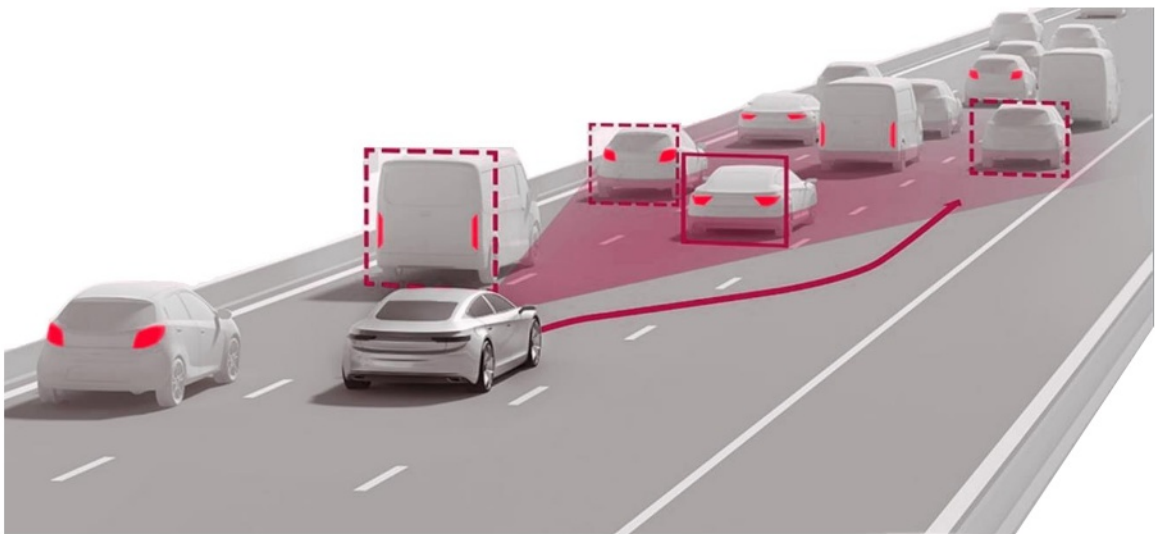


Masterarbeit

Entwicklung eines Gesamtfahrzeugmodells für hochdynamische,
autonome Ausweichmanöver

Master Thesis

Development of an overall vehicle model for highly dynamic,
autonomous evasive manoeuvres



Wissenschaftliche Arbeit zur Erlangung des Akademischen Grades Master of Science
in der Fakultät für Maschinenbau, Fahrzeugtechnik und Flugzeugtechnik
an der Hochschule für angewandte Wissenschaften München

Verfasser:

Can Polat

Matrikelnummer:

██████████

Studiengang:

Fahrzeugtechnik M. Sc.

Semester:

3. Semester (WiSe 2021/2022)

Betreuer:

Prof. Dr.-Ing. Bo Yuan
Hochschule München (Fakultät 03)

Abgabedatum:

01.12.2021

Erklärung zur Abschlussarbeit

Hiermit wird erklärt, dass die Arbeit mit obigem Thema selbständig verfasst und noch nicht anderweitig für Prüfungszwecke vorgelegt wurde. Weiterhin sind keine anderen als die angegebenen Quellen oder Hilfsmittel verwendet und wörtliche sowie sinngemäße Zitate als solche gekennzeichnet worden.

München, den _____

Can Polat

Danksagung

Für die Betreuung meiner Arbeit und vier wundervolle und lehrreiche Semester möchte ich mich bei meinem Ausbilder und Professor Dr.-Ing. Bo Yuan bedanken. Seit meinem ersten Hochschultag hat er mich in Technische Mechanik und Dynamik begleitet und in jeder Vorlesung motiviert, mein Wissen in Bereich Fahrdynamik, Fahrdynamikregelsysteme sowie Sensoren und Aktoren zu vertiefen. Als angehender Ingenieur bin ich geehrt und stolz, meinen Weg zum Erreichen meines Zieles Sie als mein Mentor zu haben.

Nach meiner ersten Anfrage beim Startup TrajectoriX am Strasczeg Center for Entrepreneurship (SCE) der Hochschule München erhielt ich die einmalige Chance, mit dieser vorliegenden Masterarbeit meine Kenntnisse und Fähigkeiten als Masterand im Bereich autonomes Fahren gezielt einzusetzen. Mein Dank gilt den Mitarbeitern des Startups, die mich unterstützten.

Kurzfassung

Eine große Unterstützung bieten Assistenzsysteme an, die eine Priorisierung der idealen Trajektorie ohne Lenk- und Bremsingriff des Fahrers zur Kollisionsvermeidung erstellen. Der Unterstützungsgrad kann mittels sensorischer Informationen definiert und das System so in der Bahnführungsebene ein hochdynamisches, autonomes Ausweichmanöver ausführen. Das Ausmaß einer unvermeidlichen Kollision lässt sich durch integrierte Fahrerassistenzsysteme deutlich reduzieren.

Vorliegende Masterarbeit stellt unterschiedliche Gesamtfahrzeugmodellvarianten zur Kollisionsvermeidung und Validierung sicherheitskonformer Trajektorienplanungen im 3D-Fahrsimulator vor. Die Vorgehensweise zur Ermittlung eines geeigneten Gesamtfahrzeugmodells ist in drei Teile gegliedert: Zu Beginn werden drei Hypothesen vorgestellt und die Randbedingungen für die Systemsteuerung definiert. Auf Basis des Ausgangszustands folgen mehrere Optimierungsansätze zur Etablierung eines Fahrerassistenzsystems während einer Normalfahrt. Im darauffolgenden Abschnitt sind realitätsnahe Referenzmodelle die Voraussetzung für eine Lenkungs-, Reifen- und Fahrzeugmodellauswahl. Die Simulationsabläufe in Model-in-the-Loop-Konzepten und Software-in-the-Loop-Simulationen basieren auf zuvor erstellten Datensätzen und sind mit realen Messwerten zu vergleichen. Folglich lassen sich Visualisierungen der definierten Manöver unter Einbehaltung der physikalischen Beschleunigungsgrenzen in der virtuellen Umgebung evaluieren. Der letzte Teil analysiert die Anwendbarkeit der generierten Fahrdynamik-Algorithmus-Datensätze unter Einbeziehung unterschiedlicher Fahrzeugparameter. Die Gegenüberstellung der physikalischen Querbeschleunigungsgrenzen beim einfachen Fahrspurwechsel zeigt, dass ein individueller Einsatz des *Polat-Python-API-Driving-Dynamics*-Programms für jeden Fahrzeugtyp realisierbar ist.

Die Bewertung der autonomen Kollisionsvermeidung im Fahrsimulator führt zu dem Ergebnis, dass die Anforderungen für hochdynamische Ausweichmanöver erfüllt sind. Die potenzielle Erstellung weiterer Datensätze für jeden Fahrzeugtyp bringt einen größeren Nutzen für simulationsbasierte Untersuchungen im Bereich der aktiven Fahrsicherheit.

Abstract

Assistance systems offer great support that create a prioritization of the ideal trajectory without the driver's steering and braking intervention to avoid collisions. The degree of support can be defined by means of sensory information and the system can thus perform a highly dynamic, autonomous evasive manoeuvre at the track guidance level. The extent of an unavoidable collision can be significantly reduced by integrated driver assistance systems.

This master's thesis presents different overall vehicle model variants for collision avoidance and validation of safety-compliant trajectory planning in a 3D driving simulator. The procedure for determining a suitable overall vehicle model is divided into three steps: At the beginning, three hypotheses are presented and the boundary conditions for the system control are defined. Based on the initial state, several optimization approaches follow to establish a driver assistance system during normal driving. In the following section, realistic reference models are the requirement for a selection of steering, tyre and vehicle models. The simulation sequences in model-in-the-loop concepts and software-in-the-loop simulations are based on previously created data sets and are supposed to be compared with real measured values. Therefore, visualizations of the defined manoeuvres can be evaluated while respecting the physical acceleration limits in the virtual environment. The last part analyses the applicability of the generated vehicle dynamics algorithm data sets with the inclusion of different vehicle parameters. The comparison of the physical lateral acceleration limits for a simple lane change show that an individual application of *Polat-Python-API-Driving-Dynamics* program is feasible for each vehicle type.

The evaluation of the autonomous collision avoidance in the driving simulator leads to the conclusion that it meets the requirements for highly dynamic, evasive manoeuvres. The potential creation of additional data sets for each type of vehicle will provide greater benefit for simulation-based investigations in the field of active driving safety.

Inhaltsverzeichnis

Erklärung zur Abschlussarbeit	ii
Danksagung	iii
Kurzfassung.....	iv
Abstract.....	v
Inhaltsverzeichnis	vi
Abkürzungsverzeichnis	ix
Formelzeichen	xi
1 Einleitung	1
1.1 Zielsetzung und Methodik	2
1.2 Grundaufgaben eines Kollisionsvermeidungssystems	3
1.3 Hintergrund der Themenwahl.....	5
2 Stand der Technik	7
2.1 Funktionsweise der aktiven Fahrsicherheitssysteme	7
2.1.1 Propriozeptive Fahrstabilisierungssysteme	7
2.1.2 Exterozeptive Fahrerassistenzsysteme	10
2.1.3 Entwicklungsvarianten autonomer Notausweichassistenzsysteme	13
2.2 Untersuchungswerkzeuge	17
2.2.1 Planung mit Software-in-the-Loop-Simulationen	17
2.2.2 Integrierte Entwicklungsumgebungen	18
2.2.3 Visualisierung im Fahrsimulator CARLA.....	21
3 Grundlagen der Fahrdynamik	23
3.1 Definition der Fahrzeugbewegung	23
3.1.1 Ortsfestes Koordinatensystem	23
3.1.2 Fahrzeugfestes Koordinatensystem	23
3.2 Trajektorien von Ausweichmanövern	25
3.2.1 Closed-Loop-Manöver: Doppelter Fahrspurwechsel	25
3.2.2 Open-Loop-Manöver: Einfacher Fahrspurwechsel	28
3.3 Gesamtfahrzeugmodell	30
3.3.1 Lenkungsmodell	31
3.3.2 Reifenmodell	32
3.3.3 Fahrzeugmodell	36

4	Modellbildung für ein Gesamtfahrzeug	45
4.1	Auswahl eines geeigneten Gesamtfahrzeugmodells	45
4.1.1	Gesamtfahrzeugmodell für die Fahrzeugführung.....	47
4.1.2	Modellierung hochdynamischer Ausweichmanöver	49
4.1.3	Fahrstreifenwechsel in drei Bereichen	51
4.2	Referenzdatengenerierung für den Modellansatz	52
4.2.1	Erstellung der Referenzfahrzeugparameter	53
4.2.2	Konzipieren einer Lookup-Table für Ausweichmanöver.....	57
4.3	Implementierung.....	60
4.3.1	Aufbau des SiL-Programms	60
4.3.2	Definition von Softwaremodulen und Funktionen	61
5	Integration der PPADD in CARLA	68
5.1	Vereinfachungen im Regelkreis	68
5.2	Qualitative Beurteilung der Simulationsergebnisse des Gesamtfahrzeugmodells	70
5.3	Verwendung von Python-APIs in CARLA	73
5.3.1	Numerische Integrationsverfahren	74
5.3.2	Datengenerierung für die Validierung	76
5.3.3	Zustandsautomat.....	77
6	Validierung des Modells.....	80
6.1	Grenzen der Programmierung	80
6.1.1	Funktionsauswertung des Post-Processing.....	81
6.1.2	Post-Processing-Ergebnisse unterschiedlicher Fahrzeugtypen	82
6.2	Gegenüberstellung der Gesamtfahrzeugmodelle.....	84
6.2.1	Analogie der Kollisionsvermeidungsmaßnahmen.....	85
6.2.2	Gesamtmodellvalidierung für die Simulation	88
6.3	Qualitative Bewertung der Gesamtfahrzeugmodelle.....	91
7	Zusammenfassung, Diskussion und Ausblick.....	93
7.1	Zusammenfassung mit Schlussfolgerung	93
7.2	Diskussion des PPADD-Reifenmodells	95
7.3	Ausblick.....	96
	Literaturverzeichnis	97
	Abbildungsverzeichnis.....	107
	Tabellenverzeichnis	109

A.1 Definition	110
A.1.1 Begriffe der Fahrzeugführung.....	110
A.1.2 Kritikalität von Verkehrssituationen.....	110
A.1.3 Weitere Definitionen.....	111
A.2 Ausweichassistent der Fraunhofer IIS	112
A.3 Referenzmodell CarMaker	113
A.3.1 Fahrdynamische Größen des Einzelsinus-Tests (ISO 7401)	113
A.3.2 VDA-Ausweichtest (ISO 3888-2).....	118
A.4 MATLAB/Simulink Programmierungen.....	119
A.4.1 Kollisionsvermeidungsmanöver	119
A.4.2 Ermittlung der Koeffizienten der Magic Formula nach Pacejka	119
A.4.3 MiL-Programmierung der Modelle A – C	119
A.5 Python-Programmierungen	123
A.5.1 PDD.....	123
A.5.2 PPADD	124
A.5.3 PPADD-Lookup-Table	126
A.5.4 PPP	128
A.5.5 Aufzeichnung vom Fahrsimulator CARLA.....	130
A.6 C++-Programmierungen	131

Abkürzungsverzeichnis

Abkürzung	Bezeichnung
ABS	Anti-Blockier-System
ACC	Adaptive Cruise Control
API	Application Programming Interface (dt. Programmierschnittstelle)
AWA	Ausweichassistent
C. G.	Center of Gravity (dt. Schwerpunkt)
Car2X	Car to x
CARLA	CAR Learning to Act (s. Dosovitskiy, 2017, S. 2)
DIN	Deutsche Institut für Normung
ESA	Evasive Steering Assist
ESM	Einspurmodell
ESP	Elektronisches Stabilitäts-Programm
FAS	Fahrerassistenzsystem
FCW	Forward Collision Warning
FTire	Flexible Ring Tire
HA	Hinterachse
HiL	Hardware-in-the-Loop
HMI	Human Machine Interface
IDE	Integrated Development Environment
IIS	Integrierte Schaltungen
ISO	Internationale Organisation für Normung
KI	Künstliche Intelligenz
MF	Magic Formula
MiL	Model-in-the-Loop
NA	Nickachse/ Nickzentrum

OEM	Original Equipment Manufacturer
P-Regler	Proportional-Regler
PDD	Polat-Driving-Dynamics (das Programm enthält sämtliche Python-Programme, um die fahrdynamischen Größen zu beschreiben, s. A.5.1 PDD)
PD-Regler	Proportional-Differential-Regler
PI-Regler	Proportional-Integral-Regler
PID-Regler	Proportional-Integral-Differential-Regler
PPADD	Polat-Python-API-Driving-Dynamics (das Programm enthält sämtliche Python-APIs, um auf den Fahrsimulator CARLA zuzugreifen und die 3D-Visualisierung plausibel zu gestalten, s. A.5.2 PPADD)
PPP	Polat-Post-Processing (das Programm dient zur Überprüfung der generierten Datensätze aus der Simulation PPADD und vergleicht die Werte mit den fahrdynamischen Größen aus PDD, s. A.5.4 PPP)
R. C.	Roll Center (dt. Wankzentrum)
RL	Reinforcement Learning
S	Schwerpunkt des Fahrzeugs
SAE	Society of Automotive Engineers
SCE	Strascheg Center for Entrepreneurship
SiL	Software-in-the-Loop
SP _A	Schwerpunkt des Fahrzeugs
TRW	Thompson Ramo Wooldridge
VA	Vorderachse
X ₁	Startzustand
X ₂	Zielzustand
ZF	Zahnradfabrik
ZSM	Zweispurmodell

Formelzeichen

Koordinatensystem

Die verwendeten Koordinatensysteme sind rechtshändige Systeme und folgen, sofern dort definiert, der DIN ISO 8855 (vgl. DIN ISO 8855, 2013, S. 6 – 9).

Zeichen	Bedeutung
x, y, z	ortsfestes Koordinatensystem in der Fahrebene zum Zeitpunkt t , x zeigt in die Fahrzeuglängsrichtung nach vorne, y nach links, z nach oben
x_0, y_0, z_0	ortsfestes Koordinatensystem in der Fahrebene zum Zeitpunkt t_0 , x_0 zeigt in die Fahrzeuglängsrichtung nach vorne, y_0 nach links, z_0 zeigt nach oben
x_A, y_A, z_A	fahrzeugfestes Koordinatensystem, x_A zeigt in die Fahrzeuglängsrichtung nach vorne, y_A nach links, z_A zeigt nach oben

Lateinische Buchstaben

Zeichen	Einheit	Bezeichnung
a	$\frac{\text{m}}{\text{s}^2}$	Beschleunigung
a_{Brems}	$\frac{\text{m}}{\text{s}^2}$	Bremsbeschleunigung
a_x	$\frac{\text{m}}{\text{s}^2}$	Längsbeschleunigung
a_y	$\frac{\text{m}}{\text{s}^2}$	Querb beschleunigung
$a_{y,\text{max}}$	$\frac{\text{m}}{\text{s}^2}$	maximale Querb beschleunigung
$a_{y,\text{h,kip}}$	$\frac{\text{m}}{\text{s}^2}$	Querb beschleunigung der Kippgrenze an der Hinterachse
$a_{y,\text{v,kip}}$	$\frac{\text{m}}{\text{s}^2}$	Querb beschleunigung der Kippgrenze an der Vorderachse
c_α	$\frac{\text{N}}{\text{rad}}$	Schräglaufsteifigkeit
$c_{\alpha,\text{h}}$	$\frac{\text{N}}{\text{rad}}$	Schräglaufsteifigkeit an der Hinterachse
$c_{\alpha,\text{v}}$	$\frac{\text{N}}{\text{rad}}$	Schräglaufsteifigkeit an der Vorderachse
$c_{\text{Feder,h}}$	$\frac{\text{N}}{\text{m}}$	Federsteifigkeit an der Hinterachse
$c_{\text{Feder,v}}$	$\frac{\text{N}}{\text{m}}$	Federsteifigkeit an der Vorderachse
$c_{\text{Stabi,h}}$	$\frac{\text{Nm}}{\text{rad}}$	Stabilisatorsteifigkeit an der Hinterachse
$c_{\text{Stabi,v}}$	$\frac{\text{Nm}}{\text{rad}}$	Stabilisatorsteifigkeit an der Vorderachse
c_κ	$\frac{\text{Nm}}{\text{rad}}$	Wanksteifigkeit
$c_{\kappa,\text{h}}$	$\frac{\text{Nm}}{\text{rad}}$	Wanksteifigkeit an der Hinterachse
$c_{\kappa,\text{v}}$	$\frac{\text{Nm}}{\text{rad}}$	Wanksteifigkeit an der Vorderachse
d_{k+1}	—	lokaler Diskretisierungsfehler
d_κ	$\frac{\text{Nms}}{\text{rad}}$	Dämpfsteifigkeit
$d_{\kappa,\text{h}}$	$\frac{\text{Nms}}{\text{rad}}$	Dämpfsteifigkeit an der Hinterachse

$d_{\kappa,v}$	$\frac{\text{Nms}}{\text{rad}}$	Dämpfsteifigkeit an der Vorderachse
f	Hz	Frequenz
g	$\frac{\text{m}}{\text{s}^2}$	Fallbeschleunigung
h	s	Schrittweite
h'	m	Wankhebelarm
h_{CG}	m	Schwerpunkthöhe
h_{h}	m	Wankpolhöhe an der Hinterachse
h_{NA}	m	Abstand des Fahrzeugschwerpunkts zum Nickschwerpunkt in der z_{A} -Achse
h_{v}	m	Wankpolhöhe an der Vorderachse
i_{S}	–	Lenkübersetzung
l	m	Radstand
l_{NA}	m	Abstand des Fahrzeugschwerpunkts zum Nickschwerpunkt in der y_{A} -Achse
l_{h}	m	Abstand von Fahrzeugschwerpunkt zur Hinterachse
l_{v}	m	Abstand von Fahrzeugschwerpunkt zur Vorderachse
m	kg	Gesamtfahrzeugmasse
m_{A}	kg	Aufbaumasse
n_{ges}	m	Gesamtnachlauf
r	m	Reifenradius
s	m	Strecke
s	m	Spurweite
$s_{\text{c,h}}$	m	Federspurweite an der Hinterachse
$s_{\text{c,v}}$	m	Federspurweite an der Vorderachse
$s_{\text{d,h}}$	m	Dämpfungsspurweite an der Hinterachse
$s_{\text{d,v}}$	m	Dämpfungsspurweite an der Vorderachse
s_{h}	m	Spurweite an der Hinterachse

s_v	m	Spurweite an der Vorderachse
t	s	Zeit
t_1	s	Zeit der ersten Lenkradwinkel-Zeit-Kombination aus dem <i>PPADD</i> -Lookup-Table
t_2	s	Zeit der zweiten Lenkradwinkel-Zeit-Kombination aus dem <i>PPADD</i> -Lookup-Table
$t_{\text{Ausweichmanöver}}$	s	Zeit für die Ausführung des Ausweichmanövers
v	$\frac{\text{m}}{\text{s}}$	Geschwindigkeit
v_1	$\frac{\text{m}}{\text{s}}$	Startgeschwindigkeit
v_2	$\frac{\text{m}}{\text{s}}$	Zielgeschwindigkeit
$v_{\text{Ego-Fzg}}$	$\frac{\text{m}}{\text{s}}$	Geschwindigkeit des Ego-Fahrzeugs
v_{Fzg}	$\frac{\text{m}}{\text{s}}$	Fahrzeuggeschwindigkeit
v_{Grenz}	$\frac{\text{m}}{\text{s}}$	Grenzgeschwindigkeit
v_h	$\frac{\text{m}}{\text{s}}$	Geschwindigkeit an der Hinterachse
v_{rel}	$\frac{\text{m}}{\text{s}}$	Relativgeschwindigkeit
v_R	$\frac{\text{m}}{\text{s}}$	Reifengeschwindigkeit
v_{R_x}	$\frac{\text{m}}{\text{s}}$	Reifenlängsgeschwindigkeit
v_{R_y}	$\frac{\text{m}}{\text{s}}$	Reifenquergeschwindigkeit
v_v	$\frac{\text{m}}{\text{s}}$	Geschwindigkeit an der Vorderachse
x_1	m	Startposition in der x_0 -Ebene
x_2	m	Zielposition in der x_0 -Ebene
x_{Brems}	m	notwendiger Abstand für ein Bremsmanöver
x_d	m	Abstand zum Hindernis
x_{Grenz}	m	Grenzabstand
x_{Lenk}	m	notwendiger Abstand für ein Lenkmanöver
Δx	m	longitudinaler Versatz (Grenzabstand zum Objekt)

y_1	m	Startposition in der y_0 -Ebene
y_2	m	Zielposition in der y_0 -Ebene
y_{Soll}	m	lateraler SOLL-Abstand
Δy	m	lateraler Versatz
Δy_{max}	m	maximaler lateraler Versatz
Δy_{min}	m	minimaler lateraler Versatz
Δy_{Spur}	m	lateraler Versatz einer Fahrspur
A_S	–	Lenkungsverstärkung
B	–	Steifigkeit
BCD	–	Steifigkeit bei Schräglaufwinkel $\alpha = 0 \text{ deg}$
C	–	Formfaktor
C_R	$\frac{\text{Nm}}{\text{rad}}$	Verdrehsteifigkeit Achsaufhängung
C_S	$\frac{\text{Nm}}{\text{rad}}$	Lenkungssteifigkeit
C_T	$\frac{\text{Nm}}{\text{rad}}$	Torsionsteifigkeit Drehstab
D	–	Dämpfungsgrad
D	N	Maximalwert
E	–	Krümmungs- bzw. Biegefaktor
F	N	Kraft
F_a	N	Beschleunigungswiderstand
F_{Luft}	N	Luftwiderstand
F_{Roll}	N	Rollwiderstand
F_{R_x}	N	Reifenlängs-/ Umfangskraft
F_{R_y}	N	Reifenquer-/ Seiten-/ Lateralkraft
F_{R_z}	N	Reifenvertikal-/ Hoch-/ Reifenaufstands-/ Normalkraft
F_S	N	Seitenkraft

F_{Steig}	N	Steigungswiderstand
$F_{\text{W,ges}}$	N	Gesamtwiderstand
F_y	N	Seitenkraft
$F_{y,h}$	N	Seitenkraft an der Hinterachse
$F_{y,h,i}$	N	kurveninnere Seitenkraft an der Hinterachse
$F_{y,h,o}$	N	kurvenäußere Seitenkraft an der Hinterachse
$F_{y_{\text{Test,FTire}}}$	N	Berechnung der Seitenkräfte mit „Vehicle Body 3DOF Dual Track“ aus MATLAB/Simulink
$F_{y_{\text{Test,MF}}}$	N	Berechnung der Seitenkräfte mit Magic Formula nach Pacejka
$F_{y,v}$	N	Seitenkraft an der Vorderachse
$F_{y,v,i}$	N	kurveninnere Seitenkraft an der Vorderachse
$F_{y,v,o}$	N	kurvenäußere Seitenkraft an der Vorderachse
F_z	N	Reifenaufstandskraft
$F_{z,h,i}$	N	kurveninnere Reifenaufstandskraft an der Hinterachse
$F_{z,h,o}$	N	kurvenäußere Reifenaufstandskraft an der Hinterachse
$F_{z,v,i}$	N	kurveninnere Reifenaufstandskraft an der Vorderachse
$F_{z,v,o}$	N	kurvenäußere Reifenaufstandskraft an der Vorderachse
$\Delta F_{y_{\text{Test}}}$	N	Differenz zwischen $F_{y_{\text{Test,MF}}}$ und $F_{y_{\text{Test,FTire}}}$
$\Delta F_{z,h}$	N	dynamische Radlast an der Hinterachse
$\Delta F_{z,v}$	N	dynamische Radlast an der Vorderachse
J_x	kg·m ²	Trägheitsmoment um die durch den Schwerpunkt gehende x_A -Achse (Wankträgheitsmoment)
J_y	kg·m ²	Trägheitsmoment um die durch den Schwerpunkt gehende y_A -Achse (Nickträgheitsmoment)
J_z	kg·m ²	Trägheitsmoment um die durch den Schwerpunkt gehende z_A -Achse (Gierträgheitsmoment)

M_H	Nm	Lenkradmoment
M_S	Nm	Lenkmoment
T_{Lenk}	Nm	Lenkmoment (vgl. Abbildung 8)

Griechische Buchstaben

Zeichen	Einheit	Bezeichnung
α	rad	Schräglaufwinkel
α_{Grenz}	rad	Grenzschräglaufwinkel für den linearen Bereich
α_{h}	rad	Schräglaufwinkel an der Hinterachse
α_{Test}	rad	Testschräglaufwinkel für den linearen Bereich
α_{v}	rad	Schräglaufwinkel an der Vorderachse
$\alpha_{\text{v,i}}$	rad	kurveninnerer Schräglaufwinkel an der Vorderachse
$\alpha_{\text{v,o}}$	rad	kurvenäußerer Schräglaufwinkel an der Vorderachse
β	rad	Schwimmwinkel
$\dot{\beta}$	$\frac{\text{rad}}{\text{s}}$	Schwimmgeschwindigkeit
$\delta_{\text{Ausweichmanöver}}$	rad	Radlenkwinkel für die Ausführung des Ausweichmanövers
δ_{h}	rad	Radlenkwinkel an der Hinterachse
δ_{H}	rad	Lenkradwinkel
$\delta_{\text{H,1}}$	rad	Lenkradwinkel der ersten Lenkradwinkel-Zeit-Kombination aus dem <i>PPADD</i> -Lookup-Table
$\delta_{\text{H,2}}$	rad	Lenkradwinkel der zweiten Lenkradwinkel-Zeit-Kombination aus dem <i>PPADD</i> -Lookup-Table
$\delta_{\text{H,IST}}$	rad	IST-Lenkradwinkel
$\delta_{\text{H,SOLL}}$	rad	SOLL-Lenkradwinkel
$\dot{\delta}_{\text{H}}$	$\frac{\text{rad}}{\text{s}}$	Lenkradwinkelgeschwindigkeit
$\dot{\delta}_{\text{H,Grenz}}$	$\frac{\text{rad}}{\text{s}}$	Grenzenlenkradwinkelgeschwindigkeit
$\dot{\delta}_{\text{H,max}}$	$\frac{\text{rad}}{\text{s}}$	maximale Lenkradwinkelgeschwindigkeit
δ_{v}	rad	Radlenkwinkel an der Vorderachse
$\delta_{\text{v,m}}$	rad	mittlerer Radlenkwinkel an der Vorderachse
$\delta_{\text{v,i}}$	rad	kurveninnerer Radlenkwinkel an der Vorderachse

$\delta_{v,o}$	rad	kurvenäußerer Radlenkwinkel an der Vorderachse
δ_{soll}	rad	SOLL-Lenkradwinkel
κ	rad	Wankwinkel
κ_A	rad	Wankwinkel
$\dot{\kappa}$	$\frac{\text{rad}}{\text{s}}$	Wankgeschwindigkeit
$\ddot{\kappa}$	$\frac{\text{rad}}{\text{s}^2}$	Wankbeschleunigung
λ	—	Schlupf
λ_{Opt}	—	optimaler Schlupf
μ	—	Kraftschlussbeiwert
μ_{Gleit}	—	Kraftschlussbeiwert im Gleitbereich
μ_{max}	—	maximaler Kraftschlussbeiwert
τ	s	Zeitkonstante
φ	rad	Nickwinkel
φ_A	rad	Nickwinkel
$\ddot{\varphi}$	rad	Nickbeschleunigung
ψ	rad	Gierwinkel
ψ_A	rad	Gierwinkel
$\dot{\psi}$	$\frac{\text{rad}}{\text{s}}$	Giergeschwindigkeit
$\ddot{\psi}$	$\frac{\text{rad}}{\text{s}^2}$	Gierbeschleunigung
ω	$\frac{\text{rad}}{\text{s}}$	Raddrehzahl
Θ	$\text{kg}\cdot\text{m}^2$	Trägheitsmoment um die durch den Schwerpunkt gehende z_A -Achse (Gierträgheitsmoment)

1 Einleitung

Der technische Fortschritt bei Fahrerassistenzsystemen (FAS) ermöglicht mit hochdynamischen Kollisionsvermeidungstrajektorien schnelle Eingriffe, damit Vollbremsungen oder Ausweichmanöver zur Kollisionsvermeidung einsetzbar sind. Diese hochdynamischen Ausweichmanöver können Szenarien, in denen andere Verkehrsteilnehmer aus einer Seitenstraße heraus die Vorfahrt missachten, deutlich entschärfen. Hierbei ist zwischen einem Überhol- und einem Ausweichvorgang grundsätzlich zu unterscheiden. Ein Überholvorgang eines Hindernisses kann mit situationsbedingter Geschwindigkeit ausgeführt werden, ohne den Gegenverkehr zu gefährden. Nach § 5 StVO ist dieses regel- bzw. wissensbasierte Fahrverhalten durchzuführen, sofern das Manöver planbar ist und ausreichender Seitenabstand zu den anderen Verkehrsteilnehmern eingehalten werden kann. Bei einem Ausweichmanöver hingegen bleibt keine Zeit für die Kollisionsvermeidungsplanung. Das fertigkeitbasierte Verhalten des Fahrers – abruptes Bremsen, ruckartiges Lenken – leitet sensorische Informationen als Signal in das FAS ein, welches anschließend eine ideale Kollisionsvermeidungstrajektorie berechnet. Vielen FAS wird aufgrund unterschiedlich komplexer Verkehrssituationen in absehbarer Zeit keine vollständige Verantwortungsübernahme genehmigt werden (vgl. DEKRA, 2016, S. 2 – 65).

