 %	1,8 %	5,5 %	11,6 %
Mittel( $d$ )	6 mm	6 mm	13 mm	26 mm	49 mm
SD( $d$ )	0,12 mm	0,02 mm	0,02 mm	0,04 mm	0,20 mm

des Positionsfehlers. Dies kann mit dem oben gezeigten Anstieg des Positionsfehlers  $d$  der optimierten Trajektorie erklärt werden. Wählt man eine Abtastzeit größer als 0,6 s, erreicht die Anzahl der fehlgeschlagenen Trajektorien die 5 %-Marke. Diese Abtastzeiten werden daher bei Verwendung des erweiterten kinematischen Modells als zu groß eingestuft, um das menschliche Fahren zu verallgemeinern. Daher sollte die Abtastzeit kleiner oder gleich 0,6 s gewählt werden. Für diese Abtastzeit ist die Oszillation der optimierten Trajektorie noch akzeptabel.

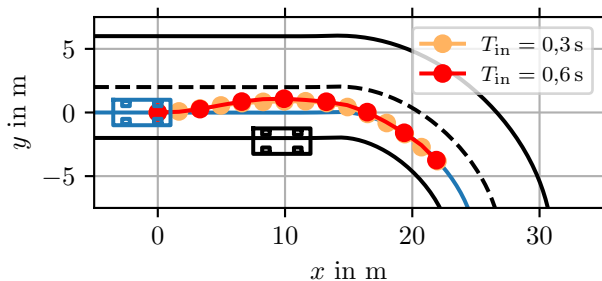
Zu weiteren Demonstrationszwecken der Ergebnisse auf die Trajektorienplanung, wird ein vereinfachter Trajektorienplaner auf Basis einer Baumsuche sowie des erweiterten kinematischen Einspurmodells aus Gleichungen (4.1) bis (4.6) implementiert (siehe Anhang A.1). In jedem Planungsschritt kann der Planer zwischen sieben verschiedenen Lenkgeschwindigkeiten wählen. In der Kostenfunktion werden der seitliche Abstand und die Orientierungsdifferenz zur Mittellinie sowie der euklidische Abstand zu anderen Objekten berücksichtigt. In Abb. 4.3 ist ein beispielhaftes Überhol szenario eines geparkten Fahrzeugs dargestellt. Zwei Trajektorien werden mit den Abtastzeiten  $T_{\text{in}} = 0,3 \text{ s}$  (orange) und  $T_{\text{in}} = 0,6 \text{ s}$  (rot) geplant. Die geplanten Trajektorien unterscheiden sich nur geringfügig, obwohl die Abtastzeit verdoppelt wird.

## 4.4 Verhaltensmodell

Das im Folgenden dargestellte Verhaltensmodell beschreibt mögliche Bewegungen eines Fahrzeugs in Abhängigkeit seines aktuellen Zustands. Ein solches Modell kann in der Trajektorienplanung nützlich sein, z. B. zur



**Abbildung 4.2:** Ergebnisse der Optimierung (4.7) für verschiedene Eingangsabstastzeiten  $T_{\text{in}}$  für eine Beispieltrajektorie (Aufnahme 28, Trajektorie 64). Dargestellt sind die zurückgelegte Strecke  $s$ , die Geschwindigkeit  $v$  und der euklidische Abstand zu den Ground-Truth-Daten  $d$ . Eine Erhöhung von  $T_{\text{in}}$  führt zu einer unflexibleren Trajektorie. Wird  $T_{\text{in}}$  zu groß gewählt, schwankt die optimierte Trajektorie um die Ground-Truth-Trajektorie, sodass der Fehler  $d$  periodisch oszilliert. [141]



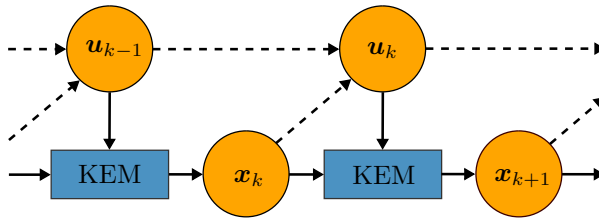
**Abbildung 4.3:** Zwei geplante Trajektorien für das Ego-Fahrzeug (blau) mit unterschiedlichen Eingangsabstastzeiten  $T_{\text{in}}$ . Das Fahrzeug versucht der Mittellinie der Straße (blau) zu folgen und dabei einen sicheren Abstand zum geparkten Fahrzeug (schwarz) einzuhalten. Obwohl  $T_{\text{in}}$  der roten Trajektorie im Vergleich zur orangenen verdoppelt ist, ist die geplante Trajektorie fast die gleiche. [141]

Reduktion des Planungsraums durch Berücksichtigung der Zustandskorrelationen des Modells. Wird das Modell von menschlichen Fahrern abgeleitet, werden nicht-menschliche Fahrweisen im Planungsraum ausgeschlossen. Daher ist das Ziel dieses Abschnitts solche Korrelationen in menschlichen Fahrdaten zu finden.

Für diese Analyse werden die Ergebnisse der Optimierung (4.7) aus Abschnitt 4.3 verwendet, welche eine weitere Analyse des Fahrverhaltens von Menschen an städtischen Kreuzungen ermöglicht. Durch das Einpassen des erweiterten Einspurmodells in den Datensatz werden Approximationen der menschlichen Trajektorien erhalten. Diese Approximationen enthalten auch Informationen über den Lenkwinkel sowie die zeitdiskreten Aktionen  $\mathbf{u}_k$  des Fahrzeugs.

Durch Verwendung der Optimierungsergebnisse aus (4.7) wird das Übergangsmodell definiert, welches den Zusammenhang zwischen einem Fahrzeugzustand  $\mathbf{x}_k$  und dem darauffolgenden Zustand  $\mathbf{x}_{k+1}$  abbildet. Dabei wird der Übergang zwischen den Fahrzeugzuständen, wie in der Optimierung, durch das erweiterte kinematische Einspurmodell nach den Gleichungen (4.1) bis (4.6) definiert, dargestellt in Abb. 4.4 als durchgezogene Linien. Der Eingang  $\mathbf{u}_k$  ist die Steuergröße, die die zukünftigen Bewegungen des Fahrzeugs (also den nächsten Zustand  $\mathbf{x}_{k+1}$ ) bestimmt und von einem Trajektorienplaner ermittelt werden muss. An dieser Stelle kommt das Verhaltensmodell zum Einsatz: Dieses Modell stellt die Korrelationen zwischen dem nächsten Eingang  $\mathbf{u}_k$  und dem aktuellen Zustand  $\mathbf{x}_k$  sowie dem zuletzt gewählten Eingang  $\mathbf{u}_{k-1}$  dar (gestrichelte Linien in Abb. 4.4). Für einen Trajektorienplaner kann dieses Modell verwendet werden, um sich auf Eingänge zu konzentrieren, die dem Menschen ähnlich sind. Für die Eingangsabtastzeit zwischen zwei Zeitpunkten  $k$  und  $k+1$  wählen wir  $T_{\text{in}} = 0,6$  s, wie zuvor in Abschnitt 4.3 argumentiert.

Wie in Abschnitt 4.1 beschrieben, ist das Verhaltensmodell unabhängig von der aktuellen Verkehrssituation. Das bedeutet, dass die Fahrzeugpose (d.h.  $w_{x_{\text{ref}}}, w_{y_{\text{ref}}}, \psi$ ) und die Umgebung (d.h. Straßenführung, andere Verkehrsteilnehmer, etc.) bei der folgenden Analyse vernachlässigt werden. Der Zustand  $\mathbf{x}$  enthält also nur die Geschwindigkeit  $v$  und den Lenkwinkel  $\delta$  des Fahrzeugs, während sich der Eingang  $\mathbf{u}$  aus der Beschleunigung  $a$  und der Lenkgeschwindigkeit  $\omega$  des Fahrzeugs zusammensetzt. In einer probabilistischen Herangehensweise ist ein Planer an der bedingten Wahrscheinlichkeit  $p(\mathbf{u}_k | \mathbf{x}_k, \mathbf{u}_{k-1}) = p(a_k, \omega_k | v_k, \delta_k, a_{k-1}, \omega_{k-1})$  interessiert, die durch Marginalisierung der Gesamtwahrscheinlichkeit  $p(\mathbf{u}_k, \mathbf{x}_k, \mathbf{u}_{k-1}) = p(a_k, \omega_k, v_k, \delta_k, a_{k-1}, \omega_{k-1})$  berechnet werden kann. Im Folgenden werden zunächst die Variablen der Fahrzeuglenkung trans-



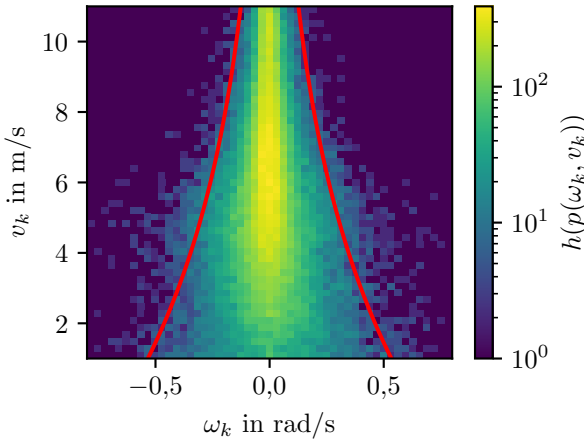
**Abbildung 4.4:** Zeitdiskretes Übergangsmodell. Der nächste Fahrzeugzustand  $\mathbf{x}_{k+1}$  kann durch das deterministische, erweiterte kinematische Einspurmodell (KEM) aus den Gleichungen (4.1) bis (4.6) berechnet werden (durchgezogene Linien). Das Verhaltensmodell stellt die Korrelation zwischen dem Eingang  $\mathbf{u}_k$  des Einspurmodells in Abhängigkeit des aktuellen Zustands  $\mathbf{x}_k$  sowie des letzten Eingangs  $\mathbf{u}_{k-1}$  dar (gestrichelte Linien). [141]

formiert, bevor das extrahierte Verhaltensmodell beschrieben wird.

#### 4.4.1 Transformation der Lenkvariablen $\omega_k, \delta_k$

Um ein einfacheres Verhaltensmodell zu ermöglichen, wird in diesem Abschnitt eine Transformation der Lenkvariablen vorgestellt. Solche Transformationen können vielfältig eingesetzt werden, z. B. bei der Diskretisierung von Lenkvariablen (vgl. Abschnitt 5.3.1). Zur Herleitung dieser werden zunächst die Abhängigkeiten zwischen der Geschwindigkeit des Fahrzeugs und der Lenkgeschwindigkeit analysiert. Abb. 4.5 zeigt ein 2D-Histogramm der aktuellen Vorderradlenkgeschwindigkeit  $\omega_k$ , aufgetragen über die aktuelle Geschwindigkeit  $v_k$ . Das Histogramm zeigt eine nichtlineare Korrelation zwischen diesen beiden Zufallsvariablen. Die Korrelation ist verständlich, da mit zunehmender Geschwindigkeit die Lenkbewegungen des Fahrers kleiner werden, um größere Querbeschleunigungen des Fahrzeugs zu verhindern. Dieser Zusammenhang ist auch in Abb. 4.6 zu sehen, wo Daten für zwei exemplarisch ausgewählte Geschwindigkeitsbereiche der Lenkgeschwindigkeit dargestellt sind. Hier ist die Standardabweichung für den Geschwindigkeitsbereich von 3–4 m/s 3,6 mal größer als für den Bereich von 12–13 m/s. Außerdem zeigt sich, dass die Daten innerhalb eines kleinen Geschwindigkeitsbereichs eine studentsche  $t$ -Verteilung darstellen. Bei niedrigeren Geschwindigkeiten nähert sich diese  $t$ -Verteilung einer Normalverteilung an, da bei Kreuzungsszenarien die Fahrzeuge bei niedrigeren Geschwindigkeiten häufiger abbiegen als bei höheren.

Um diesen Zusammenhang im Verhaltensmodell abzubilden, wird eine



**Abbildung 4.5:** 2D-Histogramm von  $\omega_k$  in rad/s über  $v_k$  in m/s, aufgetragen in logarithmischer Skala. Für größere Geschwindigkeiten nimmt der absolute Wert der Lenkwinkelgeschwindigkeit ab. Die extrahierte Transformationsfunktion (4.8) ist in Rot dargestellt. [141]

nichtlineare Transformation durchgeführt. Aus den 98.0%-Quantilen jedes Geschwindigkeitsbereiches wird die Exponentialfunktion

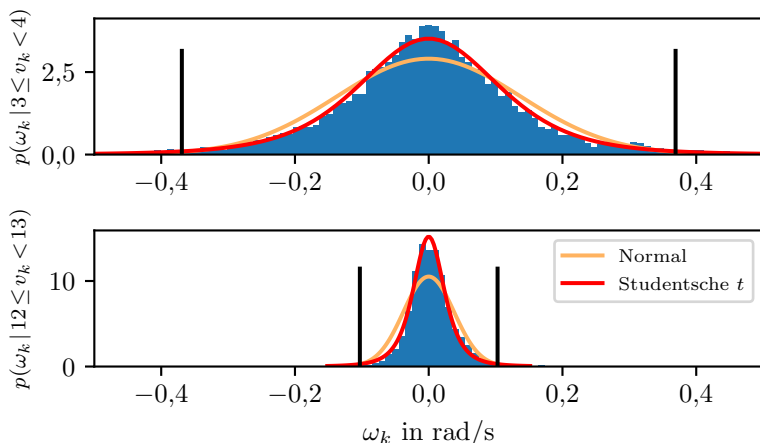
$$|\omega_{\max, k}(v_k)| = p_1 e^{-\frac{v_k}{p_2}} \quad (4.8)$$

extrahiert, welche die Korrelation zwischen einer Geschwindigkeit  $0 < v_k < 15 \text{ m/s}$  und der maximalen Lenkwinkelgeschwindigkeit  $\omega_{\max, k}(v_k)$  in rad/s darstellt. Dabei sind die Werte der Parameter  $p_1 = 0,6164$ ,  $p_2 = 6,9401$ . Die daraus entstehende Korrelation ist in Abb. 4.5 durch die roten Linien dargestellt. Durch

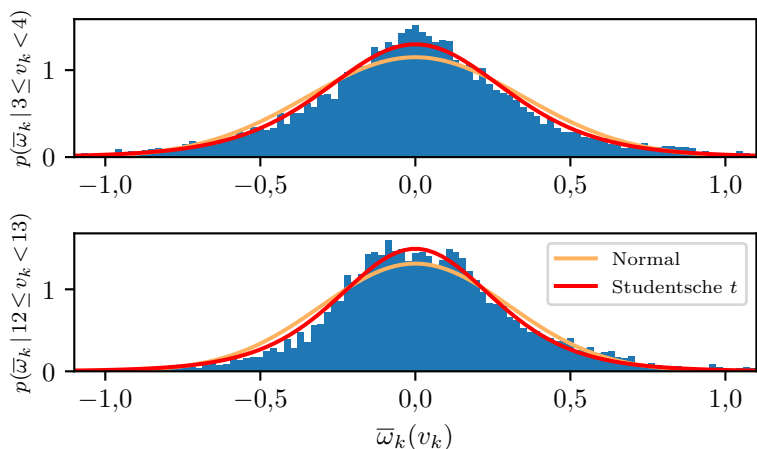
$$\bar{\omega}_k(v_k) = \frac{\omega_k}{|\omega_{\max, k}(v_k)|}, \quad (4.9)$$

erhält man eine transformierte Lenkwinkelgeschwindigkeit  $\bar{\omega}_k(v_k)$ , welche über  $v_k$  näherungsweise gleichverteilt ist (vgl. Abb. 4.9a). Dies ist ebenso in Abb. 4.7 zu sehen, wo die transformierten Daten aus den zwei Geschwindigkeitsbereichen aus Abb. 4.6 aufgetragen sind. Durch die nichtlineare Transformation kann eine einzige Lenkwinkelgeschwindigkeitsverteilung über den gesamten Geschwindigkeitsbereich angenommen werden.

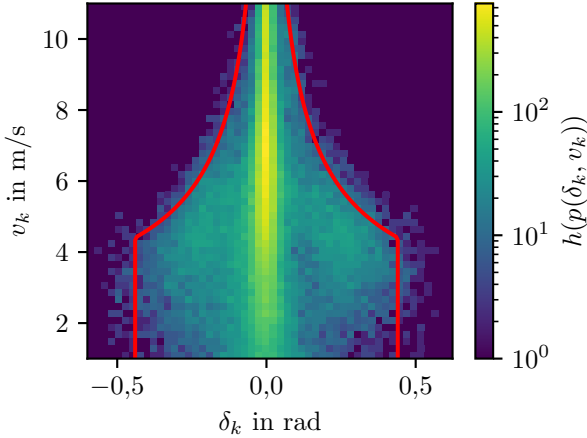
Ähnlich wie die Lenkgeschwindigkeit  $\omega_k$  ist auch der Lenkwinkel  $\delta_k$  abhängig von der Geschwindigkeit (vgl. Abb. 4.8). Dies ergibt sich aus den



**Abbildung 4.6:** Verteilungsdichte von  $\omega_k$  in rad/s für zwei unterschiedliche Geschwindigkeitsbereiche  $v_k$  in m/s. Das Histogramm der Ground-Truth-Daten (blau), die zugehörige Normalverteilung (orange) sowie die studentische  $t$ -Verteilung (rot) werden dargestellt. Die schwarzen Linien begrenzen das 98,0%-Quantil welches zur Extraktion der Transformationsfunktion (4.8) verwendet wird. [141]



**Abbildung 4.7:** Verteilungsdichte von  $\bar{\omega}_k(v_k)$  für zwei unterschiedliche Geschwindigkeitsbereiche  $v_k$  in m/s. Das Histogramm der mittels Gleichung (4.8) skalierten Ground-Truth-Daten (blau), die zugehörige Normalverteilung (orange) sowie die studentische  $t$ -Verteilung (rot) werden dargestellt.



**Abbildung 4.8:** 2D-Histogramm von  $\delta_k$  in rad über  $v_k$  in m/s, aufgetragen in logarithmischer Skala. Für größere Geschwindigkeiten nimmt der absolute Wert des Lenkwinkels ab. Die extrahierte Transformationsfunktion (4.10) ist in Rot dargestellt. [141]

gleichen Erklärungen wie für  $\omega_k$ . Zur Begrenzung der Querbewegung des Fahrzeugs wird eine nichtlineare Transformation abgeleitet, indem das 98,0%-Quantil jedes Geschwindigkeitsbereiches extrahiert wird:

$$|\delta_{\max, k}(v_k)| = \min \left( \delta_{\max}, \arcsin \left( \frac{a_{\text{lat}, \max} l}{v_k^2} \right) \right), \quad (4.10)$$

$$\bar{\delta}_k(v_k) = \frac{\delta_k}{|\delta_{\max, k}(v_k)|}, \quad (4.11)$$

mit dem maximalen Lenkwinkel  $\delta_{\max} = 0,44$  rad, der maximalen Querbewegung  $a_{\text{lat}, \max} = 2,96 \text{ m/s}^2$  sowie dem Radstand  $l = 2,79$  m. Im weiteren Verlauf werden die transformierte Lenkwinkelgeschwindigkeit und der transformierte Lenkwinkel verwendet, sodass ein einfacheres Modell als Verhaltensmodell benutzt werden kann.

#### 4.4.2 Verhaltensmodell menschlicher Fahrer

Nach der Transformation der Zustandsvariablen der Fahrzeugquerbewegung wird im Folgenden das extrahierte Verhaltensmodell vorgestellt.



Das Verhaltensmodell stellt die Korrelationen zwischen dem aktuellen Zustand  $\mathbf{x}_k$ , dem letzten Eingang  $\mathbf{u}_{k-1}$  sowie dem nächsten Eingang  $\mathbf{u}_k$  dar (vgl. Abb. 4.4). Daher ist der Zustandsvektor des Modells definiert als  $\mathbf{x}_M = [a_{k-1}, \bar{\omega}_{k-1}(v_{k-1}), \bar{\delta}_k(v_k), a_k, \bar{\omega}_k(v_k)]^\top$ , während die Geschwindigkeit des Fahrzeugs nur indirekt über die Transformationen in Gleichungen (4.9) und (4.11) berücksichtigt wird. Durch die Transformation dieser Zustände können die Datenkorrelationen mit einer multivariaten Gauß-Verteilung

$$M_N = N(\mathbf{x}_M \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad (4.12)$$

dargestellt werden, mit dem Mittelwertsvektor  $\boldsymbol{\mu}$  und der Kovarianzmatrix  $\boldsymbol{\Sigma}$ . Durch die Transformation werden die Nichtlinearitäten der Variablen der Querbewegung weiterhin berücksichtigt, obwohl das Modell aus einer multivariaten Gauß-Verteilung besteht.

Für einen Trajektorienplaner kann dieses Modell zur Berechnung der bedingten Wahrscheinlichkeit für den nächsten Eingang

$$p(a_k, \bar{\omega}_k(v_k) \mid \bar{\delta}_k(v_k), a_{k-1}, \bar{\omega}_{k-1}(v_{k-1})) \quad (4.13)$$

verwendet werden. Je nach Planungsalgorithmus kann dann entweder die verbleibende Verteilung direkt verwendet werden oder es können mögliche Eingänge aus der Verteilung abgetastet werden. Um die bedingte Wahrscheinlichkeit zu berechnen, wird angenommen, dass eine multivariate Gauß-Verteilung  $N(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$  vorliegt, die wie folgt aufgeteilt werden kann:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{bmatrix}, \quad \boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{bmatrix}.$$

Dann ist die bedingte Wahrscheinlichkeit  $p(\mathbf{x}_a \mid \mathbf{x}_b) = N(\mathbf{x} \mid \boldsymbol{\mu}_{a|b}, \boldsymbol{\Sigma}_{a|b})$  weiterhin eine Gauß-Verteilung und kann mit

$$\boldsymbol{\mu}_{a|b} = \boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{ab} \boldsymbol{\Sigma}_{bb}^{-1} (\mathbf{x}_b - \boldsymbol{\mu}_b), \quad (4.14)$$

$$\boldsymbol{\Sigma}_{a|b} = \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab} \boldsymbol{\Sigma}_{bb}^{-1} \boldsymbol{\Sigma}_{ba}, \quad (4.15)$$

berechnet werden. Die bedingte Wahrscheinlichkeit des hier vorgestellten Verhaltensmodells (Gleichung (4.13)) kann durch die Matrixmultiplikationen in Gleichungen (4.14) und (4.15) bestimmt werden. Das hergeleitete Modell ist daher einfach und schnell.

Die Performanz des Modells resultiert wesentlich aus den Transformationen der Lenkgeschwindigkeit und des Lenkwinkels. Dadurch wird

die Geschwindigkeit des Fahrzeugs nur indirekt berücksichtigt und der transformierte Zustandsraum kann durch eine multivariate Verteilung dargestellt werden. Durch die Wahl einer multivariaten Gauß-Verteilung ist eine schnelle Marginalisierung und Inferenz möglich, welche mit den präsentierten Matrixmultiplikationen berechnet werden kann. Außerdem wird das Modell aus realen, menschlichen Fahrdaten abgeleitet.

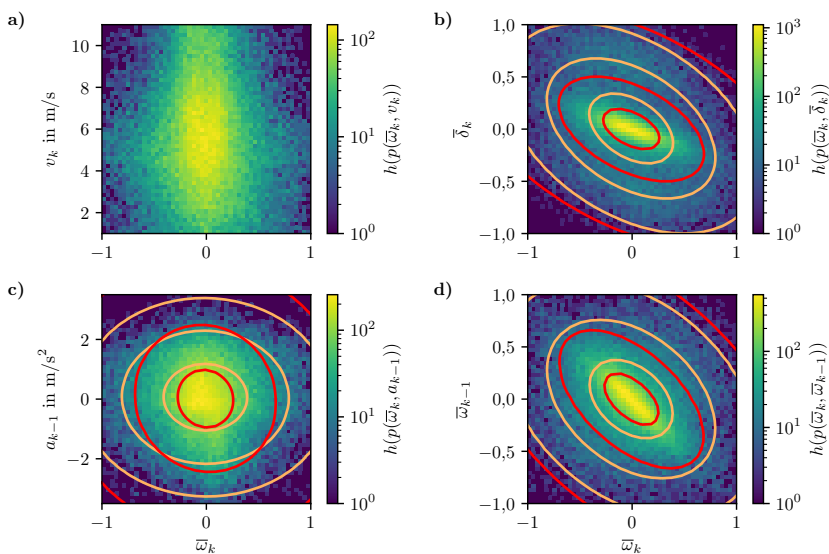
### 4.4.3 Modellanalyse

Nachdem das Modell im letzten Abschnitt präsentiert wurde, wird es nun im Hinblick auf den gegebenen Datensatz analysiert. Von allen Modellvariablen  $\mathbf{x}_M$  ist ein Planer am meisten an der transformierten Lenkgeschwindigkeit  $\bar{\omega}_k$  und der Beschleunigung  $a_k$  interessiert. Sie sind die Eingänge des Modells in Abb. 4.4 und geben die zukünftige Bewegung des Fahrzeugs vor. Da die Gesamtdaten von  $\bar{\omega}_k$  und  $a_k$  eine studentsche  $t$ - oder eine approximierte Gaußverteilung darstellen, wird nicht nur eine multivariate Gauß-Verteilung  $M_N$  wie in Gleichung (4.12) in die Daten eingepasst, sondern auch eine multivariate studentsche  $t$ -Verteilung  $M_{St} = St(\mathbf{x}_M \mid \boldsymbol{\mu}, \boldsymbol{\Lambda}, \nu)$ . Hierzu wird der Erwartungsmaximierungsalgorithmus (EM) [30] verwendet.

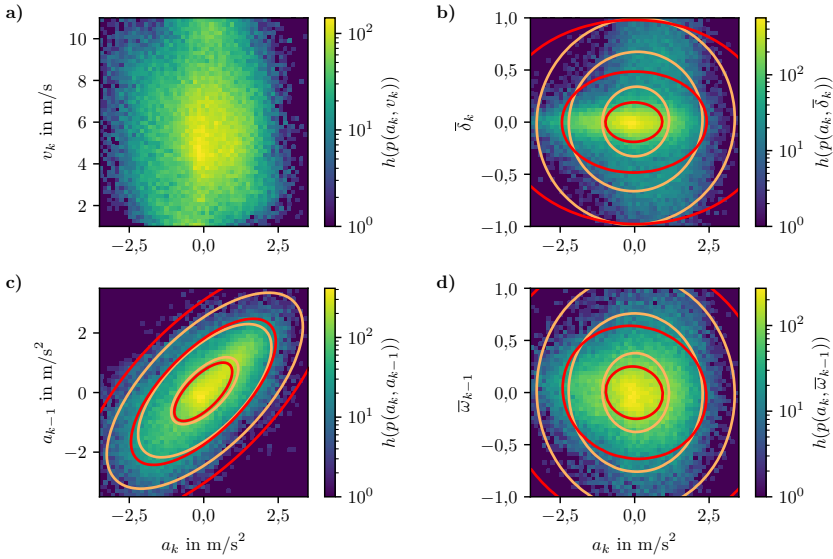
In Abb. 4.9 und 4.10 sind die 39,35%-, 86,47%- und 98,89%-Quantil-Ellipsen der marginalisierten Verteilungen als orangefarbene (Gauß) und rote (studentsche  $t$ ) Linien dargestellt. Während die robuste studentsche  $t$ -Verteilung sich mehr auf die Datenspitzen konzentriert, kann die Orientierung der marginalisierten Quantil-Ellipsen im Vergleich zur Orientierung der Gauß-Verteilung (z. B. Abb. 4.9c oder Abb. 4.10d) abweichen. Um die Modelle zu vergleichen, wird der Likelihood-Quotient (engl.: *Likelihood ratio*) [3, S. 1511]  $LR(M_N, M_{St})$  verwendet, welches insgesamt 3,19 beträgt. Außerdem zeigt Tabelle 4.3 die Likelihood-Quotienten der Modellmarginalisierungen. Ein Likelihood-Quotient größer eins gibt an, dass die Gauß-Verteilung besser zu den Daten passt, während ein Wert kleiner eins angibt, dass die studentsche  $t$ -Verteilung besser korreliert. Die multivariate Gauß-Verteilung schneidet etwas besser ab und wird daher gewählt. Darüber hinaus qualifiziert die schnelle Marginalisierung und Inferenz dieses einfache Modell zusätzlich. Mit diesem können nicht alle Abhängigkeiten in den Daten (z. B.  $\bar{\delta}_k$  auf  $a_k$ ) dargestellt werden. Allerdings ist bei der Trajektorienplanung Einfachheit und Recheneffizienz wichtiger als ein genaueres, aber komplexeres Modell. Die Parameter des endgültigen multivariaten Gauß-Modells  $M_N$  sowie des multivariaten studentschen  $t$  Modells  $M_{St}$  sind im Anhang A.3 dargestellt.

**Tabelle 4.3:** Likelihood-Quotient  $LR(M_N, M_{St})$  der marginalisierten multivariaten Gauß-Verteilung im Vergleich zur marginalisierten, multivariaten studentischen  $t$ -Verteilung, ausgewertet für alle Datenpunkte des Datensatzes. Die Variablen in den Zeilen und Spalten der Tabelle geben die verbleibenden Variablen nach der Marginalisierung an, ebenso wie die in Abb. 4.9 und 4.10 gezeigten Quantil-Ellipsen. Ein Wert größer eins zeigt an, dass die Daten besser zur multivariaten Gauß-Verteilung passen. Der gesamte Likelihood-Quotient ohne Marginalisierung beträgt 3,19.

$LR(M_N, M_{St})$	$\bar{\delta}_k$	$a_{k-1}$	$\bar{\omega}_{k-1}$
$a_k$	2,76	2,73	2,74
$\bar{\omega}_k$	2,89	2,76	2,87



**Abbildung 4.9:** 2D-Histogramm der transformierten Lenkwinkelgeschwindigkeit  $\bar{\omega}_k$  in Abhängigkeit von (a) der aktuellen Geschwindigkeit  $v_k$ , (b) dem aktuellen transformierten Lenkwinkel  $\bar{\delta}_k$ , (c) der letzten Beschleunigung  $a_{k-1}$  und (d) der letzten transformierten Lenkwinkelgeschwindigkeit  $\bar{\omega}_{k-1}$ . Die Linien zeigen die 39,35%-, 86,47%- und 98,89%-Quantil-Ellipsen der marginalisierten multivariaten Gauß- (orange) und studentischen  $t$ -Verteilung (rot) an. [141]



**Abbildung 4.10:** 2D-Histogramm der Beschleunigung  $a_k$  in Abhängigkeit von (a) der aktuellen Geschwindigkeit  $v_k$ , (b) dem aktuellen transformierten Lenkwinkel  $\bar{\delta}_k$ , (c) der letzten Beschleunigung  $a_{k-1}$  und (d) der letzten transformierten Lenkwinkelgeschwindigkeit  $\bar{\omega}_{k-1}$ . Die Linien zeigen die 39,35%, 86,47% und 98,89%-Quantil-Ellipsen der marginalisierten multivariaten Gauß- (orange) und studentischen  $t$ -Verteilung (rot) an. [141]

## 4.5 Zusammenfassung

In diesem Kapitel werden Fahrverläufe menschlicher Fahrer an städtischen Kreuzungen im Kontext der Trajektorienplanung analysiert. Zunächst wird die Wahl der Eingangsabtastzeit  $T_{\text{in}}$  eines zeitdiskreten, erweiterten Fahrradmodells mit der Fahrzeugbeschleunigung  $a$  und der Lenkgeschwindigkeit  $\omega$  als Eingänge analysiert. Durch Anpassung des Fahrradmodells an die menschlichen Fahrdaten wird herausgefunden, dass menschlich gefahrene Trajektorien mit diesem Modell durch eine Eingangsabtastzeit von 0,6s oder weniger dargestellt werden können. Größere Abtastzeiten begrenzen die möglichen Bewegungen des Fahrzeugs, sodass der Anteil nicht abzubildener Trajektorien steigt (vgl. Tabelle 4.2).

Die Untersuchung der Abtastzeit zeigt, dass aktuelle Trajektorienplaner in der Literatur deutlich kleinere Abtastzeiten verwenden (vgl. Ta-

belle 4.1). Mit der hier präsentierten, neuartigen Analyse soll motiviert werden, die Verwendung längerer Abtastzeiten in Trajektorienplanern zu testen. Durch die Erhöhung der Abtastzeit wird der Suchraum und damit die Komplexität des Planungsproblems reduziert, während die geplante Trajektorie zusätzlich geglättet wird. Dies bedeutet auch eine geringere Flexibilität, die dazu führen könnte, dass keine Lösung für das Planungsproblem gefunden wird. Aufgrund des Analysedesigns sowie des vorgestellten Beispielplaners wird angenommen, dass dies kein Problem darstellt. Weitere Vorteile sind das Ermöglichen längerer Planungshorizonte und damit effizientere und vorausschauendere Trajektorien.

In einem nächsten Schritt werden die Eingangsverteilungen (d.h. die Verteilungen der Beschleunigung und der Lenkgeschwindigkeit) von menschlichen Fahrern analysiert. Hier zeigt sich eine nichtlineare Korrelation zwischen der Querbewegung (Lenkwinkel und Lenkgeschwindigkeit) und der Längsbewegung (Geschwindigkeit) des Fahrzeugs. Diese Punkte unterstreichen die Notwendigkeit, Pfad- und Trajektorienplanung in einem einzigen Schritt zu kombinieren. Weiterhin werden geschwindigkeitsabhängige, nichtlineare Transformationen für die Lenkgeschwindigkeit und den Lenkwinkel eingeführt. Im transformierten Raum wird ein neuartiges Übergangsmodell extrahiert, das nur vom letzten und aktuellen Zeitschritt abhängt. Dabei können die Korrelationen durch eine multivariate Gauß-Verteilung modelliert werden. Dieses Modell bildet alle wesentlichen Korrelationen genügend genau ab und ist dennoch einfach und kann schnell ausgeführt werden.

## 5 CarPre-Trajektorienplanung

Die Trajektorienplanung ist das Schlüsselement autonomer Fahrzeuge, da durch sie das Verhalten des autonomen Agenten mit seiner Umgebung maßgeblich bestimmt wird. In diesem Kapitel wird ein neuartiger Ansatz zur Trajektorienplanung vorgestellt, die CarPre-Planung (Monte-Carlo modellprädiktive Trajektorienplanung, engl.: *Monte-Carlo model predictive planner*). Hierzu wird zuerst eine kurze Einleitung und Motivation für Trajektorienplaner gegeben. Im Anschluss wird der Stand der Technik präsentiert, bevor der neuartige Trajektorienplanungsansatz auf Basis der Monte-Carlo Baumsuche vorgestellt wird. Der echtzeitfähige Algorithmus verwendet Heuristiken sowie Simulationen von möglichen Bewegungen zur Berechnung der Trajektorie. Hierbei wird auf diskrete Verhaltensklassen wie in der Planungshierarchie aus Abschnitt 2.1 sowie deren Vor- und Nachteile verzichtet. Nach einer Evaluation des Planers an Beispielszenarien, schließt das Kapitel mit einer Zusammenfassung. Die hier vorgestellten Aspekte wurden teilweise vom Autor in [140] veröffentlicht und werden hier erneut dargestellt.

### 5.1 Einleitung und Motivation

Die Trajektorienplanung bestimmt maßgeblich das Verhalten des autonomen Fahrzeugs und ist damit ein Schlüsselement des autonomen Fahrens. Bei diesem komplexen Planungsproblem müssen sowohl die statische Umgebung als auch dynamische Verkehrsteilnehmer berücksichtigt werden, sodass ein gewünschtes Systemverhalten, auch im Einklang mit den geltenden Verkehrsregeln, entsteht. Das Verhalten wird dabei von den Passagieren besonders deutlich wahrgenommen und bewertet. Ein Planungsergebnis muss nicht nur sicher, sondern auch möglichst vorausschauend und komfortabel sein, damit das Vertrauen und die Akzeptanz von autonomen Fahrzeugen wachsen.

In diesem Kapitel wird der Anwendungsbereich des Planers auf innerstädtische Szenarios beschränkt (vgl. Kapitel 6). Für den hier betrachteten Planer bedeutet dies eine Beschränkung auf eine Maximalgeschwin-

digkeit von 30 km/h. Dies ist allerdings keine allgemeine Einschränkung des Konzeptes. Vielmehr kann es ebenso für höhere Geschwindigkeiten erweitert werden. Des Weiteren wird von einer strukturierten Umgebung ausgegangen, d.h. der fahrbare Raum ist durch eine Straße mit einer eigenen und einer entgegengesetzten Fahrspur eingeschränkt.

Für die mathematische Definition des hier zu lösenden Planungsproblems wird auf Abschnitt 2.2 verwiesen. Die Trajektorienplanung kann in der Planungshierarchie autonomer Fahrzeuge (vgl. Abschnitt 2.1) in die dritte (Bewegungsplanung) Ebene eingeordnet werden. Durch diese hierarchische Aufteilung kann der Verhaltensplaner (2. Ebene) dem Bewegungsplaner eine erste Suchrichtung vorgeben. Da die Kostenfunktion eines Trajektorienplaners durch mögliche Multimodalitäten<sup>1</sup> nicht konvex ist, endet die Optimierung des Bewegungsplaners meist im lokalen Minimum des zugehörigen Verhaltens. Somit kann sich der Bewegungsplaner auf den Bereich um die Initialisierung fokussieren und muss nicht den globalen Planungsraum absuchen. Dies ist eine Reduktion der Planungskomplexität, welche allerdings den Nachteil mit sich bringt, dass sichergestellt werden muss, dass die Bewegungsplanung das berechnete Verhalten des Verhaltensplaners umsetzt [32]. Das kombinierte Lösen der beiden Ebenen schließt solche Planungskonflikte per Design aus, erhöht jedoch die Planungskomplexität erheblich, da nun der gesamte Planungsraum abgesucht werden muss. Um dieses kombinierte Problem in der verfügbaren Rechenzeit lösen zu können, müssen neuartige Algorithmen entwickelt und/oder bestehende optimiert werden.

Die derzeitigen Planungsalgorithmen besitzen alle unterschiedliche Stärken und Schwächen, sodass weiterhin nach neuen Lösungsmöglichkeiten gesucht wird, welche sich von den bestehenden Algorithmen absetzen. Ein vielversprechender Ansatz ist die Monte-Carlo Baumsuche [25]. Es handelt sich dabei um einen Algorithmus zum Lösen von Entscheidungsproblemen, der insbesondere zum Lösen kombinatorischer Spiele eingesetzt wird. In der Kategorie der kombinatorischen Spiele wird das Spiel Go als das anspruchsvollste angesehen, da es einen hohen Verzweigungsfaktor<sup>2</sup> und ca.  $2,08 \cdot 10^{170}$  mögliche Spielpositionen

---

<sup>1</sup>Multimodalitäten treten in vielen Verkehrsszenarien auf. Ein Beispiel hierfür ist eine verengte Straße mit entgegenkommendem Verkehr. Nach [145] existieren zwei mögliche Lösungsvarianten: entweder man fährt vor oder nach dem Gegenverkehr durch die Verengung. Somit existieren zwei lokale Minima in der Kostenfunktion, welche damit nicht konvex sein kann.

<sup>2</sup>Der Verzweigungsfaktor gibt die Anzahl an möglichen Aktionen in einem Spielzug an.

besitzt [126]. Durch die Kombination der Monte-Carlo Baumsuche mit Reinforcement Learning gelang es AlphaGo [116] und seiner Weiterentwicklung AlphaGo Zero [117] menschliches Spielniveau zu übertreffen, ein Meilenstein für die künstliche Intelligenz.

Die Idee des CarPre-Trajektorienplaners ist es, durch Wahl und Anpassung eines potenziell geeigneten Algorithmus, der Monte-Carlo Baumsuche, einen Lösungsansatz für das kombinierte Planungsproblem der Verhaltens- sowie Bewegungsplanung zu finden. Da dies ein Jederzeit-Algorithmus (engl.: *anytime algorithm*) ist, steht zu jedem Zeitpunkt ein erstes Planungsergebnis zur Verfügung, welches mit zusätzlicher Laufzeit immer weiter verbessert wird. Somit kann garantiert werden, dass eine Lösung<sup>3</sup> innerhalb der vorgeschriebenen Zeit ausgegeben wird. Dies ist besonders für sicherheitskritische Anwendungen wie die Trajektorienplanung essenziell. Durch weitere Optionen, wie z. B. die Inkorporation expliziter Verhaltensregeln<sup>4</sup>, das Einbinden diskreter Variablen in den Entscheidungsprozess sowie eine Vielzahl an möglichen Anpassungen bietet die Monte-Carlo Baumsuche Vorteile für die Trajektorienplanung. Im Weiteren wird der neuartige CarPre-Trajektorienplaner auf Basis der Monte-Carlo Baumsuche beschrieben. Bevor der Algorithmus vorgestellt wird, wird zuerst ein Überblick über den Stand der Technik präsentiert.

## 5.2 Stand der Technik

Der aktuelle Stand der Technik wird im Folgenden, wie in [5, 97], in die beiden Ebenen der Verhaltens- und Bewegungsplanung unterteilt. Der interessierte Leser sei auf [71] für grundlegende Planungsalgorithmen verwiesen. Ein erweiterter Überblick über den Stand der Technik ist in [44, 97, 112] dargestellt. Abb. 5.1 bildet exemplarische Planungsansätze aus der Literatur ab.

### Verhaltensplanung

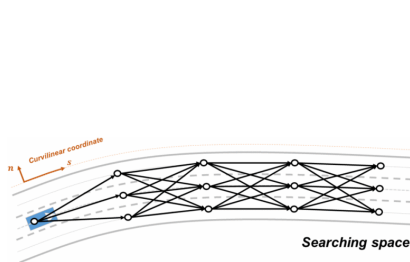
Ziel der Verhaltensplanung ist das Vereinfachen der nachfolgenden Bewegungsplanung. Um dieses Ziel zu erreichen, existieren in der Literatur zwei unterschiedliche Ansätze, die Bestimmung von Verhaltensklassen sowie die

---

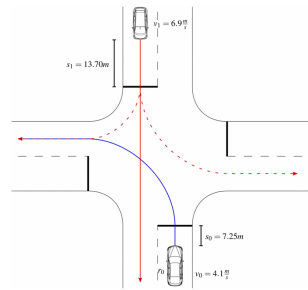
<sup>3</sup>Diese Lösung muss nicht optimal sein, konvergiert allerdings mit steigender Berechnungszeit gegen die optimale Lösung. [64]

<sup>4</sup>Explizite Verhaltensregeln werden unter anderem durch die geltenden Straßenverkehrsordnung vorgegeben. Bei der Extraktion der Verhaltensregeln kann dabei z. B. die verhaltenssemantische Szeneriebeschreibung nach [42, 43] helfen.

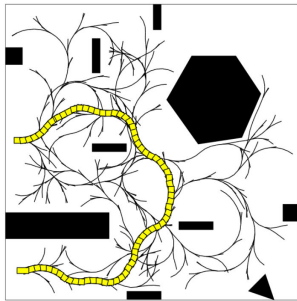




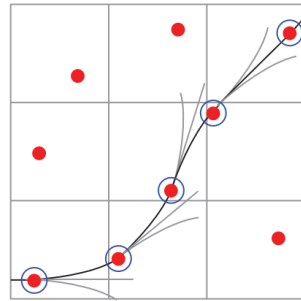
(a) Grobes Frenet-Gitter für Verhaltenstrajektorien. [77]



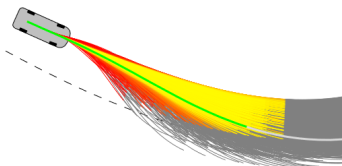
(b) Fahrspur-Verhaltensplaner. [55]



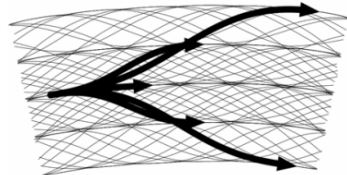
(c) RRT mit Bewegungsmodell. [74]



(d) Hybrid A\*. [91]



(e) Sternplanung im Frenet-Raum. [133]



(f) Zustandsgitter im Frenet-Raum. [86]

**Abbildung 5.1:** Verschiedene Verhaltens- (a, b), Bewegungs- (c–e) sowie kombinierte Planungsansätze (f) aus der Literatur. Die Bilder sind aus den entsprechenden Quellen entnommen.

Berechnung von Verhaltenstrajektorien. Um ein Verhalten berechnen zu können, werden vereinfachende Annahmen getroffen. Ein Beispiel hierfür ist die Annahme, dass sich alle Fahrzeuge nur mittig in ihrem Fahrstreifen bewegen (vgl. Abb. 5.1b) [55, 94].

Bei der Berechnung von Verhaltensklassen wird ein abstraktes, ausführbares Verhalten aus einer vordefinierten, endlichen, potenziell möglichen Menge bestimmt. Hierdurch kann für jedes abstrakte Verhalten eine spezialisierte Bewegungsplanung verwendet werden [96]. Während bestehende Systeme meist auf endlichen Zustandsmaschinen basieren [4, 13, 46, 91, 131], beschäftigen sich aktuelle Forschungsansätze mit Planungsunsicherheiten<sup>5</sup>, mit der Risikominimierung sowie mit Kooperationen mit anderen Verkehrsteilnehmern [7, 16, 20, 28, 39, 45, 55, 135]. Ebenso existieren erste Ansätze [66, 67, 76], welche die Monte-Carlo Baumsuche zur Bestimmung der abstrakten Verhaltensklasse benutzen.

Bei der Berechnung von Verhaltenstrajektorien wird dagegen eine erste Trajektorie in einem vereinfachten Planungsraum bestimmt. Diese Trajektorie stellt dabei das gewünschte Verhalten dar, wie z. B. *der Straße folgen* oder *die Fahrspur wechseln*. Dabei ist die Trajektorie z. B. durch sehr grobes Abtasten des Planungsraumes stark vereinfacht und muss in der darauffolgenden Bewegungsplanung konkretisiert werden (vgl. Abb. 5.1a). Beispiele hierfür sind [54, 77].

## Bewegungsplanung

Anhand des Ergebnisses der Verhaltensplanung wird in der Bewegungsplanung eine ausführbare (d.h. vom Fahrzeug umsetzbare) Trajektorie berechnet. Die hierfür verwendeten Algorithmen lassen sich nach [97] in drei Klassen einteilen, welche im Folgenden vorgestellt werden.

### 1. Optimierungsansätze:

Das Planungsproblem wird als Optimierungsproblem im kontinuierlichen Planungsraum, wie z. B. in der modellprädiktiven Regelung, formuliert. Durch die Verwendung eines Fahrzeugmodells (vgl. Abschnitt 2.3) in den Nebenbedingungen der Optimierung wird die Fahrzeugkinematik berücksichtigt. Beispiele hierfür sind [20, 32, 53, 54, 78, 80, 115, 131, 145]. Der Vorteil dieser Methoden ist die (meist) schnelle Lösung im kontinuierlichen Planungsraum. Nachteile sind,

---

<sup>5</sup>Planungsunsicherheiten beinhalten z. B. die Bewegungsunsicherheit anderer Verkehrsteilnehmer oder die Perzeptionsunsicherheit der Umgebung.

dass erst dann eine Lösung vorliegt, wenn die Optimierung konvergiert ist. Dies kann insbesondere in komplexen, kritischen Situationen problematisch werden, wenn einzelne Optimierungen deutlich mehr Rechenzeit benötigen als erwartet (z. B. Peaks in Abb. A.1 aus [131]). Ein weiterer Nachteil ist die Konvergenz in ein lokales Minimum. Somit ist eine vorgelagerte Planungsebene, wie die Verhaltensplanung, zwingend erforderlich, damit die Optimierung in das gewünschte lokale Minimum konvergiert. Darüber hinaus können diskrete Zustände<sup>6</sup> nicht direkt eingebunden werden.

## 2. Graph-basierte Ansätze:

Durch die Diskretisierung des Planungsraumes in eine endliche Anzahl an Zuständen wird ein Graph erzeugt. Durch Graphsuchalgorithmen, wie z. B. Dijkstra [31], A\* [48] oder D\* [119], kann die optimale Trajektorie im diskretisierten Graphen berechnet werden. Zur Erstellung des hierfür benötigten Graphen existieren unterschiedliche Methoden. Die häufigste Variante ist das Abtasten des Planungsraumes, sodass eine Baumstruktur [1, 38, 91, 133] oder eine Gitterstruktur [86, 87, 118] entsteht. Der Vorteil von Graph-basierten Ansätzen ist, dass die Diskretisierung ein globales Absuchen des Graphen ermöglicht. Durch eine globale Suche bleibt die Planung nicht in lokalen Minima stecken, sondern das (diskretisierte) globale Minimum wird gefunden. Darüber hinaus wird keine explizite Aufzählung von Planungsbedingungen wie bei den Nebenbedingungen in Optimierungsansätzen benötigt. Stattdessen ermöglicht die Diskretisierung das Überprüfen sehr komplexer Bedingungen der möglichen Trajektorien bzw. jedes einzelnen Trajektorienpunktes [69, S.1106]. Der Nachteil dabei ist die Diskretisierung: wird sie zu groß gewählt, vereinfacht sich durch den kleineren Planungsraum das Planungsproblem, allerdings wird unter Umständen keine Lösung gefunden, obwohl im kontinuierlichen Raum eine solche existiert<sup>7,8</sup>. Wird die Diskretisierung dagegen zu klein gewählt, explodiert der Graph durch den „Fluch der Dimension“ [10] und die Planungskomplexität steigt.

---

<sup>6</sup>Ein diskreter Zustand kann z. B. eine Ampel sein, welche zwischen *Halten* und *Fahren* des autonomen Systems entscheidet.

<sup>7</sup>Ein Beispiel hierfür kann eine Verengung sein, welche auf Grund einer zu groben Diskretisierung nicht durchquert werden kann.

<sup>8</sup>Wird der Graph nicht durch Abtasten sondern durch kombinatorische Methoden (auch: geometrische Methoden) wie in [71, S.249ff] erstellt, kann ein solcher Fall nicht eintreten. Eine Erstellung eines kombinatorischen Graphen ist in der Praxis unter anderem durch die hohe benötigte Rechenzeit nicht relevant [97, S.12].

### 3. Inkrementelle Ansätze:

Inkrementelle Ansätze versuchen den Nachteil einer festen Diskretisierung wie bei Graph-basierten Ansätzen zu umgehen. Dies wird erreicht, indem der Planungsraum inkrementell mit immer feineren Diskretisierungsschritten abgetastet wird und zeitgleich versucht wird eine Lösung zu finden [97, S. 44]. Beispiele hierfür sind RRT [73, 74] (vgl. Abb. 5.1c), RRT\* [59, 60], RRM [2] oder CL-RRT [69]. Der Vorteil dieser Ansätze ist, dass eine Lösung gefunden wird, falls sie existiert (vgl. Graph-basierte Ansätze). Der Nachteil ist allerdings, dass die hierfür benötigte Rechenzeit unbegrenzt ist, d.h. es ist möglich, dass die Lösung erst nach viel zu langer Zeit gefunden wird. Dies schließt eine Echtzeitanwendung in der Regel aus.

Zusammengefasst haben sich in den letzten Jahren zwei Klassen an gängigen Lösungsansätzen herausgebildet: die kontinuierliche Optimierung sowie die diskrete Graphsuche. Unabhängig von den vorgestellten Lösungsalgorithmen existieren wiederkehrende Lösungsansätze, die im Weiteren vorgestellt werden.

- **Pfad-Geschwindigkeits-Zerlegung:**

Zur Vereinfachung des Planungsproblems kann dieses nach [58] in zwei Probleme unterteilt werden: Zuerst wird ein Pfad durch die statische Umgebung geplant, bevor in einem zweiten Schritt die Geschwindigkeit über den berechneten Pfad bestimmt wird. Hierdurch vereinfacht sich das Planungsproblem enorm, da der Planungsraum massiv eingeschränkt wird. Daher ist dies eine beliebte Methode, auch bei aktuellen Ansätzen [53, 80, 87, 101, 118, 139]. Der Vorteil dieser Zerlegung ist ebenso ihr Nachteil, der Planungsraum wird ggf. zu stark beschränkt, was zu zwei wesentlichen Problemen führen kann. Erstens sind die lateralen Fahrzeugbewegungen abhängig von der Geschwindigkeit (vgl. Abschnitt 4.4.1). Wird die aktuelle Geschwindigkeit des Fahrzeugs nicht berücksichtigt, können stark gekrümmte Pfade geplant werden, welche nach der Geschwindigkeitsplanung immer noch zu hohe laterale Beschleunigungen aufweisen. Zweitens kann der optimale Pfad in dynamischen Verkehrssituationen nicht zur optimalen Trajektorie führen<sup>9</sup>, was zu kritischen Fahr-

---

<sup>9</sup>Ein Beispiel hierfür ist eine Ausweichsituation. Das autonome Fahrzeug kann auf zwei Seiten um ein statisches Objekt herumfahren. Auf der Seite des optimalen Pfades kommt allerdings ein Fahrzeug entgegen, welches in der statischen Pfadplanung nicht berücksichtigt wird. Die optimale Trajektorie (Weg um die Seite ohne Gegenverkehr) kann somit durch die Pfad-Geschwindigkeits-Zerlegung nicht gefunden werden.

manövern führen kann.

- **Frenet-Koordinatentransformation:**

Für strukturierte Umgebungen, wie ein Straßennetzwerk, kann das kartesische Koordinatensystem in das Frenet-Koordinatensystem umgewandelt werden. Die Frenet-Transformation hängt vom jeweiligen Straßenverlauf ab, sodass die Frenet-Koordinaten in Längs- und Querrichtung zur Straße verlaufen (vgl. Abb. 5.1e). Die Transformation ermöglicht damit ein einfaches Planen in Relation zur Straße. Beispiele aus der Literatur sind [53, 77, 86, 87, 118, 133]. Der Nachteil der Transformation ist, dass die physikalische Größenaufteilung zwischen longitudinaler und lateraler Fahrzeugbewegung erst nach der Rücktransformation in kartesischen Koordinaten bekannt ist. Somit kann z. B. nicht die laterale Beschleunigung oder die Pfadkrümmung im Frenet-Raum berücksichtigt werden. Darüber hinaus treten Probleme mit sehr stark gekrümmten Kurven auf, wie z. B.  $90^\circ$  Kurven an innerstädtischen Kreuzungen [51, S. 68].

In der Klasse der Bewegungsplanung existieren, nach dem Wissen des Autors, bisher keine Ansätze welche die Monte-Carlo Baumsuche verwenden. Ein Ende-zu-Ende-Ansatz<sup>10</sup> [23] existiert, bei welchem mit Hilfe einer tiefen Monte-Carlo Baumsuche ein diskretisierter Lenkwinkel aus einem Kamerabild sowie dem Fahrzeugzustand bestimmt wird. Durch den Aufbau von Ende-zu-Ende-Ansätzen ist das Ergebnis des Systems nicht nachvollziehbar, da keine Zwischenergebnisse (z. B. Umfeldrepräsentation) existieren. Eine Untergliederung in Teilprobleme wie Perzeption, Planung und Regelung ist nicht vorhanden, weswegen auf diesen Ansatz nicht weiter eingegangen wird.

### **Kombinierte Ansätze**

Zusätzlich zu der klassischen Aufteilung in Verhaltens- und Bewegungsplanung existieren kombinierte Ansätze, welche das Planungsproblem in einem Schritt lösen. Dabei wird die erhöhte Planungskomplexität in Kauf genommen, um eine mögliche Planungskomplexität zwischen den beiden Ebenen auszuschließen.

---

<sup>10</sup>Bei einem Ende-zu-Ende-Ansatz wird durch tiefes, mehrschichtiges Lernen versucht alle Systemparameter gleichzeitig zu bestimmen. Auf das autonome Fahren bezogen bedeutet das, dass als Eingang die Sensorrohdaten (z. B. Bilddaten, Lidar-Punktwolke) verwendet werden. Der Ausgang des gelernten Systems sind die Stellgrößen der Aktoren (z. B. Lenkwinkel, Beschleunigung).

Bestehende Ansätze [86, 144] berechnen, ebenso wie z. B. [54, 77] bei der Berechnung der Verhaltenstrajektorien, für die strukturierte Straßenumgebung ein diskretes Zustandsgitter (vgl. Abb. 5.1f). Durch Durchsuchen des Gitters kann eine optimale Trajektorie gefunden werden. Die Schwierigkeit liegt dabei darin, dass im Gegensatz zu der Berechnung der Verhaltenstrajektorien das Zustandsgitter feiner aufgelöst ist. Dadurch steigt die Größe des Zustandsraums und damit auch die benötigte Rechenzeit zur Auswertung des Zustandsgitters sehr schnell an („Fluch der Dimension“ [10]). [86] versucht das Problem durch Parallelisierung und Berechnung auf einer Grafikkarte zu lösen.

Ein alternativer Ansatz von [51] setzt ebenso auf die parallele Rechenleistung aktueller Grafikkarten. In einem ersten Schritt wird ein Graph durch zufälliges Abtasten des Zustandsraumes erzeugt. Im Anschluss werden Trajektorien in alle Kanten des Graphen optimiert und mit einer Kostenfunktion bewertet. Durch die hohe Parallelisierung auf der Grafikkarte können so über 50.000 Trajektorien berechnet und die beste Trajektorie ausgewählt werden.

## Zusammenfassung

Das Trajektorienplanungsproblem ist komplex und muss zeitkritisch gelöst werden. Hinzu kommt die Schwierigkeit, dass die Kostenfunktion nicht konvex ist, sodass weitere Planungsstrategien benutzt werden müssen, um das globale Minimum des Planungsproblems zu finden. Daher wird das Problem oft hierarchisch in die Ebenen der Verhaltens- und Bewegungsplanung unterteilt. Kombinierte Ansätze existieren auf Grund der hohen Lösungskomplexität kaum. Ein möglicher Ansatz hierfür ist die Monte-Carlo Baumsuche, welche bisher nur in der Verhaltensplanung eingesetzt wird und hier durch die Vorteile der Jederzeit-Planung sowie der Flexibilität [76] punktet. Daher wird im Folgenden ein erster Ansatz vorgestellt, welcher das Potenzial hat, das komplette Planungsproblem aus Verhaltens- und Bewegungsplanung zu lösen.

## 5.3 Planung mittels Monte-Carlo Baumsuche

Im Folgenden wird die im Rahmen dieser Dissertation neu entwickelte CarPre-Trajektorienplanung auf Basis der Monte-Carlo Baumsuche beschrieben. Zuerst wird hierzu die klassische Monte-Carlo Baumsuche vorgestellt, bevor im Anschluss auf die Anpassungen zur CarPre-

Trajektorienplanung eingegangen wird.