Ausschließlich die leistungsfähigeren Fahrerassistenzsysteme und -funktionen gewährleisten eine in Echtzeit präzise Frühwarnung und greifen im Notfall selbst ein. Bis zu 45 Prozent weniger Unfälle würden auf Autobahnen entstehen, wäre jedes zweite Fahrzeug mit einem Abstandsregeltempomat samt Auffahrkollisionswarnung ausgestattet (vgl. DEKRA, 2016, S. 9). Die gleichzeitigen Meldungen verschiedener Assistenzsysteme in kurzer Abfolge stellen allerdings eine Informationsflut dar, deren fehlende Abstimmung zur Priorisierung den Fahrer belasten.

Fahrsimulatoren ermöglichen die Entwicklung autonomer Fahrmanöver mit Testverfahren, damit Autofahrer im alltäglichen Verkehr mit weniger Aufwand, Gefahr und Kosten abgesichert werden. Die meisten Simulatoren benötigen eine physische Eingabe vom Nutzer, um mit dem virtuellen Ego-Fahrzeug den erwünschten SOLL-Kurs zu fahren. Hierbei fehlen bei manchen Open-Source-lizenzierten Simulatoren aber Gesamtfahrzeugmodelle, welche die fahrdynamischen Fahrzeugeigenschaften in kritischen Fahrsituationen berücksichtigen. Die folgenden Untersuchungen dienen dazu, bei einem plötzlich auftauchenden Hindernis auf der Fahrbahn eine Kollisionsvermeidung mit einem autonomen, hochdynamischen Ausweichmanöver zu entwickeln.

1.1 Zielsetzung und Methodik

Ziel dieser Masterarbeit ist die Entwicklung eines Gesamtfahrzeugmodells für hochdynamische und autonome Ausweichmanöver in der Simulationssoftware CARLA. Für die Eingabe plausibler Parameter müssen Fahrzeugdaten und Modellierungsdatensätze mit Referenzwerten aus realen Versuchen entnommen werden. Die im lizenzierten Softwareprogramm MATLAB/Simulink hinterlegten Parametersätze zu Lenkungs- und Reifenmodellen lassen sich nach Optimierungsmaßnahmen für den erwünschten Programmcode ableiten. Der Fahrsimulator ist jedoch trotzdem nicht in der Lage, eine Programmchnittstelle (API) aufzubauen und eine Visualisierung darzustellen. Der Grund für die Programmierung in MATLAB/Simulink und anschließende Integration in Open-Source-Programmiersprachen wie Python und C++ ist die fehlende API-Kompatibilität. Abbildung 1 zeigt den Arbeitsplan, der die schrittweise Entwicklung qualifizierbarer und quantifizierbarer hochdynamischer Ausweichmanöver im Fahrsimulator CARLA visualisierbar macht.

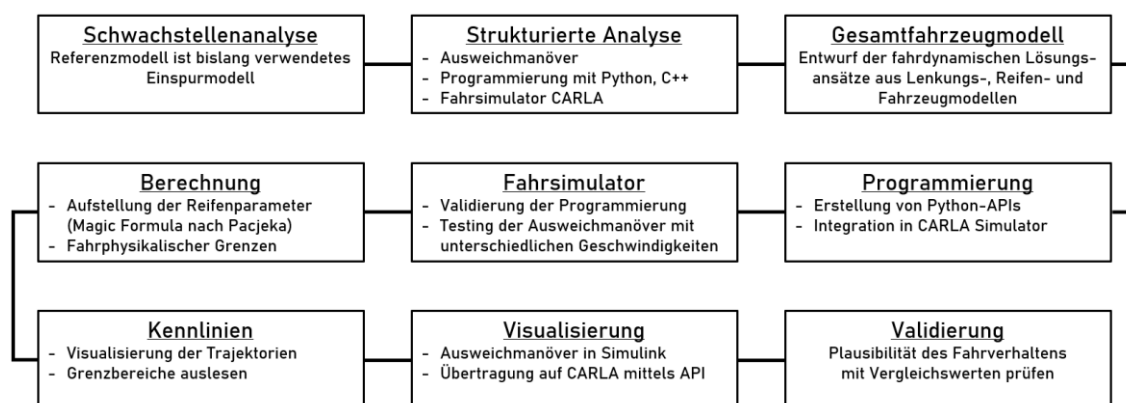


Abb. 1: Ablauf für die Programmierung eines geeigneten Gesamtfahrzeugmodells [Eigene Darstellung]

Im Fokus dieser Prozessabfolge steht die frühzeitige Erkennung von Schwachstellen der Gesamtfahrzeugmodelle beim Ausführen eines Ausweichmanövers mit CARLA im direkten Vergleich. So lassen sich beispielsweise unterschiedliche Varianten von Lenkungs-, Reifen- und Fahrzeugmodellen bewerten und miteinander vergleichen. Die Darstellung der Kollisionsvermeidung ist sowohl visuell als auch numerisch durch Referenzwerte verifizierbar, potenzielle Kollisionsgefahren sind wegen fehlender physikalischer Parameter schneller erkennbar und behebbar. Ein API-kompatibles Softwareprogramm kann somit innerhalb kurzer Zeit programmiert und getestet werden (vgl. Abbildung 1 und A.5.3 PPADD).

1.2 Grundaufgaben eines Kollisionsvermeidungssystems

Für die Komfort- und Sicherheitserhöhung werden FAS-Funktionen verschiedener Automatisierungsstufen implementiert, hierbei ist das Ziel die selbstständige Bewältigung aller Verkehrssituationen durch zukünftige FAS-Systeme. Wenn Antikollisionssysteme funktionssicher integriert sind, kann das System die Fahrzeugführung in kritischen Fahr-situationen beherrschen. Im Rahmen der Sensordatenerfassung werden beispielsweise Bedrohungen aufgrund unbeabsichtigten Verlassens der Fahrspur oder zu dichten Auf-fahrens der Verkehrsteilnehmer erkannt und für eine Situationsanalyse verwendet. Fol-glich überwachen Kollisionsvermeidungssysteme die Gefahren einer potenziellen Koll-i-sion aus und bewerten ununterbrochen die vor ihnen liegende Fahrbahn (vgl. Kallmeyer, 2019). In einem weiteren Schritt unterstützt die künstliche Intelligenz das autonome Fahr-zeug bei der Entscheidungsfindung und der Ausführung von priorisierten Trajektorien.

Ein Ausweichmanöver ist laut dem Urteil vom BGH VI ZR 263/09 notwendig, sofern es aus objektiver Sicht unverzichtbar erscheint (vgl. Bundesgerichtshof, 2010, S. 5f.). Die Unfallursache des gewählten Fahrverhaltens ist bei Haftungsfragen ausschlaggebend, die Reaktionsmöglichkeiten des Antikollisionssystems hingegen bedeutungslos. Aktive Si-cherheitssysteme können Unfälleintritte verhindern, indem sie in schwierigen Situationen oder bei unvorhersehbarem Lenk- oder Bremsbedarf den Fahrer unterstützen. Autonome Fahrzeuge unterliegen physikalischen Gesetzen und können abhängig von der Relativge-schwindigkeit und vom Abstand zum Objekt Kollisionen meiden. In der nachfolgenden Darstellung können die Kollisionsvermeidungsmaßnahmen in folgende Flächenbereiche (A – D) untergliedert werden:

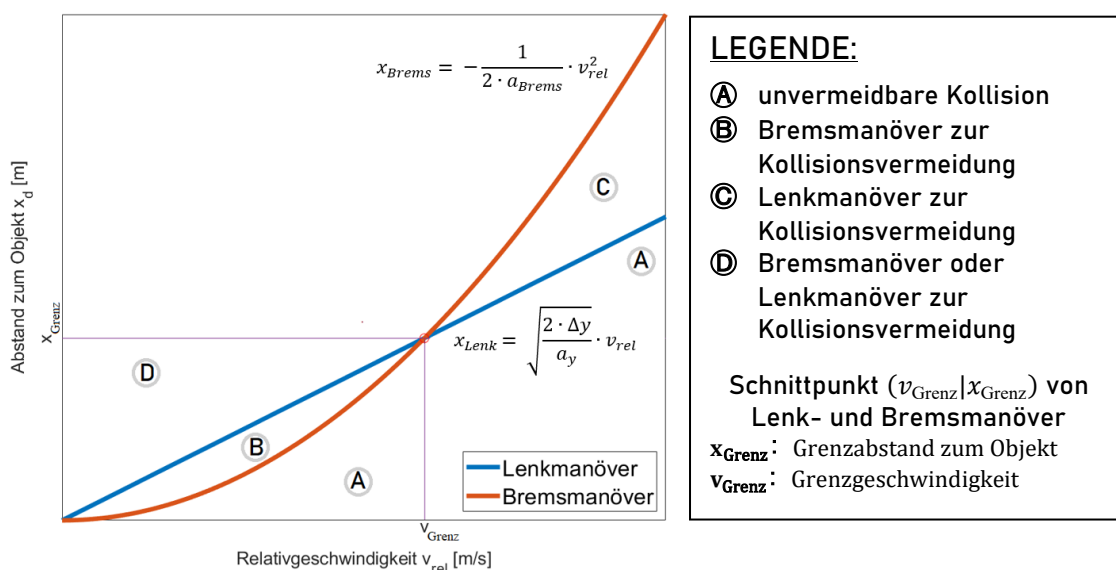


Abb. 2: Kollisionsvermeidungsmanöver [Eigene Darstellung] (vgl. A.4.1 Kollisionsvermeidungsmanöver, in Anlehnung an Winner, 2015, S. 920 und Eckert, 2011, S. 2)

Bereich A, unterhalb der Bremsmanöver-Kurve, legt nahe, dass in Anbetracht des technischen Fortschritts Eingriffe in die Längs- und Querdynamik von Fahrzeugen erforderlich sind. Im Diagramm ist außerdem zu erkennen, dass das Bremsen bei niedrigen Geschwindigkeiten (Bereich B) und das Lenken bei höheren Relativgeschwindigkeiten (Bereich C) effizienter sind. Mehr als 80 % aller PKW-Schäden können aber mit einem Bremsmanöver nicht verhindert werden und führen zu unvermeidbaren Kollisionen (Bereich B → Bereich A) (vgl. Winner, 2015, S. 62). Sobald der Fahrer die Grenzggeschwindigkeit v_{Grenz} überschreitet, ist für die Vermeidung der Kollision im Bereich C ein Lenkmanöver erforderlich. Das Kollisionsvermeidungssystem kann aber trotz des Einsatzes von Notbremsassistenten ihren Anwendungszweck nicht erfüllen, denn für den Bremsvorgang wird abhängig von der Relativgeschwindigkeit ein definiter Bremsweg vorausgesetzt. Verliefe der Graph des Bremsmanövers signifikant flacher, würde die Bremsung einem Unfall vorbeugen. Die Unfallfolgen können für die Verkehrsteilnehmer nur dann gering ausfallen, wenn die Auslösekriterien in Abhängigkeit vom Umfeld vorausschauend in Echtzeit prognostiziert werden.

1.3 Hintergrund der Themenwahl

Der Ausgangszustand führt derzeit automatische Kollisionsvermeidungs-Simulationen aus, bis keine Unfallgefahr mehr besteht. Grundlage hierfür sind mathematische Simulationsmodelle, wie etwa das lineare Einspurmodell, die in Fahr simulatoren zur Entwicklung von autonomen Fahrerassistenzsystemen verwendet werden. Vereinfacht programmierte Ausweichmanöver lassen die virtuellen 2D-Fahrzeuge, deren Exterieur einem Rechteck gleicht, in einer Kreisbahn fahren. In Abbildung 3 ist zu sehen, dass sich dieses Ego-Fahrzeug bei einer Kreisfahrt in Richtung Hindernis im roten Kollisionsgefahrbereich befindet. Der grüne Pfad nach rechts entspricht dabei keinem realitätsnahen verkehrssicheren Ausweichmanöver, da die geometrische Straßenbegrenzung im Verkehrsgeschehen vernachlässigt wird. Hierbei werden – bedingt durch die vereinfachte Simulationsumgebung – nachweislich Wetter- und Sichtverhältnisse sowie Fahrerreaktionen nicht berücksichtigt.

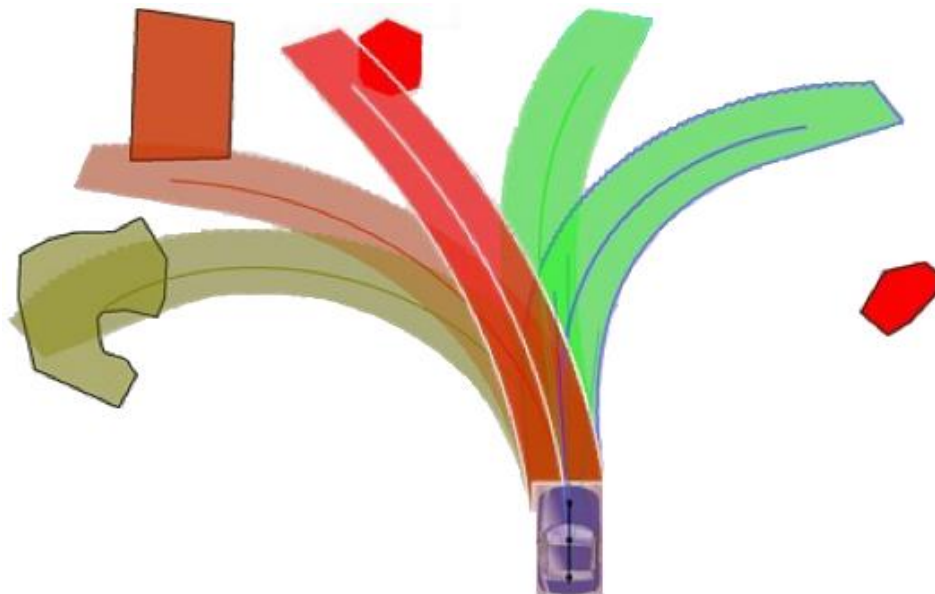


Abb. 3: Skizze zur Illustration möglicher Ausweichmanöver (Ausgangszustand) (Stand: 01.05.2021)
(vgl. Startup TrajectoriX, 2021)

Ein Fahrspurwechsel ist ebenfalls nicht illustriert, obwohl ein reales Fahrmanöver mit einem Aus- und Einschervorgang des Fahrers erfolgt. Die Optimierung des Notfallausweichmanövers lässt sich durch einen doppelten Fahrspurwechsel verbessern, der fahrphysikalische Grenzen berücksichtigt. Die erste Hypothese beschäftigt sich deshalb mit der Frage, ob eine lineare Einspurmodell-Simulation einen realen Fahrversuch ersetzen kann oder vielmehr ein unzureichendes Fahrverhalten aufweist. Folglich muss das Fahrzeugverhalten mit detaillierten Teilmodellen verbessert werden. Ziel ist, ein autonomes, hochdynamisches Ausweichmanöver im Simulationsmodell zu erreichen.

Im Rahmen der zweiten Hypothese ist zu klären, ob die Auswertungsergebnisse der simulierten Ausweichmanöver bei Eingabe von unterschiedlicher Fahrzeugdaten variieren und ob ein universeller Einsatz überhaupt möglich ist. Um dies herauszufinden, sind Daten über affektbehaftete Auswirkungsparameter beim Spurwechselfvorgang aus Versuchsfahrten oder numerischen Simulationen zu sammeln. Aus diesem Grund werden im Folgenden realitätsnahe, autonome und hochdynamische Ausweichmanöver in einem komplexen Gesamtfahrzeugmodell entwickelt. Die letzte Hypothese hinterfragt, ob sich eine schlechte Fahrbahnoberfläche negativ auf das Fahrverhalten während der Kollisionsvermeidung auswirkt.

Die Unfallvermeidung mithilfe eines Ausweichmanövers ist zuvor im Fahrsimulator zu konzeptionieren. Für die Validation werden physikalische Grenzwerte und Kriterien festgelegt und untersucht, was wiederum die systematische Abarbeitung der Meilensteine in der MiL- und SiL-Softwareentwicklung voraussetzt (vgl. Abbildung 1). Eine zielgerichtete Beurteilung des Standes der Technik und der verwendeten Gesamtfahrzeugmodelle dient als Stütze, um etwaigen Handlungsbedarf zu erkennen.

2 Stand der Technik

In diesem Kapitel werden die Grundlagen zur idealen Kollisionsvermeidung der Fahrsicherheitsysteme dargestellt, die als Ausgangspunkt für die nachfolgenden Untersuchungen dienen. Zu Beginn wird darauf eingegangen, welche propriozeptiven Fahrstabilisierungssysteme und exterozeptiven Assistenzsysteme in Bezug auf die Unfallvermeidung relevant sind (vgl. Bengler, 2014, S. 6 – 9). Überwiegend beschäftigt sich dieses Kapitel mit autonomen Notausweichassistenten, die bereits auf dem Markt existieren und einen deutlichen Unterschied zum KI-basierten Ausweichmanöver im Ausgangszustand aufweisen (vgl. Hochschule München, 2021). Ausgehend von den Grundlagen der FAS werden die Untersuchungswerkzeuge vorgestellt. Die verwendeten Programmierungs- und Simulationsumgebungen sowie deren Bedeutung bei der Realisierung hochdynamischer Ausweichmanöver des geeigneten Gesamtfahrzeugmodells stehen dabei im Vordergrund.

2.1 Funktionsweise der aktiven Fahrsicherheitsysteme

Mit zunehmender Verkehrsdichte sowie Weiterentwicklung leistungsfähigerer Kraftfahrzeuge und den damit einhergehenden kritischen Fahrsituationen im alltäglichen Straßenverkehr entsteht immer mehr Bedarf an aktiven Fahrsicherheitsystemen. Autofahrer müssen aufgrund der notwendigen Fahrzeugstabilität bei Vollbremsungen und ungeplanten Ausweichmanövern häufiger Bremspedal-, Lenkrad- und Gaspedalbetätigungen durchführen (vgl. Isermann, 2006, S. 169 – 172). Aktive Fahrsicherheitsysteme sind Regelsysteme im Fahrzeug, die der Vermeidung von Kollisionen dienen und kritische Fahrzustände verhindern. Die propriozeptiven Fahrstabilisierungssysteme genügen diesem Ausmaß an Verkehrsteilnehmern nicht mehr hinreichend, weshalb exterozeptive und autonome Fahrerassistenzsysteme zunehmend an Bedeutung gewinnen (vgl. Bengler, 2014, S. 6 – 9).

2.1.1 Propriozeptive Fahrstabilisierungssysteme

Unter *Propriozeption* ist die Wahrnehmung der Bewegung und Lage des Fahrzeugs durch messbare systeminterne Größen zu verstehen. Wie wichtig die Messwerte von Sensoren für den Fahrstabilisierungsbetrieb des Anti-Blockier-Systems und des Elektronischen Stabilitäts-Programms sind, wird nun erläutert.

Anti-Blockier-System (ABS)

Das Blockieren der Räder bei Bremsmanövern ist zu verhindern, damit das Fahrzeug lenk- und kontrollierbar bleibt. Abhängig von den Parametern Fahrzeuggeschwindigkeit v_{Fzg} und Raddrehzahl ω wirkt das Anti-Blockier-System gegen das Überschreiten des optimalen Schlupfes λ_{Opt} am Rad, wie in der Gleichung (Gl.) (1) dargestellt (vgl. Isermann, 2006, S. 37 und Reif, 2010, S. 37 – 39):

$$\lambda = \frac{v_{\text{Fzg}} - r \cdot \omega}{v_{\text{Fzg}}} \quad (1)$$

Der Reibwert der Reifen-Fahrbahn-Kontaktzone befindet sich im Nullpunkt und steigt bis zum Kraftschlussbeiwertmaximum $\mu_{\text{max}}(\lambda)$ an, Lage und Höhe des Reibwertmaximums werden durch die Fahrbahn, die Reifen, den Fahrzustand und die Umgebungsbedingungen bestimmt. Das Überschreiten des Kraftschlussmaximums würde, wie links in Abbildung 4 dargestellt, für die Reibpaarung von Reifen und Straßenoberfläche dem Übergang von der Haftreibung (stabiler Bereich) in die Gleitreibung (instabiler Bereich) entsprechen. Erst eine optimierte Bremskraft am jeweiligen Rad gewährleistet die gewünschte Fahrstabilität bei einer Vollbremsung mit Lenkmanövern oder unter μ -Split-Bedingungen (vgl. Abbildung 4 rechts). Im Eingriffsfall würde ein Rad, wie z. B. in der Darstellung mit $\mu = 0,1$, seine Bodenhaftung beim Bremsen verlieren oder zu Rutschen beginnen. In kurzen Intervallen muss daher eine ABS-Bremsung erfolgen. Auf diese Weise verhindert das ABS das Blockieren der Räder und somit das Über- und Untersteuern des Fahrzeugs (vgl. Isermann, 2006, S. 178).

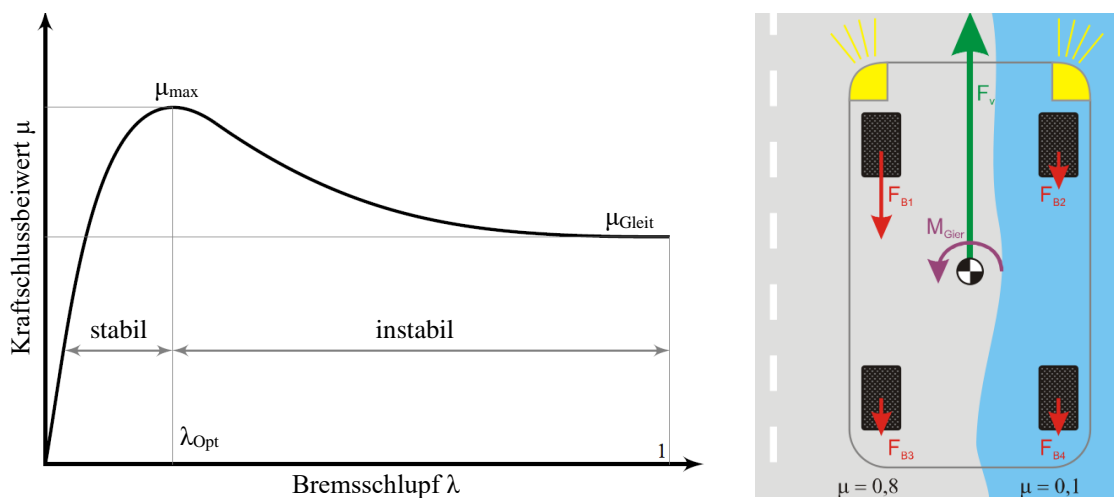


Abb. 4: Typischer Kraftschlussbeiwert-Schlupf-Verlauf eines Fahrzeugreifens [Eigene Darstellung] (in Anlehnung an Ammon, 1997, S. 107) (links) und Bremsengriffe bei μ -Split-Bedingungen (Wikipedia, 2021) (rechts)

Neben der beschriebenen Grundfunktion werden während des Bremsvorgangs weitere Fahrzeugstabilitätsmaßnahmen verwendet, die eine Spurführung und Lenkbarkeit des Fahrzeugs ermöglichen (vgl. Isermann, 2006, S. 139 – 167 und Winner, 2015, S. 727 – 730). Somit wird eine Verkürzung des Bremsweges erreicht, wenn das ABS aufgrund einer guten Regelstrategie den maximalen Kraftschluss nahezu vollständig benutzt. Die Funktion der radindividuellen Bremsschlupfregelung des ABS wird in anderen Fahrdynamikregelungen eingesetzt, da sie in konventionellen Antriebssystemen mit Zentralmotor einen Stelleingriff anbietet (vgl. Schönemann, 2013, S. 520 – 527).

Elektronisches Stabilitäts-Programm (ESP)

Während das ABS grundsätzlich die Längsdynamik des Fahrzeugs beeinflusst, realisiert das ESP eine Regelung der Querdynamik zur Verbesserung des Spurhaltevermögens und der Lenkbarkeit in allen Betriebszuständen. Die Abbremsung einzelner Räder zur Erzeugung eines Giermomentes nutzt das ESP. Die entstehende Giermomentänderung nimmt bei großen Lenkwinkeländerungen ab. Ein zu starker Lenkeingriff und die damit einhergehende Instabilität lassen in diesem Fall das Fahrzeug zum Über- und Untersteuern neigen (vgl. Breuer, 2015, S. 196f. und Isermann, 2006, S. 169 – 197).

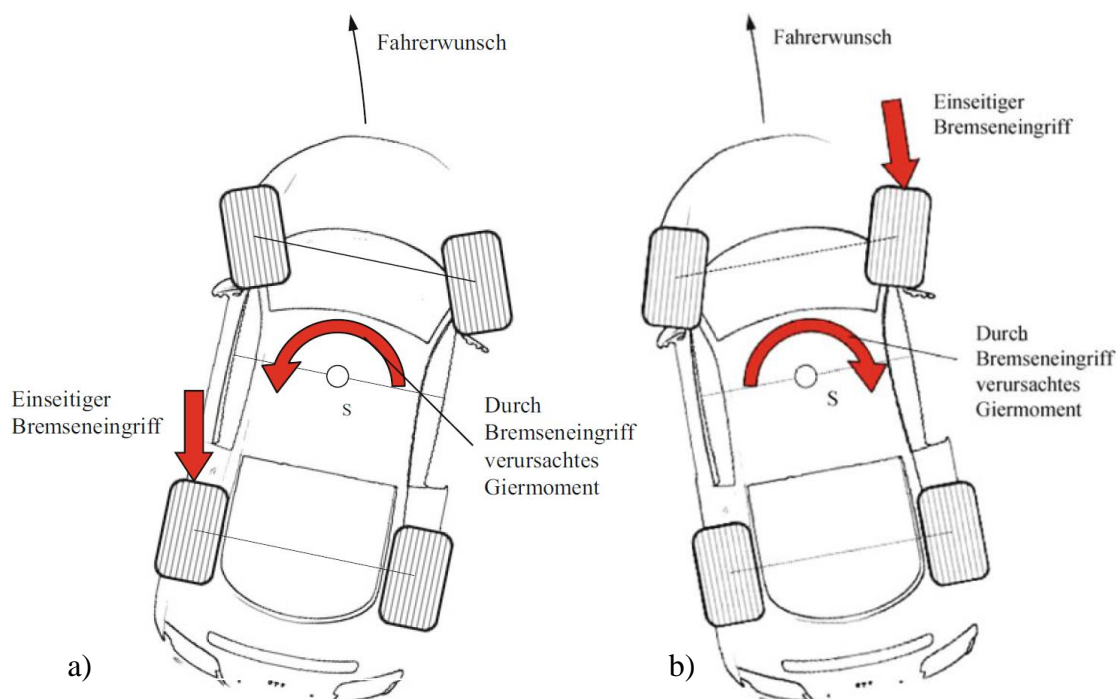


Abb. 5: Bremseneingriff bei Fahrdynamikregelung bei a) untersteuerndem und b) übersteuerndem Fahrzeug (s. Breuer, 2015, S. 196f.)

Um untersteuerndes Fahrverhalten zu reduzieren, wird nach Abbildung 5 a) an das kurveninnere Rad an der Hinterachse gebremst. Dies verhindert ein Verlassen der gewünschten Fahrtroute in Richtung des fahrbahnäußeren Rades. Bei übersteuerndem Fahrverhalten giert das Fahrzeug um die Hochachse und droht in einen unkontrollierbaren Schleuderzustand zu geraten. In Abbildung 5 b) wird deswegen durch gezielten Bremsingriff am kurvenäußeren Vorderrad ein Giermoment um die Hochachse erzeugt, dass der übersteuernden Gierbewegung entgegenwirkt. Ein Ausbrechen des Fahrzeughecks bzw. ein eintretender Schleuderzustand wird somit vermieden.

Das Regelungskonzept des ESP basiert auf einer Antriebs- und Bremsschlupfregelung unter Verwendung der bereits vorgestellten Funktionen des ABS. Typische Fahrsituationen für den notwendigen Eingriff des ESP sind der Doppelspurwechsel oder der Spurwechsel bei Vollbremsung. Diese Fahrmanöver resultieren zumeist aus zu später Erkennung oder dem plötzlichen Auftreten von Hindernissen. Derartige ESP-Eingriffe erfassen Giergeschwindigkeit, Schwimm- und Lenkwinkel sowie weitere fahrdynamische Kenngrößen und stellen sie für einen erweiterten Beobachter bereit. Das stabile Fahrverhalten ist mit einem Sollwertrechner des Fahrzeugmodells zu optimieren, der die fahrdynamischen Parameter für Gierrate und Schwimmwinkel bestimmt (vgl. Isermann, 2006, S. 169 – 212 und Reif, 2010, S. 58 – 73).