Die Monte-Carlo Baumsuche [25] ist ein Suchalgorithmus, um diskrete Entscheidungsprobleme<sup>11</sup> zu lösen. Um ein solches zu lösen, baut der Algorithmus einen Suchbaum auf, welcher aus Knoten (Zustände des Suchraumes) und Kanten (mögliche Aktionen in einem Zustand) besteht. Für jeden Zustand wird die Anzahl der bisherigen Besuche  $n_b$  sowie seine summierte Erfolgsabschätzung  $R_s$  gespeichert. Durch das Ausprobieren verschiedener Aktionen wird nach der erfolgversprechendsten<sup>12</sup> gesucht, d.h. nach der Aktion mit der höchsten mittleren Erfolgsabschätzung  $R_m = \frac{R_s}{n_b}$ . Eine Iteration des Algorithmus besteht nach [22] aus vier Schritten, welche in Abb. 5.2 dargestellt sind:

### I. Selektion:

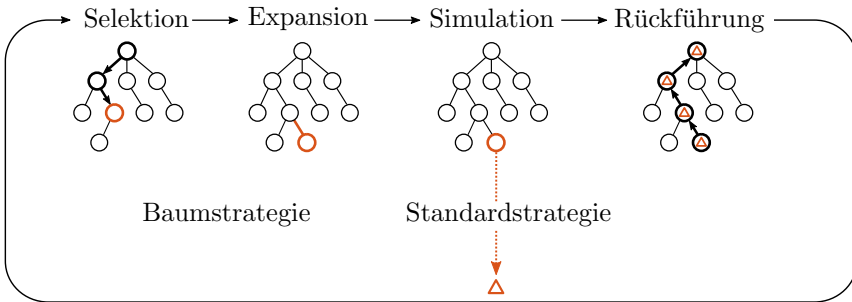
In dieser Phase wird ein Kinderknoten ausgewählt, der in dieser Iteration untersucht wird. Dabei wird strategisch zwischen Aktionen, welche bisher am erfolgversprechendsten aussehen (Ausbeutung, engl.: *exploitation*) und zwischen weniger erfolgversprechenden, aber auch nicht oft untersuchten Aktionen (Erforschung, engl.: *exploration*) abgewogen. Letztere sollten auch untersucht werden, da sie unter Umständen auf Grund der zufälligen Auswertung bisher schlechte Ergebnisse geliefert haben, obwohl sie eigentlich erfolgversprechend sind. Um das Exploitation/Exploration-Problem zu lösen, wird oft die UCT-Formel (Oberes Konfidenzintervall für Bäume, engl.: *upper confidence bound applied to trees*) herangezogen:

$$\text{UCT} = \frac{R_{s,j}}{n_{b,j}} + c \cdot \sqrt{\frac{\ln n_b}{n_{b,j}}}, \quad (5.1)$$

wobei  $n_b$  die Anzahl der Besuche des derzeitigen Knotens,  $n_{b,j}$  und  $R_{s,j}$  die Besuche bzw. die summierte Erfolgsabschätzung des Kinderknotens durch Auswahl der  $j$ -ten Aktion ist. Mittels des Parameters  $c > 0$  kann der Fokus der Selektion mehr auf die Ausbeutung oder mehr auf die Erforschung gelegt werden. Angefangen beim Wurzelknoten des Baums wird der Folgezustand zur Aktion mit dem höchsten UCT-Wert gewählt. Im selben Schema wird durch die Ebenen des Baums iteriert, bis eine Aktion ausgewählt wird, zu der noch

<sup>11</sup>Bei diskreten Entscheidungsproblemen kann zwischen einer endlichen Anzahl an Aktionen gewählt werden. Beispiele sind z.B. Brettspiele wie Tic-Tac-Toe, Go oder Schach.

<sup>12</sup>Es wird somit ein Maximierungsproblem gelöst.



**Abbildung 5.2:** Überblick über die vier Phasen einer Iteration der Monte-Carlo Baumsuche. Bild nach [19].

kein Folgeknoten im Baum existiert. Dieses Vorgehen wird dabei Baumstrategie genannt.

## II. Expansion:

Zur ausgewählten Aktion aus Schritt I wird der passende Kinderknoten erstellt.

## III. Simulation:

Der Erfolg der ausgewählten Aktion wird durch eine Simulation abgeschätzt. Dabei werden nach einer definierter Standardstrategie<sup>13</sup> Aktionen ausgewählt und simuliert bis ein Zustand erreicht wird, an welchem keine weiteren Aktionen zur Verfügung stehen (Spielende). Dieser Endzustand wird bewertet und an den nächsten Schritt übergeben.

## IV. Rückführung:

Das geschätzte Ergebnis  $R$  der Simulation wird vom ausgewählten Knoten bis zum Wurzelknoten durch den Baum zurückgeführt. Hierzu wird bei jedem dieser Knoten die Anzahl der Besuche  $n_b$  sowie die Erfolgsabschätzung wie folgt aktualisiert:

$$\begin{aligned} n'_b &= n_b + 1, \\ R'_s &= R_s + R. \end{aligned}$$

Sobald eine definierte Zeit  $t_{\text{calc}}$  verstrichen ist oder eine bestimmte Anzahl an Iterationen ausgeführt wurde, endet der Algorithmus. Es wird vom

<sup>13</sup>Die einfachste Standardstrategie ist das zufällige Auswählen von Aktionen.



Wurzelknoten aus die Aktion gewählt, welche am erfolgversprechendsten<sup>14</sup> ist. Für eine Zusammenfassung der Eigenschaften sowie möglichen Modifikationen wird auf [19] verwiesen.

Im Folgenden werden die Anpassungen, welche für die CarPre-Trajektorienplanung benötigt werden, vorgestellt.

### 5.3.1 Diskretisierung des Suchraumes

Das Problem der Trajektorienplanung ist nach Abschnitt 2.2 ein kontinuierliches Planungsproblem. Für die Monte-Carlo Baumsuche müssen allerdings diskrete Aktionen vorliegen.<sup>15</sup> Daher wird im Folgenden der Aktions- sowie Zustandsraum diskretisiert. Damit nur physikalisch mögliche Trajektorien geplant werden können, wird die Fahrzeugbewegung durch das erweiterte kinematische Einspurmodell beschränkt (vgl. Gleichungen (4.1) bis (4.5)), sodass eine Aktion aus einem Beschleunigungswert sowie einer Lenkwinkelgeschwindigkeit besteht. Der Referenzpunkt des Modells wird, wie in Kapitel 3 motiviert, auf die Vorderachse gelegt. Dabei kann das Fahrzeug nur vorwärtsfahren ( $v \geq 0$ ). Der Suchbaum wird in äquidistanten Zeitschritten aufgebaut, d.h. um von einem in den nächsten Zustand zu kommen wird die gewählte Aktion für die Abtastzeit  $T_{\text{in}}$  ausgeführt. Die folgende Diskretisierung hängt demnach von dieser gewählten Abtastzeit  $T_{\text{in}}$  ab.

Zuerst werden die longitudinalen Zustände (d.h. Beschleunigung und Geschwindigkeit) diskretisiert. Die Menge der diskreten Beschleunigungswerte wird aus dem Diskretisierungsabstand  $\Delta a$  sowie der minimalen  $a_{\text{min}}$  und der maximalen Beschleunigung  $a_{\text{max}}$  zu

$$a_{\text{d}} \in \{a_{\text{min}} \leq p \cdot \Delta a \leq a_{\text{max}} \mid p \in \mathbb{Z}\}, \quad (5.2)$$

bestimmt. Durch die feste Abtastzeit  $T_{\text{in}}$  sowie  $a_{\text{d}}$  als Vielfaches von  $\Delta a$  ergibt sich für die Geschwindigkeit:

$$v_{\text{d}} \in \{p \cdot \Delta a T_{\text{in}} \leq v_{\text{max}} \mid p \in \mathbb{N}_0\},$$

<sup>14</sup>Dies kann z. B. der Knoten mit dem bestem durchschnittlichen Simulationsergebnis  $R_{\text{m}}$  oder der Knoten mit den meisten Besuchen  $n_{\text{b}}$  sein.

<sup>15</sup>Es existieren auch kontinuierliche MCTS-Ansätze, wie z. B. [22, 24, 75]. Durch die unendliche Anzahl an Aktionen ist das Planungsproblem sehr komplex. Da das hier vorliegende Planungsproblem in Echtzeit gelöst werden soll, wird in dieser Arbeit der Suchraum diskretisiert.

bis eine Maximalgeschwindigkeit  $v_{\max}$  erreicht ist. Durch eine passende Wahl von  $\Delta a$  können gängige Verhaltensmuster<sup>16</sup> dargestellt werden. Eine passende Wahl ist hierfür z. B.  $a_{\min} = -3 \text{ m/s}^2$ ,  $a_{\max} = 2 \text{ m/s}^2$  und  $\Delta a = 1 \text{ m/s}^2$ .

Für die Zustände der lateralen Bewegung (d.h. Lenkwinkel und Lenkwinkelgeschwindigkeit) wird auf die Lenkwinkel-Transformation aus Abschnitt 4.4.1 zurückgegriffen (Gleichung (4.10)). So wird sichergestellt, dass die lateralen Beschleunigungen bei höheren Geschwindigkeiten nicht zu groß werden. Für den geschwindigkeitsabhängigen Lenkwinkel  $\delta_d(v_d)$  mit  $n_\delta \in \{2p - 1 \mid p \in \mathbb{N}\}$  diskretisierten Zuständen ergibt sich

$$\delta_d(v_d) \in \left\{ q \cdot \Delta\delta(v_d) \mid q \in \mathbb{Z} \wedge |q| < \frac{n_\delta}{2} \right\}, \quad (5.3)$$

$$\text{mit } \Delta\delta(v_d) = \frac{2 \cdot \min \left( \arcsin(\kappa_{\max} l), \arcsin \left( \frac{a_{\text{lat,max}} l}{v_d^2} \right) \right)}{n_\delta - 1},$$

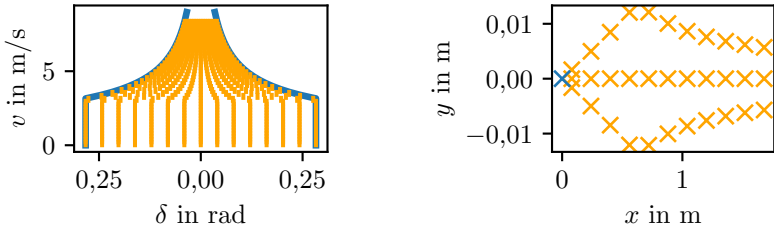
wobei  $\kappa_{\max}$  die maximal fahrbare Pfadkrümmung,  $a_{\text{lat,max}}$  die maximale laterale Beschleunigung sowie  $l$  der Radstand des Fahrzeugs ist. Eine beispielhafte Diskretisierung ist in Abb. 5.3a dargestellt. Die diskrete Lenkwinkelgeschwindigkeit  $\omega_d(\delta_{d,i}(v_d), a_d)$  mit  $n_\omega \in \{2p - 1 \mid p \in \mathbb{N}\}$  diskretisierten Zuständen wird aus dem aktuellen Lenkwinkel  $\delta_{d,i}(v_d)$  abgeleitet:

$$\omega_d(\delta_{d,i}(v_d), a_d) \in \left\{ q \cdot \frac{\delta_{d,i+q}(v_d + a_d T_{\text{in}}) - \delta_{d,i}(v_d)}{T_{\text{in}}} \mid \right. \\ \left. q \in \mathbb{Z} \wedge |q| < \frac{n_\omega}{2} \wedge 0 \leq i + q < n_\delta \right\}, \quad (5.4)$$

sodass für die lateralen Bewegungen ein um null symmetrischer Zustandsraum entsteht.

Die bisherige Diskretisierung führt zu einem Zustandsgitter, d.h. das Lösen der Bewegungsgleichungen (4.4) und (4.5) mit den diskretisierten Eingängen  $a_d$ ,  $\omega_d(\delta_{d,i}(v_d), a_d)$  sowie der Abtastzeit  $T_{\text{in}}$  führt ohne Diskretisierungsfehler zu den Zuständen  $v_d$ ,  $\delta_d(v_d)$ . Dies ist für die verbleibenden Zustände des kinematischen Fahrzeugmodells (Position, Orientierung) durch die nichtlinearen Funktionen in Gleichungen (4.1) bis (4.3) nicht möglich. Daher entsteht durch die Diskretisierung der Zustände ein

<sup>16</sup>Ein Beispiel hierfür sind die in [26, S. 116] dargestellten longitudinalen Verhaltensklassen: Notbremsen, komfortables Bremsen, Ausrollen, Geschwindigkeit halten, komfortables Beschleunigen sowie Notbeschleunigung.



(a) Diskretisierter Lenkwinkel  $\delta_d(v_d)$  (gelb) in Abhängigkeit der Geschwindigkeit. Die blaue Kurve stellt die Lenkwinkeltransformation aus Gleichung (4.10) dar und beschränkt  $\delta_d(v_d)$ , sodass eine um null symmetrische Lenkwinkeldiskretisierung entsteht.

(b) Erreichbare Positionen (gelb) aus einer Startposition (blau) mit unterschiedlichen Startgeschwindigkeiten  $v_{\text{start}} \in [0.4 \text{ m/s}, 8.4 \text{ m/s}]$  sowie  $\psi_{\text{start}} = 0$ ,  $\delta_{\text{start}} = 0$ . Die Aktionen werden zu  $a_d = 0$  und  $\omega_d(\delta_{d,i}(v_d), a_d) \in \{-\Delta\delta(v_d) T_{\text{in}}, 0, \Delta\delta(v_d) T_{\text{in}}\}$  gewählt.

**Abbildung 5.3:** Geschwindigkeitsabhängige Diskretisierung des Lenkwinkels  $\delta_d(v_d)$  mit den exemplarisch gewählten Parametern  $T_{\text{in}} = 0,2 \text{ s}$ ,  $v_{\text{max}} = 8,4 \text{ m/s}$ ,  $\kappa_{\text{max}} = 0,1 \text{ m}^{-1}$ ,  $a_{\text{lat,max}} = 1 \text{ m/s}^2$ ,  $l = 2,791 \text{ m}$ ,  $n_\delta = 12$ ,  $n_\omega = 3$ .

Diskretisierungsfehler, welcher nur vermieden werden kann, wenn die jeweiligen Zustände im kontinuierlichen Wertebereich bleiben. Dies ist generell möglich, da für die Baumsuche nur diskrete Aktionen und keine diskreten Zustände benötigt werden. Im Folgenden wird eine optionale Diskretisierung der Orientierung  $\psi$  vorgestellt, bei der  $\psi$  in  $n_\psi$  Zustände diskretisiert wird:

$$\psi_d \in \left\{ p \cdot \frac{2\pi}{n_\psi} < 2\pi \mid p \in \mathbb{N}_0 \right\}.$$

Diese Diskretisierung kann, je nach Computer, Rechenleistung beim Erstellen des Suchbaumes einsparen. Hierzu werden im Vorfeld für alle möglichen Zustände die kinematischen Bewegungsgleichungen gelöst und in einer Lookup-Tabelle abgespeichert. Durch die Diskretisierung der Orientierung wächst die Größe der Lookup-Tabelle um den Faktor  $n_\psi$ , im Gegenzug müssen die in der Lookup-Tabelle abgespeicherten Werte nicht mehr um die Fahrzeugorientierung gedreht werden.

Die Position des Fahrzeugs wird nicht diskretisiert, da sie nur ein Ergebnis der Bewegungsgleichungen ist und keine weiteren Gleichungen von ihr abhängen. Dies ist in Abb. 5.3b dargestellt. Durch das Lösen der Bewegungsgleichungen (4.1) – (4.5), z. B. mit dem Runge-Kutta Verfahren 4. Ordnung, werden kontinuierliche Positionen bestimmt. Ebenso lässt sich der Einfluss der Lenkwinkeltransformation beobachten. Obwohl die Start-

geschwindigkeiten variieren, ist die laterale Bewegung (hier: in  $y$ -Richtung) beschränkt.

Anhand der oben beschriebenen Diskretisierung kann ein Suchbaum konstruiert werden. Bei der Auswahl der Diskretisierungsparameter ist zu beachten, dass sie maßgeblich die Komplexität des Planungsraumes festlegt. Dabei sind vor allem die Anzahl an Eingängen  $n_{\text{in}} = n_a n_\omega$  sowie die Abtastzeit  $T_{\text{in}}$  entscheidend, da diese die Anzahl an möglichen Aktionen innerhalb eines Planungshorizontes  $T_{\text{hor}} \in \{p \cdot T_{\text{in}} \mid p \in \mathbb{N}\}$  exponentiell ansteigen lassen (vgl. Fluch der Dimension [10]):

$$n_{\text{in,ges}} = (n_a n_\omega)^{\frac{T_{\text{hor}}}{T_{\text{in}}}}. \quad (5.5)$$

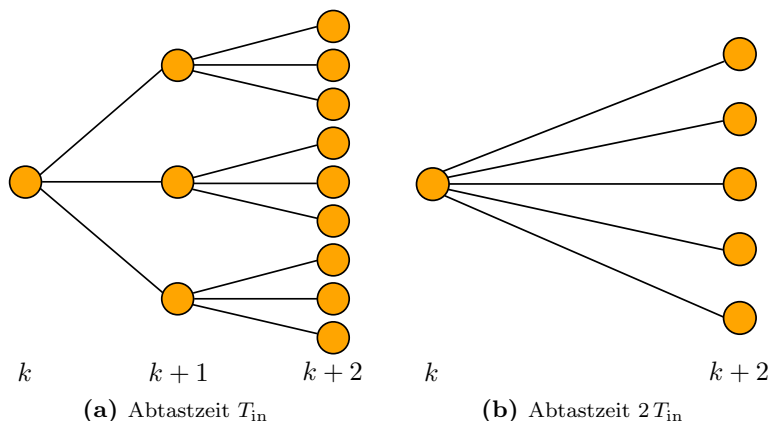
Um dieses exponentielle Wachstum abzuschwächen, kann die Anzahl der möglichen Aktionen in Abhängigkeit der zuletzt gewählten Aktion beschränkt werden. Ein Beispiel hierfür ist die Beschränkung der Beschleunigung  $a_k$  in Abhängigkeit der letzten Beschleunigung  $a_{k-1}$ :

$$a_k \in \{a_{k-1} - \Delta a; a_{k-1}; a_{k-1} + \Delta a\}, \quad (5.6)$$

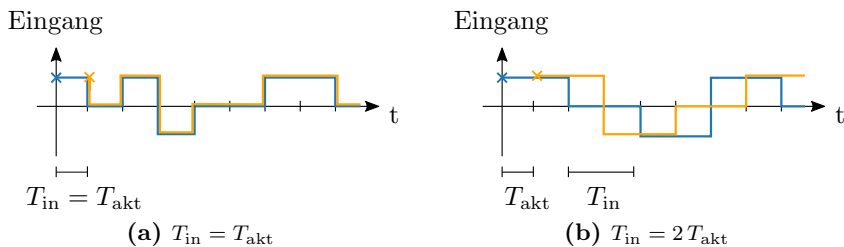
sodass die Anzahl an möglichen Beschleunigungsaktionen  $n_a$  auf drei reduziert wird. Diese Beschränkung gleicht einer Limitierung des Rucks, erhöht somit den Fahrkomfort und reduziert gleichzeitig die Planungskomplexität.

Eine weitere Möglichkeit zur Komplexitätsreduktion ist in Abb. 5.4 dargestellt. Durch Erhöhung der Abtastzeit wird das exponentielle Wachstum verlangsamt (vgl. Gleichung (5.5)). Hierdurch wird ebenso der Aktionsraum des Planungsproblems eingeschränkt. Dies kann dazu führen, dass keine Lösung des Planungsproblems mehr gefunden wird, obwohl eine Lösung im kontinuierlichen Raum existiert. Um dem entgegenzuwirken kann, wie im Beispiel, die Anzahl an Aktionen pro Zeitschritt erhöht werden. Das Ziel ist letztendlich eine möglichst große Abtastzeit mit möglichst wenigen Aktionen zu finden, ohne den Aktionsraum des Planungsproblems zu sehr einzuschränken.

Bei einer Erhöhung der Abtastzeit zur Komplexitätsreduktion sollte außerdem die Aktualisierungszeit  $T_{\text{akt}}$ , also die Zeit zwischen zwei aufeinanderfolgenden Planungen, betrachtet werden. Wie in Abschnitt 4.3 beschrieben, beeinflusst  $T_{\text{akt}}$  maßgeblich die Fähigkeit, auf andere dynamische Objekte in der Umgebung zu reagieren. Wird die Abtastzeit groß gewählt und das System soll weiterhin schnell auf dynamische Objekte reagieren können, muss  $T_{\text{akt}} < T_{\text{in}}$  gewählt werden. Dabei ergibt sich das in Abb. 5.5 dargestellte Problem. Während für den Fall  $T_{\text{in}} = T_{\text{akt}}$  das



**Abbildung 5.4:** Vereinfachtes Beispiel zur Reduktion der Planungskomplexität. Während in (a) ein Entscheidungsbaum einer gewählten Abtastzeit vorliegt, wird im Entscheidungsbaum (b) die Abtastzeit verdoppelt. Der durch die Verdopplung eingeschränkte Aktionsraum wird durch Erhöhung der möglichen Aktionen von drei auf fünf Aktionen pro Knoten kompensiert. Die fehlenden vier Aktionen ergeben die Komplexitätsreduktion und sind z. B. Aktionskombinationen die nicht relevant sind. Für dieses Beispiel ergeben sich für einen Baum bis  $T_{hor} = 10 T_{in}$  für (a) 59049 mögliche Aktionskombinationen während für (b) nur 3125 Kombinationen möglich sind.



**Abbildung 5.5:** Einfluss des Verhältnisses der Abtastzeit  $T_{in}$  zur Aktualisierungszeit  $T_{akt}$  auf die Trajektorienplanung. (a) Ist  $T_{in} = T_{akt}$  kann in einem nächsten Planungsschritt dieselbe Eingangsabfolge (orange) und damit dieselbe Trajektorie geplant werden wie im vorherigen Planungsschritt (blau). (b) Ist dagegen  $T_{in} > T_{akt}$  ist die Planung derselben Eingangsabfolge in einem nächsten Planungsschritt ausgeschlossen. Die geplante Trajektorie wird sich von der vorherigen unterscheiden.

Ergebnis der Neuplanung exakt der Planung des vorherigen Zeitschritts entsprechen kann, ist dies für  $T_{\text{akt}} < T_{\text{in}}$  nicht gegeben. Durch die erhöhte Abtastzeit wird sich das Ergebnis nach  $T_{\text{akt}}$  zwingend unterscheiden<sup>17</sup> (vgl. Abb. 5.5b).

### 5.3.2 Definition des Spielendes

Das Spielende gibt Zustände vor, an denen keine weiteren Aktionen mehr ausgewählt werden können. Dies betrifft sowohl Zustände im Entscheidungsbaum als auch Zustände bei der Simulation zur Erfolgsabschätzung. Bei Brettspielen ist ein solches Spielende z. B. durch die Spielregeln bestimmt. Im Falle der Trajektorienplanung kann das Erreichen eines Zielzustandes ein Spielende sein. Mit Hilfe des kinematischen Bewegungsmodells können immer weitere Aktionen gewählt werden, bis ein solcher Zielzustand erreicht wird. In der Regel liegt dieser Zielzustand allerdings weiter entfernt und wird daher nur durch die Routenplanung verfolgt (vgl. Abschnitt 2.1). Somit müssen weitere Endzustände für den Trajektorienplaner definiert werden:

1. Erreichen eines festgelegten Zeithorizonts  $t = T_{\text{hor}}$  (auch: Planungshorizont).
2. Verlassen des Bereichs der verfügbaren Umgebungsinformationen, sodass keine weitere Planung möglich ist.
3. Verursachen einer Kollision mit anderen Objekten.
4. Anhalten des Fahrzeugs für  $t > 0$ .

Die Limitierung des Planungshorizonts in Punkt 1 gleicht hierbei dem Prinzip des fortschreitenden Zeitfensters (engl.: *receding horizon control* [82]), welches in der modellprädiktiven Regelung eingesetzt wird. Die Planung betrachtet somit nur ein begrenztes Zeitfenster zur Bestimmung der nächsten Trajektorie. Nach der Berechnung einer Trajektorie wird der erste Teil dieser ausgeführt, bevor eine neue Planung mit einem verschobenen Zeitfenster gestartet wird.

---

<sup>17</sup>Die einzige Ausnahme hierfür ist, dass der Eingang in beiden Planungen über den gesamten Planungshorizont konstant auf demselben Wert gehalten wird.

### 5.3.3 Auswahl des Startpunktes

Die geplante Trajektorie wird, wie im vorherigen Abschnitt beschrieben, nach dem Prinzip des fortschreitenden Zeitfensters zyklisch neu berechnet. Ist eine Aktualisierungszeit  $T_{\text{akt}}$  verstrichen, wird neu geplant und im Anschluss das Ergebnis an einen Trajektorienfolgeregler weitergegeben. Damit der Regler aus der übergebenen Trajektorie die benötigten Stellgrößen berechnen kann, muss die Trajektorie stetig sein (vgl. [142, S. 55]). Dies bedeutet, dass der Startpunkt (auch: Wurzelknoten des Suchbaums) der nächsten Planung auf der zuletzt geplanten Trajektorie liegen muss. Wäre dies nicht der Fall und die Planung würde dagegen im aktuellen Fahrzeugzustand starten, würde der Eingang des Reglers dauerhaft Sollgrößensprünge enthalten. Dies kann dazu führen, dass Planer und Regler gegeneinander arbeiten, sodass das System aufschwingt und instabil wird.

Damit ein solches Verhalten nicht auftritt, muss der Startzustand der nächsten Planung abhängig von der Aktualisierungszeit  $T_{\text{akt}}$  sowie der Berechnungszeit aller Algorithmen<sup>18</sup>  $t_{\text{verarb}}$  aus der letzten Trajektorie gewählt werden. Es ergibt sich folgende Bedingung

$$n_{\text{start}} > \frac{t_{\text{verarb}}}{T_{\text{akt}}},$$

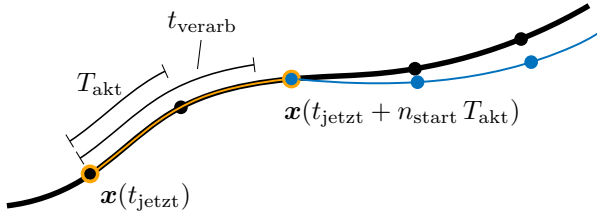
wobei  $n_{\text{start}}$  die benötigte Mindestanzahl an Zeitschritten für eine stetige Trajektorie ist (vgl. Abb. 5.6). Somit besteht die nächste Trajektorie aus einem kurzen Stück der vorhergehenden Trajektorie (orange) sowie der neu geplanten Trajektorie (blau).

### 5.3.4 Standardstrategie

Um eine gewählte Aktion besser bewerten zu können, wird mittels der Simulation ein Erfolg abgeschätzt. Damit in dieser nicht zufällig Kombinationen aus Beschleunigungs- und Lenkwinkelgeschwindigkeitswerten gezogen werden müssen, wird im Folgenden eine Standardstrategie vorgestellt, welche aus der Analyse menschlicher Fahrdaten aus Kapitel 4 abgeleitet wird. Bei der Analyse können zwei Hauptverhaltensmuster festgestellt werden. Erstens versuchen menschliche Fahrer ihrer Fahrspur zu folgen (vgl. Abb. 5.7). Zweitens wird in der Regel nur über einen längeren Zeitraum beschleunigt (vgl. Abb. 4.2 und 4.10). Daher wird das Standardverhalten wie folgt definiert:

---

<sup>18</sup>Dies umfasst sowohl die Berechnungszeit des Planungsalgorithmus als auch die Zeit zur Vorverarbeitung der Eingangsdaten.



**Abbildung 5.6:** Auswahl des Startzustands der Trajektorienplanung. Die schwarze Linie stellt die zuletzt geplante Trajektorie dar. Damit der Regler des Fahrzeugs eine stetige Trajektorie erhält, muss der Startzustand auf dieser letzten Trajektorie und hierbei mindestens  $n_{\text{start}}$ -Zeitschritte in der Zukunft liegen, sodass die benötigte Berechnungszeit  $t_{\text{verarb}}$  der Trajektorie kleiner als die Zeit zum neuen Startpunkt  $n_{\text{start}} T_{\text{akt}}$  ist. Die Trajektorie besteht somit aus einem nicht veränderbaren Teil der letzten Trajektorie (orange), sowie einem neugeplanten Teil (blau). Das Bild ist an [142, S. 55] angelehnt.

- **Beschleunigung:**

Der vorherige Beschleunigungswert wird für die nächsten  $k_{\text{acc}}$ -Zeitschritte erneut gewählt (Modell der konstanten Beschleunigung). Im Anschluss werden Beschleunigungswerte um null (d.h.  $a_d \in \{-\Delta a; \Delta a\}$ ) auf  $a_d = 0$  gesetzt.

- **Lenkwinkelgeschwindigkeit:**

Die Lenkwinkelgeschwindigkeit wird für die nächsten  $k_{\omega}$ -Zeitschritte auf null gesetzt. Im Anschluss wird die Lenkwinkelgeschwindigkeit anhand einer Richtungskarte so gewählt, dass das Fahrzeug dem Straßenverlauf möglichst folgt. Eine solche Richtungskarte ist in Abb. 5.8 dargestellt.

### 5.3.5 Initialisierung der Knotenerfolgswerte

Nach der UCT-Formel (5.1) wird an einem Knoten jede Aktion bevorzugt, die noch nie ausgewählt wurde, da

$$\lim_{n_{b,j} \rightarrow 0} \text{UCT} = \lim_{n_{b,j} \rightarrow 0} \left( \frac{R_{s,j}}{n_{b,j}} + c \cdot \sqrt{\frac{\ln n_b}{n_{b,j}}} \right) = \infty.$$

Dies bedeutet, dass zuerst alle möglichen Aktionen eines Knotens exploriert werden, bevor ein Knoten der nächsten Bauebene exploriert werden



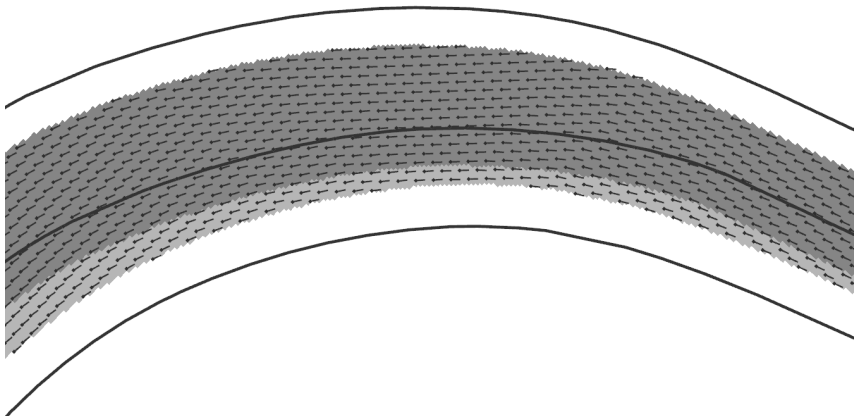


(a) Pfade Geradeausfahrten.