2.1.2 Exterozeptive Fahrerassistenzsysteme

Zwar unterstützen die Fahrstabilisierungssysteme ABS und ESP den Fahrer bei einem Notmanöver, jedoch werden die Umfeldinformationen und Fahrerfehler dabei nicht berücksichtigt. Sensorik zur Wahrnehmung des Fahrzeugumfeldes sind z. B. Kameras, Radar, Lidar oder Ultraschallsensoren, die die Position und Bewegung umliegender Objekte vermessen. Umfeldobjekte und Umweltbedingungen zur Außenwahrnehmung können mit *exterozeptiven Assistenzsystemen* erfasst werden (vgl. Bengler, 2014, S. 6 – 9). Denn ein hoher Automatisierungsgrad der Fahrzeuge erfordert zunehmend sichere Funktionalität für Notbrems- und Notausweichassistenten (vgl. Sippl, 2018, S. 2 – 26).

Notbremsassistent

Ein Notbremsassistentensystem wird zur selbstständigen Erkennung potenzieller Frontalzusammenstöße und für das Abbremsen des Fahrzeugs verwendet, um einen Zusammenstoß zu verhindern oder abzumildern (vgl. Verordnung [EU] der Kommission, 2012, S. 1 – 24). Diese standardmäßig in Fahrzeugen integrierte Unfallvermeidungsmaßnahme reduziert ohne aktiven Eingriff des Fahrers autonom die Fahrzeuggeschwindigkeit auf ein Minimum.

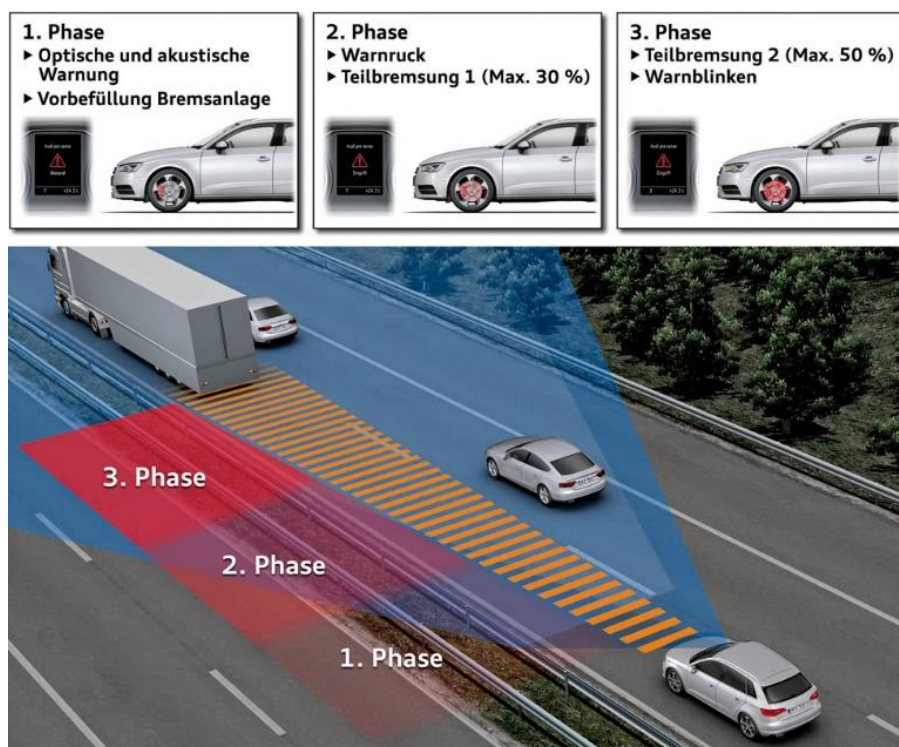


Abb. 6: Bremsassistent Audi pre sense plus (vgl. Audi AG, 2011)

Eine Fahrgeschwindigkeitsreduktion wird bspw. von Audi mit Notbremsassistenten und Abstandsmesstechniken, bestehend aus zwei Front-Radarsensoren und einer Videokamera, realisiert. Bei nahender Kollision ertönt, wie in Abbildung 6 visualisiert, in der *ersten Phase* im Fahrzeuginnenraum, ein akustisches Warnsignal. In der *zweiten Phase* ist mit einem kurzen Warnruck, der die Aufmerksamkeit des Fahrers auf das Verkehrsgeschehen lenken soll, eine 30 %-Teilbremsung zu erwarten. Löst der Fahrer daraufhin selbst einen Bremsvorgang aus, so wird dieser in der *dritten Phase* vom System in Form einer 50 %-Teilbremsung unterstützt. Falls er nicht reagiert, aktiviert das FAS die Warnblinkanlage des Fahrzeugs, schließt die Fenster und strafft den Gurt. Als letzte Schutzmaßnahme leitet das System im Falle eines unvermeidbaren Unfalls in der *vierten Phase* selbstständig eine Vollbremsung ein. Dies kann zu einer Reduktion der Aufprallgeschwindigkeit um bis zu $40 \frac{\text{km}}{\text{h}}$ führen (vgl. Audi AG, 2011).

Notausweichassistent

Potenzielle Kollisionsgefahren müssen von exterozeptiven Fahrerassistenzsystemen frühzeitig erkannt und der Fahrer durch Brems- und Lenkeingriffe aktiv unterstützt werden (vgl. Bengler, 2014, S. 6 – 19). Unter dem Begriff „Notausweichassistent“ wird im Rahmen dieser Arbeit ein Notfall-Fahrerassistenzsystem verstanden, das dem Fahrer in unfallgeneigten Fahrsituationen zur erfolgreichen Manöverdurchführung verhilft. Die Aufgabe dieses Assistenten ist es, die Gefahr einer Kollision innerhalb kurzer Zeit zu erkennen und zu vermeiden. Hierzu muss das autonome System eine Reaktionsverbesserung durch frühzeitige Entscheidungsfähigkeit im Vergleich zum Reaktionsvermögen des Fahrers ausweisen. Das komfortable Umfahren eines Hindernisses mittels eines Überholmanövers ist nicht Untersuchungsgegenstand.

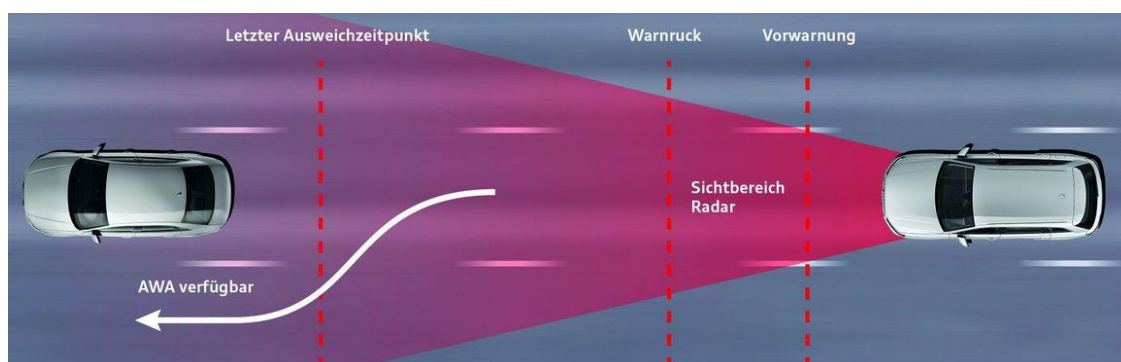


Abb. 7: Ausweichassistent Audi pre sense front (vgl. Audi AG, 2016)

In Abbildung 7 sind die Daten von Vermessungstechniken der Audi pre sense front ersichtlich, die durch Radarsensor und Frontkamera einen potenziellen Auffahrunfall erkennen. Eine drohende Kollision und entsprechende Bremsung wird innerhalb der definierten Systemgrenze analysiert und hängt von der Situation und Geschwindigkeit des Fahrzeugs ab. Die Grenze hat als Eingangsgrößen den Abstand, die Breite und den Versatz des Vorderfahrzeugs. Das System erkennt Verkehrsteilnehmer zwischen 30 bis $150 \frac{\text{km}}{\text{h}}$ und warnt den Fahrer in einem mehrstufigen Konzept optisch, akustisch und haptisch. Abhängig von der Geschwindigkeit ist eine Bremskraftunterstützung zur Geschwindigkeitsreduktion oder eine Vollverzögerung zur Kollisionsvermeidung realisierbar. Die Maßnahmen, die in der Abbildung zu sehen sind, dienen dem Insassenschutz. So werden z. B. die Aktivierung der Warnblinkanlage, die reversible Straffung der Sicherheitsgurte, die Optimierung der Sitzposition sowie das Schließen der Fenster eingeleitet. Der Notausweichassistent kann den Fahrer mittels einer Überlagerungslenkung in Form eines zusätzlichen Lenkmoments beim aktiven Lenkvorgang unterstützen (vgl. Audi AG, 2017).

2.1.3 Entwicklungsvarianten autonomer Notausweichassistenzsysteme

Selbstfahrende Fahrzeuge benötigen Assistenzsysteme bei fertigkeitbasierten Entscheidungen bezüglich der Geschwindigkeit, der Abstandshaltung und des Bremsvorgangs. Damit steigt die Verantwortung der FAS mit zunehmender Entlastung der passiven Fahrer. In Deutschland ist die Haftungsfrage im Falle eines Unfalls noch ungeklärt, weshalb im Bereich der Kollisionsvermeidung mithilfe von Ausweichmanövern Entwicklungs- und Forschungsarbeiten existieren (vgl. Bundesgerichtshof, 2010, S. 5f.).

Regelungstechnik: Fahrdynamik- und Positionsregelung

Von der Systemidentifikation über die Reglersynthese können Regler der ZF TRW Automotive Inc. dazu beitragen, die Auswertung des Regelungskonzepts in Echtzeit umzusetzen (vgl. Günnel, 2014).

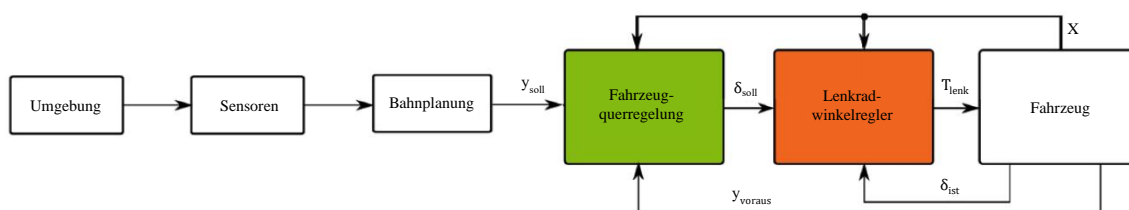


Abb. 8: Modellprädiktives Regelungskonzept: Fahrzeugquer- und Lenkradwinkelregelung [Veränderte Darstellung] (vgl. Seewald, 2015, S. 29)

Die Umgebung wird mittels der Datenfusion aus vorausschauenden Radar- und Kamerasensoren zum Einleiten eines Notausweichvorgangs erfasst. Wie in Abbildung 8 skizziert, berechnet sich eine Bahnplanung hier aus den Randbedingungen der Verkehrssituation unter Verwendung eines Polynoms 5. Grades und gibt die Regelgröße der seitlichen Fahrzeugsteuerung y_{soll} vor (vgl. Hansen, 2016, S. 67, Keller, 2014, S. 3008, Keller [1], 2015, S. 369 und Wissing, 2016, S. 11). Der SOLL-Wert für die Fahrzeugquerregelung führt das Fahrzeug entlang der vorgegebenen Bahn und vermeidet Kollisionen. Als Ausgangsgröße wird im ersten Regler der SOLL-Lenkradwinkel δ_{soll} ausgegeben und dem Lenkradwinkelregler zur Verfügung gestellt. Der zweite Regler übergibt das Lenkmoment T_{lenk} als Outputgröße an das Fahrzeug (vgl. Keller, 2014, S. 3008 – 3013, Seewald, 2015, S. 27 – 30, Keller [1], 2015, S. 369 – 374, Keller [2], 2015, S. 176 – 196, Wissing, 2016, S. 9 – 14 und Götte, 2016, S. 1908 – 1913). Bei diesem Regelkreis ist eine analytische Durchführung des Manövers mit einem Punktmassenmodell realisierbar, um die Komplexität der Berechnung zu reduzieren (vgl. Keller, 2014, S. 3009, Keller [1], 2015, S. 369, Wissing, 2016, S. 10 und Hansen, 2016, S. 67).

Neben dem bereits erwähnten Regelungskonzept der ZF TRW Automotive Inc. existieren noch weitere Regelungsverfahren, die in der folgenden Tabelle aufgelistet sind:

Tab. 1: Überblick über unterschiedliche Regelungsverfahren für Ausweichmanöver

Regelungsverfahren	Funktionsweise
P-Regler PD-Regler PID-Regler	<ul style="list-style-type: none"> ▪ Parametrisierung nach bestimmten Einstellregeln oder Versuchsweisen ▪ Kaskaden- bzw. Parallelregelkreisstruktur für Bahn- und Trajektorienfolgeregelung (Krümmungsvorsteuerung) <p>(vgl. Schorn [1], 2006, S. 378 – 383, Bevan, 2008, S. 2 – 132, Eskandarian, 2008, S. 1049 – 1054, Talvala, 2011, S. 137 – 148, Ackermann, 2014, S. 10694 – 10699, Iwano, 2014, S. 6338 – 6343, Pfeffer, 2015, S. 199 – 213, Schramm, 2018, S. 293 – 296, Isermann, 2006, S. 291 – 301 und Kramer, 2008, S. 107 – 109)</p>
Zustandsregler	<ul style="list-style-type: none"> ▪ modellbasierten Entwürfe ▪ Verknüpfung mit oder ohne einen PI-Regler und einem optionalen Beobachter <p>(vgl. Smith, 1995, S. 3586 – 3591, Schorn [2], 2006, S. 5837 – 5842, König, 2007, S. 314 – 321, König, 2009, S. 15 – 152, Menhour, 2014, S. 135 – 146 und Wang, 2011, S. 46 – 62)</p>
Spezielle Regler	<ul style="list-style-type: none"> ▪ nichtlineare Zustandsregler in Kombination mit einem PD-Regler <p>(vgl. Kehl [1], 2007, S. 43 – 80 und Kehl, 2007, S. 306 – 313)</p>
Fuzzy Regler	<ul style="list-style-type: none"> ▪ geschwindigkeitsadaptive Regelung (gesteuerte Adaption) <p>(vgl. Fernández-Llorca, 2011, S. 390 – 399)</p>
Ein- und Ausgangs-linearisierung	<ul style="list-style-type: none"> ▪ Linearisierung des Ein- und Ausgangsverhaltens der Strecke durch Berechnung von (nichtlinearen) Rückführungen <p>(vgl. Söhnitz, 2001, S. 116, Mayr, 2001, S. 31 – 154 und Werling, 2008, S. 585 – 591)</p>
Modellprädiktive Regelung	<ul style="list-style-type: none"> ▪ Verwendung modellprädiktiver Planungs- und Regelungsverfahren für die Bahn- und Trajektorienfolgeregelung <p>(vgl. Seewald, 2015, S. 26 – 30, Falcone, 2007, S. 566–580, Choi, 2012, S. 393 – 398, Gao, 2012, S. 1 – 6, und Katriniok, 2013, S. 974 – 979)</p>

Entscheidungsbaum mit Domain-Randomisierung

Das autonome Notausweichmanöver des Fraunhofer IIS verwendet zum Trainieren neuronaler Netze Reinforcement Learning (RL) und Domain-Randomisierung. Reinforcement Learning berücksichtigt eine aktionsbezogene Belohnung, die Entscheidungsfindung abhängig von der Umgebung wählt. RL-Algorithmen untersuchen die auszuführende Aktion und entwickeln eine Strategie, durch die das autonome Fahrzeug die Ausweichmanöver selbstständig durchführen kann. Die Domänen-Randomisierung dient zur systematischen Datengenerierung, was die Verallgemeinerung von RL-Algorithmen in neuen Umgebungen verbessert (vgl. Kontes, 2020, S. 1 – 8).

Das agentenorientierte Programm erzeugt – vergleichbar mit objektorientierten Programmen – ein eigenständiges Programmierparadigma und eine Entwurfsmethodik. Die Sensoren des Fahrzeugs erfassen den IST-Zustand der Umgebung und registrieren, welche Auswirkungen durch die Aktion, z. B. die Betätigung der Bremse und des Lenkrads bei Erreichen der Systemgrenze, entstehen. In Abbildung 9 ist ersichtlich, dass sich der Agent autonom weiterentwickelt, mit physikalischen Domänen operiert sowie interagiert und sich anschließend adaptiert und reproduziert. Dadurch kann er den Nutzen eines Ausweichvorgangs registrieren und weiterleiten, woraufhin der Aktuator das Manöver ausführt. Daraus lässt sich erschließen, dass ein agentenorientiertes Programmieren mit bestimmten Fähigkeiten und Zielen bestrebt ist, die Anforderung des Nutzers zu erfüllen und auf Veränderungen im Programm zu reagieren (vgl. Kontes, 2020, S. 1 – 8).

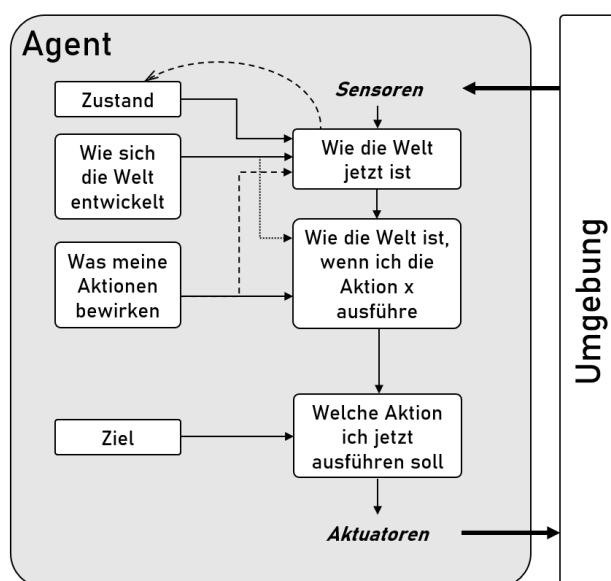


Abb. 9: Interaktionsschleife zwischen Agent und Umgebung
[Eigene Darstellung] (vgl. Russel, 2012, S. 80)

Ein Ausweichmanöver nach links zeigt das Entscheidungsbaumdiagramm des Fraunhofer IIS (vgl. Abbildung 10). Eingangswert, dessen Entscheidungsmöglichkeiten „True“ (A) oder „False“ (Subtree) ergeben, ist zu Beginn der Abstand $x \leq 60$ m mit Relativgeschwindigkeit. Der Subtree stellt hierbei einen alternativen Entscheidungsbaum dar, der eine andere Einsatzmethode beinhaltet. Sofern der relative Geschwindigkeitsbereich $v_{\text{rel}} \leq 4,2 \frac{\text{km}}{\text{h}}$ dem Wert von B und dem Abstandsbereich $x \leq 43$ m der C-Entscheidung entspricht, leitet das System bei einer Distanz $x \leq -0,8$ m die E-Entscheidung ein. Dies gleicht einem Ausweichmanöver (*LEFT*) auf die linke Fahrbahnspur, das den Lenkvorang ausführt, bis keine Kollisionsgefahr mehr besteht. Als weitere Optionen stehen *FASTER* (Geschwindigkeitsaufbau) und *BRAKE* (Geschwindigkeitsabbau) zur Verfügung (vgl. Bastani, 2018, S. 1 – 13 und Kontes, 2020, S. 1 – 8). Durch das Trainieren des Agenten kann somit das System auf Basis vorhandener Parameter, wie z. B. der Abstand zu einem Hindernis oder die Relativgeschwindigkeit, selbstständig Entscheidungen treffen.

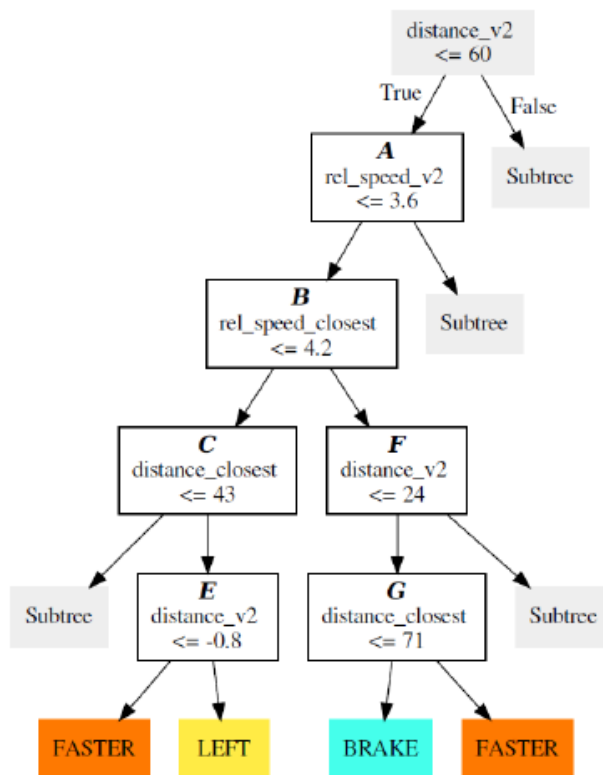


Abb. 10: Generierung eines Entscheidungsbaumes durch neuronale Netze (s. A.2 Ausweichassistent der Fraunhofer IIS und vgl. Bastani, 2018, S. 2)

2.2 Untersuchungswerkzeuge

Zum Aufbau und zur objektiven Validierung von Simulationsmodellen sind begleitende Referenzwerte durch reale Messungen eine zwingende Voraussetzung. Im Rahmen dieser Arbeit wird ein Überblick über die erforderlichen Modelle, die verwendete Software, die Entwicklungsumgebung sowie die Schnittstellen gegeben. Anschließend wird durch den Vergleich der erarbeiteten Referenzwerte und Simulationsergebnisse aus ausgewählten Validierungsmanövern die Plausibilität des Gesamtfahrzeugmodells überprüft.

2.2.1 Planung mit Software-in-the-Loop-Simulationen

Aufgrund zunehmenden Einsatzes von fahrdynamischen Fahrerassistenzsystemen wächst die Komplexität der Softwarefunktionalitäten (vgl. Schöttle, 2011, S. 9 – 15). Für die Steigerung der Herausforderungen bei der Gesamtbeherrschbarkeit ist die Planung der Vorgehensweise vor der Tätigkeitsausführung entscheidend. Vor dem Entwicklungsprozess von Softwarefunktionalitäten zum Einleiten eines autonomen Ausweichmanövers wird ein V-Modell (vgl. Abbildung 11) erstellt. Die grün markierten Felder stellen die behandelten Prozesse im Rahmen dieser Masterarbeit dar. Nach diesem Modell können die anfangs festgelegten Anforderungen vom Startup definiert und in die System- und Funktionsspezifikation unterteilt werden. In der Vorentwicklungsphase von Fahrzeugfunktionen wird im Model-in-the-Loop (MiL) die Umwelt sowie das Fahrzeug mit spezifischen Parametern als Regelstrecke im Umgebungsmodell modelliert. Mit der grafischen Programmiersprache MATLAB/Simulink lassen sich an entsprechenden Schnittstellen Blockschaltbilder mit Umgebungsmodellen verknüpfen.

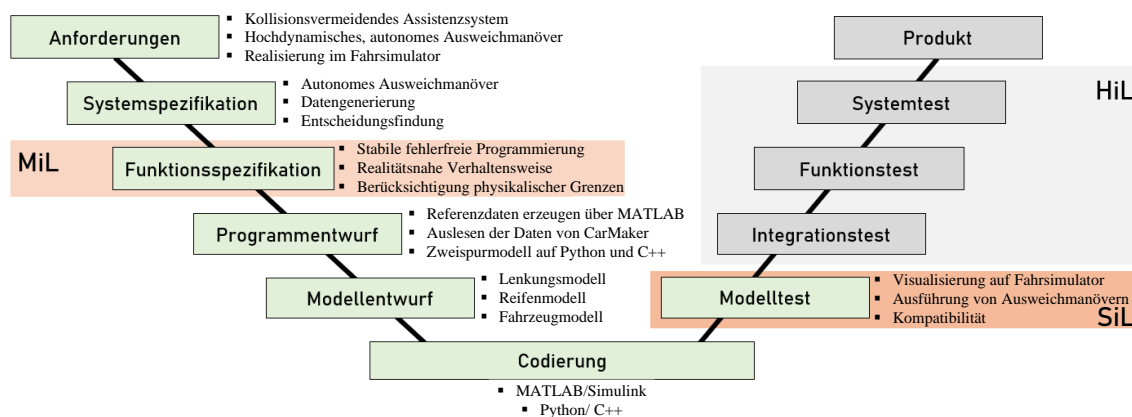


Abb. 11: V-Modell [Veränderte Darstellung] (vgl. Heißing, 2013, S. 569 und Winner, 2015, S. 130)

Für die Entwicklung und zum Testen der Systeme werden realitätsnahe und ausführbare Seriensoftwarecodes für das Ausweichmanöver in einer Simulationsumgebung ausgewählt. Diese Softwaretests sollen im Rahmen einer Software-in-the-Loop (SiL) mit Integration in die zu testenden technischen Systeme umfassender, wiederholbar, automatisierbar, rückverfolgbar sowie kostengünstiger durchführbar sein. Mit den Programmiersprachen Python und C++ ist die in der MiL definierte Vorgehensweise adaptierbar und kann durch entsprechende Schnittstellen am Simulationsprogramm CARLA verknüpft werden. So lässt sich bei der hochdynamischen Kollisionsvermeidung eine Evaluation unterschiedlicher Testszenarien und in rechenintensiven Teilmodellen für komplexere Untersuchungen anwenden (vgl. Brand, 2012, S. 649 – 660). „Da weder bei der MiL noch bei der SiL-Simulation reale Steuergeräte verwendet werden, besteht [...] nicht die Notwendigkeit, dass die verwendeten Simulationsmodelle echtzeitfähig sein müssen.“ (s. Isermann, 2006, S. 105). Dadurch ist es möglich, mit den vorhandenen Sensordaten Tests im Gesamtfahrzeugmodell mit verschiedenen Auslegungsvarianten ohne Hardware an einem Computer mit entsprechender Entwicklungssoftware durchzuführen. Je nach Ergebnis müssen Anpassungen an der Modellierung vorgenommen werden. Das Ziel der Untersuchung bei SiL-Simulationen ist die Funktionsüberprüfung und -optimierung der entwickelten Algorithmen. Nach erfolgreicher Programmierauslegung und -überprüfung wird das Assistenzsystem auf der Zielplattform implementiert und steht im nächsten Schritt als reales Steuergerät für HiL-Simulationen zur Verfügung.

2.2.2 Integrierte Entwicklungsumgebungen

Simulationen zielen darauf ab, Fehler zu identifizieren und zu vermeiden, indem Modelle in frühen Phasen getestet werden, noch bevor das System implementiert und bereitgestellt wird. Die Gesamtfahrzeugmodelle durchlaufen in diesen Prozessen mehrere Simulations- und Verfeinerungsrunden (vgl. Abbildung 11). Der Ausarbeitung hochdynamischer Ausweichmanöver stehen lizenzpflichtige Entwicklungsumgebungen (IDE) wie MATLAB/Simulink (vgl. MathWorks [1], 2021) sowie Open-Source-lizenzierten Entwicklungsumgebungen wie Geany (vgl. Geany, 2021) und Qt Creator (vgl. Qt Creator, 2021) zur Verfügung. Die Wahl fällt auf die zuvor genannten IDE, da diese vom Startup und von der Hochschule München bereitgestellt werden. Sie werden im Folgenden näher beleuchtet.

Lizenzpflichtige Entwicklungsumgebung: MATLAB/Simulink

Insbesondere Simulink bietet neben MATLAB zum Modellieren von Gesamtfahrzeugmodellen eine grafische Programmieroberfläche als anpassbare Blöcke vom Simulink Library Browser (Vehicle Dynamics Blockset) (vgl. MathWorks [1], 2021). Diese Blöcke ermöglichen es, individuelle sowie numerische und kombinatorische Operationen auszuführen oder konstante Werte darzustellen. Darüber hinaus können Simulink-Modelle hierarchisch sein und die Kapselung von Blöcken in Subsysteme mit eigenen Eingangs- und Ausgangsports ermöglichen.

Vehicle Dynamics Blockset – Vehicle Body 3DOF Dual Track

Das in Simulationen verwendete dynamische Modell eines Fahrzeugs ist innerhalb des Vehicle Body 3DOF-Blocks codiert und implementiert ein starres zweiachsiges Fahrzeugmodell, um Längs-, Quer- und Gierbewegung zu berechnen. Zur Vereinfachung werden Nick- und Wankbewegungsdynamiken vernachlässigt. Der Simulink-Block wurde mit den Optionen „Vehicle track: Dual“ und „Axle forces: External longitudinal velocity“ konfiguriert. Als Eingang dienen somit die Geschwindigkeit und das Lenkwinkelsignal. Die Codierung des Modells richtet sich, wie im Unterkapitel 3.1 näher erläutert, nicht nach der DIN ISO 8855 (vgl. DIN ISO 8855, 2013, S. 6f.), sondern nach der SAE J670E-Konvention (vgl. SAE J670E, 1976, S. 23). Der Vehicle Body 3DOF-Block berücksichtigt zur Verarbeitung der Daten die Gesamtmasse, den Luftwiderstand und die Gewichtsverteilung zwischen den Achsen aufgrund von Beschleunigung und Lenkung. Die Ausgangsgrößen sind Positions- und Geschwindigkeitskomponenten des Massenmittelpunkts in kartesischen Trägheitskoordinaten, Geschwindigkeitskomponenten in Ego-Koordinaten, Gierwinkel etc.

2D- und 3D-Visualisierung der Fahrmanöver in MATLAB/Simulink

Zur Ermittlung der Trajektorie visualisiert Simulink den 2D-Trajektorienverlauf und die 3D-Fahrmanöversimulation. Bei der 2D-Visualisierung wird ein „Bird’s-Eye Scope“ verwendet, das eine Draufsicht auf das Fahrzeug sowie eine Darstellung der Fahrmanöversimulation bietet. Dieses Tool zeigt zu jedem Zeitpunkt die Ausrichtung und Position des Fahrzeugs an. Eine zusätzliche Dimension ist mit den Blöcken „Simulation 3D Vehicle with Ground Following“ und „Simulation 3D Scene Configuration“ möglich. Diese dreidimensionale Visualisierung mit repräsentativen Kollisionsvermeidungstrajektorien auf der Simulationsfahrbahn ist die Vorlage für die Nachstellung im Simulationsprogramm CARLA.

Open-Source-lizenzierte Entwicklungsumgebung: Geany und Qt Creator

Als stabile und leistungsstarke IDE ermöglichen Geany und Qt Creator, mit einer großen Unabhängigkeit von bestimmten Desktopumgebungen den gleichzeitigen Betrieb anzubieten. Geany und Qt Creator sind im Vergleich zu MATLAB/Simulink Open-Source zugänglich. Qt Creator ermöglicht als plattformübergreifendes IDE die Erstellung von Anwendungen für mehrere Desktop-Geräteplattformen und Programmiersprachen wie z. B. C++. Geanys Texteditor besitzt mehr als 50 Programmiersprachen, bspw. Python, und läuft mit den gängigen Betriebssystemen Windows, Linux und MacOS (vgl. Woyand, 2019, S. 1). Python eignet sich im Vergleich zu C++ für schnelle Entwicklungen (Rapid Application Development [vgl. A.1.3 Weitere Definitionen]) in SiL-Projekten. Wie in Abbildung 12 dargestellt, ist die Erstellung und Erweiterung von Programmierschnittstellen (API) zum Simulationsprogramm CARLA und die Visualisierung in der Unreal Engine mit beiden Entwicklungsumgebungen realisierbar (vgl. Dosovitskiy, 2017, S. 2f.). Mit Python-Skripten ist die Verarbeitung, Berechnung, CARLA-Rohdatenauslesung sowie die Steuerung von Gas, Bremse und Lenkung am CARLA-Simulator ausführbar.

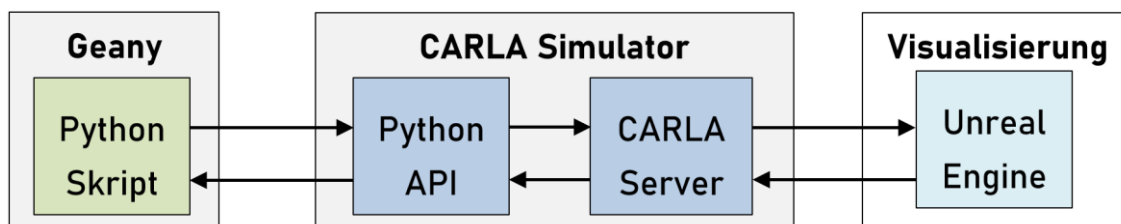


Abb. 12: Implementierung aus verschiedenen Entwicklungsumgebungen [Eigene Darstellung]

2.2.3 Visualisierung im Fahr Simulator CARLA

Die in C++ geschriebene Software CARLA ist ein auf der Spiel-Engine (Unreal Engine) fotorealistischer Open-Source-Simulator, der entwickelt wurde, um autonome Fahralgorithmen zu trainieren, zu validieren und zu testen. Er bietet digitale Assets und eine vollständige Kontrolle über statische und dynamische Verkehrsteilnehmer, über Umgebungsbedingungen, Sensormodelle und Kartengenerierung an (vgl. Dosovitskiy, 2017, S. 1 – 16). Ein weiterer Vorteil von CARLA sind die Sensorsimulationen, welche bereits in der Forschungs- und Entwicklungsphase verifizierbar sind. Da im Fahr Simulator wegen der simulierten Welt alle Informationen der Umgebung vorhanden sind, lassen sich verschiedene Sensoren modellieren (vgl. Sievers, 2018, S. 22 – 25). Dadurch kann der Schwerpunkt auf das Testen von datenverarbeitenden Algorithmen gelegt werden. Folglich können virtuelle Ego-Fahrzeuge in der ausgewählten Verkehrssituation auf einem Computerbildschirm von einem Fahrer gesteuert werden (vgl. Abbildung 13). Zusätzlich vorhandene physische Geräte – wie das in der nachfolgenden Illustration abgebildete Gaming-Lenkrad samt dazugehörigem Pedalerie-Set – gewährleisten die aktive Interaktion in der Fahrzeugumgebung.




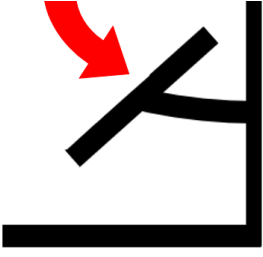
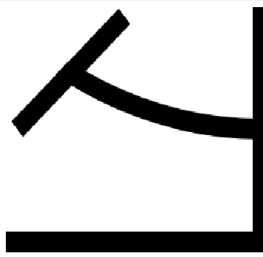


Abb. 13: Darstellung des Fahr Simulators CARLA [Eigene Darstellung]

Nach Simulationsstart öffnet die virtuelle Testumgebung ein Fenster, in welchem die Verbindung zum CARLA Server hergestellt und die Simulationsumgebung gerendert wird. Unter Verwendung weiterer Programmierschnittstellen ist eine aktive Steuerung mit den grün gekennzeichneten Hardwareprodukten (s. Abbildung 13) des Ego-Fahrzeugs in CARLA bedienbar.

Eine Übertragung der physischen Bewegung in die Simulation setzt eine Eingabeverarbeitung voraus. Die in der Programmierung hinterlegten Python-APIs verarbeiten die Eingaben der Lenkrad- und Pedalstellung des Fahrmodulators in Form von Attributen. Eigenschaften speichert der Programmierer mit bestimmten Informationen in Attributen ab. Jedes optionale Attribut besitzt, wie in Tabelle 2 ersichtlich, einen dazugehörigen Wert zwischen 0 und 1 (vgl. Rupp, 2021, S. 442).

Tab. 2: CARLA Attribute

	Maximal- und Minimalwerte der Attribute		
Attribut ID „steer“	 Linker Lenkradanschlag	 Mittellage	 Rechter Lenkradanschlag
Lenkradstellung	+1	0	-1
Attribut ID „brake“/ „throttle“	 Pedalausschlag	 Ausgangslage	
Brems- bzw. Gaspedalstellung	+1	0	

3D-Visualisierung der Fahrmanöver in der Spiel-Engine „Unreal Engine“

Die Unreal Engine besitzt verschiedene Bibliotheken, die als Module für spezielle Fahrzeugsimulationen bereitstehen (vgl. Unreal Engine, 2021). „PhysXVehicles“ ist eines der Module, welche die Physik eines Fahrzeugs vereinfacht als Punktmasse wiedergeben (vgl. Dosovitskiy, 2017, S. 3). Diese Modellvorstellung mit einem einfachen Fahrzeugmodell führt zu unplausiblen Werten der fahrdynamischen Parameter, die sich außerhalb der physikalischen Grenzen befinden.

3 Grundlagen der Fahrdynamik

Die Grundlagen der Fahrdynamik zur Trajektorienplanung eines Gesamtfahrzeugmodells sind Inhalt dieses Kapitels. Die Fahrdynamik beschäftigt sich mit der Fahrzeugbewegung unter Berücksichtigung der Einflüsse bei Momenten und Kräften. Das Fahrverhalten wird gezielt mit genormten Analysetechniken in Testverfahren oder realitätsnahen Simulationen bewertet. Dazu gehören die Definition der Fahrzeugbewegung und jene der objektiven sowie subjektiven Fahrmanöver. Danach wird auf die Teilmodelle zur Erstellung eines potenziellen Gesamtfahrzeugmodells des Ego-Fahrzeugs eingegangen.

3.1 Definition der Fahrzeugbewegung

Die Simulationssoftware CARLA verwendet das linkshändige Koordinatensystem analog zu SAE J670E und widerspricht der deutschen Norm (vgl. CARLA, 2021). Der Unterschied zum rechtshändigen Koordinatensystem ist, dass die y -Achse nach SAE J670E im Vergleich zur DIN ISO 8855 spiegelverkehrt liegt (vgl. DIN ISO 8855, 2013, S. 6). Eine Konvertierung vom linkshändigen in das rechtshändige Koordinatensystem ist daher für die Simulierung der Fahrzeugbewegung in Deutschland unabdingbar.

3.1.1 Ortsfestes Koordinatensystem

Die Achsen x und y spannen die horizontale Fahrbahnebene auf und die Achse z zeigt entgegen der Richtung der Erdbeschleunigung g nach oben, wie in Abbildung 14 dargestellt. Der Ursprung des Koordinatensystems wird mit 0 bezeichnet (vgl. DIN ISO 8855, 2013, S. 6 und Mitschke, 2014, S. 3f.).

3.1.2 Fahrzeugfestes Koordinatensystem

Das fahrzeugfeste Koordinatensystem liegt mit seinem Ursprung SP_A im Fahrzeugreferenzpunkt. Die translatorischen Freiheitsgrade werden in Richtung der x_A -Achse als Längsbewegung, bei Querbewegung in Richtung der y_A -Achse als Schieben in Fahrzeugquerrichtung und in Richtung der z_A -Achse als Hubbewegung wahrgenommen. Die rotatorischen Freiheitsgrade sind mit den Kardanwinkeln beschrieben und werden als Wanken bei Drehung um die x_A -Achse mit κ_A , Nicken während der Drehung um die y_A -Achse mit φ_A und Gieren bei Drehung um die z_A -Achse mit ψ_A bezeichnet (vgl. DIN ISO 8855, 2013, S. 6f. und Mitschke, 2014, S. 3f.). Die Fahrzeuglage im Raum wird somit eindeutig mit sechs Starrkörperfreiheitsgraden bestimmt (vgl. Abbildung 14).

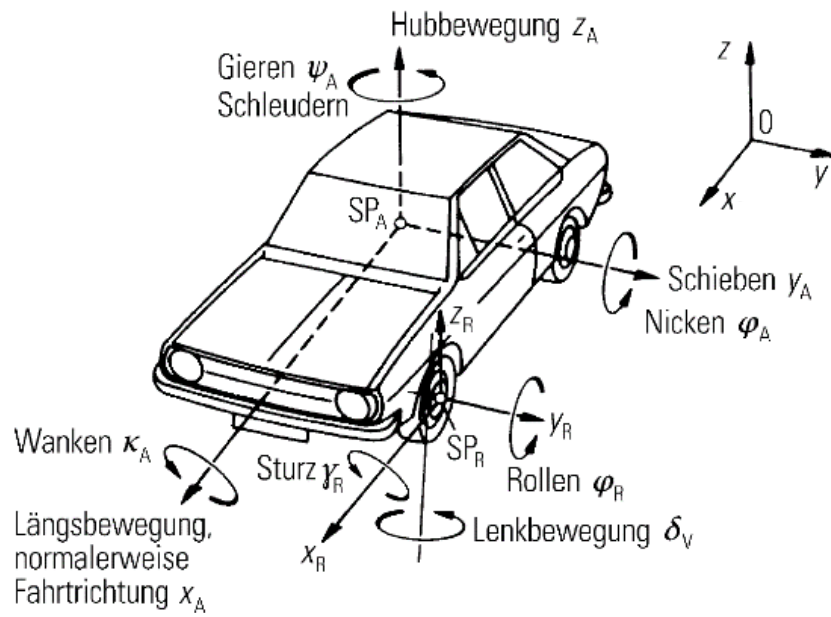


Abb. 14: Fahrzeugfestes und ortsfestes Achsenystem
[Veränderte Darstellung] (vgl. Mitschke, 2014, S. 4)

3.2 Trajektorien von Ausweichmanövern

Fahrdynamische Indikatoren ermöglichen eine subjektive oder objektive Bewertung kritischer Fahrsituationen. Bei einer subjektiven Beurteilung eines Sachverhaltes legt zuvor der Testführer bestimmte Kriterien fest und begutachtet die Voraussetzungen, wohingegen die objektive Bewertung keine persönliche Wertung beinhaltet und unabhängig vom Subjekt beurteilt (vgl. Schindler, 2019, S. 107). Zur Durchführung von Ausweichmanövern sind daher Messdatensätze zur objektiven Bewertung hinterlegt, die anhand physikalischer Werte ohne den Fahrereingriff aus Zeitdaten beurteilt werden (vgl. Heißing, 2013, S. 126f. und Winner, 2015, S. 126f.). Diese standardisierten Fahrspurwechsel nach ISO 3888-1, ISO 3888-2 sowie ISO 7401 haben viele Anforderungen und lassen sich grundsätzlich in Closed- und Open-Loop einteilen.

3.2.1 Closed-Loop-Manöver: Doppelter Fahrspurwechsel

Der geschlossene Regelkreis im Closed-Loop-Manöver beinhaltet den Fahrer, das Ego-Fahrzeug sowie die Umwelt (vgl. Heißing, 2013, S. 126f.). Als Regler ist der Fahrer vorgesehen, der die Fahrzeugführung und Lenkeingaben, bspw. für die Nachstellung von realen Fahrsituationen mit doppeltem Fahrspurwechsel, ausführt (vgl. Zomotor, 1991, S. 220). Der Test dient zur Prüfung des dynamischen Fahrverhaltens in einer unfallkritischen Situation, die ein unvorhersehbares Ausweichmanöver mit anschließendem Wechsel auf die ursprüngliche Fahrspur erfordert. Die Miteinbeziehung des Fahrers stellt einen großen Vorteil dar, da eine Analyse der Zusammenhänge zwischen Fahrerreaktion und Fahrmanöver realisierbar ist (vgl. Winner, 2015, S. 126 – 130). Im Nachfolgenden sind die Zustands- und Lenkwinkelangaben beim Ausführen eines Doppelspurwechsels in Abbildung 15 illustriert, die zur Fahrverhaltensbeurteilung dienen.

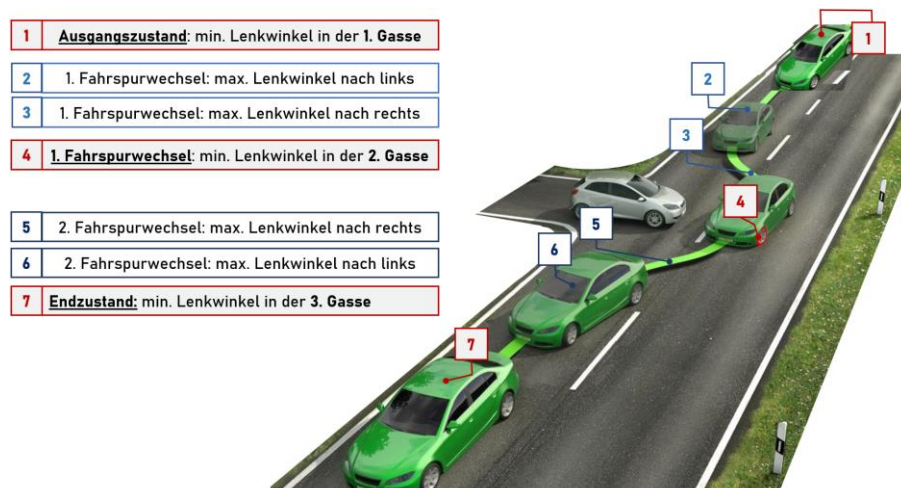


Abb. 15: Doppelspurwechsel im Trajektorienverlauf [Eigene Darstellung] (vgl. D'Ambrosi, 2020)

Es wird ein Ausweichmanöver von der ersten Gasse (Ausgangszustand) auf die linke Fahrspur (zweite Gasse) und anschließend das Zurücklenken auf die ursprüngliche rechte Fahrbahnseite (dritte Gasse) visualisiert. Abbildung 15 beschreibt die durchgeführte Kollisionsvermeidung und verdeutlicht im Übergang von Zustand 3 zu 4 mit Rot die starke Belastung des linken Vorderreifens infolge hoher Querbewegung. Die Fahrstabilität des Fahrzeugs wird ständig überwacht, um bei einem Closed-Loop-Manöver eine Kippgefahr vor dem Zustand 4 zu meiden. Die Beherrschbarkeitsuntersuchung beim schnellen Lenkvorgang gibt daraufhin Aufschluss darüber, welche Parameter und Grenzwerte für das Ausbrechen des Fahrzeugs verantwortlich sind. Standardisierte Verfahren zur Untersuchung der objektiven und subjektiven Beurteilung der Fahrmanöver beim Fahrspurwechsel sind in den Normen der ISO 3888-1 und ISO 3888-2 festgelegt.

ISO Wedeltest (ISO 3888-1)

In Fahrversuchen lassen sich mit dem Wedeltest Fahrerreaktionen anhand der Fahrweise beschreiben und identifizieren. Die identischen Randbedingungen der Ausgangsgrößen – gleiche Fahrzeuggeschwindigkeit und Trajektorie – werden vorausgesetzt. Abhängig vom Fahrzeugtyp liegt der Geschwindigkeitsbereich bei ca. $80 \frac{\text{km}}{\text{h}}$ für Personenkraftfahrzeuge und bei ca. $130 \frac{\text{km}}{\text{h}}$ für sportliche Fahrzeuge (vgl. Trzesniowski, 2019, S. 391). Die fahrdynamischen Größen Querbewegung, Giergeschwindigkeit, Schwimm-, Wank- und Lenkwinkel sowie Lenkmoment dürfen zu bestimmten Zeit- und Wegpunkten nicht als objektive vergleichbare Kriterien herangezogen werden, da diese vom Fahrer abhängig sind (vgl. Heißing, 2013, S. 129).

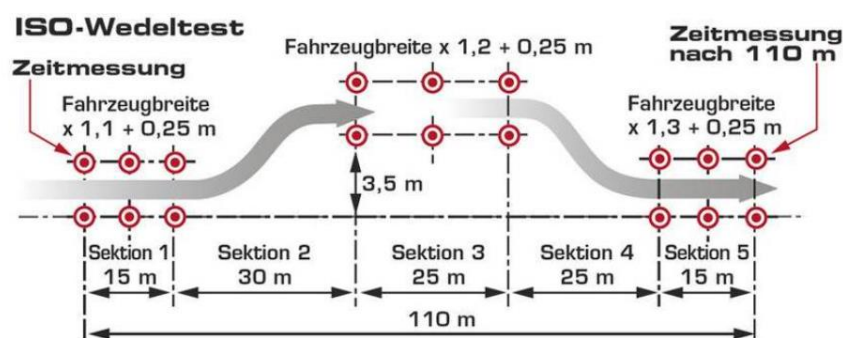


Abb. 16: Abmaße des ISO Wedeltests nach ISO 3888-1 (s. Frenet, 2021)

Der Wedeltest nach ISO 3888-1 liefert nur Erkenntnisse über Spurtreue und Lenkverhalten – dies betrifft Manöver, die unabhängig von fahrdynamischen Größen bei Versuchsfahrten von geübten Fahrern erprobt werden. Wie in Abbildung 16 verdeutlicht, wird die

Durchfahrtzeit mit Lichtschranken am Anfang und Ende der 110 m langen Messstrecke gemessen, genauso wie die fahrzeugabhängige Gassenbreite und der seitliche Versatz von 3,5 m. Zur Beurteilung werden aus mindestens drei fehlerfreien Fahrten, in denen nicht die Markierungskegel berührt wurden, nur die Mittelwerte der Durchfahrtzeit akzeptiert (vgl. ISO 3888-1, 2018, S. 1 – 5 und Trzesniowski, 2019, S. 391).

VDA-Ausweichtest (ISO 3888-2)

In der ISO 3888 wurde im Teil 2 der ursprüngliche „Elchtest“ nach einer Überarbeitung zur Kollisionsvermeidung als VDA-Ausweichtest veröffentlicht (vgl. ISO 3888-2, 2011, S. 1 – 5). Dieser Spurwechselstest mit einer anschließenden Rückkehr auf die Ausgangspur wird in der Fahrzeugentwicklung zur Beurteilung des Handhabungsverhaltens herangezogen. An drei Pylonengassen mit insgesamt 61 m Länge soll mit maximaler Geschwindigkeit ein doppelter Spurwechsel, wie in Abbildung 17 visualisiert, ausgeführt werden. Der ISO-Spurwechsel enthält eine Einfahr- und Ausweichgasse mit fahrzeugabhängiger Breite und Länge von 12 m und 11 m. Die Ausfahrgasse mit fahrzeugunabhängiger Breite (3 m) hat eine Länge von 12 m.

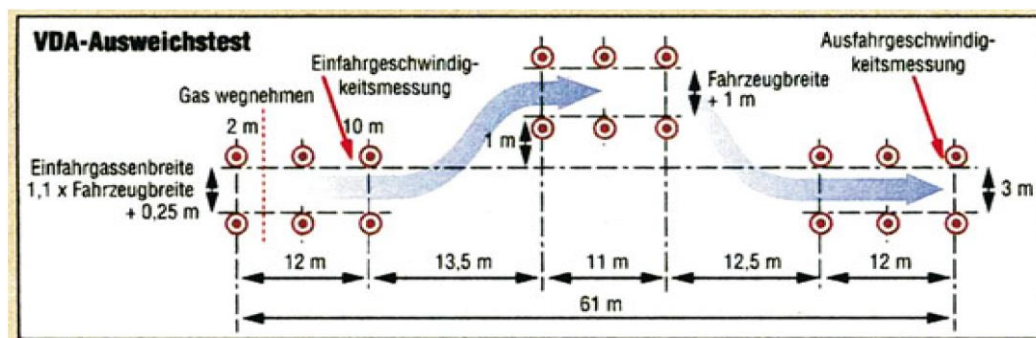


Abb. 17: Pylonengasse und Fahrzeugtrajektorie des VDA-Ausweichtests nach ISO 3888-2 (s. Heißing, 2013, S. 129)

Nach Beginn der Einfahrgasse wird die Fahrgeschwindigkeit ab 2 m sukzessiv erhöht, um den Ausweichvorgang im Schubbetrieb im höchsten Gang ausführen zu können. Die Bewertung erfolgt mit der maximalen Eingangsgeschwindigkeit, mit der die Gasse ohne Berührung der Pylonen durchfahren wird. Ziel ist hierbei, gegen Ende des Manövers eine möglichst hohe Querbewegung zu erreichen. Mit diesem VDA-Ausweichtest ist die kipp- und schleuderkritische Anregung der Fahrzeuge feststellbar, sodass die Auftretenswahrscheinlichkeit potenzieller Fahrzeugüberschläge bei Fahrversuchen erkannt werden kann (vgl. ISO 3888-2, 2011, S. 1 – 5 und Schramm, 2018, S. 372 – 374).

3.2.2 Open-Loop-Manöver: Einfacher Fahrspurwechsel

Die fahrdynamischen Ansprüche nach Reproduzier- und Messbarkeit erfordern bei Open-Loop-Manövern genormte Versuchsdurchführungen. Im offenen Regelkreis bzw. Open-Loop lassen sich die Versuche auf der Fahrdynamikfläche abbilden und durch Eingabe von Längsgeschwindigkeit und Querdynamik ein Lenkradwinkelverlauf festlegen. Ein Lenkroboter kann zur Verbesserung der Fahrzeugreaktion im unfallkritischen Fahrmanöver beitragen (vgl. Pfeffer, 2011, S. 135f. und Heißing, 2013, S. 126f.). Nachweise über den Fortschritt können mit Versuchen erbracht werden, bspw. durch Einleitung von Manövern, die nicht zu typischen, rein stationären Bedingungen zuzuordnen sind. So können Einflussparameter für den Lenkvorgang bei Durchführungen von Open-Loop-Manövern objektiv ermittelt werden, weshalb in manchen Versuchen fahrdynamische Zustände eine Relevanz für die reguläre Fahraufgabe darstellen. Beispiele für Tests der Open-Loop-Klasse aus der ISO 7401 sind der Weave-Test, der Lenkradwinkelsprung oder der Einzelsinus, der im Folgenden vorgestellt wird (vgl. ISO 7401, 2011, S. 4 – 9).

Sinusoidal input — one period (ISO 7401)

Der Einzelsinus-Test (Single Sine Test) ist nach ISO 7401 ein Open-Loop-Fahrmanöver für transientes, querdynamisches Übertragungsverhalten. Wie in Abbildung 18 ersichtlich, wird hierbei das Ziel verfolgt, die sinusförmige Lenkwinkleingabe mithilfe von ISO/ TR 8725 zu beschreiben. Der entstandene Zeitverzug zwischen der Lenkradwinkleingabe und den Fahrzeugreaktionsgrößen gibt Auskunft über das Eigenschwingverhalten des Fahrzeugs. Dies kann z. B. bei einem doppelten Fahrspurwechsel, vergleichbar mit dem VDA-Ausweichtest, untersucht werden. Bei der Einzelsinus-Lenkradwinkleingabe ist allerdings zu beachten, dass das Fahrzeug aus der Geradeausfahrt im stationären Zustand durch eine Lenkradwinkleinleitung einer Sinusperiode wieder in die Geradeausfahrt gebracht wird und somit einen einfachen Fahrspurwechsel ausführt (vgl. Pfeffer, 2011, S. 136f. und Graf, 2014, S. 34f., 78f. und 97). Die für die Modellierung gewählten standardisierten Basisparameter sind $f = 0,5 \text{ Hz}$, $v_{\text{Fzg}} = 100 \frac{\text{km}}{\text{h}}$ und $|a_{y,\text{max}}| = 4 \frac{\text{m}}{\text{s}^2}$ (vgl. ISO 7401, 2011, S. 7 – 10). Somit werden wichtige Erkenntnisse zur objektiven Beschreibung des Lenkverhaltens im Zusammenhang mit Lenkradwinkel und Lenkradmoment gezogen (vgl. Pfeffer, 2011, S. 136f.).



Abb. 18: Einzelsinus nach ISO 7401 [Eigene Darstellung]

3.3 Gesamtfahrzeugmodell

Bei konzeptionellen Modellen sind in Modellierungsprozessen mathematische und physikalische Grundlagen wiederzufinden (vgl. Isermann, 2006, S. 32). Vor der Entwicklung dieser Modelle müssen spezifische fahrdynamische Analysen beispielsweise vom Lenk-, Reifensystem sowie vom Fahrzeugaufbau für den Anwendungsfall spezifiziert und mit reellen Parametern aus Erfahrungswerten programmiert werden. Das reale Fahrzeug wird zu Untersuchungszwecken mit Dekomposition von Ebene 1 zu 2 in reale Bauteile unterteilt (vgl. Abbildung 19). Dennoch ist dies nicht hinreichend, um den gewünschten Anwendungszweck vollkommen zu erfüllen. Die Teilsysteme können dann, wie in Abbildung 19 in der 3. Ebene ersichtlich, in Teilmodelle abstrahiert und einzeln programmiert werden.

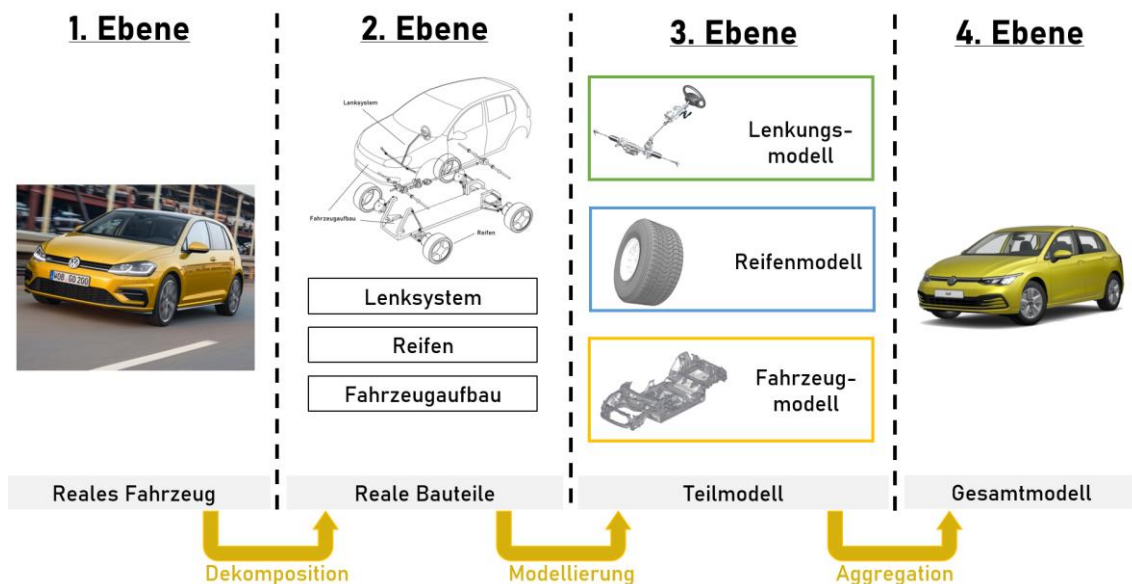


Abb. 19: Planung in vier Ebenen für die Modellierung eines virtuellen Gesamtfahrzeugs [Eigene Darstellung] (vgl. Isermann, 2006, S. 32, Carwow, 2021, Braess, 2013, S. 502, Sokolova, 2011, Modellboard, 2013, Volkswagen, 2021 und Schramm, 2018, S. 12)

Modellierungen auf Komponenten- und Subsystemebene ermöglichen es, ein komplexes Gesamtfahrzeugmodell als gleichwertige Systemabbildung, bestehend aus Lenkungs-, Reifen- und Fahrzeugmodellen, zu simulieren – wie in der 4. Ebene abgebildet. Dabei sind Prognosen zu physikalischen Veränderungen einzelner Subsysteme nicht über das Fahrzeugverhalten treffbar, da die Teilmodelle Mischformen aus Verhaltens- und Strukturmodellen sind. Hierbei können die Teilmodelle das virtuelle Äquivalent zum realen Fahrzeug sein und für Versuchszwecke als Gesamtmodell verwendet und mit der entsprechenden Parametrisierung unabhängig von der Modellstruktur erweitert werden.