(b) Pfade Kurvenfahrten.

**Abbildung 5.7:** Selektierte Pfadverläufe (blau) menschlicher Autofahrer. Die Autofahrer halten die Spur, welche sie gemäß ihrer Routenplanung zu ihrem Ziel bringt. Das Standardverhalten der Autofahrer ist somit meist die Spur zu halten. Die Pfadverläufe sowie das Hintergrundbild stammen aus dem inD-Datensatz [12]. [140]



**Abbildung 5.8:** Diskretisierte Richtungskarte (schwarze Richtungszeiger) zur Auswahl der Lenkwinkelgeschwindigkeiten zum Spurhalten. Ein solches Vektorfeld kann anhand einer Straßenkarte bzw. eines sensorbasierten Umfeldmodells erstellt werden (schwarze Linien). Die Fahrbahnbereiche sind in die eigene Fahrbahn (hellgrau) sowie die entgegengesetzte Fahrbahn (dunkelgrau) eingeteilt und werden um den Platzbedarf des Ego-Fahrzeugs verkleinert (vgl. Abschnitt 3.4.2). [140]

kann. Bei begrenzter Rechenzeit kann dies dazu führen, dass der Suchbaum nicht die benötigte Tiefe erreicht. Um die Konvergenz des Algorithmus zu beschleunigen, wird eine Initialisierung der Knotenerfolgswerte

$R_{s,j} = R_{\text{ini}}$  vorgenommen. Ebenso wird die Anzahl der Besuche des Knotens auf  $n_{b,j} = 1$  gesetzt. Anhand der Richtungskarte wird die beste Lenkwinkelgeschwindigkeit  $\omega_{\text{best}}$  zum Spurhalten für den derzeitigen Zustand ermittelt und dementsprechend die Initialisierung  $R_{\text{ini}}$  zu

$$R_{\text{ini}} = \begin{cases} R_{\text{ini},1} & \text{für } \omega_k = \omega_{\text{best}} \text{ und } a_k = a_{k-1}; \\ R_{\text{ini},2} & \text{für } \omega_k \neq \omega_{\text{best}} \text{ und } a_k = a_{k-1}; \\ 0 & \text{sonst.} \end{cases}$$

gesetzt, wobei  $0 \leq R_{\text{ini},2} < R_{\text{ini},1} \leq 1$ . Mit dieser Wahl wird eine erste Präferenz zukünftiger Aktionen gesetzt, sodass bei der Exploration Aktionen mit gleichen Beschleunigungen sowie Lenkwinkelgeschwindigkeiten zum Folgen der Straße bevorzugt werden. Bei der Auswahl der jeweiligen Aktion wird standardmäßig bis zum Spielende simuliert und die Erfolgsabschätzung zurückgeführt. Durch die Mittelwertbildung verliert der initiale Wert mit steigenden Durchläufen an Einfluss und nähert sich dem realen Erfolg an.

### 5.3.6 Erfolgskfunktion

Wird ein Spielende erreicht, muss die bis dahin geplante Trajektorie bewertet werden, um eine Abschätzung der gewählten Aktion zu ermöglichen. Um eine eindeutige Priorisierung der Planungsziele festzulegen, wird eine um jeweils eine Größenordnung unterschiedliche Erfolgskfunktion  $R(\mathbf{x}_k)$  für jeden Zustand  $\mathbf{x}_k$  der Trajektorie gewählt:

$$R(\mathbf{x}_k) = \frac{1}{\sum_{i=0}^{m-1} b_{\text{R}}^i} \sum_{i=0}^{m-1} b_{\text{R}}^i R_i(\mathbf{x}_k),$$

mit den  $m$  Erfolgstermen  $R_i(\mathbf{x}_k) \in [0, 1]$  und der Basis der Größenordnung  $b_{\text{R}}$ . Die Gesamtbewertung der Trajektorie  $R_{\text{traj}}$  wird als die normierte Summe aller  $n_{\text{traj}} = \frac{T_{\text{hor}}}{T_{\text{in}}}$  möglichen Zustände zusammengefasst:

$$R_{\text{traj}} = \frac{1}{n_{\text{traj}}} \left( \sum_{k=1}^p R(\mathbf{x}_k) + (n_{\text{traj}} - p) R(\mathbf{x}_p) \right),$$

wobei  $p \leq n_{\text{traj}}$  die Anzahl der zu bewertenden Trajektorienpunkte beschreibt. Wird die Simulation durch das oben beschriebene Spielende frühzeitig ( $p < n_{\text{traj}}$ ) beendet, wird die Zustandsbewertung des Endzustands  $\mathbf{x}_p$  so häufig wiederholt, bis  $n_{\text{traj}}$  Bewertungen in die Trajektorienbewertung eingeflossen sind. Dies ist wichtig, damit jeder Zustand  $\mathbf{x}_k$  in jeder Simulation gleich gewichtet wird.

Im Folgenden wird eine beispielhafte Priorisierung mit zugehörigen Erfolgskfunktionen vorgestellt:

1. Vermeiden einer Kollision

$$R_{\text{koll}}(\mathbf{x}_k) = \begin{cases} 1 & \text{falls keine Kollision;} \\ 0 & \text{falls Kollision.} \end{cases}$$

2. Halten der eigenen Fahrspur

$$R_{\text{spur}}(\mathbf{x}_k) = \begin{cases} 1 & \text{falls eigene Fahrspur;} \\ R_{\text{spur,g}} & \text{falls entgegengesetzte Fahrspur;} \\ 0 & \text{falls fahrbaren Bereich verlassen,} \end{cases}$$

mit dem Erfolgswert  $R_{\text{spur,g}} \in [0, 1]$  für die entgegengesetzte Fahrspur.

3. Erreichen der Zielgeschwindigkeit

$$R_v(\mathbf{x}_k) = \begin{cases} 1 - \frac{|v_{\text{ziel}} - v|}{v_{\text{max}}} & \text{falls im fahrbaren Bereich;} \\ 1 - \frac{|v|}{v_{\text{max}}} & \text{falls Kollision oder Verlassen} \\ & \text{des fahrbaren Bereichs.} \end{cases}$$

4. Vermeiden von starken Beschleunigungen

$$R_a(\mathbf{x}_k) = \begin{cases} 1 - \frac{|a|}{a_{\text{max}}} & \text{für } |a_{\text{min}}| < a_{\text{max}}; \\ 1 - \frac{|a|}{|a_{\text{min}}|} & \text{sonst.} \end{cases} \quad (5.7)$$

5. Minimierung des Abstands zur Fahrspurmitte  $d_{\text{mitte}}$

$$R_{\text{mitte}}(\mathbf{x}_k) = \begin{cases} 1 - \frac{d_{\text{mitte}}}{d_{\text{max}}} & \text{für } d_{\text{mitte}} < d_{\text{max}}; \\ 0 & \text{sonst.} \end{cases}$$

mit dem Maximalabstand  $d_{\text{max}}$  als obere Grenze der Abstandsminimierung.

6. Fahren in Richtung der Straße

$$R_{\psi}(\mathbf{x}_k) = 1 - \frac{|\Delta\theta|}{\pi},$$

mit der Differenz  $\Delta\theta$  zwischen der Bewegungsrichtung  $\theta = \psi + \delta$  des Fahrzeugs sowie der Richtung  $\theta_{\text{karte}}$  der Richtungskarte.

Wie an den beispielhaften Erfolgskfunktionen zu sehen ist, können diskrete Variablen, wie z. B. die binäre Kollisionsvariable, direkt in den Erfolgstermen berücksichtigt werden. So kann die Zielgeschwindigkeit z. B. davon abhängen, ob das Fahrzeug kollidiert ist oder nicht. Schließlich sollte im Fall, dass eine Kollision unvermeidbar ist, das Ziel sein, sich möglichst im Stillstand zu befinden. Diese Eigenschaften, also die Auswertung komplexer Erfolgsterme mit diskreten Entscheidungsvariablen kombiniert mit der Gewichtung unterschiedlicher Größenordnung führt zu einer flexiblen Vorgabe des gewünschten Verhaltens. Daher lässt sich in der CarPre-Trajektorienplanung eine verhaltenssemantische Szeneriebeschreibung (vgl. [42, 43]) integrieren, um die Verhaltensregeln der Straßenverkehrsordnung umzusetzen. Eine solche verhaltenssemantische Szeneriebeschreibung weist visuellen Indikatoren, wie z. B. Straßenschildern oder Fahrbahnmarkierungen, Verhaltensattribute zu, welche dann vom autonomen Fahrzeug umgesetzt werden.

### 5.3.7 Bestimmung von Kollisionen

Die Überprüfung von Kollisionen nimmt nach [118, 123] einen erheblichen Anteil der Planungsrechenzeit in Anspruch und muss bei der CarPre-Trajektorienplanung in jedem Zeitschritt des Suchbaumes sowie der Simulationen durchgeführt werden. Durch die sehr große Anzahl an benötigten Kollisionsüberprüfungen sollte der Ansatz möglichst wenig Rechenoperationen benötigen. Hierzu bietet sich die rotationsinvariante Abstandsbestimmung zweier Kreise an. Daher wird auf den Ansatz aus Abschnitt 3.4.1 zurückgegriffen, bei dem das autonome Fahrzeug als ein einzelner Kreis approximiert wird. Andere Verkehrsteilnehmer werden in Objektkreise zerlegt (siehe Abb. 5.9). Eine Kollision des Ego-Fahrzeugs an der Position  $w\mathbf{p}_{\text{ref}}$  ist nicht gegeben, falls für alle Objektkreise gilt:

$$d_{\text{sicher}} + r_{\text{ego}} + r_{\text{obj},i} < \left\| \begin{bmatrix} wx_{\text{ref}} \\ wy_{\text{ref}} \end{bmatrix} - \begin{bmatrix} wx_{\text{obj},i} \\ wy_{\text{obj},i} \end{bmatrix} \right\|_2, \quad (5.8)$$

mit dem Sicherheitsabstand  $d_{\text{sicher}}$ , dem Radius des Ego-Fahrzeugkreises  $r_{\text{ego}}$  und dem Radius  $r_{\text{obj},i}$  sowie der Position  $w\mathbf{p}_{\text{obj},i}$  des  $i$ -ten Objektkreises.

Die Zerlegung der Verkehrsteilnehmer in Objektkreise variiert je nach Form und Größe des Verkehrsteilnehmers. Hierzu wird angenommen, dass ein Objekt durch ein Begrenzungsrechteck (engl.: *bounding box*) mit der Länge  $l_{\text{obj}}$  und der Breite  $b_{\text{obj}}$  dargestellt wird, wobei  $l_{\text{obj}} \geq b_{\text{obj}}$ . Der Referenzpunkt des objektfesten Koordinatensystems  $\circ\mathcal{F}$  liegt im Mittelpunkt

des Rechtecks. Unterschreiten die Objektmaße eine Mindestlänge  $l_{\text{obj},\min}$  bzw. -breite  $b_{\text{obj},\min}$ , wird das Objekt vergrößert:

$$\begin{aligned}\overline{l_{\text{obj}}} &= \max(l_{\text{obj},\min}, l_{\text{obj}}), \\ \overline{b_{\text{obj}}} &= \max(b_{\text{obj},\min}, b_{\text{obj}}).\end{aligned}\quad (5.9)$$

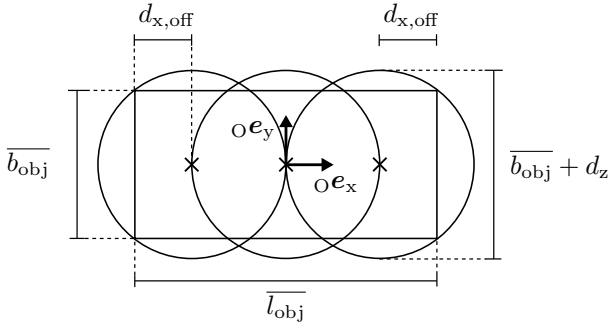
Ansonsten kann für sehr schmale Objekte ( $l_{\text{obj}} \gg b_{\text{obj}}$ ) die Anzahl an Kreisen sehr groß werden.

Im Anschluss kann das Rechteck in  $n_{\text{obj}}$  Kreise mit jeweils einem Radius  $r_{\text{obj}}$  zerlegt werden, siehe Abb. 5.9. Hierbei wird zwischen zwei Fällen unterschieden. Ist das Verhältnis zwischen Länge und Breite kleiner als ein gewisses Mindestverhältnis  $p_{\min}$  wird ein einzelner Kreis in das Objekt eingepasst, welcher durch alle vier Eckpunkte des Objektes verläuft. Ist das Verhältnis dagegen größer, werden mindestens zwei Kreise in das Objekt gelegt. Der Radius dieser Kreise wird durch die Objektbreite  $b_{\text{obj}}$  sowie einer Zusatzlänge  $d_z$  bestimmt, sodass die Kreise über das Objekt herausragen. Die beiden äußeren Kreise werden so positioniert, dass sie durch die Eckpunkte des Objektes verlaufen. Der Abstand der Kreismittelpunkte zur Außenkante  $d_{x,\text{off}}$  beträgt daher in Längsrichtung

$$d_{x,\text{off}} = \frac{1}{2} \sqrt{d_z^2 + 2d_z \overline{b_{\text{obj}}}}.$$

Abhängig von der Länge des Objektes und einem Überlappungsfaktor  $p_{\text{ü}} \in [0, 1)$  werden weitere Kreise in gleichen Abständen eingepasst. Für die Anzahl der Kreise  $n_{\text{obj}}$ , deren Radien  $r_{\text{obj}}$  sowie deren Mittelpunktspositionen  $o_{x_{m,i}}$  mit  $i \in [0, 1, \dots, n_{\text{obj}} - 1]$  ergibt sich:

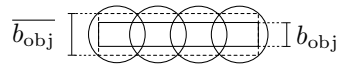
$$\begin{aligned}n_{\text{obj}} &= \begin{cases} 1 & \text{für } \frac{\overline{l_{\text{obj}}}}{\overline{b_{\text{obj}}}} < p_{\min}; \\ 2 + \max\left(0, \left\lceil \frac{\overline{l_{\text{obj}}} - 2d_{x,\text{off}}}{(d_z + \overline{b_{\text{obj}}})(1 - p_{\text{ü}})} \right\rceil - 1\right) & \text{sonst.} \end{cases}, \\ r_{\text{obj}} &= \begin{cases} \sqrt{\left(\frac{\overline{l_{\text{obj}}}}{2}\right)^2 + \left(\frac{\overline{b_{\text{obj}}}}{2}\right)^2} & \text{für } \frac{\overline{l_{\text{obj}}}}{\overline{b_{\text{obj}}}} < p_{\min}; \\ \frac{d_z + \overline{b_{\text{obj}}}}{2} & \text{sonst.} \end{cases}, \\ o_{x_{m,i}} &= \begin{cases} 0 & \text{für } \frac{\overline{l_{\text{obj}}}}{\overline{b_{\text{obj}}}} < p_{\min}; \\ d_{x,\text{off}} - \frac{\overline{l_{\text{obj}}}}{2} + i \cdot \frac{\overline{l_{\text{obj}}} - 2d_{x,\text{off}}}{n_{\text{obj}} - 1} & \text{sonst.} \end{cases}.\end{aligned}$$



(a) Parameter der Objektzerlegung



(b) Zerlegung eines fast quadratischen Objekts.



(c) Zerlegung eines Radfahrers.

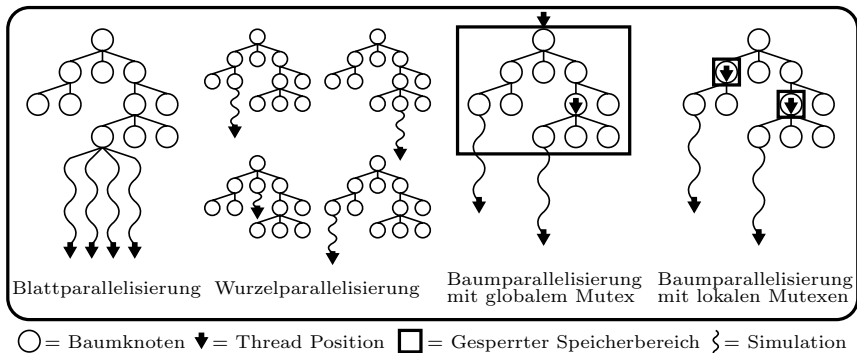
**Abbildung 5.9:** Kreiszerlegung von Rechtecken zur Kollisionsbestimmung. (a) stellt die Parameter sowie das Koordinatensystem der Zerlegung dar. In (b) wird ein nahezu quadratisches Objekt als ein einzelner Kreis dargestellt, während in (c) ein längliches Objekt (z. B. Radfahrer) in mehrere Kreise zerlegt wird. Hierfür wird das Objekt auf Grund der geringen Breite nach Gleichung (5.9) vergrößert (gestrichelte Linien) bevor die Kreise eingepasst werden.

Je nach gewähltem Überlappungsfaktor  $p_{\ddot{u}}$  überschneiden sich die Kollisionskreise eines Objektes mehr oder weniger stark. Ein zerlegtes Objekt wird daher nicht zwingend komplett von Kreisen abgedeckt. Das bedeutet, dass je nach gewähltem Faktor  $p_{\ddot{u}}$ , Abstand  $d_z$  sowie gewähltem Sicherheitsabstand  $d_{\text{sicher}}$  nicht alle Kollisionen erkannt werden. Indem das Ego-Fahrzeug seitlich in ein anderes Objekt fährt, wäre theoretisch eine nicht erkannte Kollision möglich. Durch eine geeignete Wahl von  $d_{\text{sicher}}$  sowie dem ohnehin schon überschätzten  $r_{\text{ego}}$  (vgl. Abschnitt 3.3.3) kann dies allerdings vermieden werden. Generell sollte ein zu weites Aufblähen der Objekte verhindert werden, da ansonsten dieser Ansatz die Praxistauglichkeit durch eine zu große Überschätzung der Objektgröße verliert.

### 5.3.8 Parallelisierung

Damit die vorhandene Rechenzeit möglichst optimal genutzt wird, wird der oben beschriebene Algorithmus auf  $n_{\text{threads}}$  Threads parallelisiert. Nach [21] existieren verschiedene Ansätze zur Parallelisierung, welche in Abb. 5.10 dargestellt sind. In dieser Arbeit wird die Baumparallelisierung mit lokalen Mutexen (engl.: *tree parallelization with local mutexes*) verwendet. Hier erkunden mehrere Threads gleichzeitig den Suchbaum, indem sie zeitgleich die Phasen I–IV des Algorithmus durchlaufen. Zur Synchronisierung der Daten zwischen den Threads wird in jedem Knoten ein Mutex (wechselseitiger Ausschluss) verwendet, sodass immer nur ein Thread gleichzeitig auf die Daten eines Knoten zugreifen kann.

Bei der Parallelisierung von Suchbäumen ist ein bekanntes Problem, dass der Suchbaum von einzelnen Threads oft ähnlich abgesucht wird. Dies liegt daran, dass die Selektion anhand einer deterministischen Funktion gewählt wird (vgl. Gleichung (5.1)). Als Lösung wird in [21] ein virtueller Verlust  $R_{\text{verlust}}$  (engl.: *virtual loss*) vorgeschlagen, welcher bei der Selektion eines Knotens vom Erfolgswert dessen abgezogen und während der Rückführung erneut hinzuaddiert wird. Dadurch werden andere Threads motiviert andere Baumpfade zu erkunden, um damit die Suche zu diversifizieren.



**Abbildung 5.10:** Mögliche Parallelisierungsmöglichkeiten der Monte-Carlo Baumsuche. Bild nach [21].

### 5.3.9 Extraktion der Trajektorie

Nachdem die definierte Berechnungszeit  $t_{\text{calc}}$  verstrichen ist, kann aus dem erstellten Suchbaum die neu geplante Trajektorie  $\Pi_k$  extrahiert werden. Um Schwankungen zwischen zwei aufeinanderfolgenden Trajektorien zu vermeiden, wird, falls vorhanden, auf die Aktionen der zuletzt geplanten Trajektorie  $\Pi_{k-1}$  zurückgegriffen. Beginnend beim Wurzelknoten des neu berechneten Suchbaums wird jeweils die  $j_{\text{best}}$ -te Aktion ausgewählt, welche den größten mittleren Erfolg  $R_{m,\text{best}}$  besitzt. Gleicht die bisher extrahierte Trajektorie noch der vorherigen Trajektorie<sup>19</sup>, wird außerdem der aktuelle Erfolgswert  $R_{m,\text{best,vor}}$  der vorherigen besten  $j_{\text{best,vor}}$ -te Aktion ausgewertet. Liegen diese beiden Erfolgswerte innerhalb einer zulässigen Abweichung  $\epsilon_R$ , wird die vorherige beste Aktion der aktuell besten Aktion vorgezogen. Es ergibt sich

$$j_{\text{best}} = \begin{cases} j_{\text{best,vor}} & \text{für } |R_{m,\text{best}} - R_{m,\text{best,vor}}| < \epsilon_R \text{ und bis-} \\ & \text{herige extrahierte } \Pi_k \text{ gleicht } \Pi_{k-1}; \\ j_{\text{best}} & \text{sonst.} \end{cases}$$

Dieses Vorgehen wird, beginnend beim Wurzelknoten, so oft ausgeführt bis entweder  $n_{\text{traj}}$  Trajektorienpunkte extrahiert werden oder ein Blattknoten, d.h. ein Knoten ohne weitere Kinderknoten, erreicht wird.

## 5.4 Evaluierung

Nach der Vorstellung der CarPre-Trajektorienplanung, folgt nun eine Evaluation des Algorithmus. Hierzu wird der Algorithmus in C++ implementiert und an den Open-Source-Simulator CARLA [35] angebunden. Der Algorithmus, der Simulator sowie zusätzliche Vorverarbeitungsmodul werden auf einem Intel Xeon E5-1620 v2 Prozessor (4 Kerne à 3,7 GHz, 8 Threads, 10MB Cache, Baujahr 2013) mit den in Tabelle 5.1 aufgelisteten Parametern ausgeführt. Zur Evaluation werden zwei Szenarien herangezogen, eine Spurfolgefahrt sowie eine Annäherung an ein statisches Objekt.

### 5.4.1 Spurfolgefahrt

Das erste Szenario ist die Spurfolgefahrt. Hier ist das Ziel des Planers, der eigenen Fahrspur zu folgen und diese nicht zu verlassen. Dabei soll

<sup>19</sup>Bei diesem Vergleich wird die Verschiebung  $n_{\text{start}}$  der Trajektorien nach Abschnitt 5.3.3 berücksichtigt. Es können nur so viele Aktionen verglichen werden, wie bereits aus dem aktuellen Suchbaum extrahiert wurden.



**Tabelle 5.1:** Parameter des CarPre-Trajektorienplaners.

Parameter	Wert	Parameter	Wert
$a_{\text{lat,max}}$	$1,3 \text{ m/s}^2$	$n_{\text{threads}}$	3
$a_{\text{max}}$	$1 \text{ m/s}^2$	$n_{\text{traj}}$	30
$a_{\text{min}}$	$-3 \text{ m/s}^2$	$n_{\delta}$	15
$\Delta a$	$1 \text{ m/s}^2$	$n_{\omega}$	3
$b_{\text{obj,min}}$	0,5 m	$n_{\psi}$	512
$b_{\text{R}}$	2	$T_{\text{akt}}$	0,2 s
$c$	0,5	$t_{\text{calc}}$	0,1 s
$d_{\text{max}}$	12,8 m	$T_{\text{hor}}$	6 s
$d_{\text{sicher}}$	0,2 m	$T_{\text{in}}$	0,2 s
$d_z$	0,2 m	$v_{\text{ziel}}$	8,4 m/s
$k_{\text{acc}}$	6	$r_{\text{ego}}$	1,31 m
$k_{\omega}$	2	$R_{\text{ini},1}$	0,1
$\kappa_{\text{max}}$	$0,13 \text{ m}^{-1}$	$R_{\text{ini},2}$	0,05
$l_{\text{obj,min}}$	0,5 m	$R_{\text{spur,g}}$	0,3
$\epsilon_{\text{R}}$	0,02	$R_{\text{verlust}}$	0,5
$n_{\text{start}}$	1	$v_{\text{max}}$	8,4 m/s

möglichst die Zielgeschwindigkeit  $v_{\text{ziel}}$  gefahren werden. Zur Evaluation wird beispielhaft eine Rechtskurve herangezogen, welche in Abb. 5.11 dargestellt ist. In dieser Situation hat das Fahrzeug bereits auf die Zielgeschwindigkeit  $v_{\text{ziel}}$  beschleunigt und nähert sich der Kurve an. Allerdings ist die Kurve so stark gekrümmt, dass das Fahrzeug vor der Kurve abbremsen muss. Durch die geschwindigkeitsabhängige Lenkwinkeldiskretisierung kann das Fahrzeug nur innerhalb der vorgegebenen physikalischen Begrenzungen planen, d.h. wenn das Fahrzeug nicht abbremsen würde, würde es aus der Kurve fliegen (vgl. rote/orange Punkte im rechten Bereich von Abb. 5.11a). Ohne, dass dem Planer extern eine reduzierte Geschwindigkeit für die Kurve vorgegeben wird, sucht sich der Planungsalgorithmus die schnellstmögliche Geschwindigkeit, um sicher durch die Kurve zu fahren. Durch die Initialisierung der möglichen Aktionen sowie der begrenzten Rechenzeit bremst das Fahrzeug im Verlauf der Trajektorie stärker ab, als für die Kurve nötig wäre (vgl. hellblaue Punkte im rechten Bereich von Abb. 5.11b). Dies ist auf die Explorationscharakteristik des Algorithmus

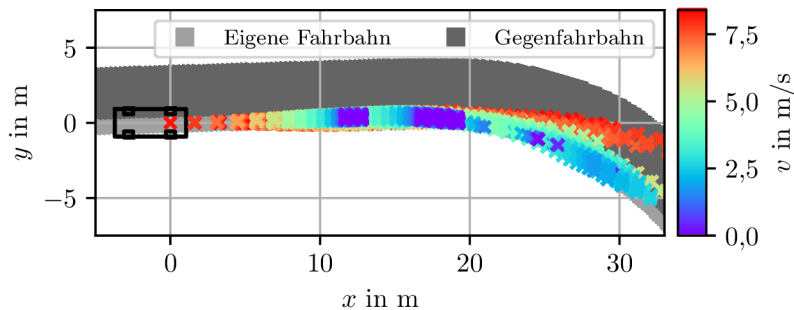
zurückzuführen. Während die Aktionen in naher Zukunft (d.h. direkt vor dem Fahrzeug) durch die hohe Anzahl an Simulationen gut abgeschätzt werden, wird auf Grund der begrenzten Rechenzeit sowie der Verzweigungen des Baumes das Verhalten in ferner Zukunft maßgeblich durch die Initialisierung der Aktionswerte und der Standardstrategie bestimmt. Durch das Prinzip des fortschreitenden Zeitfensters wird allerdings nur die erste geplante Aktion ausgeführt, bevor die Planung für den nächsten Zeitschritt gestartet wird. Der geplante Fahrverlauf über die gesamte Kurve wird in Abb. 5.12 dargestellt und besteht aus 60 einzelnen Planungsschritten.

In Abb. 5.12b sind einige Zustandsgrößen der Planung über die Zeit dargestellt. Wie an den gelben Kurven zu erkennen ist, kann der Eingang des Einspurmodells ( $a$  und  $\omega$ ) nur zu diskreten Zeitschritten geändert werden und ist dazwischen konstant. Darüber hinaus kann der Einfluss der Lenkwinkeldiskretisierung beobachtet werden. Während im Suchbaum mit  $\omega_d(\delta_{d,i}(v_d), a_d)$  geschwindigkeitsunabhängige Aktionen ausgewählt werden, sind die transformierten Lenkwinkelgeschwindigkeiten  $\omega$  sowie der Lenkwinkel  $\delta$  geschwindigkeitsabhängig und variieren demnach (vgl. zweiter und dritter Plot in Abb. 5.12b).

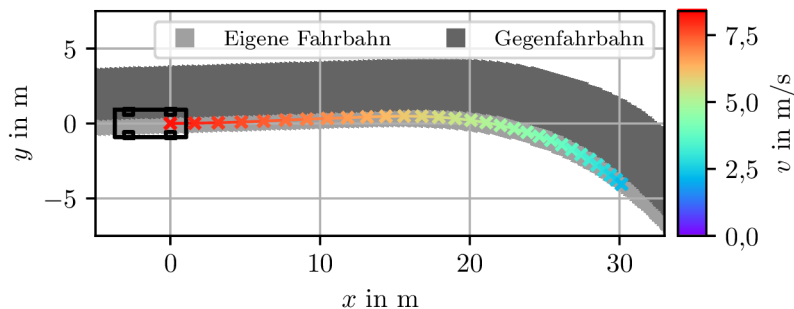
### 5.4.2 Annäherung an ein statisches Objekt

Das zweite Szenario ist eine Annäherungsfahrt an ein statisches Objekt in der eigenen Fahrspur. Das Ziel des Planers ist hierbei das Folgen der eigenen Fahrspur sowie das Vermeiden von Kollisionen. Zur Evaluation wird ein Objekt im eigenen Fahrstreifen platziert, wie in Abb. 5.13 durch die beiden roten Kollisionskreise dargestellt. Das Ego-Fahrzeug startet mit ausreichend Abstand, sodass es zuerst auf die Zielgeschwindigkeit  $v_{\text{ziel}}$  beschleunigt. Nachdem sich das Objekt in Planungsreichweite befindet, reagiert das Ego-Fahrzeug auf dieses, bremst ab und bleibt letztendlich vor dem Objekt stehen.

In Abb. 5.13a ist ein Suchbaum zum Abbremsen vor dem Objekt exemplarisch aufgezeigt. Hier stellen die lila Punkte Positionen dar, an denen das Fahrzeug zum Stillstand gekommen ist. Bremst das Fahrzeug dagegen nicht ab, kollidiert es mit dem Objekt (hellgrüne Punkte im linken roten Kreis). Durch die langgezogene lila Punktewolke ist zu erkennen, dass der Planer verschiedene Bremsmanöver zu unterschiedlichen Zeitpunkten mit unterschiedlichen Verzögerungen exploriert. Während im linken Bereich der lila Punktewolke das Abbremsen mit der Maximalverzögerung von  $a_{\min} = -3 \text{ m/s}^2$  exploriert wird, wird im rechten Bereich der lila Punktewolke das Abbremsen mit der Minimalverzögerung von  $-1 \text{ m/s}^2$  exploriert.

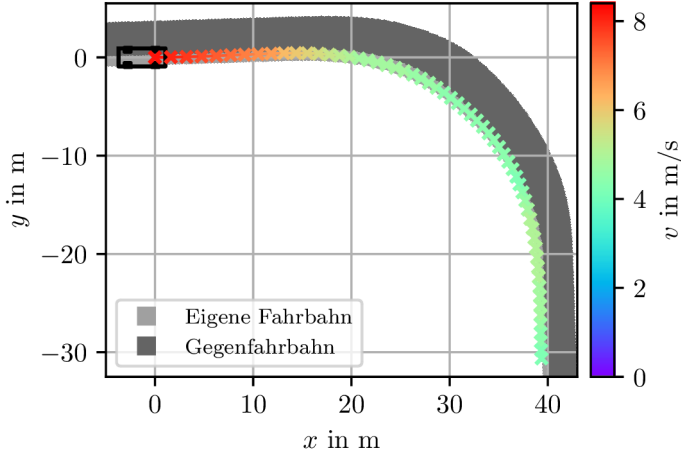


(a) Monte-Carlo Suchbaum mit 12969 Knoten.

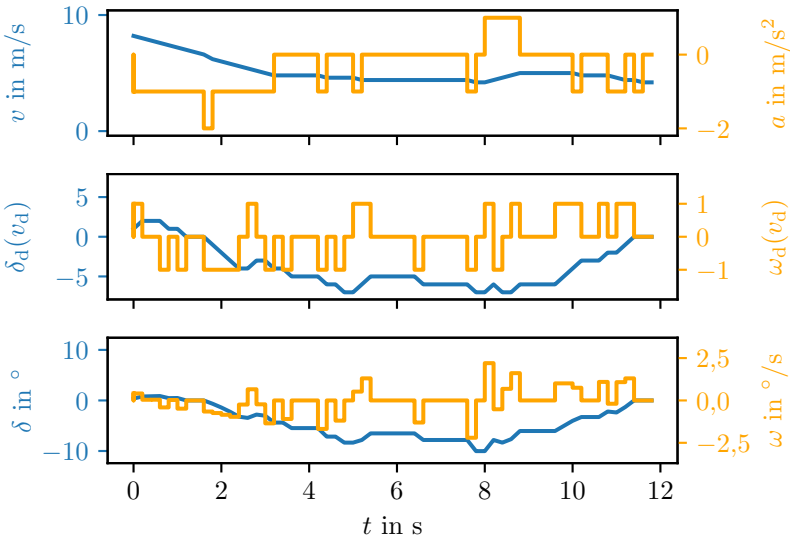


(b) Beste Trajektorie, welche sich aus dem in (a) dargestelltem Suchbaum ergibt.

**Abbildung 5.11:** Beispielhafte Trajektorienplanung für einen Zeitschritt. Die Startgeschwindigkeit des Fahrzeugs ist für die dargestellte Kurvenfahrt zu groß (vgl. rote/orange Knoten mit hoher Geschwindigkeit im rechten Bereich von (a)), sodass das Fahrzeug abbremsen muss, um nicht aus der Kurve zu fliegen. Durch das kinematische Einspurmodell sowie die geschwindigkeitsabhängige Lenkwinkeldiskretisierung können nur komfortable, regelbare Bewegungen erkundet werden. Somit werden in diesem Beispiel verschiedene Bremsmanöver exploriert, bevor die in (b) dargestellte Trajektorie ausgegeben wird. [140]



(a) Geplante Trajektorie durch die Kurve über 60 Planungsschritte.



(b) Geplante Zustandsgrößen des Einspurmodells.

**Abbildung 5.12:** Die geplante Trajektorie durch die gesamte Kurve nach dem Prinzip des fortschreitenden Zeitfensters. Dabei wird von jedem Planungsschritt die erste Aktion ausgeführt, bevor ein neuer Suchbaum erstellt wird. Durch die geschwindigkeitsabhängige Lenkwinkeldiskretisierung variiert die Lenkwinkelgeschwindigkeit  $\omega$  in ihren Werten, obwohl die diskretisierte Lenkwinkelgeschwindigkeit  $\omega_d(\delta_{d,i}(v_d), a_d)$  nur Werte zwischen -1 und 1 annimmt (vgl. (b)).

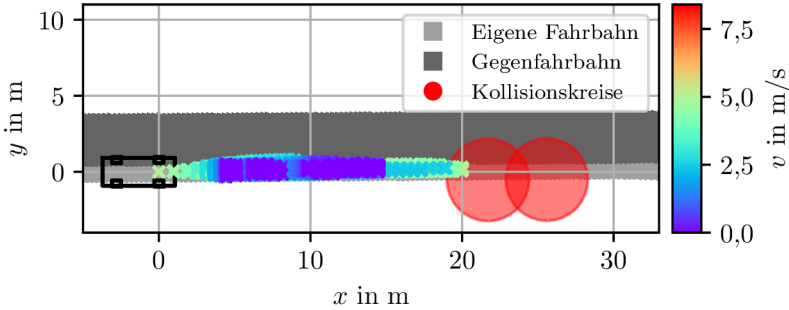
Da in der Kostenfunktion ein stärkeres Abbremsen stärker bestraft wird (vgl. Gleichung (5.7)), wird letztendlich eine Trajektorie mit Verzögerung von  $-1 \text{ m/s}^2$  ausgegeben (vgl. geplante Trajektorie in Abb. 5.13b). Allerdings wird die Distanz zu stehenden Objekten nicht berücksichtigt, sodass der Abstand zwischen dem stehenden Ego-Fahrzeug und einem statischen Objekt variieren kann. Soll hier ein gewünschtes Verhalten vorgegeben werden, muss ein weiterer Term in der Kostenfunktion hinzugefügt werden.

## 5.5 Zusammenfassung

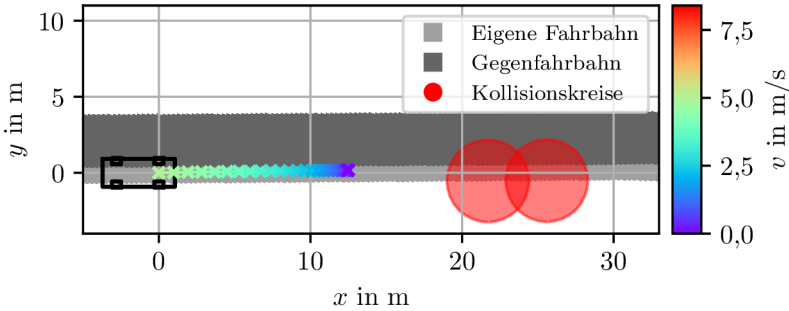
In diesem Kapitel wird ein neuartiger Planungsansatz, die CarPre-Trajektorienplanung, auf Basis der Monte-Carlo Baumsuche vorgestellt. Das Ziel ist es, einen Planungsalgorithmus zu entwickeln, welcher ohne einen vorgeschalteten Verhaltensplaner auskommt. Die Erfüllung dieses Ziels kann nicht nachgewiesen werden, da hierzu der Planer an weiteren, komplexeren Verkehrsszenarien mit dynamischen Verkehrsteilnehmern validiert werden muss. Hierzu fehlen derzeit weitere Softwarekomponenten wie die Prädiktion anderer Verkehrsteilnehmer (vgl. Abschnitt 6.2). Trotz alledem hat der vorgestellte Planungsalgorithmus das Potenzial solche komplexen Situationen zu lösen.

Der Algorithmus verwendet in der Planung kinematische Bewegungsgleichungen, sodass nur stetige, kinematisch mögliche Trajektorien geplant werden. Dabei wird auf die in Kapitel 4 hergeleitete geschwindigkeitsabhängige Lenkwinkeltransformation zurückgegriffen. Es entsteht ein diskreter Aktionsraum, welcher nur fahrbare laterale Bewegungen enthält. Der Planungsalgorithmus kann in jedem Zeitschritt zwischen einer endlichen Menge an diskreten Aktionen auswählen. Dabei besteht eine diskrete Aktion aus einer diskretisierten Beschleunigung  $a_d$  sowie einer diskretisierten, geschwindigkeitsabhängigen Lenkwinkelgeschwindigkeit  $\omega_d(\delta_{d,i}(v_d), a_d)$ . Im Anschluss werden die weiteren Anpassungen der Monte-Carlo Baumsuche an das Problem der Trajektorienplanung beschrieben. Hierbei ist die Flexibilität des Algorithmus hervorzuheben. Durch die Verwendung von diskreten Aktionen zu diskreten Zeitschritten können komplexe Kostenfunktionen oder bestimmte Verhaltenslimitationen umgesetzt werden. Weiterhin kann durch das Standardverhalten während des Simulationsschrittes das Ausführen einer gewählten Aktion abgeschätzt werden.

In einem nächsten Schritt wird der Algorithmus anhand einer Spurfolgefahrt sowie einer Annäherung an ein statisches Objekt evaluiert. Hierbei



(a) Monte-Carlo Suchbaum mit 17239 Knoten.



(b) Beste Trajektorie, welche sich aus dem in (a) dargestellten Suchbaum ergibt.

**Abbildung 5.13:** Beispielhafte Trajektorienplanung für einen Zeitschritt. Ein statisches Hindernis (rote Kreise) blockiert die Straße. Um eine Kollision zu vermeiden, exploriert der in (a) gezeigte Suchbaum Möglichkeiten vor dem Hindernis zum Stillstand zu kommen. So werden verschiedene Bremsmanöver exploriert, bevor die in (b) dargestellte Trajektorie ausgegeben wird, welche das Fahrzeug zum Stillstand bringt. Der Radius der eingezeichneten Kollisionskreise beträgt  $d_{\text{sicher}} + r_{\text{ego}} + r_{\text{obj},i}$ , d.h. sobald ein eingezeichneter Trajektorienpunkt innerhalb der Kreise liegt, entsteht nach Gleichung (5.8) eine Kollision.

zeigt sich, dass das Konzept der CarPre-Trajektorienplanung grundsätzlich funktioniert. Durch die getroffenen Anpassungen ist eine Trajektorienplanung in Echtzeit möglich. Allerdings sind auch mögliche Probleme des Ansatzes aufgefallen. So kann bei der Annäherungsfahrt die Distanz zum Objekt vor dem Fahrzeug variieren, da sie nicht explizit in der Kostenfunktion berücksichtigt wird. Gleiches gilt für ein mögliches Überholmanöver des statischen Objektes. Soll das Ego-Fahrzeug, statt sich anzunähern und anzuhalten, das statische Objekt überholen, muss auch dieses Verhalten in der Kostenfunktion widerspiegelt werden. Zwar ist die Herleitung einer solchen Kostenfunktion prinzipiell möglich, allerdings sehr schwierig, vor allem dann, wenn das Ziel ein Planungsalgorithmus ohne vorgeschalteten Verhaltensplaner sein soll. Ein möglicher zukünftiger Lösungsansatz hierfür ist der Einsatz von inversen Reinforcement-Learning-Methoden, um aus Verhaltensregeln sowie menschlichem Verhalten eine passende Kostenfunktion herzuleiten.

# 6 Experimente im Rahmen des Projekts PRORETA 5

In diesem Kapitel wird auf das Projekt „PRORETA 5: Urbanes autonomes Fahren“ eingegangen. Im Rahmen des Projekts wird ein prototypisches Fahrzeug zur Verfügung gestellt, auf welchem die bisherigen Konzepte dieser Dissertation zur Anwendung gebracht werden. Nach einer kurzen Einleitung wird auf den Systemaufbau sowie die Softwarearchitektur des Versuchsträgers eingegangen. Im Anschluss werden die Versuche und Ergebnisse präsentiert, welche innerhalb des Projekts durchgeführt wurden. Das Kapitel endet mit einer kurzen Zusammenfassung.

## 6.1 Einleitung und Motivation

Wie bereits in Kapitel 1 ausgeführt, ist die hauptsächliche technische Herausforderung für autonome Fahrzeuge die Software [95]. Gerade Verkehrsszenarien in städtischen Gebieten können sehr komplex und herausfordernd für das autonome System sein (vgl. Abb. 1.1). Daher liegt der Fokus des Kooperationsprojekts PRORETA 5 auf der Entwicklung neuartiger Algorithmen und Methoden der künstlichen Intelligenz für das autonome Fahren in innerstädtischen Gebieten.

PRORETA 5 ist die fünfte Auflage der Forschungskooperation zwischen der Continental AG und der TU Darmstadt, welche um die Universität Bremen sowie die TU Iași erweitert wird. Die vorherigen Projekte sind: PRORETA 1 [56] (Notbrems- und Ausweichassistenzsystem), PRORETA 2 [108] (Überholassistenzsystem), PRORETA 3 [134] (Integrales Sicherheitskonzept und kooperative Automatisierung) und PRORETA 4 [114] (Stadtassistenzsystem). Weitere Informationen zu den einzelnen Projekten sind auf der Website <https://www.proreta.tu-darmstadt.de> verfügbar. Im Folgenden wird ein Überblick über den zur Verfügung gestellten Versuchsträger sowie die Softwarearchitektur des Projekts gegeben.



## 6.2 Systemaufbau und Softwarearchitektur

Im Folgenden wird das prototypische Fahrzeug des Projekts sowie die im Projekt entwickelte Softwarearchitektur vorgestellt. Der Versuchsträger ist ein Volkswagen Passat (B8) und wird in Abb. 6.1 abgebildet. Die Softwarearchitektur orientiert sich an den drei Kernkompetenzen eines autonomen Systems, der Wahrnehmung, Planung und Ausführung und ist in Abb. 6.2 dargestellt. Die einzelnen Systemfunktionalitäten sind im Projekt auf mehrere Entwickelnde aufgeteilt:

- **Lokalisierung und Objekterkennung:**  
Lino Giefer und Andreas Serov  
(Universität Bremen, Prof. Dr. rer. hum. biol. K. Schill)
- **Prädiktion anderer Verkehrsteilnehmer:**  
Andreea-Iulia Patachi (TU Iași, Prof. F. Leon, PhD)
- **Trajektorienplanung:**  
Christoph Ziegler (TU Darmstadt, Prof. Dr.-Ing. J. Adamy)
- **Sicherheitsüberprüfung und Regelung:**  
Christoph Popp (TU Darmstadt, Prof. Dr. rer. nat. H. Winner,  
Prof. Dr.-Ing. S. Peters)
- **Systemüberblick und Absicherung:**  
Felix Glatzki (TU Darmstadt, Prof. Dr. rer. nat. H. Winner,  
Prof. Dr.-Ing. S. Peters)

In der Wahrnehmung wird aus einer HD-Karte<sup>1</sup> kombiniert mit einer Fahrzeuglokalisierung ein Straßenmodell erstellt. Weiterhin werden durch verschiedene Sensoren Objekte im Umfeld des Fahrzeugs erkannt. In einem nächsten Schritt werden für alle detektierten Objekte die Fahrverläufe der Zukunft prädiziert. Der Trajektorienplaner erhält all diese Informationen und berechnet daraus eine komfortable und sichere Trajektorie. Durch eine Sicherheitsüberprüfung, basierend auf klassischen Algorithmen, wird das System abgesichert und im Notfall ein Notstopp eingeleitet.

---

<sup>1</sup>Eine HD-Karte beinhaltet ein genaues Abbild der statischen Umgebung. Ist die exakte Position des autonomen Systems bekannt, kann der zukünftige Straßenverlauf aus der Karte extrahiert werden. Somit entfällt z. B. die Erkennung von Straßenmarkierungen und es muss lediglich die Differenz zwischen Karte und Realität bestimmt werden.



Abbildung 6.1: PRORETA 5-Versuchsträger: Volkswagen Passat (B8).

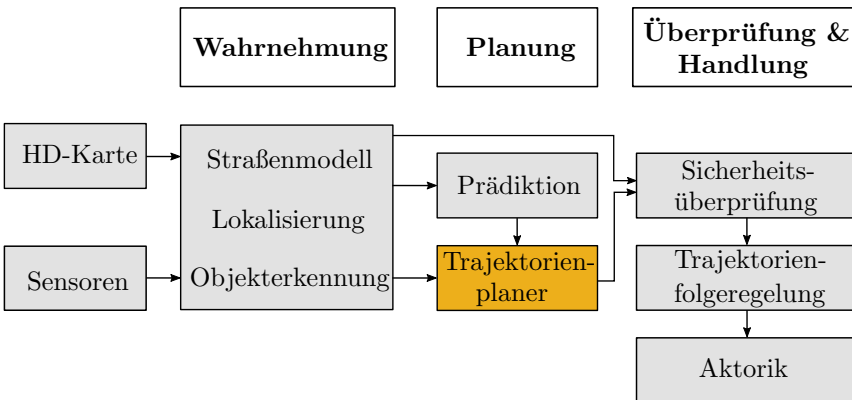


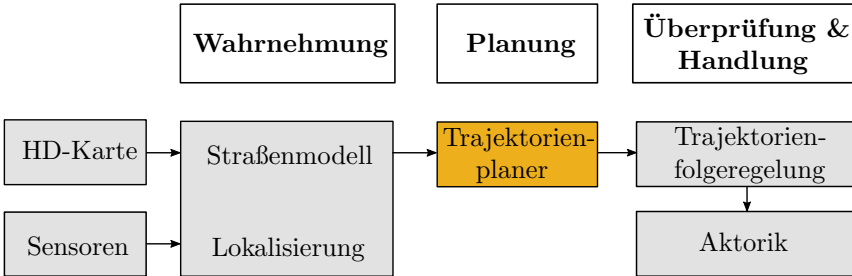
Abbildung 6.2: Überblick über die PRORETA 5-Softwarestruktur. Das orange markierte Modul wird vom Autor entwickelt und entspricht der CarPre-Trajektorienplanung.

Die überprüfte Trajektorie wird an den Trajektorienfolgeregler weitergeleitet, welcher die benötigten Beschleunigungs- und Lenkwinkelanforderungen für die Aktorik berechnet und an diese übermittelt. Hierbei wird aus Sicherheitsgründen das maximale Lenkmoment geschwindigkeitsabhängig beschränkt, sodass ein Sicherheitsfahrer immer die Möglichkeit hat sich gegen die Automatisierung durchzusetzen. Eine Freigabe zum autonomen Fahren hat das Fahrzeug für den August-Euler-Flugplatz der TU Darmstadt erhalten, die verfügbare Teststrecke ist in Abb. 6.3 dargestellt.

Da zum derzeitigen Stand (Juni 2022) die in Abb. 6.2 gezeigten Softwarekomponenten noch nicht alle im Zusammenspiel umgesetzt werden konnten, wird im Rahmen dieser Dissertation auf eine vereinfachte Struktur zurückgegriffen, welche in Abb. 6.4 dargestellt ist. Dabei entfällt die Objektdetektion sowie Prädiktion, sodass zur Planung der Trajektorie allein ein Straßenmodell, welches aus der HD-Karte berechnet wird, benötigt wird. Das Straßenmodell liegt dabei, wie in Abschnitt 3.4.2 motiviert, in einer Grid-basierten Darstellung vor. Für eine direkte Verwendung innerhalb des Planungsalgorithmus wird die Karte um eine diskrete Richtungskarte erweitert. Ebenso wird die Straßenbreite um den Platzbedarf des Ego-Fahrzeugs reduziert, um eine schnelle Fahrspurzuordnung zu gewährleisten (vgl. Abb. 5.8). Für die hier vorgestellten Versuche wird die Sicherheitsüberprüfung deaktiviert, sodass das Fahrzeug immer die geplante Trajektorie abfährt. Durch diese Reduktion können im weiteren Verlauf des Kapitels nur statische Verkehrsszenarien, wie z. B. das Spurhalten, ohne weitere Objekte betrachtet und analysiert werden.



**Abbildung 6.3:** Verfügbare Teststrecke (orange) auf dem August-Euler-Flugplatz der TU Darmstadt (Bild aus Google Earth Pro). Die Kurven der Strecke sind für die Auswertung in Abschnitt 6.3 nummeriert. [140]



**Abbildung 6.4:** Überblick über die reduzierte Softwarestruktur für die folgenden Versuche. Das orange markierte Modul wird vom Autor entwickelt und entspricht der CarPre-Trajektorienplanung. [140]

### 6.3 Ergebnisse für ein prototypisches Fahrzeug

Im folgenden Abschnitt werden die Ergebnisse der Trajektorienplanung am prototypischen Versuchsträger vorgestellt. Um die Machbarkeit in einem realen System bewerten zu können, wird der Algorithmus auf den Versuchsträger gebracht. Erste Tests bezüglich des Planers werden im Folgenden dargestellt.

Für alle Versuche wird die CarPre-Trajektorienplanung ohne weitere Anpassungen aus Kapitel 5 übernommen und auf einem Intel Xeon E5-2698 v4 Prozessor (20 Kerne à 2,2 GHz, 40 Threads, 50 MB Cache, Baujahr 2016) ausgeführt. Die Messfahrten sind am 02. Juni 2022 aufgezeichnet worden und beinhalten insgesamt ca. 45 Minuten Fahrzeit. Dies entspricht 13513 Planungen, wobei eine Planung alle 0,2 s berechnet wird. Das Ziel des Planungsalgorithmus ist es hierbei, der in Abb. 6.3 gezeigten Strecke mit der vorgegebenen Maximalgeschwindigkeit von  $8,4 \text{ m/s} = 30,24 \text{ km/h}$  zu folgen. Die Strecke wird im Uhrzeigersinn abgefahren, da sich hierfür engere Kurveninnenradien, also schwierigere Fahrverläufe, ergeben. Insgesamt wird die Runde zehnmals gefahren, wobei alle Parameter, wie in Tabelle A.2 dargestellt, gewählt werden. Lediglich die Eingangsabstastzeit  $T_{\text{in}}$  wird während der Versuchsdurchführung von 0,2 bis 0,6 s variiert. Ein Überblick über die einzelnen Messungen ist in Tabelle A.3 gegeben. Während der Messungen wird mehrmals die geschwindigkeitsabhängige Lenkmomentenbegrenzung des Fahrzeugs überschritten, obwohl die Dynamik des Fahrzeugs nicht im Grenzbereich war (fünfmal in Kurve 5, sowie

einmal in Kurve 2). In diesem Fall stoppt die Trajektorienfolgeregelung, das Fahrzeug wird vom Sicherheitsfahrer in den Stillstand gebracht und die Automatisierung erneut freigeschaltet. Eine solche Überschreitung ist zwar unerwünscht, allerdings wird ein späteres vollautomatisiertes System eine solche Begrenzung nicht enthalten. In diesem Testfahrzeug dient sie lediglich dazu, dass der Sicherheitsfahrer jederzeit in der Lage ist, das autonome System zu übersteuern.

Im weiteren Verlauf des Kapitels wird zuerst auf die Laufzeit des Algorithmus eingegangen, bevor der Fahrkomfort bei  $T_{\text{in}} = 0,2\text{s}$  analysiert wird. Zuletzt wird die Eingangsabtastzeit  $T_{\text{in}}$  des Algorithmus variiert und die Auswirkungen auf den Planungsalgorithmus ausgewertet.

### 6.3.1 Analyse der Laufzeit

Ein wichtiges Merkmal eines Planungsalgorithmus in einem autonomen Fahrzeug ist die Echtzeitfähigkeit. Um diese Eigenschaft zu besitzen, muss ein Algorithmus immer innerhalb einer fest definierten Zeit ein Ergebnis liefern. Dies ist für sicherheitskritische Anwendungen wie das autonome Fahren essentiell, da sonst von diesem System potentielle Risiken ausgehen. Zwar kann eine Sicherheitsüberprüfung, wie in Abb. 6.2 dargestellt, gefährliche Situationen durch einen Notstopp abfangen, allerdings sollte dies nur als ein zusätzlicher Notfallmechanismus angesehen werden. Daher wird im Folgenden die Laufzeit der Trajektorienplanung betrachtet.

Während bei den häufig verwendeten Optimierungsansätzen (vgl. Abschnitt 5.2) auf die Konvergenz des Lösungsproblems gewartet werden muss, ist die CarPre-Trajektorienplanung ein Algorithmus der zu jedem Zeitpunkt das aktuell beste berechnete Ergebnis ausgeben kann. Somit können Laufzeitspitzen wie in Abb. A.1 aus [131] vermieden werden. Dies zeigt die folgende Auswertung. Der Planungsalgorithmus wird entsprechend Abschnitt 5.3 nach einer definierten Berechnungszeit  $t_{\text{calc}}$  abgebrochen. Ist  $t_{\text{calc}}$  verstrichen, werden nur noch alle bereits angefangenen Iterationen des Algorithmus zu Ende ausgeführt. Im schlimmsten Fall bedeutet dies, dass kurz vor dem Verstreichen von  $t_{\text{calc}}$  eine neue Iteration angefangen wird. Wird eine nach Abschnitt 5.3.8 implementierte Parallelisierung verwendet, kann zusätzlich das Warten auf die Freigabe eines Knoten-Mutexes durch andere Prozesse hinzukommen.

Tabelle 6.1 zeigt den Mittelwert, die Standardabweichung (SD) sowie den minimalen und maximalen Wert aller Laufzeiten für  $t_{\text{calc}} = 100\text{ms}$ . Hierbei zeigt sich, dass die meisten Laufzeiten nahe an der gewählten Berechnungszeit  $t_{\text{calc}}$  liegen (99,3 % liegt bei oder unter 102 ms) und nur ver-

**Tabelle 6.1:** Laufzeiten der CarPre-Trajektorienplanung aller 13513 Planungen für  $t_{\text{calc}} = 100$  ms.

	Mittel	SD	Min	Max
$t$ in ms	100,64	0,63	100	111

einzelte Ausreißer bis zu 11 ms existieren. Diese Ausreißer können durch die Erklärung im vorangehenden Abschnitt begründet werden. Durch Modifikation des Algorithmus könnten diese weiterhin verkleinert werden, z. B. indem die aktuellen Iterationen nicht zu Ende gerechnet werden, wenn  $t_{\text{calc}}$  verstrichen ist. Das Konzept der CarPre-Trajektorienplanung ist unter realen Bedingungen echtzeitfähig und das Ergebnis kann insgesamt positiv bewertet werden.

### 6.3.2 Analyse des Fahrkomforts

In diesem Abschnitt wird der Fahrkomfort der gefahrenen Trajektorie analysiert. Zwar ist „Fahrkomfort“ generell eine subjektive Metrik, kann aber trotzdem durch eine Analyse von Messwerten objektiv beschrieben werden [36]. Nach [9, S. 90] ist hierfür hauptsächlich die Fahrzeugbeschleunigung zusammen mit dem Beschleunigungsverlauf verantwortlich. Je nach Szenario wird beim Beschleunigungsverlauf ein früheres oder späteres Abbremsen bevorzugt. Ein symmetrischer Beschleunigungsverlauf (d.h. mit minimalem Ruck) liefert allerdings für alle Szenarien gute Ergebnisse. Daher ist häufig ein Ziel des Planers, wie z. B. in [90], die Beschleunigung sowie den Ruck zu minimieren. Weitere Komfortfaktoren nach [40], wie Innenraumtemperatur, Luftqualität, Lautstärke der Fahrgeräusche oder Fahrzeugvibrationen werden im Folgenden nicht berücksichtigt.

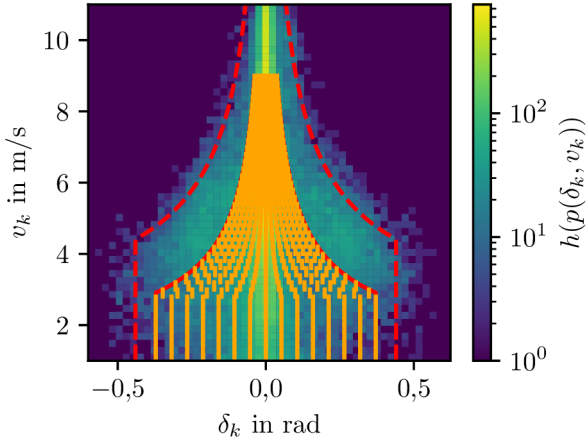
Entscheidend für den Komfort der mittels CarPre geplanten Trajektorie ist die Diskretisierung des Zustandsraumes nach Abschnitt 5.3.1. Denn diese beeinflusst maßgeblich die möglichen Aktionen und damit die möglichen Trajektorien. Daher wird in einem ersten Schritt die gewählte Diskretisierung analysiert und mit den aus Abschnitt 4.4 extrahierten, menschlichen Fahrdaten verglichen. Ein solcher Vergleich ist in Abb. 6.5 für die Diskretisierung der Lenkvariablen dargestellt. Da bei der Diskretisierung des Lenkwinkels  $\delta$  dieselbe Transformationsfunktion wie in Abschnitt 4.4 gewählt wird, ergibt sich für die Diskretisierung derselbe relative Verlauf. Durch die Wahl einer kleineren lateralen Maximalbeschleunigung  $a_{\text{lat,max}}$  sowie einer kleineren maximal fahrbaren Krümmung  $\kappa_{\text{max}}$  als in Abschnitt 4.4 werden

nur komfortable Lenkwinkel zugelassen und die laterale Beschleunigung beschränkt.