3.3.1 Lenkungsmodell

Lenkungsbestandteile wie Lenksäule, Lenkgetriebe, Lenkunterstützung und Lenkungssteifigkeit sind in einem Lenkungsmodell adaptierbar und implementierbar. Ein realitätsnahes Fahrzeugverhalten in der MiL-Software bildet das Lenksystem für Fahrdynamikuntersuchungen mit hohem Detaillierungsgrad nach und wird im SiL-Modelltest für die nachfolgende Berechnung der Gleichungen verwendet (vgl. Abbildung 11).

Der Subtrahend der Gl. (2) berücksichtigt die Lenkelastizität und die mechanische Verformung an den Vorderräderlenkwinkel $\delta_{v,i/o}$ unabhängig von der Fahrzeuggeschwindigkeit v_{Fzg} . Der Lenkwinkel unterscheidet sich vom Lenkradwinkel δ_H durch die konstante Lenkübersetzung i_s , den Gesamtnachlauf n_{ges} , die gesamte Lenkungssteifigkeit C_s und die schräglaufwinkelabhängigen Seitenkraft $F_{y,v,i/o}$ (vgl. Pfeffer, 2011, S. 35 und 81f.):

$$\delta_{v,i/o} = \frac{\delta_H}{i_s} - \frac{F_{y,v,i/o} \cdot n_{ges}}{C_s} \quad (2)$$

Für die Lenkungssteifigkeit gilt (vgl. Pfeffer, 2011, S. 35 und 82):

$$C_s = \left(\frac{1}{C_R} + \frac{1}{i_s^2 \cdot C_T \cdot A_s} \right)^{-1} \quad (3)$$

Für das Lenkmoment gilt (vgl. Pfeffer, 2011, S. 82):

$$M_H = \frac{M_s}{i_s \cdot A_s} \quad (4)$$

Im Rahmen der Modellierung und Verarbeitung der Modelldatensätze hat der Fahrer die Möglichkeit, den Lenkradwinkel δ_H als Eingabegröße mit dem Gaming-Lenkrad zu bestimmen (vgl. Abbildung 13). Das Lenksystem ist wie die anderen realen Systeme durch nichtlineare partielle Differenzialgleichungen gekennzeichnet. Wichtig ist die Reaktion des zu regelnden Prozessparameters auf seine Stellgröße, z. B. wie stark und wie schnell der Lenkradwinkel auf eine definitive Veränderung reagiert. Der Lenkradwinkel δ_H wird im Fahrzeug durch einen unterlagerten Lenkradwinkelregler eingestellt. Bei der Betrachtung dynamischer Systeme muss in der Regelungstechnik von der tatsächlichen Realisierung abstrahiert und ein Lenkungsmodell konstruiert werden. Ein idealisierter Lenkradwinkelsprungverlauf weist zu Beginn des Manövers eine unendliche Lenkradwinkelgeschwindigkeit $\dot{\delta}_H$ auf. Im realen Fall wird die Lenkradwinkelgeschwindigkeit jedoch durch die Leistungsfähigkeit des Lenkaktors begrenzt. Gemäß dem Parameter verbauter Lenkaktoren wird im Folgenden von einer maximalen Lenkradwinkelgeschwindigkeit

$|\dot{\delta}_{H,\max}| \approx 1500 \frac{\text{deg}}{\text{s}}$ ausgegangen (vgl. Bootz, 2004, S. 43f.). Das Lenksystem beider Längsaktoren verhält sich wie ein Verzögerungselement erster Ordnung und dient als Ausgangspunkt für die Modellierung (vgl. Söhnitz, 2001, S. 42). Dieses PT1-Verzögerungsglied wird mit einer Zeitkonstante $\tau = 0,1 \text{ s}$ für das Lenkungsmodell approximiert. Damit ist der Endwert des SOLL-Lenkradwinkels $\delta_{H,\text{SOLL}}$ bei einem Lenkradwinkelsprung nach ca. $5 \cdot \tau = 0,5 \text{ s}$ erreicht. Projiziert auf einen Lenkradwinkelsprung stellt sich für die Lenksystemmodellierung ein resultierender IST-Lenkradwinkel $\delta_{H,\text{IST}}$ ein.

3.3.2 Reifenmodell

Kräfte und Momente können durch Oberflächenprofile und Reibungsniveau der Fahrbahn von der Radnabe an den Latsch und zum Reifen übertragen werden. Schnittstellen zwischen Reifen und Straße sind zu beachten, um Parameter abhängig von den Reifen-Fahrbahn-Kontaktflächen zu definieren (vgl. Pfeffer, 2011, S. 2 – 6). Reifen sind die einzigen Bindeglieder, die die Fahrbahn mit dem Gesamtfahrzeugmodell koppeln. Sie nehmen in vertikaler Richtung die Radlast auf und bestimmen daher das längs- und querdynamische Fahrverhalten, weshalb sie als räumliches Kraftelement betrachtet werden (vgl. 3D-Visualisierung des Reifens in Abbildung 20). Die Reifeneigenschaften beeinflussen das Fahrverhalten signifikant, da sie für die Kraftübertragung und damit für die Kurshaltung des Fahrzeugs verantwortlich sind. Zunehmend kann daher auf Reifenmodelle zurückgegriffen werden, für die in Simulationsprogrammen einerseits Koordinatensysteme und andererseits Ein- und Ausgangsgrößen als standardisierte physikalische Parameter verwendet werden (vgl. Meywerk, 2007, S. 245 – 250). Die Simulationsmodellergebnisse müssen das quasistationäre ebenso wie das transiente Verhalten der Fahrdynamik beschreiben, damit die Realität gut abgebildet wird.

Schräglaufwinkel

Zur Bestimmung der Seitenkraft existieren Reifenmodelle mit verschiedenen Modellkomplexitäten, die grundsätzlich in die Kategorien „mathematische Modelle“, „physikalische Modelle“ und „Mischformen“ unterteilt werden (vgl. Meywerk, 2007, S. 245 – 248 und Schramm, 2018, S. 145f.). An der Kontaktfläche zwischen Reifen und Fahrbahn (Latsch) wirkt, wie in Abbildung 20 dargestellt, die Seitenkraft F_y als kinematische Kraft. Diese Seitenkräfte können in Abhängigkeit des Schräglaufwinkels, der von dem blauen Graphen entlang der gestrichelten Linie unterteilt wird, in einen linearen und nichtlinearen Bereich unterteilt werden. Beim Latscheinlauf steigt die Seitenkraft zunächst linear an (Haftung) und führt mit steigendem Schräglaufwinkel zur Auslenkung der Profilelemente und somit zur Tangentialspannung zum hinteren Latschrand (vgl. Abbildung 20). Ab der maximalen Seitenkraft geht das Haften ins Gleiten über und stellt einen degressiven Verlauf dar.

Zwischen der Reifenlängsgeschwindigkeit v_{R_x} und dem Reifengeschwindigkeitsvektor v_R im Radmittelpunkt befindet sich der Schräglaufwinkel α (vgl. Abbildung 20). Der Schräglaufwinkel ist im normalen Fahrbetrieb auf $\alpha_{v/h} < 12$ deg festgelegt (vgl. Trzesniowski, 2014, S. 234 und Schramm, 2018, S. 162) und wird wie folgt berechnet (vgl. Schramm, 2018, S. 148):

Schräglaufwinkel der Vorderachse:

$$\alpha_v = \delta_v - \tan^{-1} \left(\frac{l_v \cdot \dot{\psi} + v \cdot \sin \beta}{v \cdot \cos \beta} \right) \quad (5)$$

Schräglaufwinkel der Hinterachse:

$$\alpha_h = \tan^{-1} \left(\frac{l_h \cdot \dot{\psi} - v \cdot \sin \beta}{v \cdot \cos \beta} \right) \quad (6)$$

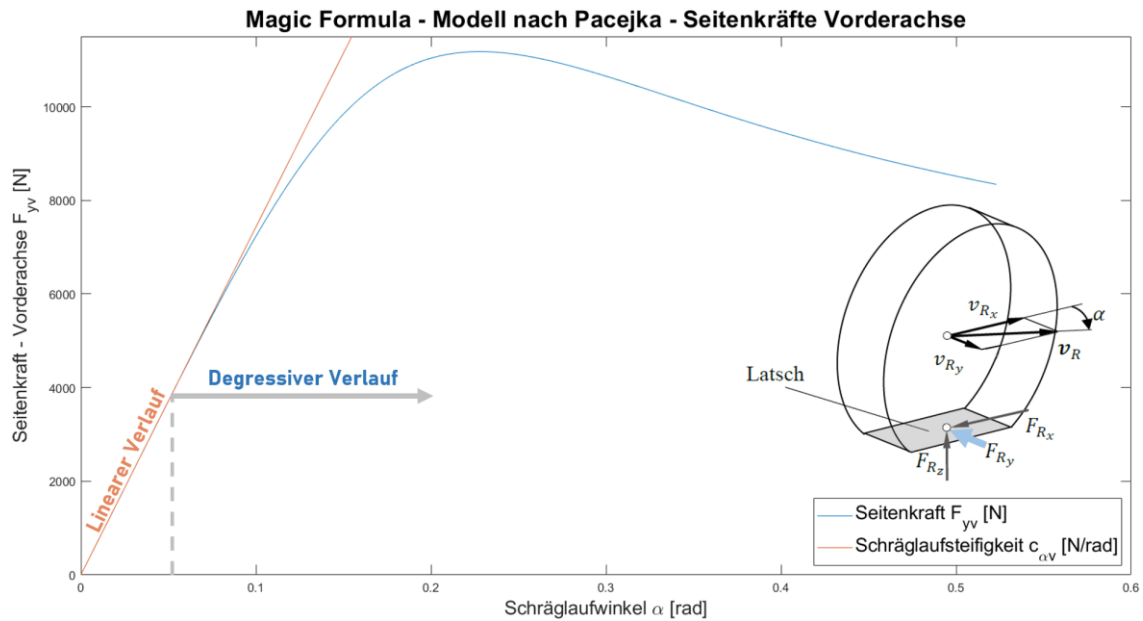


Abb. 20: Reifenkräfte im Reifenlatsch und kinematische Größen am Reifen
[Eigene Darstellung] (vgl. Schramm, 2018, S. 148 und 165 und Mitschke, 2014, S. 51)

Lineares Reifenmodell

Für eine Vereinfachung der Modellierung der Seitenführungskräfte werden bei der Reifenmodellierung lineare Zusammenhänge verwendet, die in normalen Verkehrssituationen wiederzufinden sind. Bei der Linearisierung im Reifenmodell gilt für die reifenspezifische Schräglaufsteifigkeit c_{α} , dass die Kräfte in Querrichtung linear abhängig vom Schräglaufwinkel α der Reifen sind (vgl. Abbildung 20, oranger Graph). Einwirkende Querkräfte können mit den nachfolgenden Gleichungen für Vorder- und Hinterreifen berechnet werden (vgl. Isermann, 2006, S. 40f. und Schramm, 2018, S. 229):

$$F_{y,v} = c_{\alpha,v} \cdot \alpha_v \quad (7)$$

$$F_{y,h} = c_{\alpha,h} \cdot \alpha_h \quad (8)$$

Das degressive Verhalten nach der Haftreibung wird zur Vereinfachung beim linearisierten Reifenmodell vernachlässigt. Diese Annahme gilt für die fahrdynamischen Bereiche $|a_y| < 0,4 \cdot g$ (ca. $3 - 4 \frac{\text{m}}{\text{s}^2}$) bzw. dazu äquivalent $\alpha < 3 \text{ deg}$ bzw. $0,052 \text{ rad}$ (vgl. Abbildung 20) und bildet somit eine lineare Systembeschreibung (vgl. Mitschke, 2014, S. 41 und Schramm, 2018, S. 226).

Nichtlineares Reifenmodell

In realitätsnahen 3D-Fahrsimulationen sind unterschiedliche fahrdynamische Untersuchungen mit nichtlinearen Reifenmodellen durchführbar. In Abbildung 20 ist ersichtlich, wie die blaue Seitenkraftkurve durch den Koordinatenursprung $\alpha = F_y = 0$ läuft und – nachdem sie den Maximalwert erreicht hat – an eine horizontale Asymptote anschmiegt. Das Magic Formula (MF) Tyre-Reifenmodell nach Pacejka ist für die Verwendung bei nichtlinearen hochdynamischen Fahrmanövern geeignet, da eine Darstellung gemessener Reifendaten mit einer sehr hohen Abbildungsgenauigkeit realisierbar ist. Die Koeffizienten für die Kraftübertragungsformel nach Pacejka werden an realen Reifen mit quasistatischen Reifenmessungen an Rollen- oder Fahrversuchen erfasst (vgl. Pütz, 2019, S. 55).

Neben der einfachen mathematischen Beschreibung aller charakteristischen Reifengrößen (z. B. Reifenseitenkräfte) können die Variablen der Magic Formula physikalisch exemplifiziert werden. Die Magic Formula Tyre beschreibt im nichtlinearen Bereich einen degressiven Verlauf (vgl. Pfeffer, 2011, S. 32) und lautet (vgl. Isermann, 2006, S. 40 und Pacejka, 2012, S. 7):

$$F_y(\alpha) = D \cdot \sin\{C \cdot \tan^{-1}[B \cdot \alpha - E \cdot (B \cdot \alpha - \tan^{-1}(B \cdot \alpha))]\} \quad (9)$$

F_y repräsentiert dabei die Seitenkraft am Reifen und α den Schräglaufwinkel.

Folgende Koeffizienten kommen in der Formel vor:

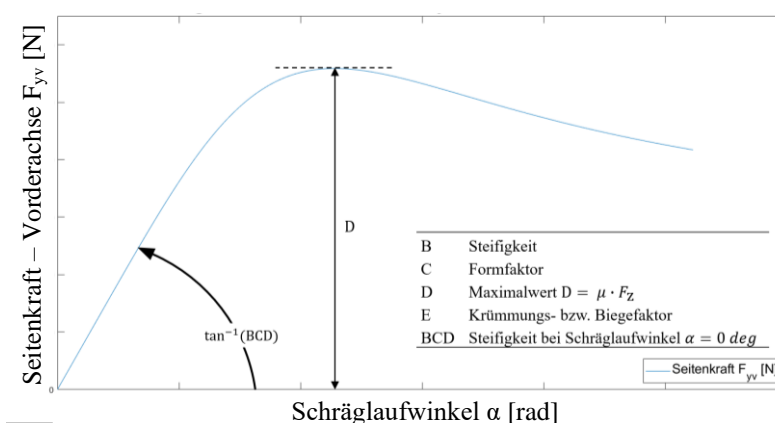


Abb. 21: Interpretation der Magic Formula-Koeffizienten
[Eigene Darstellung] (vgl. Schramm, 2018, S. 180)

Die mathematische Approximation des nichtlinearen Reifenmodells ist unabhängig von physikalischen Gesetzen. Daher benötigt die Implementierung und Auswertung der stetig numerischen Funktionsstabilität bei der Verwendung dieser Programmierung eine kurze Berechnungszeit.

3.3.3 Fahrzeugmodell

Komplexe mathematische Gesamtfahrzeugmodelle werden in der Entwicklung von Fahrzeugsystemen zunehmend etabliert. Aus diesem Grund existieren eine Vielzahl unterschiedlicher Fahrzeugmodelle, abhängig vom Anwendungsbereich und Detaillierungsgrad. Diese reichen von Punktmassenmodellen mit wenigen mathematischen Fahrdynamikgleichungen bis hin zu komplexen nichtlinearen Zweispurmodellen, die eine Beschreibung der rotatorischen und transienten Fahrzeugbewegung erlaubt (vgl. Schramm, 2018, S. 74f. und 268 – 292). In Fahrzeugmodellen sind Ego-Fahrzeuge mit diskreten mathematischen Gleichungssystemen programmiert, deren Übertragungsverhalten aus realen Fahrzeugen analysiert wurde. Das Gesamtfahrzeugmodell setzt sich, wie in Abbildung 22 dargestellt, aus Lenkungs-, Reifen- und Fahrzeugmodellen zusammen. Diese beeinflussen die Stellsysteme sowie die Vertikal- und Querdynamik des Fahrzeugs abhängig von den eingegebenen Parametern. Das Fahrsimulationsmodell in CARLA wiederum ist je nach Fahraufgabe beeinflussbar (vgl. Abbildung 22).

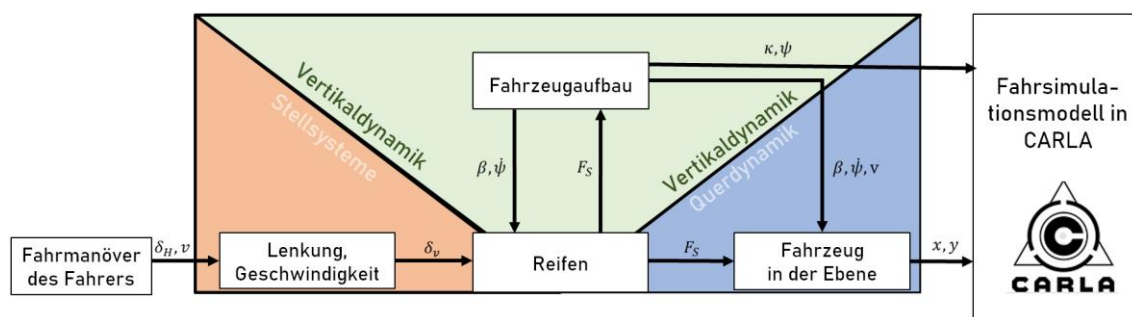


Abb. 22: Struktur des Gesamtfahrzeugmodells [Eigene Darstellung] (vgl. Halfmann, 2003, S. 22)

Wie in der Abbildung 22 zu sehen ist, gibt der Fahrer Lenkwinkel und Fahrzeuggeschwindigkeit vor und lenkt die Reifen auf die gewünschte Route. CARLA visualisiert mit dem Fahrsimulationsmodell das autonome Ausweichmanöver in einer Ebene und stellt die ausgewählte Trajektorie dar. Somit ist die realitätsnahe Funktionsweise des Gesamtfahrzeugmodells durch Fahrereingaben an das Fahrsimulationsmodell wie im realen Fahrzeug qualitativ prüfbar (vgl. Abbildung 13).

Nichtlineares Einspurmodell

Zur Vereinfachung von Fahrzeugmodellen werden bei Einspurmodellen (ESM) von jeder Achse je zwei Räder zu jeweils einem virtuellen Rad auf der Längsachse zusammengefasst. Der Fahrzeugschwerpunkt befindet sich auf der Straßenoberflächenebene und besitzt drei Freiheitsgrade (vgl. Freikörperbild in Abbildung 23), die Bewegungen translatorisch in x_0 - und y_0 -Fahrzeugebene sowie eine Rotation um die orthogonale z_0 -Achse ermöglichen. Diese Modellvorstellung ähnelt einem Fahrradmodell, daher werden hierbei keine Hub-, Nick- und Wankbewegungen, d. h. Bewegungen translatorisch in der z_0 -Achse und Drehungen um die y_0 - und x_0 -Achse, berücksichtigt (vgl. Schindler, 2019, S. 29 – 31 und Schramm, 2018, S. 225). Die Modellvereinfachung realisiert eine Verringerung des Rechenaufwands unter Berücksichtigung der fahrdynamischen Grenzbereiche als einfaches dynamisches Fahrzeugmodell.

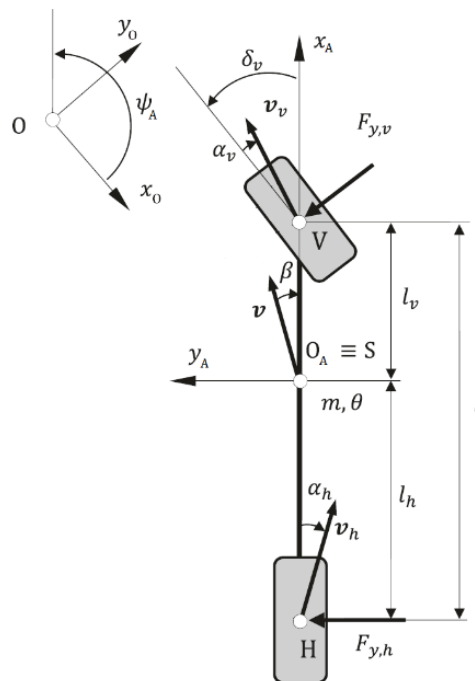


Abb. 23: Mathematische Beschreibung des Einspurmodells
[Veränderte Darstellung] (vgl. Schramm, 2018, S. 227)

Die Bewegungsgleichung zur Bestimmung des Einspurmodells setzt sich aus dem Kräfte- und Momentengleichgewicht zusammen (vgl. Abbildung 23). Alle Kräfte in Längsrichtung zusammen mit der Summe aller Momente orthogonal zur Fahrzeugebene bilden ein Gleichgewicht. Dieses Kräftegleichgewicht besteht aus kinematischen Parametern, abgesehen von den kinetischen Reifenseitenkräften $F_{y,v/h}$ und der Fahrzeugmasse m , die eine fahrzeugspezifische Größe ist. Die kinematische Größe δ_h wird mit null gleichgesetzt, da eine aktive Hinterachslenkung in dieser Arbeit und in Abbildung 23 keine Betrachtung findet.

Kräftegleichgewicht orthogonal zur Fahrzeuggeschwindigkeit (vgl. Schramm, 2018, S. 230 und Heißing, 2013, S. 103f.):

$$m \cdot [v \cdot (\dot{\beta} + \dot{\psi}) \cdot \cos(\beta)] = \cos(\delta_v) \cdot F_{y,v}(\alpha_v) + \cos(\delta_h) \cdot F_{y,h}(\alpha_h) \quad (10)$$

m : Gesamtfahrzeugmasse im Aufbauswerpunkt

v : Fahrzeuggeschwindigkeit

β : Schwimmwinkel (Winkel zwischen Geschwindigkeitsvektor und Fahrzeuglängsachse)

ψ : Gierwinkel (Winkel zwischen der Fahrzeuglängsachse und der x -Achse des ortsfesten Koordinatensystems)

δ_v : Lenkwinkel am Vorderrad

δ_h : Lenkwinkel am Hinterrad, wobei ein Fahrzeug ohne Hinterachslenkung mit $\delta_h = 0$ modelliert werden soll (eine aktive Hinterachslenkung wird daher in dieser Arbeit nicht betrachtet)

$F_{y,v/h}$: Reifenseitenkräfte (Wechselwirkung des Reifens mit der Fahrbahnoberfläche)

Beim Momentengleichgewicht setzen sich die fahrzeugcharakteristischen Größen aus kinetischen und kinematischen Parametern zusammen. Das Momentengleichgewicht erfolgt stets um den Fahrzeugschwerpunkt. Unter Verwendung des Impuls- bzw. Drallsatzes ergibt sich für das querdynamische Momentengleichgewicht folgende Gleichung (vgl. Schramm, 2018, S. 230 und Heißing, 2013, S. 103f.):

$$J_z \cdot \ddot{\psi} = \cos(\delta_v) \cdot F_{y,v}(\alpha_v) \cdot l_v - \cos(\delta_h) \cdot F_{y,h}(\alpha_h) \cdot l_h \quad (11)$$

J_z : Trägheitsmoment um die durch den Schwerpunkt gehende z_A -Achse

$l_{v/h}$: Abstand vom Fahrzeugschwerpunkt zur Vorder-/ Hinterachse

Das MF Tyre-Modell nach Pacejka wird für das nichtlineare Einspurmodell als nichtlineares Reifenmodell herangezogen, um die Reifenseitenkräfte $F_{y,v}(\alpha_v)$ und $F_{y,h}(\alpha_h)$ des realen Fahrzeugs nachzubilden. Somit ist eine Berücksichtigung der Schräglaufwinkel im Bereich des degressiven Seitenkraftverlaufs möglich. Reifennachläufe, Rückstellmomente und Umfangskräfte an den Reifen infolge der Schräglaufwinkel werden vernachlässigt und für das Einspurmodell nicht berücksichtigt (vgl. Schramm, 2018, S. 225).

Lineares Einspurmodell

Bei Vereinfachung der nichtlinearen Einspurmodellgleichungen (10) und (11) entstehen linearisierte Gleichungen mit Kleinwinkelnäherungen für die trigonometrischen Funktionen. Die Fahrzeugwinkel, wie bspw. der Lenkwinkel δ_v , sind folglich als klein anzunehmen (Linearisierung der Winkelfunktionen, z. B. $\sin(x \ll 1 \text{ deg}) \approx x$ und $\cos(x \ll 1 \text{ deg}) \approx 1$ [vgl. Heißing, 2013, S. 96]), weshalb die Querbeschleunigungsgrenze beim linearen Reifenmodell auf ca. $|a_y| = 4 \frac{\text{m}}{\text{s}^2}$ begrenzt wird (vgl. Mitschke, 2014, S. 41). Die Zustandsdifferentialgleichungen ergeben sich durch Einsetzen der Gleichungen (7), (8), (10) und (11), durch die Berücksichtigung der Kleinwinkelnäherung und durch Auflösung nach $\dot{\beta}$ und $\ddot{\psi}$ (vgl. Schramm, 2018, S. 230):

$$\dot{\beta} = \left(-\frac{c_{\alpha,v} + c_{\alpha,h}}{m \cdot v} \right) \cdot \beta + \left(-1 - \frac{c_{\alpha,v} \cdot l_v - c_{\alpha,h} \cdot l_h}{m \cdot v^2} \right) \cdot \dot{\psi} + \left(\frac{c_{\alpha,v}}{m \cdot v} \right) \cdot \delta_v \quad (12)$$

$$\ddot{\psi} = \left(-\frac{c_{\alpha,v} \cdot l_v - c_{\alpha,h} \cdot l_h}{J_z} \right) \cdot \beta + \left(-\frac{c_{\alpha,v} \cdot l_v^2 + c_{\alpha,h} \cdot l_h^2}{J_z \cdot v} \right) \cdot \dot{\psi} + \left(\frac{c_{\alpha,v} \cdot l_v}{J_z} \right) \cdot \delta_v \quad (13)$$

Nichtlineares Zweispurmodell

Beim Entwicklungsprozess eines geeigneten Gesamtfahrzeugmodells für fahrdynamische Grenzsituationen sind in dieser Arbeit nichtlineare Zweispurmodelle (ZSM) unumgänglich. Die Verwendung des MF nach Pacejka ist bei nichtlinearen Verhältnissen für die Berechnung der Seitenkräfte zwischen Reifen und Fahrbahn Voraussetzung, deren Nutzung von linearer Reifenmodellen unplausible Werte ergeben würden. Im Vergleich zum Einspurmodell wird die Nick- und Wankbewegung beim nichtlinearen Zweispurmodell über eine entkoppelte Betrachtung der Aufbaumasse berücksichtigt, da sich der Fahrzeugschwerpunkt nicht auf der Fahrbahnoberfläche befindet (vgl. Schramm, 2018, S. 255 – 262 und Mitschke, 2014, S. 807). Des Weiteren werden in der folgenden Darstellung des Modells vier Radaufstandspunkte verwendet, sodass eine Zusammenfassung der Räder auf einer virtuellen Längsachse nicht mehr gegeben ist.

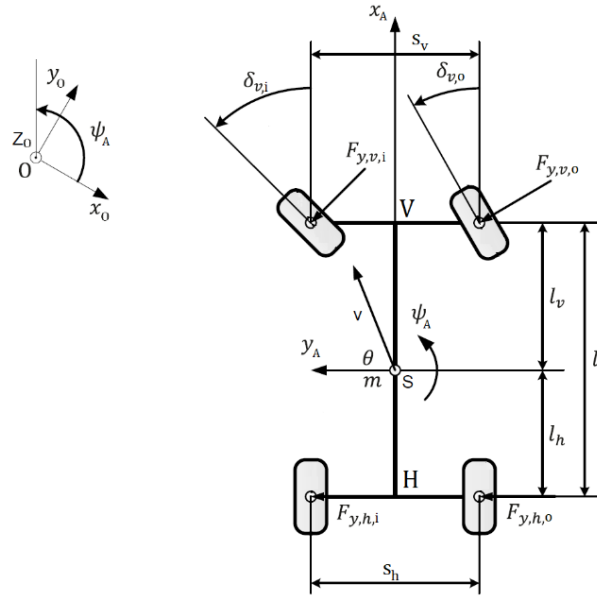


Abb. 24: Mathematische Beschreibung des Zweispurmodells

[Eigene Darstellung] (in Anlehnung an Mitschke, 2014, S. 834 und Schramm, 2018, S. 227 und 258)

Wie in Abbildung 24 ersichtlich, unterscheiden sich die kurvenäußeren und -inneren Lenkwinkel rechnerisch und können für weitere Berechnungen als $\delta_{v,m}$ berücksichtigt werden (vgl. Trzesniowski, 2014, S. 478):

$$\delta_{v,m} = \frac{\delta_{v,i} + \delta_{v,o}}{2} \quad (14)$$

Die Reifenseitenkräfte $F_{y,v}$ bzw. $F_{y,h}$ des Einspurmodells splitten sich beim Zweispurmodell in $F_{y,v,i}$ und $F_{y,v,o}$ bzw. $F_{y,h,i}$ und $F_{y,h,o}$ auf, da nun eine Vorder- und eine Hinterachse mit jeweils zwei Reifen vorhanden sind. Die Berechnung des Kräfte- und Momentengleichgewichts wird entsprechend um eine zweite Spur und um die Wank- und Nickachse erweitert:

Kräftegleichgewicht orthogonal zur Fahrzeuggeschwindigkeit (vgl. Schramm, 2018, S. 258):

$$m \cdot [v \cdot (\dot{\beta} + \dot{\psi}) \cdot \cos(\beta)] = \cos(\delta_{v,m}) \cdot (F_{y,v,i}(\alpha_{v,i}) + F_{y,v,o}(\alpha_{v,o})) + (F_{y,h,i}(\alpha_h) + F_{y,h,o}(\alpha_h)) \quad (15)$$

Momentengleichgewicht um den Fahrzeugschwerpunkt auf der (x_A, y_A) -Achse (vgl. Schramm, 2018, S. 258):

$$J_z \cdot \ddot{\psi} = \cos(\delta_{v,m}) \cdot (F_{y,v,i}(\alpha_{v,i}) + F_{y,v,o}(\alpha_{v,o})) \cdot l_v - (F_{y,h,i}(\alpha_h) + F_{y,h,o}(\alpha_h)) \cdot l_h \quad (16)$$

Jedes einzelne Rad hat Einfluss auf die horizontale Bewegung des Gesamtfahrzeugs, die aus Längs- und Querkräften resultiert. Die Anzahl der Freiheitsgrade erhöht sich aufgrund der Vernachlässigung der Hubbewegung in der z-Achse mit zunehmender Komplexität des Fahrzeugmodells von drei auf fünf.