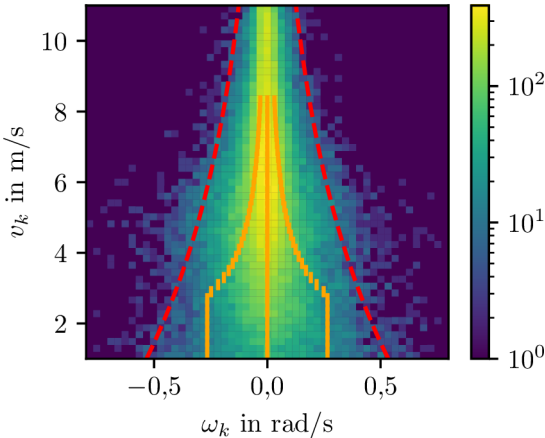
Für die diskrete Lenkwinkelgeschwindigkeit ergibt sich in Abb. 6.5b durch die Bildung des Zustandsgitters in Gleichung (5.4) ein anderer Verlauf (orange) als durch die extrahierte Transformationsfunktion in (4.8) (rote gestrichelte Linien). Gerade im niedrigen Geschwindigkeitsbereich ist die diskrete Lenkwinkelgeschwindigkeit deutlich größer als bei hohen Geschwindigkeiten. Dies liegt an der gleichförmigen Diskretisierung mit einer konstanten Anzahl  $n_\delta$  an diskreten Lenkwinkeln über den gesamten Geschwindigkeitsbereich hinweg. Durch die beiden Parameter  $a_{\text{lat,max}}$  und  $\kappa_{\text{max}}$  kann die Begrenzungsfunktion verändert werden, und zwar wird mit  $a_{\text{lat,max}}$  der Teil bei höheren Geschwindigkeiten und mit  $\kappa_{\text{max}}$  der Teil bei niedrigen Geschwindigkeiten begrenzt. Zur Erweiterung des hier präsentierten Ansatzes ist es für die Zukunft eine Möglichkeit  $n_\delta$  über die Geschwindigkeit zu variieren. Somit stehen im niedrigen Geschwindigkeitsbereich mehr diskrete Lenkwinkel zur Auswahl, wodurch die Lenkwinkelgeschwindigkeit reduziert wird. Mit einer solchen Adaption könnten ggf. auch die Lenkmomentüberschreitungen, die während der Versuche aufgetreten sind, verhindert werden.

Für den Komfort in Fahrtrichtung wird im Folgenden die Beschleunigung betrachtet. In Abb. 6.6 wird durch die orangenen Kreuze die gewählte Diskretisierung nach Gleichung (5.2) über den menschlichen Fahrdaten dargestellt. Durch die Ruckbeschränkung in Gleichung (5.6) stehen in Abhängigkeit der zuletzt gewählten Beschleunigung maximal drei diskrete Beschleunigungswerte zur Verfügung, welche den menschlichen Beschleunigungen ähneln. Eine Ausnahme sind die starken Verzögerungen in der unteren linken Ecke ( $-2 \text{ m/s}^2$  und  $-3 \text{ m/s}^2$ ). Diese Abweichung kann auf den Datensatz zurückgeführt werden. Schließlich versuchen menschliche Fahrer starke Beschleunigungen zu vermeiden und verwenden diese nur in Notfällen.

Generell gilt für die Diskretisierung, dass zwischen einer feineren oder gröberen abgewogen werden muss. Wird die Diskretisierung zu fein gewählt, wächst der diskrete Zustandsraum sehr schnell an. Weiterhin kommt hinzu, dass der Planungsalgorithmus schlechter eine Lösung findet, da der Unterschied zwischen zwei gewählten Aktionen nun kleiner ist und ggf. beide Aktionen ähnlich bewertet werden. Wird die Diskretisierung dagegen zu grob gewählt, ist ggf. die gewünschte Lösung gar nicht erreichbar. Für den Fall, dass sie erreichbar ist, findet der Planungsalgorithmus im kleineren Zustandsraum nun schneller eine Lösung. Allerdings besteht dann die Möglichkeit, dass die Lösung dann ebenso zu grob ist,



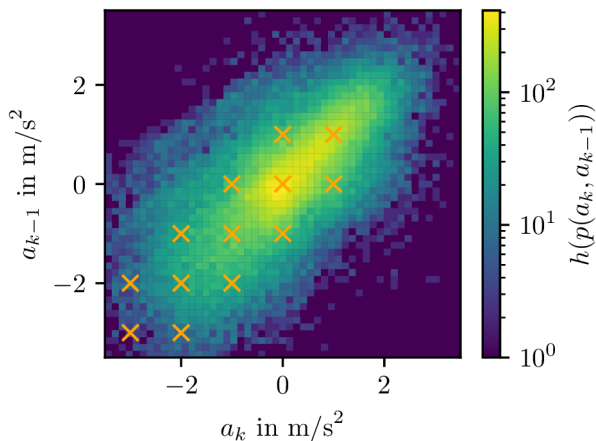
(a) 2D-Histogramm aus Abb. 4.8 von  $\delta_k$  in rad über  $v_k$  in m/s, aufgetragen in logarithmischer Skala. Die gestrichelten roten Linien stellen die extrahierte Transformationsfunktion (4.10) dar. Zur Diskretisierung des Planers werden die Bewegungsgrenzen angepasst (rote Linien), sodass durch die gleichverteilte Einteilung nach Gleichung (5.3) die in orange dargestellten diskreten Lenkwinkel für  $n_\delta = 15$  entstehen. [140]



(b) 2D-Histogramm aus Abb. 4.5 von  $\omega_k$  in rad/s über  $v_k$  in m/s, aufgetragen in logarithmischer Skala. Die gestrichelten roten Linien stellen die extrahierte Transformationsfunktion (4.8) dar. Durch die in Abb. 6.5a dargestellten diskreten Lenkwinkel ergeben sich für  $a_{cl} = 0$  nach Gleichung (5.4) die in orange dargestellten diskreten Lenkwinkelgeschwindigkeiten für  $n_\omega = 3$ .

**Abbildung 6.5:** Diskretisierung (a) des Lenkwinkels  $\delta$  sowie (b) der Lenkwinkelgeschwindigkeit  $\omega$  im Vergleich zu den in Abschnitt 4.4 extrahierten menschlichen Fahrdaten.



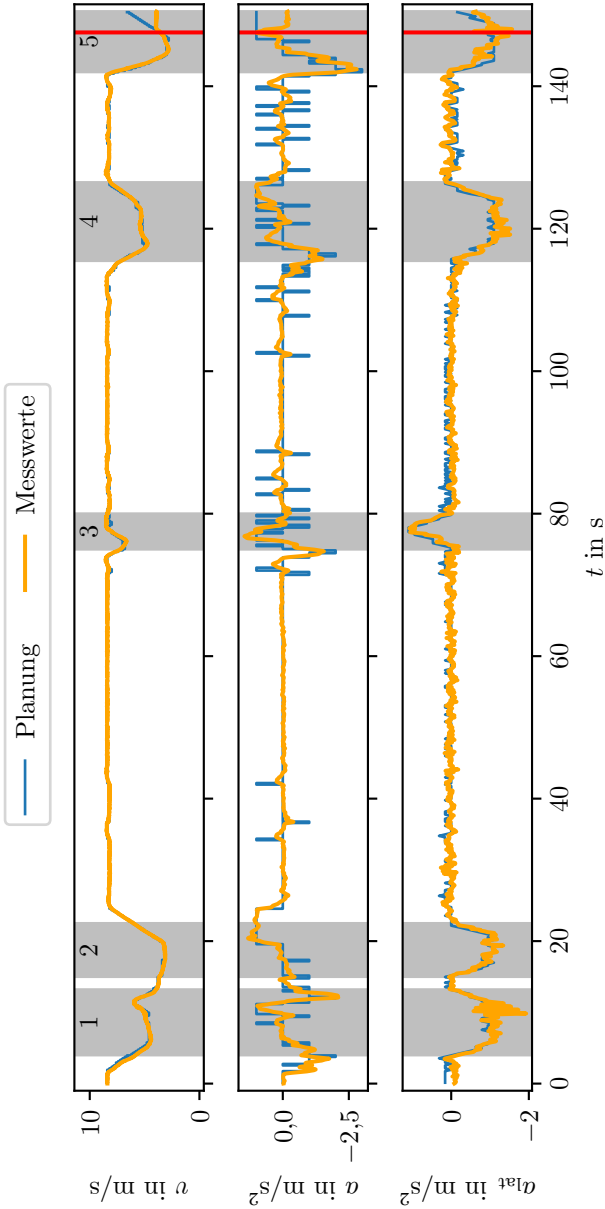


**Abbildung 6.6:** 2D-Histogramm aus Abb. 4.10c von der Beschleunigung  $a_k$  in Abhängigkeit der letzten Beschleunigung  $a_{k-1}$ , aufgetragen in logarithmischer Skala. Die möglichen diskreten Beschleunigungswerte sind in orange abgebildet und werden durch Gleichungen (5.2) und (5.6) bestimmt. [140]

sodass eine weitere Nachbearbeitung z. B. durch eine Glättung benötigt wird, um eine fahrbare Trajektorie zu generieren.

Nach dieser Diskussion der Diskretisierung, wird in einem zweiten Schritt eine Messung des prototypischen Fahrzeugs analysiert. Hierzu wird eine Messung gewählt, bei der das Fahrzeug die Lenkmomentenbegrenzung in Kurve 5 überschritten hat, sodass die Trajektorienfolgeregelung abgebrochen wird. Die Messung ist in Abb. 6.7 dargestellt und zeigt eine fast vollständige Runde auf der Teststrecke, lediglich der lange, gerade Teil zwischen Kurve 1 und 5 (untere Gerade in Abb. 6.3) ist nicht dargestellt. Das Überschreiten der Lenkmomentenbegrenzung in Kurve 5 ist rot markiert und es ist zu erkennen, dass das Fahrzeug danach nicht mehr beschleunigt, obwohl die Planung dies vorgibt.

Generell ist zu erkennen, dass der Trajektorienfolgeregler, in Kombination mit der Trägheit der Aktorik, die geplante Trajektorie, bestehend aus den blau dargestellten zeitdiskreten Eingängen, glättet. Einzelne kleine Ausreißer in der longitudinalen Beschleunigung  $a$  der geplanten Trajektorie wie zwischen 30 und 40 s oder 130 und 140 s werden herausgefiltert. Zwar sind solche hochfrequenten Schwankungen prinzipiell unerwünscht und sollten durch weiterführende Anpassungen vermieden werden, aller-



**Abbildung 6.7:** Geplante Trajektorie (blau) gegenüber den Messwerten der gefahrenen Trajektorie (gelb). Die in Abb. 6.3 markierten Fahrbahnkurven sind im Fahrerlauf grau hinterlegt und entsprechend nummeriert. Das Überschreiten der Lenkmomentenbegrenzung in Kurve 5 ist rot markiert. Die mit 50 Hz aufgenommenen Beschleunigungsmesswerte sind durch einen Tiefpassfilter (Binomialfilter der Ordnung 40, Grenzfrequenz: 120 Hz) geglättet. [140]

dings werden sie vom Fahrer nicht wahrgenommen.

Die laterale Beschleunigung  $a_{\text{lat}}$  des Fahrzeugs oszilliert bei Geradeausfahrten leicht um Null, was im Fahrzeug allerdings nicht wahrnehmbar ist. In den Kurven limitiert die Lenkwinkeldiskretisierung die maximale laterale Beschleunigung, welche vom Fahrzeug eingehalten wird. Lediglich in der ersten Kurve hat der Regler Probleme, der Trajektorie zu folgen. Bei 10 s fängt  $a_{\text{lat}}$  an zu schwingen, obwohl die Trajektorie stetig geplant war. Da dem Autor die verwendete Reglerstruktur zum Einstellen der Solllenkwinkel nicht bekannt ist, ist es schwierig hierzu eine Erklärung abzugeben.

Weiterhin lässt sich das folgende Verhalten beobachten. Das Fahrzeug beginnt vor einer Kurve bereits abzubremsen, ähnlich zu menschlichen Fahrverhalten. Die benötigte Geschwindigkeit wird daraufhin innerhalb der Kurve erreicht, bevor gegen deren Ende bereits wieder beschleunigt wird. Insgesamt entsteht hierdurch ein natürliches Fahrgefühl.

### 6.3.3 Analyse der Abtastzeit

Nach der Untersuchung des Fahrkomforts im vorherigen Abschnitt, wird in diesem Abschnitt der Einfluss der Abtastzeit auf den Planungsalgorithmus analysiert. Hierzu wird, anknüpfend an die Analyse der Abtastzeit bei menschlichen Fahrern aus Abschnitt 4.3, die Abtastzeit des Systems zwischen  $T_{\text{in}} \in \{0,2\text{ s}; 0,4\text{ s}; 0,6\text{ s}\}$  bei gleichbleibender Aktualisierungszeit  $T_{\text{akt}} = 0,2\text{ s}$  variiert. Für alle drei Konfigurationen werden dieselben CarPre-Parameter nach Tabelle A.2 gewählt, lediglich die Abtastzeit wird geändert. Hierdurch lässt sich der Einfluss von  $T_{\text{in}}$  besser analysieren. Allerdings ist anzumerken, dass diese Parameterkonfiguration auf  $T_{\text{in}} = 0,2\text{ s}$  angepasst wird und demnach für die beiden anderen Abtastzeiten nicht ideal ist. Für eine vollständige Evaluation sollten bei zukünftigen Versuchen auch die Parameter der erhöhten Abtastzeiten optimiert werden.

Das Planen von Trajektorien mit erhöhten Abtastzeiten erfolgt ebenso wie das Planen mit  $T_{\text{in}} = T_{\text{akt}}$ . Lediglich für die Bestimmung des nächsten Startzustands (vgl. Abb. 5.6) sowie zur Berechnung von Kollisionen<sup>2</sup> werden Zwischenzustände benötigt. Daher liegt für alle drei Abtastzeiten dieselbe Zustandsraumdiskretisierung nach Abschnitt 5.3.1 zu Grunde, es wird lediglich eine Aktion zwei- bzw. dreimal ausgeführt, um

---

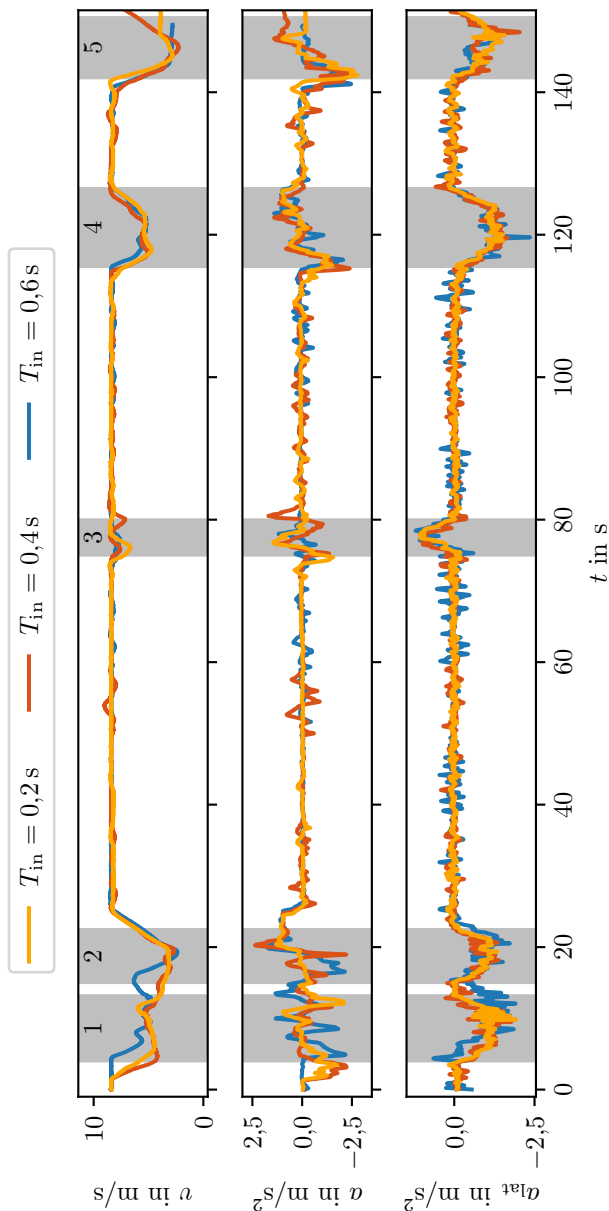
<sup>2</sup>Zwar kann eine Aktion über einen bestimmten Zeitraum, wie z. B. 0,6 s, ausgeführt werden, allerdings sollte während der konstanten Aktion trotzdem, wie in Abschnitt 5.3.7 beschrieben, nach möglichen Kollisionen gesucht werden. Wird dies dagegen nur alle  $T_{\text{in}}$  durchgeführt, kann es vorkommen, dass eine Kollision mit einem anderen Objekt übersehen wird.

zum nächsten Baumknoten zu gelangen. Im Folgenden wird zuerst das Systemverhalten analysiert, bevor in einem zweiten Schritt die Auswirkungen auf den Suchbaum untersucht werden.

### Auswirkungen auf das Systemverhalten

Abb. 6.8 zeigt für unterschiedliche Abtastzeiten  $T_{\text{in}}$  die Geschwindigkeit  $v$ , die longitudinale Beschleunigung  $a$  sowie die laterale Beschleunigung  $a_{\text{lat}}$  entlang der Teststrecke. Generell ist das Spurfolgen mit allen drei gewählten  $T_{\text{in}}$  möglich. Für größere  $T_{\text{in}}$  nehmen allerdings die lateralen Beschleunigungen zu, das Fahrzeug fängt an in der Fahrspur zu schwanken (vgl.  $a_{\text{lat}}$  z. B. zwischen 60 und 75 s). Während für  $T_{\text{in}} = 0,2\text{ s}$  kaum laterale Beschleunigungen auftreten, werden die Amplituden der Oszillationen für größere  $T_{\text{in}}$  größer. Hierdurch nimmt der Fahrkomfort des Systems stark ab. Besonders für  $T_{\text{in}} = 0,6\text{ s}$  sind die lateralen Schwankungen im Fahrzeug spürbar und werden vom Autor als störend empfunden.

Des Weiteren ist das Verhalten innerhalb der ersten Kurve interessant. Hier ergibt sich für den longitudinalen Verlauf bei  $T_{\text{in}} = 0,6\text{ s}$  ein oszillierendes Verhalten, das Fahrzeug beschleunigt mehrmals und bremst im Anschluss direkt wieder ab. Sowohl die lateralen Schwankungen auf der Geraden als auch die longitudinalen Schwankungen innerhalb der ersten Kurve lassen auf eine zu grobe Diskretisierung des lateralen Aktionsraumes (d.h.  $\delta_d(v_d)$ ,  $\omega_d(\delta_{d,i}(v_d), a_d)$ ) schließen. Denn während der Planung ist es das Ziel des Fahrzeugs, der Fahrspur möglichst gut zu folgen, da hier der Erfolgswert nach Abschnitt 5.3.6 am größten ist. Auf der Geraden bedeutet dies für den Fall, dass das Fahrzeug fast parallel zur Fahrspur fährt, dass eigentlich nur eine kleine Lenkwinkeländerung benötigt wird, um das Fahrzeug auf Kurs zu halten. Durch die zu grobe Diskretisierung kann dem System allerdings entweder keine oder nur eine zu große Lenkwinkeländerung vorgegeben werden, das System fängt an zu schwingen. Hiermit lässt sich auch die longitudinale Schwankung innerhalb einer Kurve erklären. Um eine Kurve möglichst gut abzufahren, wird ein bestimmter Lenkwinkel benötigt, welcher durch die zu grobe Diskretisierung nicht gestellt werden kann. Da die Diskretisierung des Lenkwinkels geschwindigkeitsabhängig ist, kann durch die Variation der Geschwindigkeit der Lenkwinkel in feineren Schritten geändert werden als durch die Wahl von  $\omega$  (vgl. Abb. 6.5). Somit findet der Planungsalgorithmus eine Trajektorie, welche der Spur folgt, auch wenn hierfür die Variation der Geschwindigkeit benötigt wird. Es ist anzunehmen, dass mit einer feineren lateralen Diskretisierung diese Schwankungen beseitigt werden können. Dieser Lösungsansatz ist be-



**Abbildung 6.8:** Messwerte dreier gefahrener Trajektorien mit unterschiedlichen Abtastzeiten  $T_{\text{in}}$ . Die in Abb. 6.3 markierten Fahrbahnkurven sind im Fahrverlauf nach  $T_{\text{in}} = 0,2 \text{ s}$  grau hinterlegt und entsprechend nummeriert. Da hier Messwerte dreier unterschiedlicher Fahrten mit variierenden Geschwindigkeiten über die Zeit dargestellt werden, können die eigentlichen Fahrzeugpositionen zueinander variieren. Auf Grund der ähnlichen Geschwindigkeitsverläufe wird dies vernachlässigt. Die mit 50 Hz aufgenommenen Beschleunigungsmesswerte sind durch einen Tiefpassfilter (Binomialfilter der Ordnung 40, Grenzfrequenz: 120 Hz) geglättet.

reits in Abb. 5.4 dargestellt. Die Einschränkungen, welche sich durch die Erhöhung von  $T_{\text{in}}$  ergeben, müssen durch eine Vergrößerung des Aktionsraumes kompensiert werden. Generell muss also darauf geachtet werden, dass die Aktionsdiskretisierung und die Abtastzeit entsprechend aufeinander abgestimmt sind, damit das Fahrzeug der Spur ausreichend schwingungsarm folgen kann.

### Auswirkungen auf den Suchbaum

Nach der Analyse der Auswirkungen unterschiedlicher  $T_{\text{in}}$  auf das Planungsergebnis, werden in diesem Abschnitt die Auswirkungen auf den vom Algorithmus erstellten Suchbaum untersucht. Der Suchbaum beeinflusst das Planungsergebnis, da dieses direkt aus dem Baum abgeleitet wird. Zur Analyse sind in Tabelle 6.2 die Anzahl an erkundeten Baumknoten sowie Iterationen des Algorithmus dargestellt. Während die Anzahl an Iterationen bei Erhöhung von  $T_{\text{in}}$  nahezu konstant bleibt, sinkt die Anzahl an erkundeten Baumknoten deutlich (bei  $T_{\text{in}} = 0,2\text{s}$  auf  $T_{\text{in}} = 0,6\text{s}$  um den Faktor 3,7). Dies kann auf die reduzierte maximale Größe des Suchbaumes zurückgeführt werden. Während für kleinere  $T_{\text{in}}$  eine größere Anzahl an möglichen Aktions-Zustands-Kombinationen existiert, welche einen guten Erfolgswert liefern, sinkt diese Anzahl an Kombinationen für größere  $T_{\text{in}}$ , da der Zustandsraum gröber diskretisiert wird. Dies lässt sich ebenso in Tabelle 6.3 beobachten. Auf Grund des exponentiellen Wachstums sinkt die maximal mögliche Baumgröße bei Erhöhung von  $T_{\text{in}}$  um mehrere Größenordnungen, wodurch die prozentuale Erkundung des Baumes in etwa im selben Maße ansteigt. Daraus lässt sich ableiten, dass je kleiner  $T_{\text{in}}$  gewählt wird, desto mehr muss die maximale Größe des Suchbaums durch andere Maßnahmen, wie z. B. durch Verwendung von Heuristiken (vgl. Abschnitte 5.3.4 und 5.3.5), kompensiert werden.

Um zu analysieren wie sich diese reduzierte Anzahl an Baumknoten auf den Suchbaum auswirkt, ist in Abb. 6.9 für jedes  $T_{\text{in}}$  ein zugehöriger Suchbaum dargestellt. Das Fahrzeug befindet sich auf derselben Straße und besitzt dieselbe Startgeschwindigkeit von  $3,2\text{ m/s}$ . Dabei kann Folgendes beobachtet werden. Wird  $T_{\text{in}}$  erhöht, werden zwar weniger Baumknoten erstellt, allerdings breiten sich diese über einen größeren Bereich aus. Während für  $T_{\text{in}} = 0,2\text{s}$  alle explorierten Knoten innerhalb des fahrbaren Bereichs, die meisten sogar innerhalb der eigenen Fahrspur, liegen, erhöht sich für größere  $T_{\text{in}}$  der Anteil an Knoten, welche außerhalb des fahrbaren Bereichs liegen. Dies kann erneut auf die reduzierte maximale Baumgröße sowie die gröbere Diskretisierung zurückgeführt werden. Eine einzelne Ak-

**Tabelle 6.2:** Mittelwert und Standardabweichung (SD) der Anzahl an Baumknoten sowie Iterationen für unterschiedliche Abtastzeiten  $T_{in}$ .

$T_{in}$	Baumknoten		Iterationen	
	Mittel	SD	Mittel	SD
0,2 s	16410	4110	29598	3582
0,4 s	10296	3043	30646	3356
0,6 s	4457	2266	31319	3432

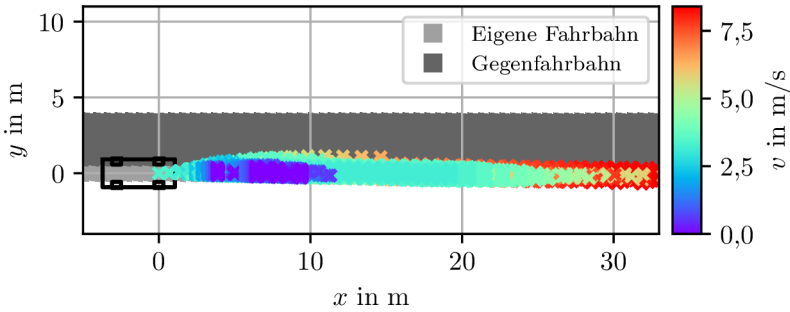
**Tabelle 6.3:** Anzahl aller möglichen Baumknoten<sup>3</sup> verglichen mit der mittleren, prozentualen Anzahl an explorierten Baumknoten (Mittelwert aus Tabelle 6.2 im Verhältnis zur möglichen Baumgröße) für unterschiedliche Abtastzeiten  $T_{in}$  bei einem Planungshorizont von  $T_{hor} = 6$  s.

$T_{in}$	Anzahl möglicher Baumknoten	Prozentuale, mittlere Erkundung
0,2 s	$4,2 \cdot 10^{28}$	$3,8 \cdot 10^{-23} \%$
0,4 s	$2,1 \cdot 10^{14}$	$5,0 \cdot 10^{-9} \%$
0,6 s	$3,5 \cdot 10^9$	$1,3 \cdot 10^{-4} \%$

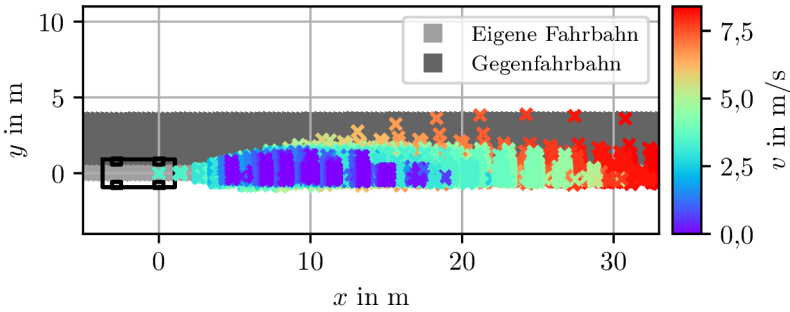
tion hat bei geringerem  $T_{in}$  weniger Einfluss auf die Fahrzeugbewegung, sodass sich der Baum für eine laterale Ausbreitung deutlich mehr verzweigen muss als bei höheren  $T_{in}$ . Dies wird für geringe  $T_{in}$  nicht umgesetzt, da Explorationen innerhalb der Gegenspür nach Abschnitt 5.3.6 schlechtere Erfolgswerte bringen. Dagegen werden für höhere  $T_{in}$  solche Explorationen ausprobiert, da der Suchbaum bereits ab dem zweiten Kinderknoten die Gegenfahrbahn erreicht.

Bei vielen verfügbaren Aktionen, die zu einem ähnlich guten Ergebnis führen, hat ein Suchalgorithmus Schwierigkeiten schnell gegen die beste Lösung zu konvergieren. Für den Algorithmus ist es einfacher, wenn die Wahl einer einzelnen Aktion einen größeren Einfluss auf das Ergebnis hat. Daher ist zu empfehlen, dass die Abtastzeit und die zugehörige Diskretisierung so groß wie möglich gewählt werden, allerdings nur so groß, dass

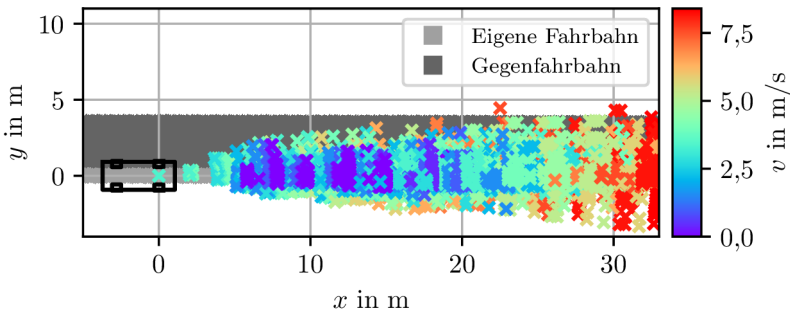
<sup>3</sup>Zur Berechnung aller möglichen Baumknoten wird hier auf Gleichung (5.5) zurückgegriffen. Hierbei wird nicht beachtet, dass der Suchbaum nach Abschnitt 5.3.2 frühzeitig beendet werden kann, oder dass einzelne Aktionen abgeschlossen sind (z. B. da bereits ein maximaler Lenkwinkel gestellt wird). Daher wird die eigentliche Anzahl an Baumknoten überschätzt.



(a) Monte-Carlo Suchbaum für  $T_{in} = 0,2\text{s}$  mit 22044 Knoten, erstellt durch 31650 Iterationen.



(b) Monte-Carlo Suchbaum für  $T_{in} = 0,4\text{s}$  mit 12945 Knoten, erstellt durch 34354 Iterationen.



(c) Monte-Carlo Suchbaum für  $T_{in} = 0,6\text{s}$  mit 7260 Knoten, erstellt durch 34326 Iterationen.

**Abbildung 6.9:** Monte-Carlo Suchbäume für unterschiedliche Abtastzeiten  $T_{in}$ .



noch eine komfortable Lösung gefunden werden kann. Da dies schwierig ist, kann ggf. auch eine gestaffelte Erhöhung der Abtastzeit Verbesserungen bringen. Bei dieser Erhöhung werden z. B. die Aktionen vom Wurzelknoten ausgehend mit 0,2s ausgeführt, bevor diese über den Planungshorizont schrittweise erhöht werden. Somit kann der Teil der geplanten Trajektorie, welcher ausgeführt wird, fein aufgelöst werden und der Teil, der in weiterer Zukunft liegt, gröber.

## 6.4 Zusammenfassung

In diesem Kapitel wird im Rahmen des Projekts PRORETA 5 eine erste Umsetzung der CarPre-Trajektorienplanung auf einem prototypischen Fahrzeug präsentiert. Zuerst wird das Projekt PRORETA 5 sowie das prototypische Fahrzeug mitsamt der verwendeten Softwarestruktur vorgestellt. Im Anschluss wird die Trajektorienplanung anhand eines Spurfolgenzenarios auf der vorhandenen Teststrecke evaluiert. Hierzu wird die Laufzeit des Algorithmus, der erreichte Fahrkomfort sowie der Einfluss der Abtastzeit  $T_{in}$  auf den Algorithmus untersucht. Durch den Jederzeit-Ansatz kann sichergestellt werden, dass die Laufzeit unter einem vorgeschriebenen Maximum liegt. Dabei kann die zur Verfügung stehende Rechenzeit voll ausgenutzt werden.

Für die Analyse des Fahrkomforts wird zuerst die in Kapitel 5 vorgestellte Diskretisierung in Relation zu den menschlichen Fahrdaten aus Kapitel 4 gesetzt. Im Anschluss wird eine Messfahrt analysiert. Hier ist zu sehen, dass das Fahrzeug, ähnlich wie ein menschlicher Fahrer, vor einer Kurve abbremst, durch die Kurve fährt und gegen Ende dieser aus der Kurve heraus beschleunigt.

In einer letzten Analyse wird der Einfluss der Abtastzeit auf den Planungsalgorithmus untersucht. Hierzu werden mehrere Messfahrten miteinander verglichen. Dabei zeigt sich, dass bei größeren Abtastzeiten eine feinere Diskretisierung gewählt werden muss. Diese wird benötigt, um die Aktionsreduktion, welche durch die Erhöhung der Abtastzeit entsteht, zu kompensieren. Durch das Verlangsamen des exponentiellen Wachstums des Suchbaums ist eine solche Erhöhung trotzdem von Vorteil und sollte in der Zukunft weiter untersucht werden. Zusammenfassend lässt sich festhalten, dass die Diskretisierung des Planungsraumes, also die Diskretisierung der Aktionen, die Diskretisierung der Zeit sowie die Diskretisierung der übrigen Zustände, von größter Bedeutung ist.

## 7 Zusammenfassung und Ausblick

In dieser Arbeit werden neue Überlegungen zur Trajektorienplanung autonomer Fahrzeuge vorgestellt. Als Basis aller Planungsgedanken wird das kinematische Einspurmodell als Bewegungsmodell für diese Arbeit ausgewählt. Im Anschluss werden drei neuartige Überlegungen zur Trajektorienplanung dargestellt, welche im Folgenden zusammengefasst werden.

Um die rechenintensive Aufgabe der Trajektorienplanung zu vereinfachen (F1), werden zunächst theoretische Überlegungen zum Referenzpunkt des kinematischen Einspurmodells vorgestellt. Der Referenzpunkt wird hierbei als ideal angesehen, wenn der benötigte seitliche Platzbedarf auf beiden Fahrzeugseiten gleich groß ist. Anhand von geometrischen Überlegungen wird für stationäre Kurvenfahrten die ideale Position hergeleitet, welche von der Krümmung der gefahrenen Kurve abhängt. Für Kurven mit variabler Krümmung existiert demnach kein idealer statischer Referenzpunkt. Basierend auf der Analyse verschiedener kritischer Manöver kann allerdings ein statischer Referenzpunkt auf der Vorderachse gewählt werden. Durch die mathematischen Überlegungen zum seitlichen Platzbedarf wird ein Kreismodell abgeleitet, welches für die vereinfachte Kollisionsüberprüfung oder Fahrspurzuordnung verwendet werden kann.

Als zweites werden Fahrverläufe menschlicher Fahrer an städtischen Kreuzungen analysiert. Dabei ist das Ziel Vereinfachungen für die Trajektorienplanung zu finden (F1). Zunächst wird versucht die von Menschen gefahrenen Trajektorien durch das kinematische Einspurmodell abzubilden. Durch die Variation der Abtastzeiten des Einspurmodells stellt sich heraus, dass die Trajektorien durch das verwendete Modell maximal mit einer Abtastzeit von  $T_{in} = 0,6s$  abgebildet werden können. Im Anschluss wird eine geschwindigkeitsabhängige, nichtlineare Transformation für die Lenkgeschwindigkeit und den Lenkwinkel eingeführt. Im transformierten Raum kann so ein einfaches Übergangsmodell auf Basis der multivariaten Gauß-Verteilung extrahiert werden.

Als drittes wird ein neuartiger Trajektorienplanungsansatz, die CarPre-Trajektorienplanung auf Basis der Monte-Carlo Baumsuche, vorgestellt.

---

Hierzu wird der Aktionsraum des autonomen Fahrzeugs mit Hilfe der nichtlinearen Lenkwinkeltransformation aus Kapitel 4 diskretisiert. Die Transformation ermöglicht eine einfache Limitierung der lateralen Fahrzeugbewegungen, sodass nur komfortable Bewegungen in der Planung berücksichtigt werden. Mit dem Ziel, einen Planungsansatz ohne darüberliegenden Manöverplaner zu entwickeln (F2), wird die Monte-Carlo Baumsuche ausgewählt, welche mit komplexen Planungsräumen gut umgehen kann. Durch seine Konfigurierbarkeit kann der Grundalgorithmus einfach an das Trajektorienplanungsproblem angepasst werden und ermöglicht so auch eine Planung in Echtzeit. Hierfür wird auch auf das Kreismodell aus Kapitel 3 zurückgegriffen. In ersten simulativen Versuchen wird die Machbarkeit des Planungsalgorithmus nachgewiesen.

Abschließend wird das im Projekt PRORETA5 verwendete prototypische Fahrzeug vorgestellt. Auf diesem sind weitere Versuche des CarPre-Trajektorienplaners durchgeführt und die Ergebnisse vorgestellt worden. Hierbei zeigt sich, dass die Diskretisierung des Planungsraumes einen großen Einfluss auf den Planungsalgorithmus und dessen Ergebnis hat. Dabei muss die Diskretisierung ausreichend fein sein, damit übliche Verkehrsszenarien planbar sind. Allerdings sollte sie nicht zu fein sein, da ansonsten der Planungsraum immer größer und damit das Finden einer geeigneten Trajektorie immer schwieriger wird.

Als weitere, zukünftige Arbeiten innerhalb des Projektes bleiben die Anpassung der Diskretisierung, sodass die Lenkmomentenbegrenzung nicht überschritten wird sowie die Einbindung der Objektprädiktion in den Planungsalgorithmus. Zukünftige Forschungsthemen umfassen die Erweiterung und Verifikation des Trajektorienplaners an komplexen, urbanen Verkehrsszenarien. Dabei kann der Planungsalgorithmus um verschiedene Aspekte erweitert werden. Um kooperatives Planen zu ermöglichen, ist eine Integration der Prädiktion anderer Verkehrsteilnehmer in den Planungsalgorithmus zu empfehlen. Ebenso kann die Verwendung von neuronalen Netzen zur Abschätzung der Trajektorienbewertung innerhalb der Monte-Carlo Baumsuche helfen, die Anzahl an Planungsexplorationen zu erhöhen und damit die Planung zu verbessern. Um die Performance des Planers weiter zu steigern, kann spezielle, an den Algorithmus angepasste, Hardware entwickelt werden. Durch solche Anpassungen können Algorithmen nicht nur schneller, sondern auch energieeffizienter arbeiten. Ein weiteres Arbeitsfeld ist die Umwandlung in einen probabilistischen Planer, sodass auch Unsicherheiten im Planungsprozess berücksichtigt werden.

# A Anhang

## A.1 Einfacher Trajektorienplaner

Zur Analyse der Auswirkungen unterschiedlicher Abtastzeiten auf die geplante Trajektorie wird für Abb. 4.3 ein vereinfachter Trajektorienplaner implementiert. Dieser wird im Folgenden vorgestellt.

Im einfachen Planungsalgorithmus wird, ebenso wie beim CarPre-Trajektorienplaner aus Kapitel 5, ein diskreter Suchbaum auf Basis des erweiterten kinematischen Einspurmodells (vgl. Gleichungen (4.1) bis (4.5)) aufgebaut. Als diskrete Aktionen stehen in jedem Baumknoten diskrete Lenkwinkelgeschwindigkeiten  $\omega_d(v)$  zur Verfügung. Dies bedeutet, dass das Fahrzeug in longitudinaler Bewegungsrichtung nicht beschleunigen oder abbremsen kann, es wird nur der laterale Trajektorienverlauf betrachtet. Hierzu wird auf die geschwindigkeitsabhängige Lenkwinkelgeschwindigkeitstransformation aus Gleichung (4.8) zurückgegriffen. Für die diskrete Lenkwinkelgeschwindigkeit  $\omega_d(v)$  mit  $n_\omega \in \{2p - 1 \mid p \in \mathbb{N}\}$  diskretisierten Zuständen ergibt sich

$$\omega_d(v) \in \left\{ q \cdot \Delta\omega(v) \mid q \in \mathbb{Z} \wedge |q| < \frac{n_\omega}{2} \right\},$$
$$\text{mit } \Delta\omega(v) = \frac{2 \cdot p_1 e^{-\frac{v}{p_2}}}{n_\omega - 1},$$

wobei die Parameter  $p_1$  und  $p_2$  die Parameter aus Abschnitt 4.4.1 darstellen. Die Geschwindigkeit  $v$  wird als konstant angenommen und für Abb. 4.3 auf 20 km/h gesetzt.

Zur Planung wird der gesamte Suchbaum mit allen möglichen Kombinationen bis zu einem festgelegten Planungshorizont  $T_{\text{hor}}$  berechnet. Im Anschluss wird aus allen  $\omega_d(v)$ -Kombinationen die beste Trajektorie extrahiert (diskretisiertes globales Optimum). Aus Gründen der Leistungsfähigkeit werden alle Knoten die weiter als 1,5 m von der Fahrspurmitte entfernt sind abgeschnitten. Nur so ist die Planung über größere  $T_{\text{hor}}$  möglich, da ansonsten der Arbeitsspeicher (32 GB) zur Zwischenspeicherung des Suchbaumes nicht ausreicht.

Dieser einfache Trajektorienplaner berechnet für die gewählte Diskretisierung die optimale Trajektorie. Um alle Möglichkeiten des exponentiell anwachsenden Suchbaums zu berechnen, werden große Rechenkapazitäten benötigt. Eine einzelne Planung kann mehrere Stunden dauern und dabei einen Großteil des Arbeitsspeichers belegen.

Im Gegensatz dazu berechnet der CarPre-Trajektorienplaner nicht die optimale Trajektorie, sondern nähert sich über die Zeit der optimalen Lösung an. Durch Heuristiken sowie der Simulation innerhalb der Monte-Carlo Baumsuche wird das Ergebnis abgeschätzt. Dadurch liegt ein erstes akzeptables Planungsergebnis, je nach System und Parameter, nach ca. 50 ms vor, da nur ein geringer Anteil des kompletten Suchbaumes berechnet wird (vgl. Tabelle 6.3).

## A.2 Fahrzeugparameter

**Tabelle A.1:** Fahrzeugparameter des Volkswagen Passat (B8).

Symbol	Beschreibung	Wert
$l$	Radstand	2,79 m
$l_{\text{th}}$	Länge zwischen Fahrzeugfront und Hinterachse	3,75 m
$b$	Fahrzeugbreite	1,83 m

## A.3 Parameter Verhaltensmodell

Parameter des multivariaten Gaußschen Modells  $M_N$

$$\boldsymbol{\mu} = \begin{bmatrix} \sigma_{a_{k-1}} \\ \sigma_{\bar{\omega}_{k-1}} \\ \sigma_{\bar{\delta}_k} \\ \sigma_{a_k} \\ \sigma_{\bar{\omega}_k} \end{bmatrix} = \begin{bmatrix} 0,0618 \\ 0,0001 \\ 0,0062 \\ 0,0350 \\ -0,0097 \end{bmatrix}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1,2640 & 0,0404 & 0,0387 & 0,8245 & -0,0060 \\ 0,0404 & 0,1441 & 0,0383 & -0,0074 & -0,0568 \\ 0,0387 & 0,0383 & 0,1123 & 0,0112 & -0,0567 \\ 0,8245 & -0,0074 & 0,0112 & 1,2656 & 0,0381 \\ -0,0060 & -0,0568 & -0,0567 & 0,0381 & 0,1611 \end{bmatrix}$$

Parameter des multivariaten studentischen  $t$  Modells  $M_{St}$ 

$$\boldsymbol{\mu} = \begin{bmatrix} \sigma_{a_{k-1}} \\ \sigma_{\bar{\omega}_{k-1}} \\ \sigma_{\bar{\delta}_k} \\ \sigma_{a_k} \\ \sigma_{\bar{\omega}_k} \end{bmatrix} = \begin{bmatrix} 0,0224 \\ -0,0006 \\ 0,0009 \\ -0,0109 \\ -0,0072 \end{bmatrix}$$

$$\boldsymbol{\Lambda} = \begin{bmatrix} 0,8332 & 0,0249 & 0,0192 & 0,5688 & -0,0114 \\ 0,0249 & 0,0554 & 0,0170 & -0,0116 & -0,0317 \\ 0,0192 & 0,0170 & 0,0315 & 0,0026 & -0,0211 \\ 0,5688 & -0,0116 & 0,0026 & 0,8190 & 0,0235 \\ -0,0114 & -0,0317 & -0,0211 & 0,0235 & 0,0604 \end{bmatrix}$$

$$\nu = 2,4920$$

## A.4 Parameter CarPre-Trajektorienplaner

Tabelle A.2: Parameter des CarPre-Trajektorienplaners.

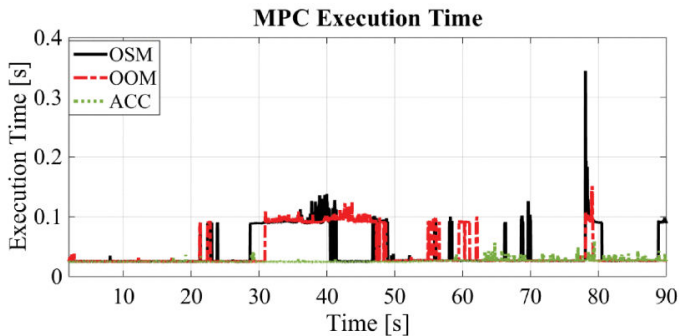
Parameter	Wert	Parameter	Wert
$a_{\text{lat,max}}$	1,3 m/s <sup>2</sup>	$n_{\delta}$	15
$a_{\text{max}}$	1 m/s <sup>2</sup>	$n_{\omega}$	3
$a_{\text{min}}$	-3 m/s <sup>2</sup>	$n_{\psi}$	512
$\Delta a$	1 m/s <sup>2</sup>	$T_{\text{akt}}$	0,2 s
$b_R$	2	$t_{\text{calc}}$	0,1 s
$c$	0,5	$T_{\text{hor}}$	6 s
$d_{\text{max}}$	12,8 m	$T_{\text{in}}$	0,2 s; 0,4 s; 0,6 s
$k_{\text{acc}}$	6	$v_{\text{ziel}}$	8,4 m/s
$k_{\omega}$	2	$r_{\text{ego}}$	1,31 m
$\kappa_{\text{max}}$	0,13 m <sup>-1</sup>	$R_{\text{ini},1}$	0,1
$\epsilon_R$	0,02	$R_{\text{ini},2}$	0,05
$n_{\text{start}}$	1	$R_{\text{spur,g}}$	0,3
$n_{\text{threads}}$	10	$R_{\text{verlust}}$	0,5
$n_{\text{traj}}$	30; 15; 10	$v_{\text{max}}$	8,4 m/s

## A.5 Experimente

**Tabelle A.3:** Überblick über die durchgeführten Versuche.

	$T_{\text{in}}$	Anzahl Runden	Überschreitung Limit
Messung 1	0,2 s	1	Ja, Kurve 5
Messung 2	0,2 s	3	Nein
Messung 3	0,2 s	1	Ja, Kurve 5
Messung 4	0,4 s	1	Ja, Kurve 5
Messung 5	0,4 s	0	Ja, Kurve 2
Messung 6	0,4 s	2	Nein
Messung 7	0,6 s	1	Ja, Kurve 5
Messung 8	0,6 s	1	Ja, Kurve 5

## A.6 Referenzierte Abbildungen



**Abbildung A.1:** Abb. 12 aus [131].

# Literaturverzeichnis

- [1] Z. Ajanovic, B. Lacevic, B. Shyrokau, M. Stolz und M. Horn. „Search-Based Optimal Motion Planning for Automated Driving“. In: **2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. IEEE, Okt. 2018. DOI: 10.1109/iros.2018.8593813.
- [2] R. Alterovitz, S. Patil und A. Derbakova. „Rapidly-Exploring Roadmaps: Weighing Exploration vs. Refinement in Optimal Motion Planning“. In: **2011 IEEE International Conference on Robotics and Automation**. IEEE, Mai 2011. DOI: 10.1109/icra.2011.5980286.
- [3] T. Arens, F. Hettlich, C. Karpfinger, U. Kockelkorn, K. Lichtenegger und H. Stachel. **Mathematik**. Springer Berlin Heidelberg, 2018. DOI: 10.1007/978-3-662-56741-8.
- [4] A. Bacha, C. Bauman, R. Faruque, M. Fleming, C. Terwelp, C. Reinholtz, D. Hong, A. Wicks, T. Alberi, D. Anderson, S. Cacciola, P. Currier, A. Dalton, J. Farmer, J. Hurdus, S. Kimmel, P. King, A. Taylor, D. V. Covern und M. Webster. „Odin: Team VictorTango’s Entry in the DARPA Urban Challenge“. In: **Journal of Field Robotics** 25.8 (Aug. 2008), S. 467–492. DOI: 10.1002/rob.20248.
- [5] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. M. Paixão, F. Mutz, L. de Paula Veronese, T. Oliveira-Santos und A. F. D. Souza. „Self-Driving Cars: A Survey“. In: **Expert Systems with Applications** 165 (März 2021), S. 113816. DOI: 10.1016/j.eswa.2020.113816.
- [6] E. Bakker, L. Nyborg und H. Pacejka. „Tyre Modelling for Use in Vehicle Dynamics Studies“. In: SAE International, Feb. 1987. DOI: 10.4271/870421.
- [7] T. Bandyopadhyay, K. S. Won, E. Frazzoli, D. Hsu, W. S. Lee und D. Rus. „Intention-Aware Motion Planning“. In: **Springer Tracts in Advanced Robotics**. Springer Berlin Heidelberg, 2013, S. 475–491. DOI: 10.1007/978-3-642-36279-8\_29.



- [8] H. Bast, D. Delling, A. Goldberg, M. Müller-Hannemann, T. Pajor, P. Sanders, D. Wagner und R. F. Werneck. „Route Planning in Transportation Networks“. In: **Algorithm Engineering**. Springer International Publishing, 2016, S. 19–80. DOI: 10.1007/978-3-319-49487-6\_2.
- [9] H. Bellem. „Comfort in Automated Driving: Analysis of Driving Style Preference in Automated Driving“. Diss. Technische Universität Chemnitz, 2018. URL: <https://nbn-resolving.org/urn:nbn:de:bsz:ch1-qucosa2-214074>.
- [10] R. Bellman. **Dynamic Programming**. Princeton, N.J: Univ. Pr, 1972. ISBN: 9780691079516.
- [11] R. Bobisse und A. Pavia. **Automatic for the City**. RIBA Publishing, Aug. 2019. DOI: 10.4324/9780429347849.
- [12] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater und L. Eckstein. „The inD Dataset: A Drone Dataset of Naturalistic Road User Trajectories at German Intersections“. In: **2020 IEEE Intelligent Vehicles Symposium (IV)**. IEEE, Okt. 2020. DOI: 10.1109/iv47402.2020.9304839.
- [13] J. Bohren, T. Foote, J. Keller, A. Kushleyev, D. Lee, A. Stewart, P. Vernaza, J. Derenick, J. Spletzer und B. Satterfield. „Little Ben: The Ben Franklin Racing Team’s Entry in the 2007 DARPA Urban Challenge“. In: **Journal of Field Robotics** 25.9 (Sep. 2008), S. 598–614. DOI: 10.1002/rob.20260.
- [14] P. S. Bokare und A. K. Maurya. „Acceleration-Deceleration Behaviour of Various Vehicle Types“. In: **Transportation Research Procedia** 25 (2017), S. 4733–4749. DOI: 10.1016/j.trpro.2017.05.486.
- [15] M. A. Branch, T. F. Coleman und Y. Li. „A Subspace, Interior, and Conjugate Gradient Method for Large-Scale Bound-Constrained Minimization Problems“. In: **SIAM Journal on Scientific Computing** 21.1 (Jan. 1999), S. 1–23. DOI: 10.1137/s1064827595289108.
- [16] S. Brechtel, T. Gindele und R. Dillmann. „Probabilistic Decision-Making Under Uncertainty for Autonomous Driving Using Continuous POMDPs“. In: **17th International IEEE Conference on Intelligent Transportation Systems (ITSC)**. IEEE, Okt. 2014. DOI: 10.1109/itsc.2014.6957722.

- [17] S. Brechtel, T. Gindele und R. Dillmann. „Probabilistic MDP-Behavior Planning for Cars“. In: **2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)**. IEEE, Okt. 2011. DOI: 10.1109/itsc.2011.6082928.
- [18] P. Brombacher, J. Masino, M. Frey und F. Gauterin. „Driving Event Detection and Driving Style Classification Using Artificial Neural Networks“. In: **2017 IEEE International Conference on Industrial Technology (ICIT)**. IEEE, März 2017. DOI: 10.1109/icit.2017.7915497.
- [19] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis und S. Colton. „A Survey of Monte Carlo Tree Search Methods“. In: **IEEE Transactions on Computational Intelligence and AI in Games 4.1** (März 2012), S. 1–43. DOI: 10.1109/tciaig.2012.2186810.
- [20] C. Burger, T. Schneider und M. Lauer. „Interaction Aware Cooperative Trajectory Planning for Lane Change Maneuvers in Dense Traffic“. In: **2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)**. IEEE, Sep. 2020. DOI: 10.1109/itsc45102.2020.9294638.
- [21] G. Chaslot, M. Winands und J. van den Herik. „Parallel Monte-Carlo Tree Search“. In: **Computers and Games**. Springer Berlin Heidelberg, 2008, S. 60–71. DOI: 10.1007/978-3-540-87608-3\_6.
- [22] G. Chaslot, M. Winands, J. v. d. Herik, J. W. Uiterwijk und B. Bouzy. „Progressive Strategies for Monte-Carlo Tree Search“. In: **New Mathematics and Natural Computation 4.03** (2008), S. 343–357. DOI: 10.1142/s1793005708001094.
- [23] J. Chen, C. Zhang, J. Luo, J. Xie und Y. Wan. „Driving Maneuvers Prediction Based Autonomous Driving Control by Deep Monte Carlo Tree Search“. In: **IEEE Transactions on Vehicular Technology 69.7** (Juli 2020), S. 7146–7158. DOI: 10.1109/tvt.2020.2991584.
- [24] A. Couëtoux, J.-B. Hoock, N. Sokolovska, O. Teytaud und N. Bonnard. „Continuous Upper Confidence Trees“. In: **Lecture Notes in Computer Science**. Springer Berlin Heidelberg, 2011, S. 433–445. DOI: 10.1007/978-3-642-25566-3\_32.

- [25] R. Coulom. „Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search“. In: **Computers and Games**. Springer Berlin Heidelberg, 2007, S. 72–83. DOI: 10.1007/978-3-540-75538-8\_7.
- [26] F. Damerow. „Situation-Based Risk Evaluation and Behavior Planning“. en. Diss. Technische Universität Darmstadt, 2018. URL: <http://tuprints.ulb.tu-darmstadt.de/6790/>.
- [27] F. Damerow und J. Eggert. „Risk-Aversive Behavior Planning under Multiple Situations with Uncertainty“. In: **2015 IEEE 18th International Conference on Intelligent Transportation Systems**. IEEE, Sep. 2015. DOI: 10.1109/itsc.2015.113.
- [28] F. Damerow, T. Puphal, Y. Li und J. Eggert. „Risk-Based Driver Assistance for Approaching Intersections of Limited Visibility“. In: **2017 IEEE International Conference on Vehicular Electronics and Safety (ICVES)**. IEEE, Juni 2017. DOI: 10.1109/icves.2017.7991922.
- [29] D. Delling, A. V. Goldberg, T. Pajor und R. F. Werneck. „Customizable Route Planning in Road Networks“. In: **Transportation Science** 51.2 (Mai 2017), S. 566–591. DOI: 10.1287/trsc.2014.0579.
- [30] A. P. Dempster, N. M. Laird und D. B. Rubin. „Maximum Likelihood from Incomplete Data Via the EM Algorithm“. In: **Journal of the Royal Statistical Society: Series B (Methodological)** 39.1 (Sep. 1977), S. 1–22. DOI: 10.1111/j.2517-6161.1977.tb01600.x.
- [31] E. W. Dijkstra. „A Note on Two Problems in Connexion with Graphs“. In: **Numerische Mathematik** 1.1 (Dez. 1959), S. 269–271. DOI: 10.1007/bf01386390.
- [32] W. Ding, L. Zhang, J. Chen und S. Shen. „EPSILON: An Efficient Planning System for Automated Vehicles in Highly Interactive Environments“. In: **IEEE Transactions on Robotics** (2021), S. 1–21. DOI: 10.1109/tro.2021.3104254.
- [33] G. Dissanayake, S. Sukkarieh, E. Nebot und H. Durrant-Whyte. „The Aiding of a Low-Cost Strapdown Inertial Measurement Unit Using Vehicle Model Constraints for Land Vehicle Applications“. In: **IEEE Transactions on Robotics and Automation** 17.5 (2001), S. 731–747. DOI: 10.1109/70.964672.

- [34] S. Dixit, U. Montanaro, M. Dianati, D. Oxtoby, T. Mizutani, A. Mouzakitis und S. Fallah. „Trajectory Planning for Autonomous High-Speed Overtaking in Structured Environments Using Robust MPC“. In: **IEEE Transactions on Intelligent Transportation Systems** 21.6 (Juni 2020), S. 2310–2323. DOI: 10.1109/tits.2019.2916354.
- [35] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez und V. Koltun. „CARLA: An Open Urban Driving Simulator“. In: **Proceedings of the 1st Annual Conference on Robot Learning**. Hrsg. von S. Levine, V. Vanhoucke und K. Goldberg. Bd. 78. Proceedings of Machine Learning Research. PMLR, Nov. 2017, S. 1–16. URL: <https://proceedings.mlr.press/v78/dosovitskiy17a.html>.
- [36] M. Elbanhawi, M. Simic und R. Jazar. „In the Passenger Seat: Investigating Ride Comfort Measures in Autonomous Cars“. In: **IEEE Intelligent Transportation Systems Magazine** 7.3 (2015), S. 4–17. DOI: 10.1109/mits.2015.2405571.
- [37] H. Eren, S. Makinist, E. Akin und A. Yilmaz. „Estimating Driving Behavior by a Smartphone“. In: **2012 IEEE Intelligent Vehicles Symposium**. IEEE, Juni 2012. DOI: 10.1109/ivs.2012.6232298.
- [38] D. Ferguson, T. M. Howard und M. Likhachev. „Motion Planning in Urban Environments“. In: **Journal of Field Robotics** 25.11-12 (Nov. 2008), S. 939–960. DOI: 10.1002/rob.20265.
- [39] E. Galceran, A. G. Cunningham, R. M. Eustice und E. Olson. „Multipolicy Decision-Making for Autonomous Driving via Change-point-Based Behavior Prediction: Theory and Experiment“. In: **Autonomous Robots** 41.6 (Aug. 2017), S. 1367–1382. ISSN: 1573-7527. DOI: 10.1007/s10514-017-9619-z. URL: <https://doi.org/10.1007/s10514-017-9619-z>.
- [40] M. C. Gameiro da Silva. „Measurements of Comfort in Vehicles“. In: **Measurement Science and Technology** 13.6 (Apr. 2002), R41–R60. DOI: 10.1088/0957-0233/13/6/201.
- [41] T. Gasser. **Rechtsfolgen zunehmender Fahrzeugautomatisierung gemeinsamer Schlussbericht der Projektgruppe**. Bremerhaven: Wirtschaftsverl. NW, Verl. für Neue Wiss, 2012. ISBN: 9783869181899.