Wank- und Nickmodell

Beim Ausführen eines Ausweichmanövers federn, wie in Abbildung 25 dargestellt, die kurveninneren Räder wegen der Fliehkraft aus, die kurvenäußeren Räder dagegen ein. Das Fahrzeug neigt sich also um die Wank- und Nickachse, wodurch eine Wank- und Nickbewegung entsteht (vgl. Mitschke, 2014, S. 816 – 818).

Nach der Bestimmung der Einzelsteifigkeiten, der Feder-/ Dämpferspurweite, der Stabilisatoren und des Schwerpunkts der Wank- und Nickpolachse kann das Wank- und Nickverhalten modelliert werden. Das Verhalten von Federn und Dämpfern wird in dieser Berechnung als linear angenommen. Zusätzlich bleibt die Lage der Wank- und Nickschwerpunkte konstant, d. h. weder die Längs- noch die Vertikaldynamik werden berücksichtigt. Die Entkopplung der Zweispurmodellbetrachtung erlaubt es, die Wank- und Nickeigenschaften als eine Aufbaumasse mit vier Einzelfedern zu modellieren. Aufgrund der auftretenden Querbeschleunigung – unter Berücksichtigung der Fahrstabilität an der Vorder- und Hinterachse hervorgerufen – entsteht ein Wankmoment am Wankhebelarm h' (vgl. Solmaz, 2008, S. 4696).

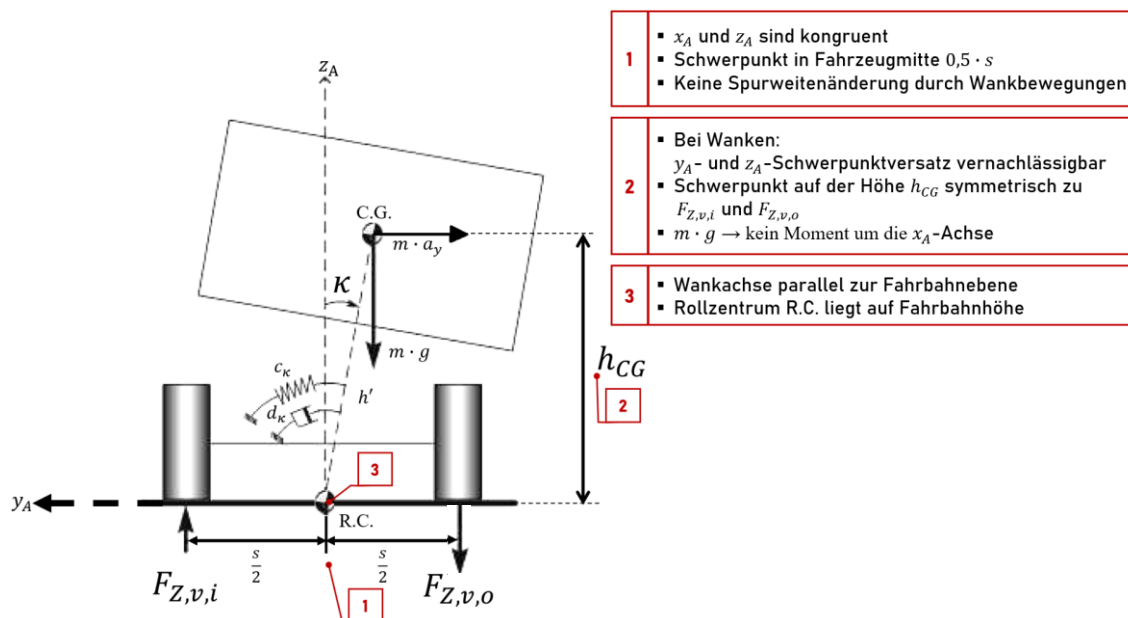


Abb. 25: Annahmen und Vereinfachungen des Zweispurmodells in der y_A - z_A -Ebene [Eigene Darstellung] (in Anlehnung an Solmaz, 2008, S. 4696 und Mitschke, 2014, S. 823)

Für die Radlastberechnung an jedem Reifen ergibt sich aus den statischen und dynamischen Radlasten (vgl. Mitschke, 2014, S. 810):

$$F_{z,v/h,i} = \frac{m \cdot g}{2} \cdot \frac{l_{h/v}}{l} - \Delta F_{z,v/h} \quad (17)$$

$$F_{z,v/h,o} = \frac{m \cdot g}{2} \cdot \frac{l_{h/v}}{l} + \Delta F_{z,v/h} \quad (18)$$

Für die querbeschleunigungsabhängige Radlast gilt (vgl. Mitschke, 2014, S. 809f., 825 und 832 und Schindler, 2019, S. 148):

$$\Delta F_{z,v} = \left(\frac{c_{k,v}}{c_{k,v} + c_{k,h} - m \cdot g \cdot h'} \cdot \frac{h'}{s_v} + \frac{l_h}{l} \cdot \frac{h_v}{s_v} \right) \cdot m_A \cdot a_y \quad (19)$$

$$\Delta F_{z,h} = \left(\frac{c_{k,h}}{c_{k,v} + c_{k,h} - m \cdot g \cdot h'} \cdot \frac{h'}{s_h} + \frac{l_v}{l} \cdot \frac{h_h}{s_h} \right) \cdot m_A \cdot a_y \quad (20)$$

$s_{v/h}$: Spurweite an der Vorder-/ Hinterachse

m_A : Aufbaumasse, die sich im Schwerpunkt des Fahrzeugs befindet

a_y : Querbeschleunigung

Es gilt für die Wanksteifigkeit $c_{k,v/h}$ an der Vorder-/ Hinterachse (vgl. Mitschke, 2014, S. 542 und 826 und Schramm, 2018, S. 219):

$$c_{k,v/h} = \left(\frac{s_{c,v/h}^2}{2} \cdot c_{Feder,v/h} + c_{Stabi,v/h} \right) \quad (21)$$

$s_{c,v/h}$: Federspurweite an der Vorder-/ Hinterachse

$c_{Feder,v/h}$: Federsteifigkeit an der Vorder-/ Hinterachse

$c_{Stabi,v/h}$: Stabilisatorsteifigkeit an der Vorder-/ Hinterachse

Eine für Aussagen über die Fahrstabilität wichtige Größe von Kraftfahrzeugen ist das Fahrverhalten unter Einfluss der maximalen Zentripetalbeschleunigung (Querbeschleunigung) im Zweispurmodell. Hohe Querkräfte können ohne Fremdeinwirkung zum Abheben eines Rades führen (vgl. Abbildung 25). Die Querbeschleunigung der Kippgrenze $a_{y,v/h,kipp}$ entsteht bei der Überschreitung der seitlichen Kraftschlussgrenze, wobei die Haftung eines kurveninneren Reifens zur Fahrbahn nicht mehr vorhanden ist. Diese berechnet sich aus der Auflösung der Gleichungen (17), (19) und (20) zu (vgl. Mitschke, 2014, S. 816 – 818):

$$a_{y,v/h,kipp} = \frac{\frac{m \cdot g}{2} \cdot \frac{l_{h/v}}{l}}{\left(\frac{c_{\kappa,v/h}}{c_{\kappa,v} + c_{\kappa,h}} \cdot \frac{h'}{s_{v/h}} + \frac{l_{h/v}}{l} \cdot \frac{h_{v/h}}{s_{v/h}} \right) \cdot m_A} \quad (22)$$

Die Aufbaufliehkraft um die Wankachse am Hebelarm h' berechnet sich aus (vgl. Schramm, 2018, S. 248):

$$h' = h_{CG} - \frac{h_v \cdot l_h + h_h \cdot l_v}{l} \quad (23)$$

h_{CG} : Höhe des Fahrzeugschwerpunkts

$h_{v/h}$: Höhe des Wankpols an der Vorder-/ Hinterachse

Die Summe der Momente um die Wankachse ergibt in Abhängigkeit von der Querbeseleunigung den Wankwinkel (vgl. Solmaz, 2008, S. 4696):

$$\kappa = \frac{m_A \cdot a_y \cdot h' - \frac{d_{\kappa,v} + d_{\kappa,h}}{2} \cdot \dot{\kappa} - (J_x + m_A \cdot h^2) \cdot \ddot{\kappa}}{\frac{c_{\kappa,v} + c_{\kappa,h}}{2} - m \cdot g \cdot h} \quad (24)$$

J_x : Trägheitsmoment um die durch den Schwerpunkt gehende x_A -Achse

Die Wankdämpfung errechnet sich aus (vgl. Mitschke, 2014, S. 542):

$$d_{\kappa,v/h} = \frac{D \cdot \sqrt{2 \cdot c_{\kappa,v/h} \cdot \left(\frac{s_{c,v/h}}{s}\right)^2 \cdot \frac{J_x}{\left(\frac{s_{v/h}}{2}\right)^2}}{\left(\frac{s_{d,v/h}}{s}\right)^2} \quad (25)$$

D : Dämpfungsgrad

$s_{d,v/h}$: Dämpfungsspurweite an der Vorder-/ Hinterachse

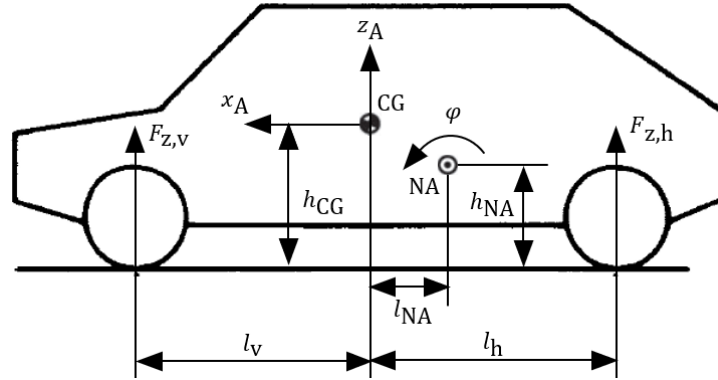


Abb. 26: Annahmen und Vereinfachungen des Zweispurmodells auf der x_A - z_A -Ebene [Veränderte Darstellung] (vgl. Halfmann, 2003, S. 45)

Der Nickwinkel wird mit folgender Formel berechnet (vgl. Halfmann, 2003, S. 46 und Abbildung 26):

$$\varphi = \iint \frac{(F_{z,h,o} + F_{z,h,i}) \cdot (l_h - l_{NA}) - (F_{z,v,o} + F_{z,v,i}) \cdot (l_v + l_{NA}) - m \cdot a_x \cdot (h_{CG} - h_{NA}) + m_A \cdot g \cdot l_{NA}}{J_y} dt dt \quad (26)$$

l_{NA} : Abstand des Fahrzeugschwerpunkts zum Nickschwerpunkt auf der y_A -Achse

h_{NA} : Abstand des Fahrzeugschwerpunkts zum Nickschwerpunkt auf der z_A -Achse

a_x : laterale (Brems-) Beschleunigung

J_y : Trägheitsmoment um die durch den Schwerpunkt gehende y_A -Achse

Fahrstreifenrelative Dynamik der Fahrzeugmodelle

Schwimm- und Gierwinkel beschreiben die Bewegungsgleichungen des (nicht-)linearen Einspurmodells sowie des nichtlinearen Zweispurmodells. Zur Überführung des fahrzeugfesten in das ortsfeste Koordinatensystem ist eine Erweiterung der Gleichungen erforderlich – nur so lässt sich eine Fahrzeugbewegung auf der Straße nachbilden. Die Ausgangslagen x_0 und y_0 des Ego-Fahrzeugs können mit der Multiplikation der Fahrzeuggeschwindigkeit berechnet werden (vgl. Mitschke, 2014, S. 619):

$$x = \int v \cdot \cos(\beta + \psi) dt + x_0 \quad (27)$$

$$y = \int v \cdot \sin(\beta + \psi) dt + y_0 \quad (28)$$

4 Modellbildung für ein Gesamtfahrzeug

Im nachfolgendem Unterkapitel wird aus den bestehenden Teilmodellkombinationen vor der API-Programmierung das geeignete Modell selektiert, um ein hochdynamisches Ausweichmanöver auszuführen. Bei der Auswahl steht stets im Fokus, ein reales Gesamtfahrzeugmodell zu erstellen, dessen Zusammenstellung der Teilsystemen übersichtlich wie möglich ist. Eine Grundlage, auf deren Basis die Anforderungen an die Kollisionsvermeidung mit selbstbestimmten Trajektorienplanungen zu erfüllen sind, soll nachvollziehbar sein. Hierfür wird im Nachfolgenden das passende Gesamtfahrzeugmodell mit zuvor definierten Kriterien und Vereinfachungen bestimmt und anschließend in die Entwicklungsumgebung implementiert.

4.1 Auswahl eines geeigneten Gesamtfahrzeugmodells

Aus den zuvor vorgestellten Teilmodellen werden nun drei Gesamtfahrzeugmodelle erstellt und anschließend ein geeignetes Gesamtfahrzeugmodell ausgewählt. Das bevorzugte Modell hat das Ziel, ein hochdynamisches Ausweichmanöver in unfallkritischen Situationen auszuführen. In jedem Modell ist das Lenkungsmodell identisch, wohingegen das Reifen- und Fahrzeugmodell variieren. Tabelle 3 bietet einen Überblick über die verschiedenen Modelle A – C an:

Tab. 3: Übersicht der modellierten Gesamtfahrzeugmodelle

	Modell A	Modell B	Modell C
Lenkungsmodell	PT1-Glied	PT1-Glied	PT1-Glied
Reifenmodell	Lin. Reifenmodell	MF (Pacejka)	MF (Pacejka)
Fahrzeugmodell	Lin. ESM	Nichtlin. ESM	Nichtlin. ZSM

Dabei bietet sich die Software CarMaker aufgrund der realitätsgetreuen Aktions- und Reaktionsvisualisierung von Fahrmanövern für Untersuchungen von vollparametrisierten und lernfähigen Gesamtfahrzeugmodellen an (vgl. CarMaker, 2021). Die Verifizierung der drei Modelle ist deswegen mit dem Simulationsprogramm CarMaker vorzunehmen, da es im Gegensatz zu anderen Simulationsprogrammen softwareintegrierte fahrdynamische Grenzbereiche besitzt. Mit den Modellen A – C sind Lenkradwinkelverläufe δ_H sowie Fahrzeuggeschwindigkeitsverläufe v in Form einer Steuerung des Open-Loop-Tests durchzuführen (vgl. Abbildung 27), damit der Graphenverlauf mit dem Referenzmodell aus CarMaker vergleichbar ist. Die anschließende Gegenüberstellung des Einzelsinus-Tests in MATLAB/Simulink mit den Datensätzen aus dem Referenzsoftwareprogramm

CarMaker ermöglicht es, die Plausibilität der fahrdynamischen Kurvenverläufe zu gewährleisten. Daraus ist abzuleiten, dass die selbstprogrammierte Befehlskette des Single Sine Tests aus ISO 7401 bei der Überprüfung der Plausibilität der fahrdynamischen Größen ein vergleichbares Kurvenverhalten anzeigt.

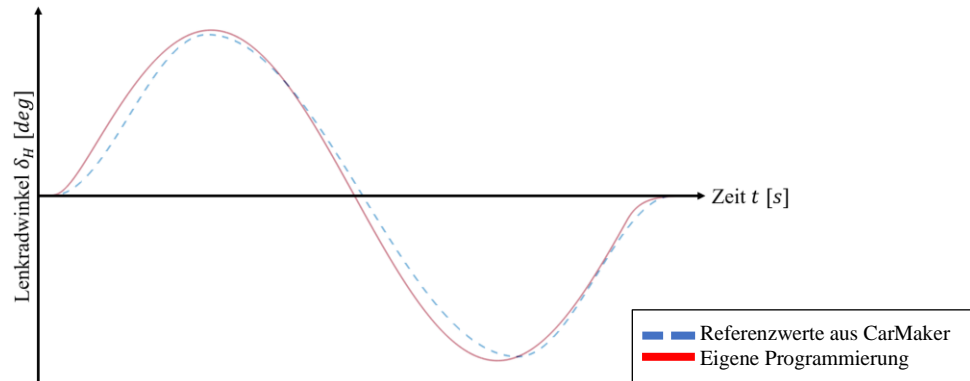


Abb. 27: Qualitativer Lenkradwinkelverlauf eines Single Sine Tests
[Eigene Darstellung] (vgl. A.3.1 Fahrdynamische Größen des Einzelsinus-Tests (ISO 7401))

Das Lenkungsmodell gibt die fahrdynamischen Größen für das Gesamtfahrzeugmodell an, das als Eingangsgröße für die Lenkeingabe dient. Die Eingabe vom Startpunkt wirkt sich auf das Fahrzeugmodell aus, indem sie bestimmt, in welche Richtung die Trajektorie verläuft. Als qualitative Vergleichsgröße dienen daher die Graphenverläufe des Schwimmwinkels β (vgl. Abbildung 28 links) und die Giergeschwindigkeit $\dot{\psi}$ (vgl. Abbildung 28 rechts). In den nachfolgenden Abbildungen ist ersichtlich, dass der dunkelblaue Referenzgraph allen drei Graphen der Modelle A – C ähnelt.

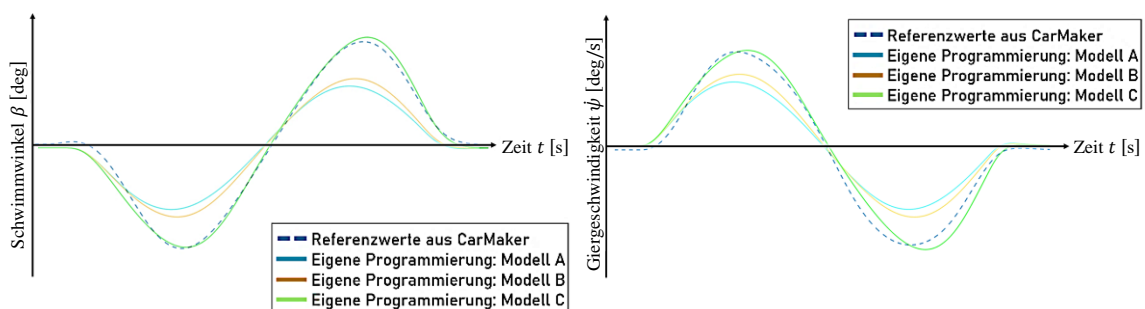


Abb. 28: Qualitative Ergebnisse des Einzelsinus: Schwimmwinkelverlauf (links) und Giergeschwindigkeitsverlauf (rechts) zur Überprüfung der Plausibilität der MiL-Simulation
[Eigene Darstellung] (vgl. A.3.1 Fahrdynamische Größen des Einzelsinus-Tests (ISO 7401))

Die Analogie der Kurvenverläufe visualisieren, dass das eigene MATLAB/Simulink-Programm mit dem Referenzmodell korrespondieren (vgl. A.4.3 MiL-Programmierung der Modelle A – C). Die Implementierung in die Software-in-the-Loop auf den Entwicklungsumgebungen Geany und Qt Creator kann somit umgesetzt werden, da das Model-in-the-Loop abgeschlossen ist (vgl. Abbildung 11).

4.1.1 Gesamtfahrzeugmodell für die Fahrzeugführung

Die kollisionsfreie Fahrzeugführung beinhaltet ein Ausweich-Fahrzeugmanöver, dass situationsabhängig einen passenden Pfad folgt. Eine tangentielle Fahrbahnorientierung sowie die komplette Belegung der Fahrbahnstreifenbreite werden bei der Betrachtung aller Fahrzeuge angenommen (vgl. Abbildung 29). Zusätzlich ist das Hindernis auf der Fahrbahn über eine Erweiterung des Programms mit einem Zustandsautomaten realisierbar. Den Zielzustand bestimmt der Fahrer durch eine Befehlseingabe in den Zustandsautomaten, sodass die Fahrzeugführung autonom eingeleitet wird (vgl. Abbildung 13).

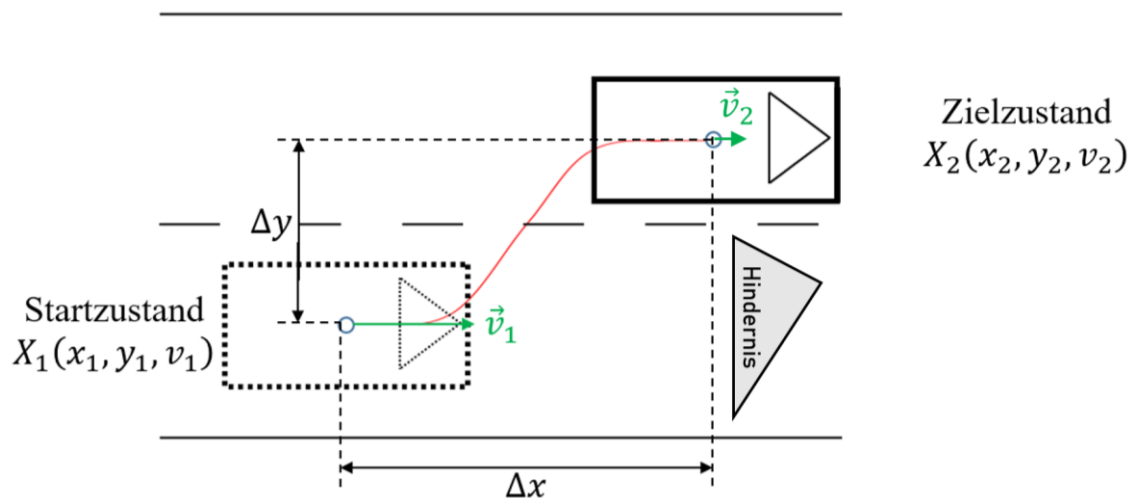


Abb. 29: Abstraktion eines Ausweichmanövers mit Start- und Zielzustand [Eigene Darstellung]

Sobald ein Hindernis auf der Simulationsfahrbahn ersichtlich ist und ein Bremsmanöver die bevorstehende Kollision nicht vermeidet, führt das FAS den Ausweichbefehl zur Kollisionsvermeidung aus. Der zuvor bestimmte und nun zu erreichende Zielzustand ist mit den entsprechenden globalen Differenzabständen Δx und Δy definiert. Die Abbildung 29 zeigt, wie das Ego-Fahrzeug vom Startzustand aus der rot gekennzeichneten kollisionsfreien und befahrbaren Trajektorie folgt. Bei der Ausführung einer idealen Kollisionsvermeidungstrajektorie ist entscheidend, dass die vom Gesamtfahrzeug abhängige Beschleunigungsgrenze sowie die Fahrzeugstabilität berücksichtigt werden.

Die Zielsetzung, realitätsnahe Simulationen mit Plausibilitätsprüfung zu verifizieren, setzt eine Rekonstruktion autonomer, hochdynamischer Ausweichmanöver voraus. Bei Kurvenfahrten mit zu hoher Querbeschleunigung z. B. hebt im Extremfall ein Rad ab und es kommt zu einem Überschlag des Fahrzeugs. Sofern die dynamische Radlast (vgl. Gl. (17)) größer als die statische Radlast (vgl. Gleichungen (19) und (20)) ist, wird die Kippgrenze überschritten. Die ISO 3888-1 und 3888-2 beschreiben Fälle, bei denen diese fahrkritischen Manöver analysiert und ausgewertet werden. Die Querbeschleunigungsgrenze während eines hochdynamischen Ausweichmanövers gibt Aufschluss darüber, ob die Gesamtfahrzeugmodelle A – C die maximalen Querbeschleunigungsgrenzen berücksichtigen. Daher sind Berechnungen der Kippgrenzen (vgl. Gl. (22)) und des Wankens (vgl. Gl. (24)) je nach Lenkradwinkelvorgabe der drei Gesamtfahrzeugmodelle geschwindigkeitsunabhängig zu ermitteln.

Tab. 4: Qualitative Bewertung der Modelle A – C nach fahrsicherheitsrelevanten Kriterien

Modell	Querbeschleunigungsgrenzen	Kriterien		Bewertungsergebnisse der Modelle
		Kippgrenze	Wanken	
A	$ a_{y,max} \leq 4 \frac{m}{s^2}$	-	-	-
B	$ a_{y,max} \leq a_{y,v/h, kipp}$	+	-	○
C	$ a_{y,max} \leq a_{y,v/h,kipp}$	+	+	+

Lineare Reifenmodelle sind wegen der konstanten Schräglauftiefigkeiten c_α auf $4 \frac{m}{s^2}$ begrenzt, was bei nichtlinearen Reifenmodellen mit fahrzeugdatenspezifischen Kippgrenzen $a_{y,v/h, kipp}$ nicht der Fall ist. Im direkten Vergleich zwischen Gesamtfahrzeugmodell A und B fällt in der Tabelle 4 auf, dass beide Modelle das Wanken nicht berücksichtigen (vgl. Tabelle 4). Das hier verwendete Einspurmodell ist wie ein Fahrradmodell, das eine Wankbewegung nicht ausführen kann (vgl. Unterkapitel 3.3.3). Obwohl das Modell B eine größere Querbeschleunigungsgrenze als $4 \frac{m}{s^2}$ erreicht, wird das verwendete nichtlineare Reifenmodell $|a_{y,max}| \leq a_{y,v/h,kipp}$ in Kombination mit einem Zweispurmodell gewählt. Bei der Bewertung der Gesamtfahrzeugmodelle fällt folglich Modell C am positivsten aus, da das Wanken als fahrdynamische Größe im zweiachsigen Fahrzeugmodell berücksichtigt wird und hohe Querbeschleunigungen bis zur Kippgrenze erreichbar sind.

4.1.2 Modellierung hochdynamischer Ausweichmanöver

Brems- und Lenkmanöver unterliegen aufgrund der Längs- und Querdynamik von Fahrzeugen physikalischen Grenzen und sind je nach Verkehrssituation begrenzt. Aus der nachfolgenden Übersicht geht hervor, dass in der vorliegenden Arbeit ein Ausweichmanöver, das eine Kombination aus Bremsen und Lenken enthält, für ein Antikollisionssystem zielführend ist (vgl. Abbildung 30). Vor allem für die SiL-Simulation autonomer Ausweichmanöver sind Daten in Abhängigkeit von der Trajektorie, der vorherrschenden Kräften und der fahrdynamischen Limitierungen des Gesamtfahrzeugmodells zu eruiieren.

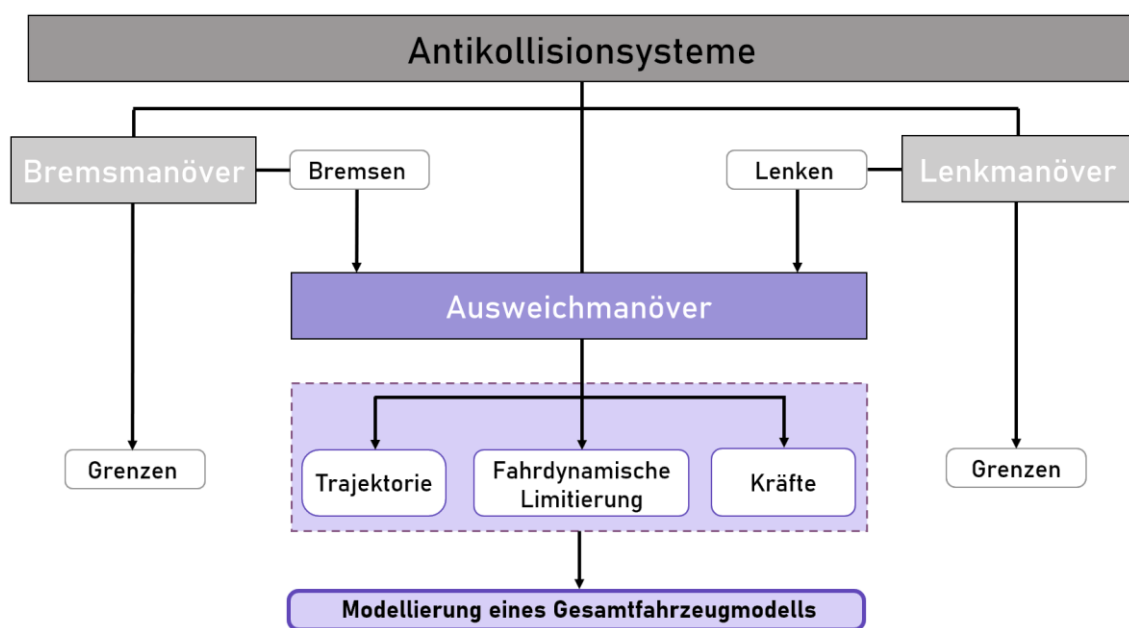


Abb. 30: Darstellung relevanter Elemente zur Kollisionsvermeidung [Eigene Darstellung]

Sobald die Umfeldsensoren ein unfallverursachendes Hindernis auf der Fahrbahn erkennen, wird ein Ausweichmanöver, bestehend aus Brems- und Lenkvorgang, eingeleitet. Beim Auftreten einer Vollbremsung ist die Grenze der Längsdynamik, beim Lenken die Annäherung an die maximale Querdynamik zu beachten. Eine fest definierte Grenze beim Lenkmanöver ist nicht $4 \frac{m}{s^2}$, sondern die Kippgrenze (vgl. Tabelle 4). Hier ist der lineare Bereich des Lenkmanövers für das hochdynamische Ausweichmanöver irrelevant und fordert einen geeigneten Lenkradwinkelverlauf. Da das zuvor betrachtete Open-Loop-Manöver Single Sine nicht für die Anwendung im hochdynamischen Bereich anwendbar ist (vgl. Unterkapitel 3.2.2), muss ein alternativer Lenkradwinkelverlauf entwickelt werden.

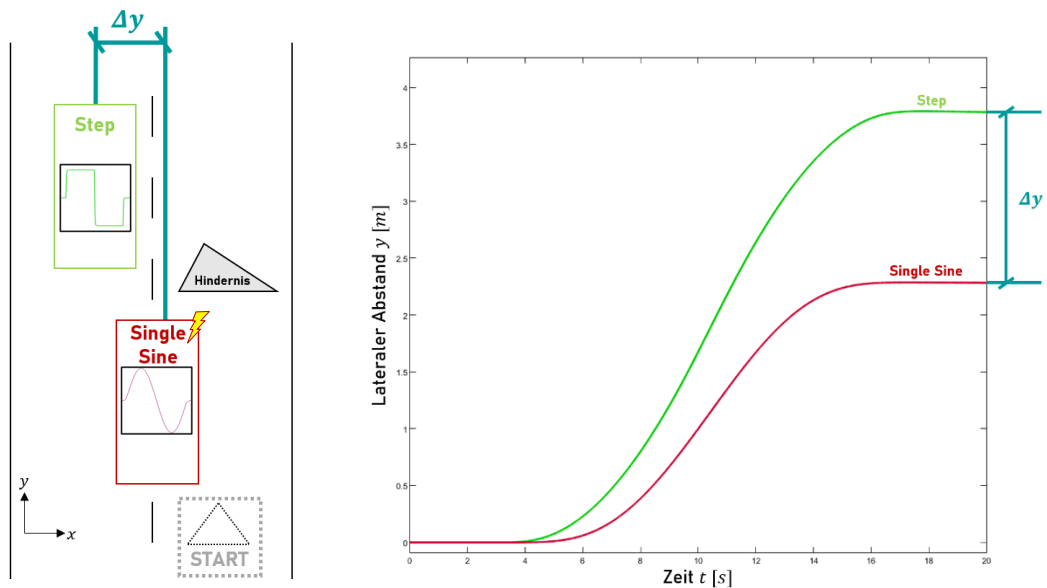


Abb. 31: Vergleich zweier verschiedener Lenkradwinkelverläufe (links) und die daraus resultierenden Positionen im Weltkoordinatensystem (rechts) [Eigene Darstellung]

Der Einzelsinus-Test wird mit der angewandten Brute-Force-Methode (Anwendung der Step-Funktion, vgl. Abbildung 31 links und A.1.3 Weitere Definitionen) verglichen. Daraus folgt, dass die höchste Ableitung unter Berücksichtigung realitätsnaher Lenkradwinkelverläufe eine Veränderung des Kurvenverlaufs voraussetzt. Wie in Abbildung 31 links dargestellt, unterscheidet sich der laterale Abstand Δy vom Einzelsinus und vom vordefinierten doppelten Lenkradwinkelsprung (vgl. Step-Funktion in Abbildung 31 links) in Form eines Rechtecksignals um den Wert Δy (vgl. Trzesniowski, 2014, S. 466). Bei einem Lenkvorgang hängen die Frequenz und die Amplitude des Rechtecksignals im jeweiligen Anwendungsfall je von den definierten Endpositionen Δx und Δy des Fahrzeugs ab. Der erforderliche Lenkeingriff sollte nicht nur das Auslenken des Fahrzeugs aus seiner ursprünglichen Fahrtrichtung, sondern das Einlenken parallel zur gewünschten Richtung des jeweiligen Straßenverlaufs umfassen. Den daraus resultierenden qualitativen SOLL-Lenkradwinkelverlauf für ein kollisionsvermeidendes Manöver zeigt Abbildung 31 rechts. Die Leistungsfähigkeit der Fahrzeug-Lenkaktoren unterliegt, wie im realen Einsatz, physikalischen Grenzen (vgl. Unterkapitel 3.3.1 und Bootz, 2004, S. 43f.), weswegen der gewünschte Lenkradwinkelverlauf nur verzögert aufgebaut wird – sein Maximalwert ist dabei beschränkt. Diese fahrdynamischen Limitierungen sind in der Trajektorie abgebildet. Somit ist in der gleichen Zeitspanne ein hochdynamisches, kollisionsfreies Ausweichmanöver mit dem erwünschten lateralen Abstand Δy möglich, während der sinusförmige Lenkradwinkelverlauf nach ISO 7401 eine Kollision nicht verhindert.

4.1.3 Fahrstreifenwechsel in drei Bereichen

Alle zu untersuchenden Ausweichmanöver beinhalten eine Längs- und Quergeschwindigkeit, die das Fahrerassistenzsystem beim Bremsen und Lenken auslöst. Die bei den Eingriffen entstehende Trajektorie setzt sich aus den Schwerpunkten des Fahrzeugs mit den entsprechenden x - y -Koordinaten zusammen. Daher lässt sich der Verlauf des Fahrmanövers in drei Bereiche untergliedern:

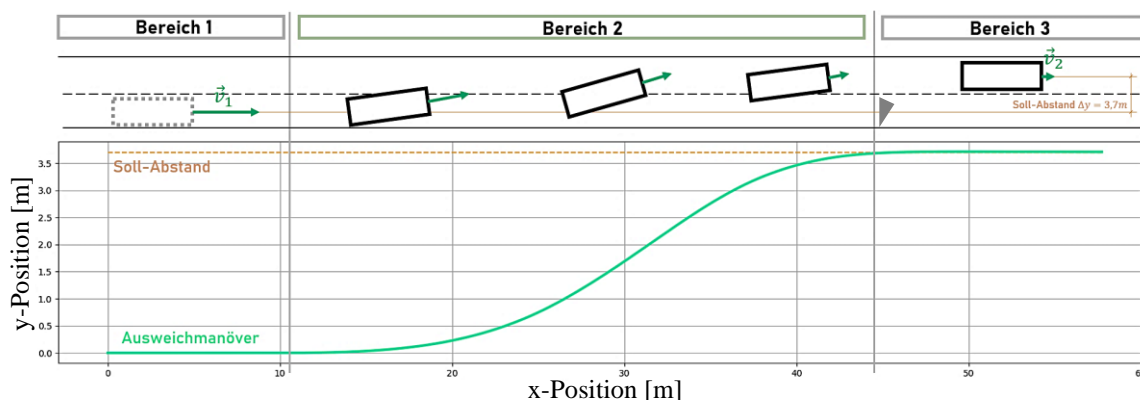





Abb. 32: Unterteilung eines Ausweichmanövers in drei Bereiche [Eigene Darstellung]

Im ersten Bereich befindet sich das Fahrzeug in Längsrichtung und führt keinen Fahrspurwechsel aus, da das Fahrerassistenzsystem keine unfallverursachende Gefahrensituation erkennt. Die Normalfahrt auf dem Ausgangsfahrstreifen wird nach dem Bremseneingriff unterbrochen, sobald ein Hindernis auf der Fahrbahn erkannt wird. Wenn ein Ausweichmanöver zwingend erforderlich ist, leitet das FAS einen zwischen den ersten beiden Bereichen ein. Dieser Eingriff in die Längs- und Querbewegung im zweiten Bereich meidet die Kollision mit dem Hindernis und führt einen Fahrspurwechsel aus. Die Kippgrenze begrenzt hierbei die maximale Querbewegung, um das Abheben eines kurveninneren Rades zu verhindern. Bevor das Ausweichmanöver beendet wird, überprüft das FAS die Kollisionsfreiheit und stellt sicher, dass eine Normalfahrt den Umständen entsprechend realisierbar ist. Ob ein Gegenlenkmanöver benötigt wird, entscheidet das System, das gegebenenfalls einen erneuten einfachen Fahrspurwechsel zurück zur ursprünglichen Fahrspur durchführen kann. Da nach Abschluss des Ausweichmanövers eine normale Fahrsituation vorherrscht und das Fahrerassistenzsystem nicht mehr eingreift, kann die Fahraufgabe vom Fahrer bzw. vom ACC-Assistenzsystem übernommen werden.

4.2 Referenzdatengenerierung für den Modellansatz

Numerische Berechnungen und Simulationen werden auf Grundlage von fahrerspezifischen Daten, die als Eingangsgröße notwendig sind, errechnet. Die Implementierung der Lösungsalgorithmen auf Parameterannahmen und -berechnungen werden mithilfe Parameterapproximation für die Lookup-Table durch Variationsrechnung nach der Brute-Force-Methode (vgl. A.1.3 Weitere Definitionen) iterativ gelöst. Die fehlenden Datensätze sind in der SiL-Simulation für das hier eigenständig entwickelte Programm *PPADD* nicht validierbar, weshalb alternative Möglichkeiten zur Erlangung der Werte erforderlich sind. Tabelle 5 listet die fehlenden Fahrzeugparameter auf, die in den jeweiligen Teilmodellen nicht vorhanden sind:

Tab. 5: Fehlende Parameter für die Erstellung eines Gesamtfahrzeugmodells

Lenkungsmodell	Reifenmodell	Fahrzeugmodell
 <ul style="list-style-type: none"> ▪ Lenkungsunterstützung ▪ Lenkungsgetriebe-übersetzung ▪ Torsionssteifigkeit – Drehstab ▪ Reifennachlauf ▪ Lenkungsverhalten (Verzögerungsglied 1. Ordnung) 	 <ul style="list-style-type: none"> ▪ Schräglaufsteifigkeit der VA/ HA ▪ Koeffizienten der Magic Formula Tyre-Modells nach Pacejka 	 <ul style="list-style-type: none"> ▪ Länge des Fahrzeuggradstands ▪ Abstand VA/ HA zum Schwerpunkt ▪ Spurweite der VA/ HA ▪ Höhe des Massenschwerpunktes zur Achsebene ▪ Massenträgheitsmoment um die $x_A/y_A/z_A$-Achse

PPADD benötigt spezifische Parameter, um die erwünschten Trajektorien zu simulieren. Einige Daten liefern der Gesamtsystemsimulator CarMaker, das numerische Berechnungsprogramm MATLAB/Simulink sowie Fachliteratur. Fahrzeugcharakteristische Daten, wie bspw. die Bremsbeschleunigung, liegen nicht vor und können ohne ein Versuchsfahrzeug ebenfalls nicht ermittelt werden. Daher sind alternativ naheliegende Parameter aus eigener Erfahrung mit der Brute-Force-Methode auszutesten und anschließend zu evaluieren. Die Kollisionsvermeidungsmaßnahmen erfüllen hierbei Sicherheitsanforderungen und keine Komfortaspekte. Aufgrund nicht vorhandener Datensätze bei einem hochdynamischen Fahrspurwechsel wird die Fahrzeugergonomie für Insassen somit nicht näher berücksichtigt. Die folgenden zwei Unterkapitel beschäftigen sich mit eben dieser Vorgehensweise zur Generierung und Ermittlung von Referenzparametern für die Modellbildung und das Konzept zum Ausführen eines hochdynamischen Ausweichmanövers.

4.2.1 Erstellung der Referenzfahrzeugparameter

Im Referenztool CarMaker sind unterschiedliche Fahrzeugtypen mit ihrer Masse und dem Massenträgheitsmoment auslesbar. Bis auf den Fahrzeugschwerpunkt sind die zu ergänzenden Angaben als Werte bzw. Wertebereiche aus Fachliteratur zu entnehmen (vgl. Pfeffer, 2011, S. 90, Heißing, 2013, S. 22, Pauwelussen, 2015, S. 285f. und Serarslan, 2015, S. 85). Die Ergänzung der Eingangswerte lässt den Vergleich der eigenen Programmierung mit dem Referenztool zu, sodass die zu verwendenden Daten treffend gewählt werden. In der nachfolgenden Tabelle sind diese Werte aufgelistet:

Tab. 6: Verwendete Fahrzeugparameter (vgl. A.4.3 MiL-Programmierung der Modelle A – C und A.5.1 PDD)

Parameter	Symbol	Wert
Antriebsart: Frontantrieb	δ_h	0 deg
Fahrzeugesamtmasse	m	2120 kg
Länge des Fahrzeugradstands	l	2,819 m
Abstand des Fahrzeugschwerpunktes zur Vorderachse	l_v	1,3 m
Abstand des Fahrzeugschwerpunktes zur Hinterachse	l_h	1,519 m
Höhe des Schwerpunkts	h_{CG}	0,63 m
Schräglaufsteifigkeit der Vorderachse (für das lineare Reifenmodell)	c_v	$7,21 \cdot 10^4 \frac{N}{rad}$
Schräglaufsteifigkeit der Hinterachse (für das lineare Reifenmodell)	c_h	$6,18 \cdot 10^4 \frac{N}{rad}$
Massenträgheitsmoment um die x_A -Achse (Wankträgheitsmoment)	J_x	$764 \text{ kg} \cdot \text{m}^2$
Massenträgheitsmoment um die y_A -Achse (Nickträgheitsmoment)	J_y	$3663 \text{ kg} \cdot \text{m}^2$
Massenträgheitsmoment um die z_A -Achse (Gierträgheitsmoment)	J_z	$3862 \text{ kg} \cdot \text{m}^2$
Gesamtnachlauf	n_{ges}	0,06 m
Lenkungsgetriebeübersetzung	i_S	15
Lenkungsunterstützung	A_S	2
Spurweite – Vorderachse	s_v	1,5 m
Spurweite – Hinterachse	s_h	1,6 m

Für die Implementierung des nichtlinearen Reifenmodells sind die unbekanntenen Koeffizienten des MF ermittelbar, die weder in CarMaker noch der Fachliteratur zu entnehmen sind. Die Koeffizienten dienen zur Simulation des *PPADD*-Programms, da kein reales Testfahrzeug mit exakten Reifen- und Fahrzeugdaten sowie Messdatensätzen herangezogen werden. Sobald ein festgelegter Fahrzeugtyp im Programm hinzugefügt wird, müssen die Werte aus Tabelle 6 angepasst werden. Für die Erstbetrachtung der autonomen Ausweichmanöver ist eine realistische Näherung an ein Audi Q5 festzustellen. Auf dieser Basis wurde ein MATLAB/Simulink-Programm erstellt, das die fehlenden Parameter für das Programm des Gesamtfahrzeugmodells mit numerischer Berechnungssoftware ermittelt. Die Ermittlung der gesuchten Koeffizienten ist im nachfolgenden Schaubild ersichtlich und wird im Anschluss erklärt:

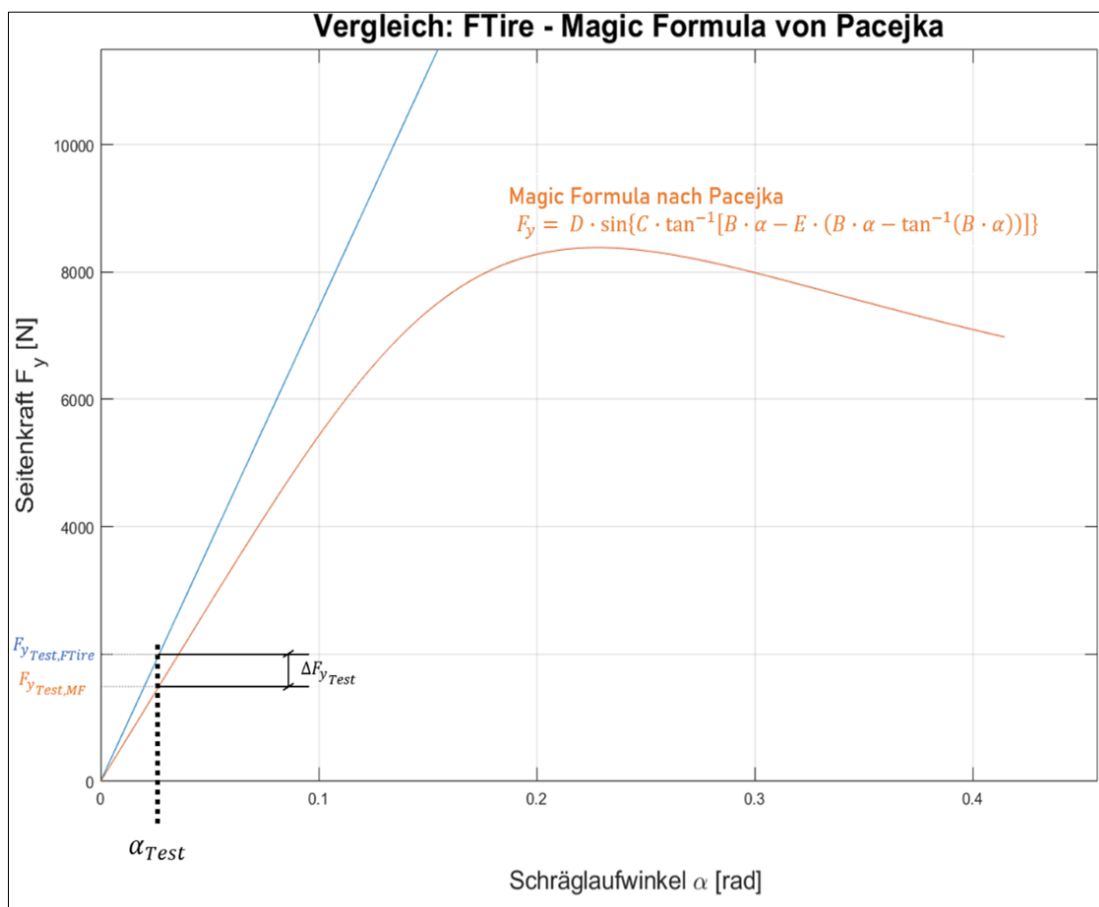
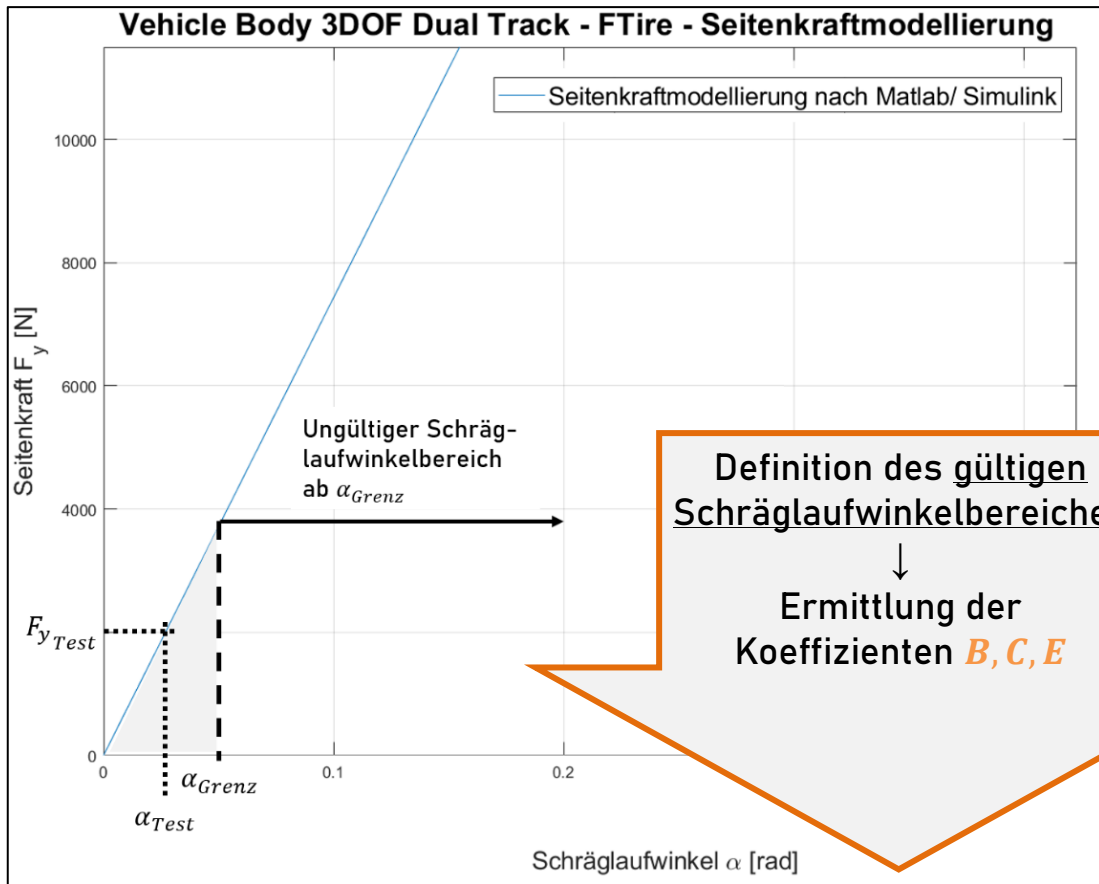


Abb. 33: Optimierungsverfahren zur Ermittlung der Koeffizienten der Magic Formula nach Pacejka [Eigene Darstellung] (vgl. A.4.2 Ermittlung der Koeffizienten der Magic Formula nach Pacejka)

Bei realen Versuchen ist der Verlauf des „Vehicle Body 3DOF Dual Track“-Graphen nicht linear, sondern hat ab dem Grenzschräglaufwinkel α_{Grenz} eine vom linearen in den degressiven Kurvenverlauf übergehende Entwicklung. Das in MATLAB/Simulink vorhandene „Vehicle Body 3DOF Dual Track“ ist zur Koeffizientenermittlung als Referenzmodell im linearen Bereich anwendbar (vgl. Abbildung 33). Hier gestrichelt hervorgehoben ist der Grenzschräglaufwinkel $\alpha_{\text{Grenz}} < 0,052$ rad, der die Grenzbeschleunigung $|\alpha_{y,\text{max}}| < 4 \frac{\text{m}}{\text{s}^2}$ festlegt. Alle Optimierungsverfahren, die auf Schräglaufwinkeln größer gleich α_{Grenz} basieren, können daher nicht für die weitere Optimierung verwendet werden. Das „Vehicle Body 3DOF Dual Track“ beinhaltet ein integriertes Reifenmodell, in dessen Codierung des Programms kein Eingriff möglich ist. An dem grauen Pfeil zwischen den Diagrammen lässt sich ablesen, dass Optimierungsverfahren mit einem Simulink-Block erforderlich sind. Der „Check Against Reference“-Block berechnet die Differenz $\Delta F_{y_{\text{Test}}}$ so lange, bis diese gegen null konvergiert und sich somit der Graph $F_{y_{\text{Test, MF}}}$ an den linearen Gültigkeitsbereich des $F_{y_{\text{Test, FTire}}}$ anschmiegt.

Im nächsten Schritt werden die Koeffizienten des Magic Formula bestimmt, damit eine Benutzung des nichtlinearen Reifenmodells in der SiL-Anwendung stattfindet. Der „Check Against Reference“-Block vergleicht zuerst die IST-Seitenkräfte von $F_{y_{\text{Test, MF}}}$ mit den SOLL-Seitenkräften von $F_{y_{\text{Test, FTire}}}$. Anschließend werden in der Optimierungsschleife die Koeffizienten B, C und E des MF mit heuristischen und darauffolgend gradientenbasierten Annäherungsverfahren ausgewählt, bis im letzten Iterationsschritt beide Kurvenamplituden und -verläufe annähernd identisch sind (vgl. MathWorks, 2015, S. 197 – 562). Hierfür nimmt die Software zuerst beliebig ausgewählte und daraufhin numerisch berechnete Zahlenwerte und setzt diese in die Magic Formula (s. Gl. (9)) ein. Der Vergleich der Zahlenwerte wird so lange in der Schleife ausgeführt, bis die Endbedingung $\Delta F_{y_{\text{Test}}} \approx 0 \pm 0,5$ erfüllt ist (vgl. Abbildung 34).

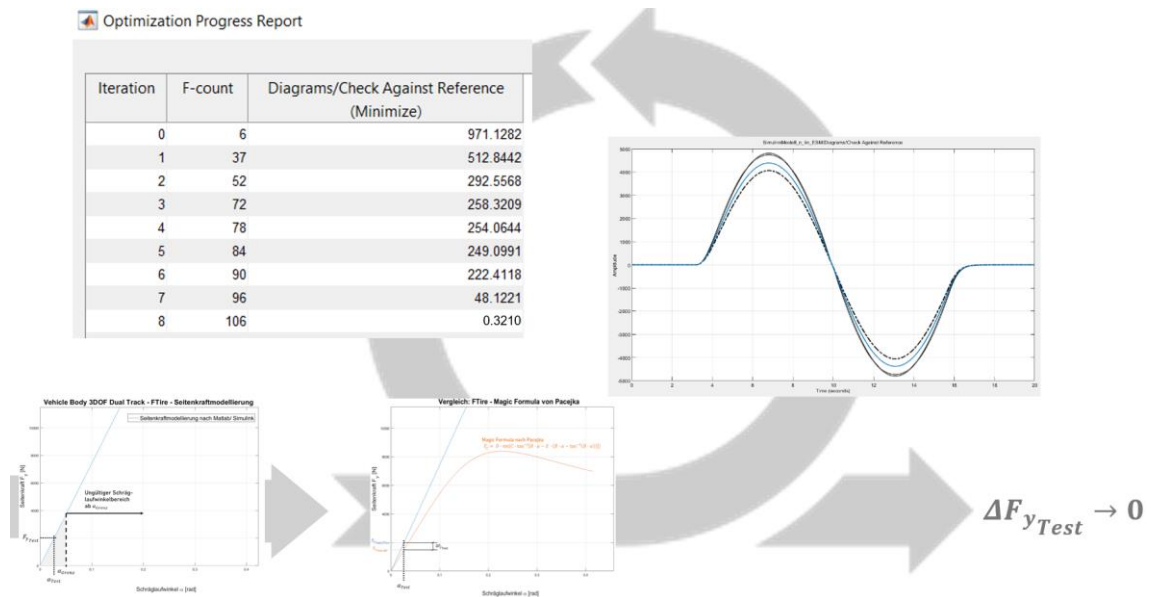


Abb. 34: Optimierungsverfahren zur Ermittlung der Koeffizienten der Magic Formula – Optimierungsschleifen mit der SOLL-Seitenkraftkurve und Anzahl der Schleifen [Eigene Darstellung] (vgl. Abbildung 33)

Das Ergebnis der durchgeführten 106 Optimierungsschleifen (vgl. Abbildung 34) ist, dass die Differenz des IST- zum SOLL-Zustand gegen null konvergiert und somit ein realitätsnaher Verlauf der Seitenkräfte für den nichtlinearen Bereich im Gesamtfahrzeugmodell annehmbar ist. Bei Abschluss des Optimierungsverfahrens ist die Auswahl der geeigneten Koeffizienten ablesbar und kann in die Entwicklungsumgebungen Geany (vgl. A.5 Python-Programmierung) und Qt Creator (vgl. A.6 C++-Programmierung) integriert werden.

4.2.2 Konzipieren einer Lookup-Table für Ausweichmanöver

Mithilfe der Referenzparameter und des ausgewählten Gesamtfahrzeugmodells ist eine Modellierung des vordefinierten Lenkradwinkelverlaufs (Rechtecksignal) realisierbar. Unter Angabe der Start- und Zielzustände $x_{1/2}$, $y_{1/2}$ und $v_{1/2}$ bzw. Δx , Δy und Δv kann eine Trajektorie generiert werden, die ein hochdynamisches Ausweichmanöver ausführt. Der Bremsvorgang des Ausweichmanövers strebt einen maximalen Energieabbau mit der Vollbremsung an, die den Abbau der Geschwindigkeit von $v_1 = 100 \frac{\text{km}}{\text{h}}$ auf v_2 realisiert (vgl. Unterkapitel 4.1.2). Die minimal erreichbare Fahrzeuggeschwindigkeit beträgt bei der Modellierung $v_2 = 1 \frac{\text{km}}{\text{h}}$, da sich die Geschwindigkeit im Nenner der Gleichungen (10) bzw. (15) befindet und für $v_2 = 0 \frac{\text{km}}{\text{h}}$ eine undefinierte Lösung resultiert. Dies berücksichtigend, verbleibt als einzige Stellgröße die Lenkradwinkeleingabe zur Realisierung des Ausweichmanövers nach links oder rechts. An dieser Stelle wird für das kollisionsvermeidende Manöver eine Steuerung des Lenkradwinkels nach Abbildung 35 verwendet.