- [42] F. Glatzki, M. Lippert und H. Winner. „Behavioral Attributes for a Behavior-Semantic Scenery Description (BSSD) for the Development of Automated Driving Functions“. In: **2021 IEEE International Intelligent Transportation Systems Conference (IT-SC)**. IEEE, Sep. 2021. DOI: 10.1109/itsc48978.2021.9564892.
- [43] F. Glatzki und H. Winner. „Inferenz von Verhaltensattributen der Verhaltenssemantischen Szeneriebeschreibung für die Entwicklung automatisierter Fahrfunktionen“. de. In: **14. Uni-DAS e.V. Workshop Fahrerassistenz und automatisiertes Fahren: 09. – 11.05.2022**. Darmstadt: Uni-DAS e.V., Mai 2022, S. 151–164. ISBN: 978-3-941543-65-2. URL: <http://tubiblio.ulb.tu-darmstadt.de/132575/>.
- [44] D. Gonzalez, J. Perez, V. Milanés und F. Nashashibi. „A Review of Motion Planning Techniques for Automated Vehicles“. In: **IEEE Transactions on Intelligent Transportation Systems** 17.4 (Apr. 2016), S. 1135–1145. DOI: 10.1109/tits.2015.2498841.
- [45] T. Gu, J. M. Dolan und J.-W. Lee. „On-Road Trajectory Planning for General Autonomous Driving with Enhanced Tunability“. In: **Intelligent Autonomous Systems 13**. Springer International Publishing, Sep. 2015, S. 247–261. DOI: 10.1007/978-3-319-08338-4\_19.
- [46] R. Guidolini, L. G. Scart, L. F. R. Jesus, V. B. Cardoso, C. Baudue und T. Oliveira-Santos. „Handling Pedestrians in Crosswalks Using Deep Neural Networks in the IARA Autonomous Car“. In: **2018 International Joint Conference on Neural Networks (IJCNN)**. IEEE, Juli 2018. DOI: 10.1109/ijcnn.2018.8489397.
- [47] B. Gutjahr, L. Groll und M. Werling. „Lateral Vehicle Trajectory Optimization Using Constrained Linear Time-Varying MPC“. In: **IEEE Transactions on Intelligent Transportation Systems** (2016), S. 1–10. DOI: 10.1109/tits.2016.2614705.
- [48] P. Hart, N. Nilsson und B. Raphael. „A Formal Basis for the Heuristic Determination of Minimum Cost Paths“. In: **IEEE Transactions on Systems Science and Cybernetics** 4.2 (1968), S. 100–107. DOI: 10.1109/tssc.1968.300136.
- [49] X. He, D. Xu, H. Zhao, M. Moze, F. Aioun und F. Guillemard. „A Human-like Trajectory Planning Method by Learning from Naturalistic Driving Data“. In: **2018 IEEE Intelligent Vehicles Symposium (IV)**. IEEE, Juni 2018. DOI: 10.1109/ivs.2018.8500448.

- [50] B. C. Heinrich, D. Fassbender und H.-J. Wuensche. „Faster Collision Checks for Car-Like Robot Motion Planning“. In: **2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. IEEE, Okt. 2018. DOI: 10.1109/iro.2018.8594454.
- [51] S. Heinrich. „Planning Universal On-Road Driving Strategies for Automated Vehicles“. Diss. Freie Universität Berlin, 2018. DOI: 10.1007/978-3-658-21954-3.
- [52] A. Herrmann, W. Brenner und R. Stadler. **Autonomous Driving**. Emerald Publishing Limited, März 2018. DOI: 10.1108/9781787148338.
- [53] X. Hu, L. Chen, B. Tang, D. Cao und H. He. „Dynamic Path Planning for Autonomous Driving on Various Roads with Avoidance of Static and Moving Obstacles“. In: **Mechanical Systems and Signal Processing** 100 (Feb. 2018), S. 482–500. DOI: 10.1016/j.ymsp.2017.07.019.
- [54] Y. Huang, H. Wang, A. Khajepour, H. Ding, K. Yuan und Y. Qin. „A Novel Local Motion Planning Framework for Autonomous Vehicles Based on Resistance Network and Model Predictive Control“. In: **IEEE Transactions on Vehicular Technology** 69.1 (Jan. 2020), S. 55–66. DOI: 10.1109/tvt.2019.2945934.
- [55] C. Hubmann, M. Becker, D. Althoff, D. Lenz und C. Stiller. „Decision Making for Autonomous Driving Considering Interaction and Uncertain Prediction of Surrounding Vehicles“. In: **2017 IEEE Intelligent Vehicles Symposium (IV)**. IEEE, Juni 2017. DOI: 10.1109/ivs.2017.7995949.
- [56] R. Isermann, E. Bender, R. Bruder, M. Darms, M. Schorn, U. Stählin und H. Winner. „Antikollisionssystem PRORETA – Integrierte Lösung für ein unfallvermeidendes Fahrzeug“. In: **Handbuch Fahrerassistenzsysteme**. Vieweg+Teubner, 2009, S. 632–646. DOI: 10.1007/978-3-8348-9977-4\_42.
- [57] J. H. Jeon, R. V. Cowlagi, S. C. Peters, S. Karaman, E. Frazzoli, P. Tsiotras und K. Iagnemma. „Optimal Motion Planning with the Half-Car Dynamical Model for Autonomous High-Speed Driving“. In: **2013 American Control Conference**. IEEE, Juni 2013. DOI: 10.1109/acc.2013.6579835.

- [58] K. Kant und S. W. Zucker. „Toward Efficient Trajectory Planning: The Path-Velocity Decomposition“. In: **The International Journal of Robotics Research** 5.3 (Sep. 1986), S. 72–89. DOI: 10.1177/027836498600500304.
- [59] S. Karaman und E. Frazzoli. „Incremental Sampling-based Algorithms for Optimal Motion Planning“. In: **Robotics: Science and Systems VI**. Robotics: Science und Systems Foundation, Juni 2010. DOI: 10.15607/rss.2010.vi.034.
- [60] S. Karaman und E. Frazzoli. „Sampling-Based Algorithms for Optimal Motion Planning“. In: **The International Journal of Robotics Research** 30.7 (Juni 2011), S. 846–894. DOI: 10.1177/0278364911406761.
- [61] C. Katrakazas, M. Qudus, W.-H. Chen und L. Deka. „Real-Time Motion Planning Methods for Autonomous On-Road Driving: State-of-the-Art and Future Research Directions“. In: **Transportation Research Part C: Emerging Technologies** 60 (Nov. 2015), S. 416–442. DOI: 10.1016/j.trc.2015.09.011.
- [62] E. Kayacan, H. Ramon und W. Saeys. „Robust Trajectory Tracking Error Model-Based Predictive Control for Unmanned Ground Vehicles“. In: **IEEE/ASME Transactions on Mechatronics** 21.2 (Apr. 2016), S. 806–814. DOI: 10.1109/tmech.2015.2492984.
- [63] S. Klingelschmitt, F. Damerow, V. Willert und J. Eggert. „Probabilistic Situation Assessment Framework for Multiple, Interacting Traffic Participants in Generic Traffic Scenes“. In: **2016 IEEE Intelligent Vehicles Symposium (IV)**. IEEE, Juni 2016. DOI: 10.1109/ivs.2016.7535533.
- [64] L. Kocsis, C. Szepesvári und J. Willemsen. „Improved Monte-Carlo Search“. In: **Univ. Tartu, Estonia, Tech. Rep 1** (2006).
- [65] J. Kong, M. Pfeiffer, G. Schildbach und F. Borrelli. „Kinematic and Dynamic Vehicle Models for Autonomous Driving Control Design“. In: **2015 IEEE Intelligent Vehicles Symposium (IV)**. IEEE, Juni 2015. DOI: 10.1109/ivs.2015.7225830.
- [66] K. Kurzer, M. Fechner und J. M. Zollner. „Accelerating Cooperative Planning for Automated Vehicles with Learned Heuristics and Monte Carlo Tree Search“. In: **2020 IEEE Intelligent Vehicles Symposium (IV)**. IEEE, Okt. 2020. DOI: 10.1109/iv47402.2020.9304597.

- [67] K. Kurzer, C. Zhou und J. M. Zollner. „Decentralized Cooperative Planning for Automated Vehicles with Hierarchical Monte Carlo Tree Search“. In: **2018 IEEE Intelligent Vehicles Symposium (IV)**. IEEE, Juni 2018. DOI: 10.1109/ivs.2018.8500712.
- [68] Y. Kuwata, G. Fiore, J. Teo, E. Frazzoli und J. How. „Motion Planning for Urban Driving Using RRT“. In: **2008 IEEE/RSJ International Conference on Intelligent Robots and Systems**. IEEE, Sep. 2008. DOI: 10.1109/iros.2008.4651075.
- [69] Y. Kuwata, S. Karaman, J. Teo, E. Frazzoli, J. How und G. Fiore. „Real-Time Motion Planning With Applications to Autonomous Urban Driving“. In: **IEEE Transactions on Control Systems Technology** 17.5 (Sep. 2009), S. 1105–1118. DOI: 10.1109/tcst.2008.2012116.
- [70] M. Lalli. **Autonomes Fahren und die Zukunft der Mobilität**. Springer Berlin Heidelberg, 2020. DOI: 10.1007/978-3-662-61812-7.
- [71] S. M. LaValle. **Planning Algorithms**. Cambridge University Press, 2006. DOI: 10.1017/cbo9780511546877.
- [72] S. M. Lavalle. **Rapidly-Exploring Random Trees: A New Tool for Path Planning**. Techn. Ber. 1998.
- [73] S. M. LaValle und J. J. Kuffner. „Randomized Kinodynamic Planning“. In: **The International Journal of Robotics Research** 20.5 (Mai 2001), S. 378–400. DOI: 10.1177/02783640122067453.
- [74] S. M. LaValle und J. J. Kuffner. „Rapidly-Exploring Random Trees: Progress and Prospects“. In: **Algorithmic and Computational Robotics**. A K Peters/CRC Press, Apr. 2001, S. 303–307. DOI: 10.1201/9781439864135-43.
- [75] J. Lee, W. Jeon, G.-H. Kim und K.-E. Kim. „Monte-Carlo Tree Search in Continuous Action Spaces with Value Gradients“. In: **Proceedings of the AAAI Conference on Artificial Intelligence** 34.04 (Apr. 2020), S. 4561–4568. DOI: 10.1609/aaai.v34i04.5885.
- [76] D. Lenz, T. Kessler und A. Knoll. „Tactical Cooperative Planning for Autonomous Highway Driving Using Monte-Carlo Tree Search“. In: **2016 IEEE Intelligent Vehicles Symposium (IV)**. IEEE, Juni 2016. DOI: 10.1109/ivs.2016.7535424.



- [77] W. Lim, S. Lee, K. Jo und M. Sunwoo. „Behavioral Trajectory Planning for Motion Planning in Urban Environments“. In: **2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)**. IEEE, Okt. 2017. DOI: 10.1109/itsc.2017.8317933.
- [78] P. F. Lima, G. C. Pereira, J. Martensson und B. Wahlberg. „Progress Maximization Model Predictive Controller“. In: **2018 21st International Conference on Intelligent Transportation Systems (ITSC)**. IEEE, Nov. 2018. DOI: 10.1109/itsc.2018.8569647.
- [79] M. V. Ly, S. Martin und M. M. Trivedi. „Driver Classification and Driving Style Recognition Using Inertial Sensors“. In: **2013 IEEE Intelligent Vehicles Symposium (IV)**. IEEE, Juni 2013. DOI: 10.1109/ivs.2013.6629603.
- [80] S. Manzinger, C. Pek und M. Althoff. „Using Reachable Sets for Trajectory Planning of Automated Vehicles“. In: **IEEE Transactions on Intelligent Vehicles** 6.2 (Juni 2021), S. 232–248. DOI: 10.1109/tiv.2020.3017342.
- [81] C. M. Martinez, M. Heucke, F.-Y. Wang, B. Gao und D. Cao. „Driving Style Recognition for Intelligent Vehicle Control and Advanced Driver Assistance: A Survey“. In: **IEEE Transactions on Intelligent Transportation Systems** 19.3 (März 2018), S. 666–676. DOI: 10.1109/tits.2017.2706978.
- [82] J. Mattingley, Y. Wang und S. Boyd. „Receding Horizon Control“. In: **IEEE Control Systems Magazine** 31.3 (Juni 2011), S. 52–65. DOI: 10.1109/mcs.2011.940571.
- [83] J. A. Matute, M. Marcano, S. Diaz und J. Perez. „Experimental Validation of a Kinematic Bicycle Model Predictive Control with Lateral Acceleration Consideration“. In: **IFAC-PapersOnLine** 52.8 (2019), S. 289–294. DOI: 10.1016/j.ifacol.2019.08.085.
- [84] J. A. Matute-Peaspan, M. Marcano, S. Diaz, A. Zubizarreta und J. Perez. „Lateral-Acceleration-Based Vehicle-Models-Blending for Automated Driving Controllers“. In: **Electronics** 9.10 (2020). ISSN: 2079-9292. DOI: 10.3390/electronics9101674. URL: <https://www.mdpi.com/2079-9292/9/10/1674>.
- [85] M. Maurer. **Autonomes Fahren : Technische, rechtliche und gesellschaftliche Aspekte**. Berlin, Germany: Springer Vieweg, 2015. ISBN: 9783662458532.

- [86] M. McNaughton, C. Urmson, J. M. Dolan und J.-W. Lee. „Motion Planning for Autonomous Driving With a Conformal Spatiotemporal Lattice“. In: **2011 IEEE International Conference on Robotics and Automation**. IEEE, Mai 2011. DOI: 10.1109/icra.2011.5980223.
- [87] Y. Meng, Y. Wu, Q. Gu und L. Liu. „A Decoupled Trajectory Planning Framework Based on the Integration of Lattice Searching and Convex Optimization“. In: **IEEE Access** 7 (2019), S. 130530–130551. DOI: 10.1109/access.2019.2940271.
- [88] C. Miyajima, Y. Nishiwaki, K. Ozawa, T. Wakita, K. Itou, K. Takeda und F. Itakura. „Driver Modeling Based on Driving Behavior and Its Evaluation in Driver Identification“. In: **Proceedings of the IEEE** 95.2 (Feb. 2007), S. 427–437. DOI: 10.1109/jproc.2006.888405.
- [89] C. Miyajima, H. Ukai, A. Naito, H. Amata, N. Kitaoka und K. Takeda. „Driver Risk Evaluation Based on Acceleration, Deceleration, and Steering Behavior“. In: **2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. IEEE, Mai 2011. DOI: 10.1109/icassp.2011.5946860.
- [90] F. Mohseni, J. Aslund, E. Frisk und L. Nielsen. „Fuel and Comfort Efficient Cooperative Control for Autonomous Vehicles“. In: **2017 IEEE Intelligent Vehicles Symposium (IV)**. IEEE, Juni 2017. DOI: 10.1109/ivs.2017.7995943.
- [91] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Etinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt und S. Thrun. „Junior: The Stanford Entry in the Urban Challenge“. In: **Journal of Field Robotics** 25.9 (Sep. 2008), S. 569–597. DOI: 10.1002/rob.20258.
- [92] Y. L. Murphey, R. Milton und L. Kiliaris. „Driver’s Style Classification Using Jerk Analysis“. In: **2009 IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems**. IEEE, März 2009. DOI: 10.1109/civvs.2009.4938719.

- [93] T. P. Nascimento, C. E. T. Dórea und L. M. G. Gonçalves. „Non-linear Model Predictive Control for Trajectory Tracking of Nonholonomic Mobile Robots“. In: **International Journal of Advanced Robotic Systems** 15.1 (Jan. 2018), S. 172988141876046. DOI: 10.1177/1729881418760461.
- [94] M. Naumann. „Probabilistic Motion Planning for Automated Vehicles“. en. Diss. Karlsruher Institut für Technologie, 2020. DOI: 10.5445/KSP/1000126389.
- [95] B. Ommer. **Autonomes Fahren**. 2020. DOI: 10.11588/HEIDOK.00027934.
- [96] P. F. Orzechowski, C. Burger und M. Lauer. „Decision-Making for Automated Vehicles Using a Hierarchical Behavior-Based Arbitration Scheme“. In: **2020 IEEE Intelligent Vehicles Symposium (IV)**. IEEE, Okt. 2020. DOI: 10.1109/iv47402.2020.9304723.
- [97] B. Paden, M. Cap, S. Z. Yong, D. Yershov und E. Frazzoli. „A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles“. In: **IEEE Transactions on Intelligent Vehicles** 1.1 (März 2016), S. 33–55. DOI: 10.1109/tiv.2016.2578706.
- [98] S. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghiani, Y. Eng, D. Rus und M. Ang. „Perception, Planning, Control, and Coordination for Autonomous Vehicles“. In: **Machines** 5.1 (Feb. 2017), S. 6. DOI: 10.3390/machines5010006.
- [99] P. Polack, F. Altche, B. d’Andrea-Novel und A. de La Fortelle. „The Kinematic Bicycle Model: A Consistent Model for Planning Feasible Trajectories for Autonomous Vehicles?“ In: **2017 IEEE Intelligent Vehicles Symposium (IV)**. IEEE, Juni 2017. DOI: 10.1109/ivs.2017.7995816.
- [100] C. Popp, C. Ziegler, M. Sippel und H. Winner. „Ideal Reference Point in Planning and Control for Automated Car-Like Vehicles“. In: **IEEE Transactions on Intelligent Vehicles** (2022). DOI: 10.1109/tiv.2022.3156370.
- [101] T. Puphal, M. Probst, M. Komuro, Y. Li und J. Eggert. „Comfortable Priority Handling with Predictive Velocity Optimization for Intersection Crossings“. In: **2019 IEEE Intelligent Transportation Systems Conference (ITSC)**. IEEE, Okt. 2019. DOI: 10.1109/itsc.2019.8917240.

- [102] R. Rajamani. **Vehicle Dynamics and Control**. New York: Springer Science, 2006. ISBN: 9780387263960.
- [103] J. Reif und M. Sharir. „Motion Planning in the Presence of Moving Obstacles“. In: **26th Annual Symposium on Foundations of Computer Science (sfcs 1985)**. IEEE, 1985. DOI: 10.1109/sfcs.1985.36.
- [104] J. H. Reif. „Complexity of the Mover’s Problem and Generalizations“. In: **20th Annual Symposium on Foundations of Computer Science (sfcs 1979)**. IEEE, Okt. 1979. DOI: 10.1109/sfcs.1979.10.
- [105] D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher und P. C. Sabeti. „Detecting Novel Associations in Large Data Sets“. In: **Science** 334.6062 (Dez. 2011), S. 1518–1524. DOI: 10.1126/science.1205438.
- [106] A. Sadat, M. Ren, A. Pokrovsky, Y.-C. Lin, E. Yumer und R. Urtasun. „Jointly Learnable Behavior and Trajectory Planning for Self-Driving Vehicles“. In: **2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. IEEE, Nov. 2019. DOI: 10.1109/iros40897.2019.8967615.
- [107] C. Samson. „Time-Varying Feedback Stabilization of Car-Like Wheeled Mobile Robots“. In: **The International Journal of Robotics Research** 12.1 (Feb. 1993), S. 55–64. DOI: 10.1177/027836499301200104.
- [108] K. Schmitt, R. Mannale und R. Isermann. „Situation Analysis, Warnings and Emergency Braking for Collision Avoidance in Overtaking Situations - PRORETA 2“. In: **IFAC Proceedings Volumes** 43.7 (Juli 2010), S. 744–749. DOI: 10.3182/20100712-3-de-2013.00061.
- [109] M. Schreier, V. Willert und J. Adamy. „Grid Mapping in Dynamic Road Environments: Classification of Dynamic Cell Hypothesis via Tracking“. In: **2014 IEEE International Conference on Robotics and Automation (ICRA)**. IEEE, Mai 2014. DOI: 10.1109/icra.2014.6907439.
- [110] R. Schubert, E. Richter und G. Wanielik. „Comparison and Evaluation of Advanced Motion Models for Vehicle Tracking“. In: **2008 11th International Conference on Information Fusion**. 2008, S. 1–6. URL: <https://ieeexplore.ieee.org/document/4632283>.

- [111] J. Schulz, C. Hubmann, N. Morin, J. Lochner und D. Burschka. „Learning Interaction-Aware Probabilistic Driver Behavior Models from Urban Scenarios“. In: **2019 IEEE Intelligent Vehicles Symposium (IV)**. IEEE, Juni 2019. DOI: 10.1109/ivs.2019.8814080.
- [112] W. Schwarting, J. Alonso-Mora und D. Rus. „Planning and Decision-Making for Autonomous Vehicles“. In: **Annual Review of Control, Robotics, and Autonomous Systems** 1.1 (Mai 2018), S. 187–210. DOI: 10.1146/annurev-control-060117-105157.
- [113] W. Schwarting, J. Alonso-Mora, L. Paull, S. Karaman und D. Rus. „Safe Nonlinear Trajectory Generation for Parallel Autonomy With a Dynamic Vehicle Model“. In: **IEEE Transactions on Intelligent Transportation Systems** 19.9 (Sep. 2018), S. 2994–3008. DOI: 10.1109/tits.2017.2771351.
- [114] J. Schwehr, S. Luthardt, H. Dang, M. Henzel, H. Winner, J. Adamy, J. Fürnkranz, V. Willert, B. Lattke, M. Höpfl und C. Wannemacher. „The PRORETA 4 City Assistant System“. In: **at - Automatisierungstechnik** 67.9 (Sep. 2019), S. 783–798. DOI: 10.1515/auto-2019-0051.
- [115] F. Seccamonte, J. Kabzan und E. Frazzoli. „On Maximizing Lateral Clearance of an Autonomous Vehicle in Urban Environments“. In: **2019 IEEE Intelligent Transportation Systems Conference (ITSC)**. IEEE, Okt. 2019. DOI: 10.1109/itsc.2019.8917353.
- [116] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel und D. Hassabis. „Mastering the Game of Go With Deep Neural Networks and Tree Search“. In: **Nature** 529.7587 (Jan. 2016), S. 484–489. DOI: 10.1038/nature16961.
- [117] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel und D. Hassabis. „Mastering the Game of Go Without Human Knowledge“. In: **Nature** 550.7676 (Okt. 2017), S. 354–359. DOI: 10.1038/nature24270.

- [118] T. Stahl, A. Wischnewski, J. Betz und M. Lienkamp. „Multilayer Graph-Based Trajectory Planning for Race Vehicles in Dynamic Scenarios“. In: **2019 IEEE Intelligent Transportation Systems Conference (ITSC)**. IEEE, Okt. 2019. DOI: 10.1109/itsc.2019.8917032.
- [119] A. Stentz. „Optimal and Efficient Path Planning for Partially-Known Environments“. In: **Proceedings of the 1994 IEEE International Conference on Robotics and Automation**. IEEE Comput. Soc. Press, 1994. DOI: 10.1109/robot.1994.351061.
- [120] J. Strohbeck, V. Belagiannis, J. Muller, M. Schreiber, M. Herrmann, D. Wolf und M. Buchholz. „Multiple Trajectory Prediction with Deep Temporal and Spatial Convolutional Neural Networks“. In: **2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. IEEE, Okt. 2020. DOI: 10.1109/iros45743.2020.9341327.
- [121] J. K. Subosits und J. C. Gerdes. „Impacts of Model Fidelity on Trajectory Optimization for Autonomous Vehicles in Extreme Maneuvers“. In: **IEEE Transactions on Intelligent Vehicles** 6.3 (Sep. 2021), S. 546–558. DOI: 10.1109/tiv.2021.3051325.
- [122] C. Sun, X. Zhang, Q. Zhou und Y. Tian. „A Model Predictive Controller With Switched Tracking Error for Autonomous Vehicle Path Tracking“. In: **IEEE Access** 7 (2019), S. 53103–53114. DOI: 10.1109/access.2019.2912094.
- [123] G. Tanzmeister, M. Friedl, D. Wollherr und M. Buss. „Efficient Evaluation of Collisions and Costs on Grid Maps for Autonomous Vehicle Motion Planning“. In: **IEEE Transactions on Intelligent Transportation Systems** 15.5 (Okt. 2014), S. 2249–2260. DOI: 10.1109/tits.2014.2313562.
- [124] The SciPy community. **scipy.optimize.least\_squares Documentation**. URL: [https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.least\\_squares.html](https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.least_squares.html) (besucht am 22.06.2022).
- [125] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrosseck, C. Koelen, C. Markey, C. Rummel, J. van Niekirk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian und P. Mahoney. „Stanley: The Robot

- That Won the DARPA Grand Challenge“. In: **Journal of Field Robotics** 23.9 (2006), S. 661–692. DOI: 10.1002/rob.20147.
- [126] J. Tromp. „The Number of Legal Go Positions“. In: **Computers and Games**. Springer International Publishing, 2016, S. 183–190. DOI: 10.1007/978-3-319-50935-8\_17.
- [127] E. Velenis, P. Tsiotras und J. Lu. „Optimality Properties and Driver Input Parameterization for Trail-Braking Cornering“. In: **European Journal of Control** 14.4 (Jan. 2008), S. 308–320. DOI: 10.3166/ejc.14.308-320.
- [128] Volkswagen. **Datenblatt VW Passat GTE**. Feb. 2018. URL: <https://www.motor-talk.de/forum/aktion/Attachment.html?attachmentId=767402> (besucht am 29.09.2020).
- [129] P. Wagner, R. Nippold, S. Gabloner und M. Margreiter. „Analyzing Human Driving Data an Approach Motivated by Data Science Methods“. In: **Chaos, Solitons & Fractals** 90 (Sep. 2016), S. 37–45. DOI: 10.1016/j.chaos.2016.02.008.
- [130] R. Wallace, A. Stentz, C. Thorpe, H. Maravec, W. Whittaker und T. Kanade. „First Results in Robot Road-Following.“ In: **Proceedings of the International Joint Conference on Artificial Intelligence**. Jan. 1985, S. 1089–1095.
- [131] Q. Wang, B. Ayalew und T. Weiskircher. „Predictive Maneuver Planning for an Autonomous Vehicle in Public Highway Traffic“. In: **IEEE Transactions on Intelligent Transportation Systems** 20.4 (Apr. 2019), S. 1303–1315. DOI: 10.1109/tits.2018.2848472.
- [132] W. Wang, J. Xi, A. Chong und L. Li. „Driving Style Classification Using a Semisupervised Support Vector Machine“. In: **IEEE Transactions on Human-Machine Systems** 47.5 (Okt. 2017), S. 650–660. DOI: 10.1109/thms.2017.2736948.
- [133] M. Werling, S. Kammel, J. Ziegler und L. Gröll. „Optimal Trajectories for Time-Critical Street Scenarios Using Discretized Terminal Manifolds“. In: **The International Journal of Robotics Research** 31.3 (Dez. 2011), S. 346–359. DOI: 10.1177/0278364911423042.

- [134] H. Winner, F. Lotz, E. Bauer, U. Konigorski, M. Schreier, J. Adamy, M. Pfromm, R. Bruder, S. Lücke und S. Cieler. „PRORETA 3: Comprehensive Driver Assistance by Safety Corridor and Cooperative Automation“. In: **Handbook of Driver Assistance Systems**. Springer International Publishing, Dez. 2015, S. 1449–1469. DOI: 10.1007/978-3-319-12352-3\_19.
- [135] K. H. Wray, S. J. Witwicki und S. Zilberstein. „Online Decision-Making for Scalable Autonomous Systems“. In: **Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence**. International Joint Conferences on Artificial Intelligence Organization, Aug. 2017. DOI: 10.24963/ijcai.2017/664.
- [136] S. Xu und H. Peng. „Design, Analysis, and Experiments of Preview Path Tracking Control for Autonomous Vehicles“. In: **IEEE Transactions on Intelligent Transportation Systems** 21.1 (Jan. 2020), S. 48–58. DOI: 10.1109/tits.2019.2892926.
- [137] K. Yang, X. Tang, Y. Qin, Y. Huang, H. Wang und H. Pu. „Comparative Study of Trajectory Tracking Control for Automated Vehicles via Model Predictive Control and Robust H-infinity State Feedback Control“. In: **Chinese Journal of Mechanical Engineering** 34.1 (Aug. 2021). DOI: 10.1186/s10033-021-00590-3.
- [138] D. Yokoyama und M. Toyoda. „Do Drivers’ Behaviors Reflect Their Past Driving Histories? - Large Scale Examination of Vehicle Recorder Data“. In: **2016 IEEE International Congress on Big Data (BigData Congress)**. IEEE, Juni 2016. DOI: 10.1109/bigdatacongress.2016.58.
- [139] Y. Zhang, H. Chen, S. L. Waslander, J. Gong, G. Xiong, T. Yang und K. Liu. „Hybrid Trajectory Planning for Autonomous Driving in Highly Constrained Environments“. In: **IEEE Access** 6 (2018), S. 32800–32819. DOI: 10.1109/access.2018.2845448.
- [140] C. Ziegler und J. Adamy. „Anytime Tree-Based Trajectory Planning for Urban Driving“. In: **IEEE Open Journal of Intelligent Transportation Systems** 4 (2023), S. 48–57. ISSN: 2687-7813. DOI: 10.1109/ojits.2023.3235986.
- [141] C. Ziegler, V. Willert und J. Adamy. „Modeling Driving Behavior of Human Drivers for Trajectory Planning“. In: **IEEE Transactions on Intelligent Transportation Systems** (2022), S. 1–10. DOI: 10.1109/tits.2022.3183204.



- 
- [142] J. Ziegler. „Optimale Bahn- und Trajektorienplanung für Automobile“. de. Diss. Karlsruher Institut für Technologie, 2015. DOI: 10.5445/IR/1000057846.
- [143] J. Ziegler und C. Stiller. „Fast Collision Checking for Intelligent Vehicle Motion Planning“. In: **2010 IEEE Intelligent Vehicles Symposium**. IEEE, Juni 2010. DOI: 10.1109/ivs.2010.5547976.
- [144] J. Ziegler und C. Stiller. „Spatiotemporal State Lattices for Fast Trajectory Planning in Dynamic On-Road Driving Scenarios“. In: **2009 IEEE/RSJ International Conference on Intelligent Robots and Systems**. IEEE, Okt. 2009. DOI: 10.1109/iros.2009.5354448.
- [145] J. Ziegler, P. Bender, T. Dang und C. Stiller. „Trajectory Planning for Bertha - A Local, Continuous Method“. In: **2014 IEEE Intelligent Vehicles Symposium Proceedings**. IEEE, Juni 2014. DOI: 10.1109/ivs.2014.6856581.