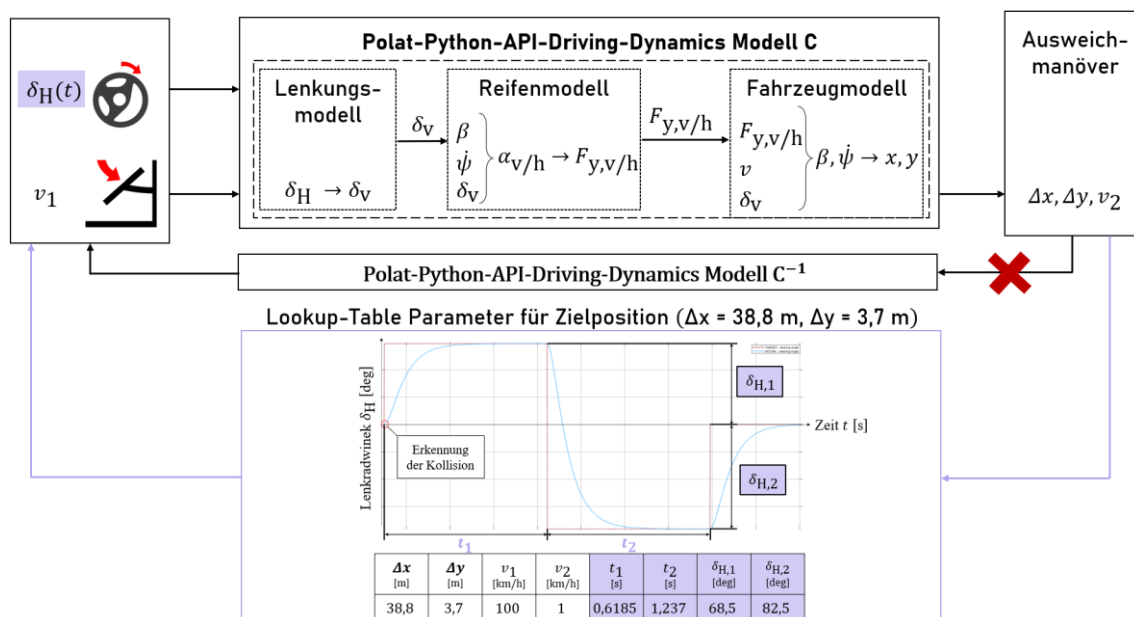


Abb. 35: Überblick über das Konzept für ein hochdynamisches Ausweichmanöver [Eigene Darstellung]

Wie dargestellt, ist eine analytische Rückwärtsrechnung wegen undefinierter mathematischer Inversion der Berechnungskette im ersten Rückwärtspfad als invertierte Matrix Polat-Python-API-Driving-Dynamics Modell C⁻¹ (vgl. Abbildung 35) nicht möglich. Obwohl der Verlauf der Trajektorie und die fahrdynamischen Limitierungen vorliegen, kann eine Rückwärtsrechnung aufgrund der Addition unbekannter Schwimm- und Gierwinkel in analytischer Form nicht durchgeführt werden (vgl. Gleichungen (27) und (28)).

Die Auflösung beider Gleichungen nach den gesuchten fahrdynamischen Größen β und ψ ergibt eine unendliche Lösungsmenge. Somit sind für verschiedene Kollisionsvermeidungssituationen Referenzdaten für das Lenkverhalten notwendig, das je nach Situation die gewünschte Trajektorie ausführen kann.

Alternativ sind Lenkradwinkelvorgaben mit dem violetten Rückwärtspfad „Lookup-Table“ (dt.: „Wertetabelle“) bestimmbar, die das analytische unlösbare Inversionsproblem des Gesamtfahrzeugmodells C umgeht. Eine Lookup-Table ist eine spezielle Datenstruktur, die vorberechnete Lenkradwinkel-Zeit-Kombinationen enthält und den Vorteil hat, dass die Rechenzeit im Anwendungsfall kurz ist. Bei den violett markierten Feldern handelt es sich um die vordefinierten Parameter, welche die Lookup-Table bei einem Schleifendurchlauf zur Verfügung stellt. Der Nachteil der Wertetabelle ist, dass sie vor der Anwendung erst zu erstellen ist und die Datensätze eingetragen sein müssen. Die Lookup-Table-Anwendung wird mit der Zielposition ($\Delta x = 38,8$ m, $\Delta y = 3,7$ m) im Schaubild dargestellt. Dieses Beispiel soll verdeutlichen, dass bei einer Anfangsgeschwindigkeit von $v_1 = 100 \frac{\text{km}}{\text{h}}$ eine Vollbremsung getätigt wird. Nach dem Bremsmanöver besitzt das Simulationsfahrzeug die Endgeschwindigkeit $v_2 = 1 \frac{\text{km}}{\text{h}}$, welche mit einem Lenkmanöver bei Erkennung einer Kollisionsgefahr einen doppelten Lenkradwinkelsprung ausführt. Wie das Diagramm in Abbildung 35 verdeutlicht, stehen die Lenkradwinkelamplituden $\delta_{H,1/2}$ sowie die Zeit $t_{1/2}$ in Abhängigkeit des longitudinalen und lateralen Versatzes. Somit ist die Lenkradwinkel-Zeit-Kombination mit einer Anfangsfahrgeschwindigkeit, ähnlich wie bei einem Open-Loop-Test, zu ermitteln, in die Wertetabelle einzutragen und im Anwendungsfall vorzugeben. Hierbei führt der doppelte Lenkradwirkungssprung zunächst einen positiven Sprung mit der Amplitude $\delta_{H,1} = 68,5$ deg bis $t_1 = 0,6185$ s aus (vgl. Abbildung 35, 2. Rückwärtspfad). Der negative Sprung folgt daraufhin mit $\delta_{H,2} = -82,5$ deg bis $t_2 = 1,237$ s, sodass mit dem Gesamtfahrzeugmodell C die gewünschte Zielkoordinate ($\Delta x = 38,8$ m, $\Delta y = 3,7$ m) erreicht wird. Die Bewertungskriterien basieren beim doppelten Lenkradwinkelsprung auf den Querbeschleunigungsreaktionen des Fahrzeugs im Zeitbereich. Die Fahrzeugreaktion auf das Rechtecksignal entspricht im Rahmen dieser Modellierung einem hochdynamischen Ausweichmanöver, wobei der An- und Abstieg der Fahrdynamikgrößen möglichst schnell erfolgen sollte.

Abhängig von der Lenkradwinkel-Zeit-Kombination der Lookup-Table variiert, wie in Abbildung 36 links dargestellt, die Querschleunigungskurve, welche die Kippgrenze nicht überschreiten darf. Der Δy -Abstand muss nach dem Planungskonzept zufolge in der Lage sein, Trajektorien für verschiedene laterale Abstände auf deutschen Straßen bereitzustellen, d. h. von $\Delta y_{\min} = 0,0$ m bis $\Delta y_{\max} = 14,5$ m (vgl. Lippold, 2006, S. 16 und Deutscher Bundestag, 2009, S. 12f.). Der Vergleich der rot gestrichelten Kippgrenze mit dem Δy_{\max} -Graphen zeigt, dass die Kippgrenze bei einem Ausweichmanöver mit einem Δy -Abstand von 14,5 m eingehalten wird. Bei einem einfachen Fahrspurwechsel Δy_{Spur} ist der Anstieg der Querschleunigung vergleichbar mit dem Δy_{\max} -Graphen, jedoch hängt hier das Nichtüberschreiten der Kippgrenze mit den Datensätzen der Lookup-Table zusammen. Durch eine Plausibilitätsprüfung mittels der zuvor genannten Vergleiche sowie mittels einer Kontrolle des reinen Bremsvorgangs mit $\Delta y_{\min} = 0,0$ m kann überprüft werden, ob die Lookup-Table die richtige Lenkradwinkel-Zeit-Kombinationen enthält.

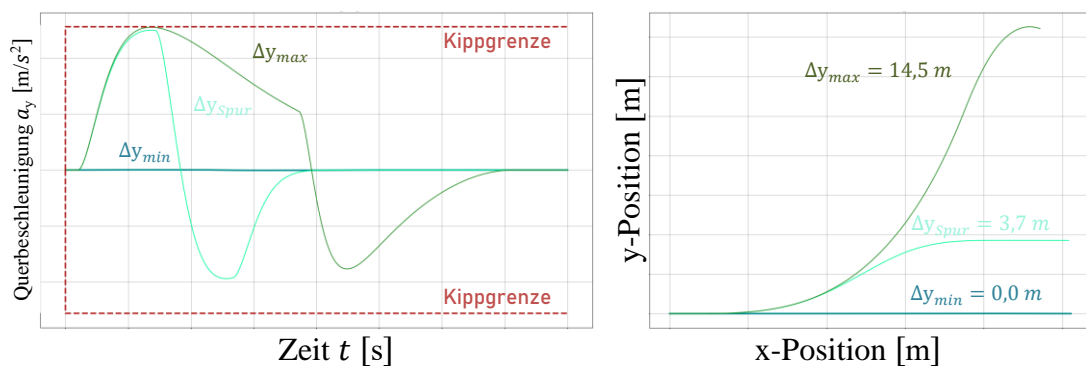


Abb. 36: Fahr-dynamische Ergebnisse aus drei verschiedenen Ausweichmanövern mit den lateralen Abständen $\Delta y_{\min} = 0,0$ m, $\Delta y_{\text{Spur}} = 3,7$ m und $\Delta y_{\max} = 14,5$ m: Querschleunigung (links) und x - y -Position im ortsfesten Koordinatensystem (rechts) [Eigene Darstellung]

4.3 Implementierung

Im Folgenden ist das Gesamtfahrzeugmodell C für die Trajektorienplanung und weitere Untersuchungen dargestellt, welches in die Entwicklungsumgebung Geany implementiert wurde (vgl. A.5 Python-Programmierung). Es werden drei Modelle näher betrachtet, die sich nicht im Lenkungsmodell, aber in der Kombination des Reifenmodells und Fahrzeugmodells unterscheiden. Modell A beinhaltet ein lineares Reifen- und Einspurmodell, wohingegen Modell B das nichtlineare enthält. Im Vergleich dazu gewährt Modell C als einziges Modell mit nichtlinearem Reifen- und Zweispurmodell eine detailliertere Beschreibung der Fahrdynamik. Eine Modellierung der Abfolge des Programms und die dahinterliegende Logik ist in einem Signalflussgraphen visualisiert und wird einer Plausibilitätsprüfung unterzogen. Ist die Prüfung der MiL erfolgreich, kann die Überführung in die Software als SiL realisiert werden (vgl. Abbildung 11). Der daraus resultierende Algorithmus für das Ausweichmanöver eines Gesamtfahrzeugmodells ist für den Einsatz in die Simulationssoftware CARLA zu übernehmen.

4.3.1 Aufbau des SiL-Programms

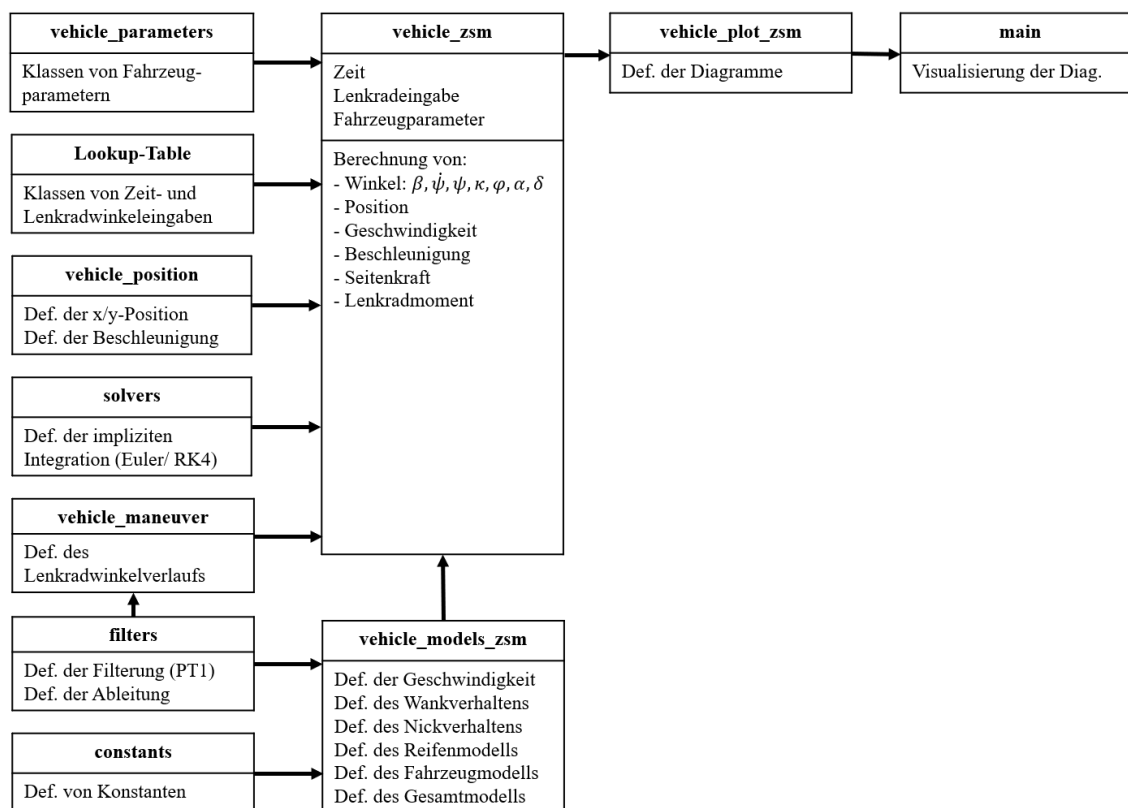


Abb. 37: Übersicht über die PDD-Softwaremodule und -funktionen [Eigene Darstellung] (vgl. A.5.1 PDD)

Im SiL werden verschiedene Programme zusammentragen, um einen Algorithmus für das autonome Ausweichmanöver mit dem Gesamtfahrzeugmodell C auszuführen. Aufgrund fehlender physikalischer Größen (vgl. Tabelle 6) sowie mathematischer und regelungstechnischer Zusammenhänge (z. B. PT1-Glied im Unterkapitel 3.3.1) sind mehrere Algorithmen für die Modellierung erforderlich. Die Softwaremodule, Funktionsweisen und Abfolgen der an dieser Stelle zum Einsatz kommenden *PDD*-Programme sind in Abbildung 37 schematisch dargestellt. Die Teilprogramme der *PDD*, z. B. **solvers**, definiert die Integration von Funktionen, wohingegen **filters** die Ableitung von fahrdynamischen Größen und die Glättung der Lenkradwinkelvorgaben übernimmt. Für die Anwendung der Modellierung sind die einzelnen Fahrzeugparameter in **vehicle_parameters** enthalten, um die Gleichungen (2) – (28) zu lösen. Die einzelnen Zeit- und Lenkradwinkeldatensätze sind im Modul **Lookup-Table** enthalten.

Die Berechnung der fahrdynamischen Gleichungen befindet sich in den Softwaremodulen **vehicle_maneuver**, **vehicle_models_zsm** und **vehicle_position**. Das Teilprogramm **vehicle_maneuver** benötigt die Daten aus der Lookup-Table, um den eigenständig generierten SOLL-Lenkradwinkelverlauf als doppelte Sprungantwort zu realisieren. Die Glättung mit **filters** als Verzögerungsglied erster Ordnung wandelt die Lenkung in einen realen Lenkradwinkelverlauf um. Lediglich **vehicle_models_zsm** benötigt für die Definition der einzelnen Gleichungen die Ableitung aus **filters** und Konstanten aus **constants**. Aus diesen einzelnen *PDD*-Programmier-Bausteinen können mittels des Zusammenwirkens der Algorithmen die einzelnen fahrdynamischen Größen des Gesamtfahrzeugmodells berechnet werden. Diese Größen können in Form von Diagrammen somit mit **vehicle_plot_zsm** definieren und in der **main-file** visualisieren.

4.3.2 Definition von Softwaremodulen und Funktionen

Die Abfolge der Algorithmen setzt sich aus den Eingangsgrößen Lenkwinkleingabe und Geschwindigkeitsvorgabe zusammen. Aufgebaut sind die nachfolgenden Verarbeitungsgrößen aus dem gewählten Gesamtfahrzeugmodell mit Lenkungs-, Reifen- und Fahrzeugmodell sowie den Ausgangsgrößen für die Visualisierung in Diagrammen. Die Unterteilung in Eingangs-, Verarbeitungs- und Ausgangsgrößen wird im Nachfolgenden definiert und mithilfe von Simulink-Signalflussgraphen anschaulich visualisiert.

Eingangsgrößen

Lenkradwinkeleingabe

Die Datensätze aus der Lookup-Table werden vor dem Eintritt einer potenziellen Kollisionsgefahr als Eingangsgrößen eingelesen. Diese setzen sich, wie in Abbildung 38 dargestellt, aus zwei Lenkradwinkelsprüngen – bestehend aus den drei Bereichen A, B und C – zusammen. Die einzelnen Parameter werden abhängig vom Δx - und Δy -Abstand aus dem Lookup-Table entnommen und in die programmierten Gleichungen eingesetzt.

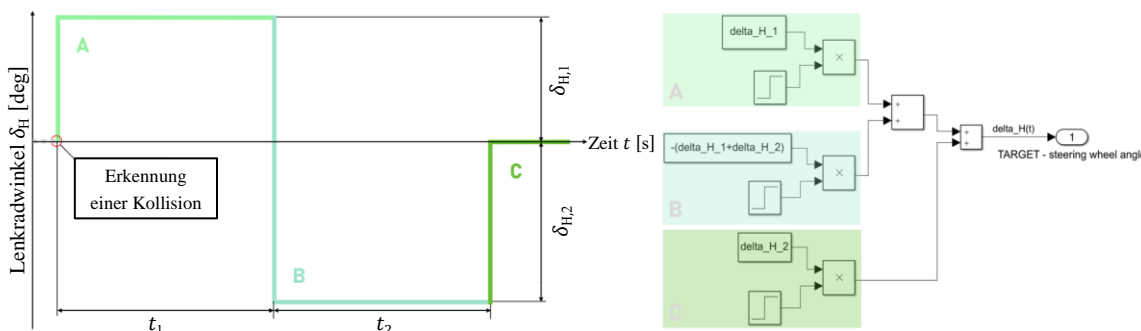


Abb. 38: Lenkradwinkelverlauf eines Ausweichmanövers mit den Lenkradwinkel-Zeit-Paaren aus der Lookup-Table [Eigene Darstellung] (vgl. A.4.3 MiL-Programmierung der Modelle A – C)

Nach der Erkennung einer bevorstehenden Kollision leitet das erste Lenkradwinkel-Zeit-Paar den Sprung für den Beginn des lateralen Abstands ein. Während der Normalfahrt wird ein Lenkradwinkel eingeschlagen, sodass das Fahrzeug sich auf einen Fahrspurwechsel vorbereiten kann. Im Bereich A wird nach Ablauf der Zeit t_1 der Lenkradwinkel $\delta_{H,1}$ auf den Bereich B überführt und auf die zweite Lenkradwinkelvorgabe $\delta_{H,2}$ reduziert. Damit die Zielposition erreicht wird, lenkt das Assistenzsystem in die entgegengesetzte Richtung. Nach Ablauf der Zeit t_2 im Bereich C ist das Lenkmanöver ausgeführt, die Lenkradwinkelvorgabe setzt sich auf 0 deg zurück und die Normalfahrt kann fortgesetzt werden. Somit wird für die Ausführung eines Ausweichmanövers die SOLL-Lenkradwinkelvorgabe aus den Summanden A, B und C definiert (vgl. Abbildung 38).

Geschwindigkeitsvorgabe

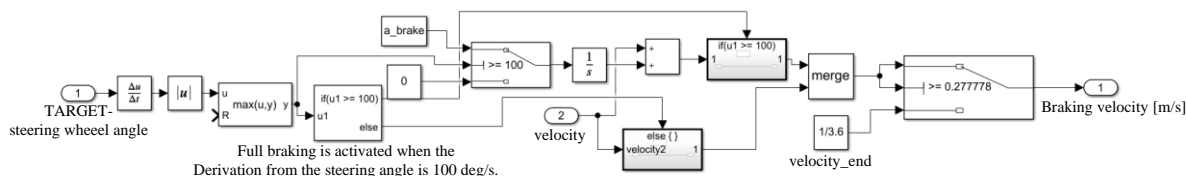


Abb. 39: Geschwindigkeitsreduzierungsprozess bei einem Ausweichmanöver [Eigene Darstellung] (vgl. A.4.3 MiL-Programmierung der Modelle A – C)

Die Lenkradwinkelgeschwindigkeit dient als Indikator für die Einleitung der Geschwindigkeitsreduzierung im Fahrerassistenzsystem. Zunächst wird der SOLL-Lenkradwinkelverlauf (vgl. Abbildung 38) als „TARGET – steering wheel angle“ abgeleitet und in Betrag gesetzt, um die Lenkradwinkelgeschwindigkeit $\dot{\delta}_H$ zu ermitteln. Die maximale Lenkradwinkelgeschwindigkeit $\dot{\delta}_{H,max} (= \max(u, y))$ wird kontinuierlich mit einem Grenzwert $if(u1 \geq 100)$ verglichen. Sobald dieser Grenzbereich $\dot{\delta}_{H,Grenz} = 100 \frac{deg}{s}$ überschritten wird, setzt eine Vollbremsung ein. Diese reduziert die Ausgangsgeschwindigkeit $v_1 = 100 \frac{km}{h}$ mit der Multiplikation aus der Integration der Bremsbeschleunigung über der Zeit linear, bis die Endgeschwindigkeit $v_2 = 1 \frac{km}{h}$ erreicht wurde (vgl. Abbildung 39). Nach Ende des Ausweichmanövers kann die Fahraufgabe vom Fahrer bzw. vom ACC-Assistenzsystem übernommen und die Geschwindigkeit erhöht werden (vgl. Unterkapitel 4.1.3). Als zusätzlicher Abfragezweig bezweckt das System keine Vollbremsung, solange der Grenzwert der $\dot{\delta}_H$ nicht überschritten wird. Somit kann das autonome System abhängig von $\dot{\delta}_H$ das Lenk- und Bremsmanöver bis zum Stillstand des Fahrzeugs ausführen.

Verarbeitungsgrößen

Lenkungsmodell

Die Formel zur Berechnung des Lenkwinkels aus dem Unterkapitel 3.3.1 (vgl. Gl. (2)) wird in zwei Terme unterteilt (vgl. Abbildung 40). Das Lenkungsmodell wandelt im ersten Schritt den „TARGET – steering wheel angle“ mit einem PT1-Glied in den „ACTUAL – steering wheel angle“ um, damit ein realitätsnaher Lenkradwinkelverlauf wiedergegeben wird.

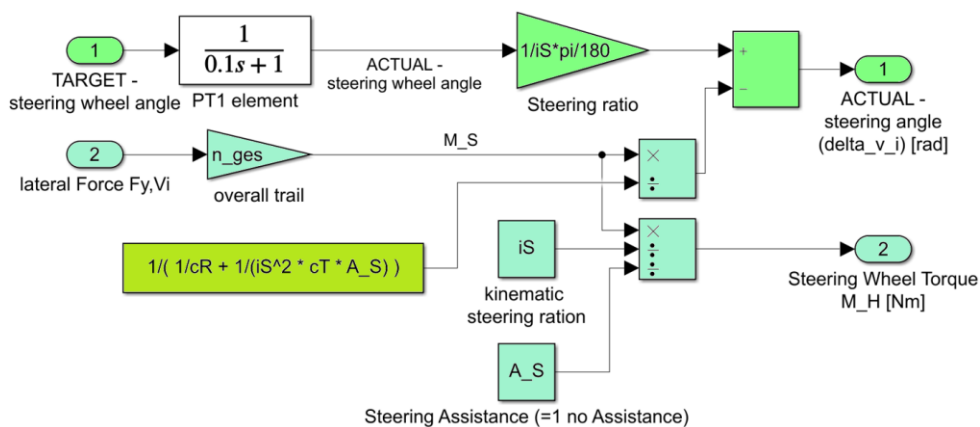


Abb. 40: Lenkungsmodell [Eigene Darstellung] (vgl. A.4.3 MiL-Programmierung der Modelle A – C)

Somit kann der Lenkwinkel am inneren und äußeren Reifen der Vorderachse unter Berücksichtigung der jeweiligen Seitenkraft berechnet werden. Des Weiteren ist die Kalkulation des Lenkradmoments M_H nötig (vgl. Gl. (4)), um das aufzubringende Moment des Fahrerassistenzsystems zu bestimmen (vgl. Abbildung 40).

Reifenmodell

Das Gesamtfahrzeugmodell C besitzt je zwei kurveninnere und -äußere Reifen, die in Vorder- und Hinterachse unterschieden werden. Da es sich hier um ein frontantriebenes Fahrzeug ohne Hinterachs- bzw. Allradlenkung handelt, wie im Freikörperbild des Zweispurmodells (vgl. Abbildung 24) ersichtlich, werden für die Berechnung der Schräglaufwinkel zwei unterschiedliche Formeln je Achse bei der Programmierung verwendet.

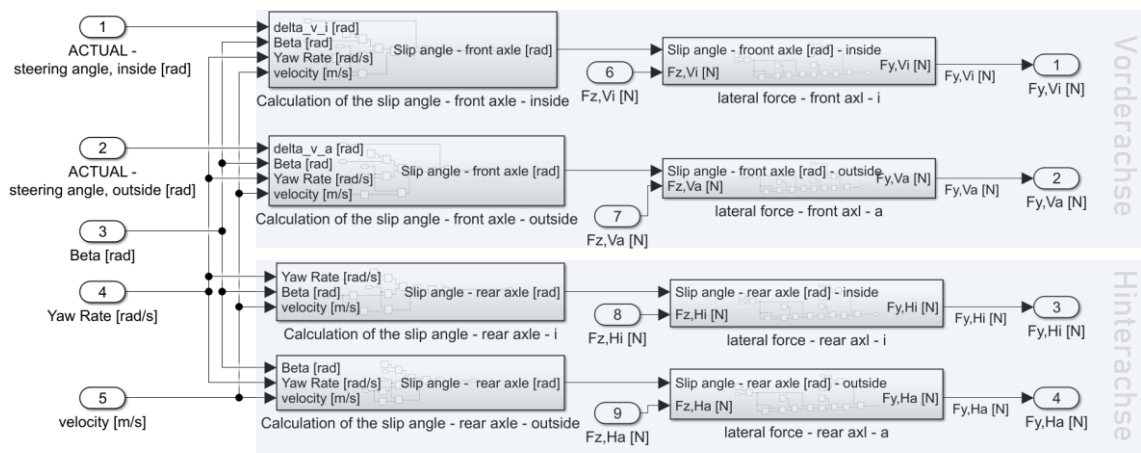


Abb. 41: Reifenmodellierung mittels Magic Formula nach Pacejka [Eigene Darstellung] (vgl. A.4.3 MiL-Programmierung der Modelle A – C)

Die fahrdynamischen Größen der Vorderachse werden mit den Eingangsgrößen Lenkwinkel, Schwimmwinkel, Giergeschwindigkeit und Fahrzeuggeschwindigkeit in den Gleichungen (2), (15) und (16) berechnet. Das Ergebnis wird für die Magic Formula nach Pacejka (vgl. Gl. (9)) benötigt, um die schräglaufwinkelabhängige Seitenkraft am kurveninneren und -äußeren Reifen kalkulieren zu können. In diesem Zusammenhang ergibt sich der Maximalwert D aus der Multiplikation des in der Simulation konstant angenommenen Kraftschlussbeiwerts μ (vgl. Abbildung 46) und der Radlast F_Z (vgl. Abbildung 21). Die Berechnung der fahrdynamischen Größen für die Hinterachse läuft analog zur Berechnung der entsprechenden Größen für die Vorderachse ab, wobei beim Schräglaufwinkel der Lenkwinkel auf null gesetzt ist.

Fahrzeugmodell

Die Fahrzeuggeschwindigkeit $velocity$, der gemittelte Vorderachslenkwinkel und die Summen der schräglafabhängigen Seitenkräfte $F_{y,v}$ und $F_{y,h}$ als Eingangsgrößen sind, wie in Abbildung 42 dargestellt, zur Berechnung der Schwimmwinkel und Giergeschwindigkeit notwendig. Diese fahrdynamischen Größen werden mit den Gleichungen (15) und (16) berechnet und nach Schwimmgeschwindigkeit $Beta Rate$ und Gierbeschleunigung $Yaw Acc$ aufgelöst. Daraus ergeben sich aus der Integration beider Parameter der Schwimmwinkel $Beta$ und die Giergeschwindigkeit $Yaw Rate$ (vgl. Abbildung 42).

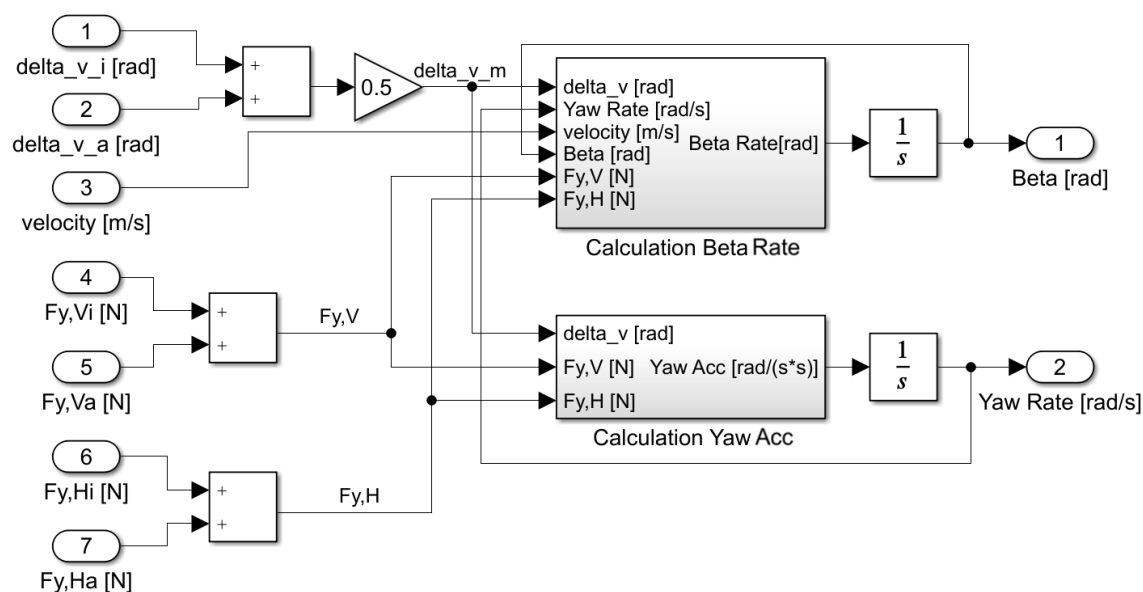


Abb. 42: Nichtlineares Zweispurmodell
[Eigene Darstellung] (vgl. A.4.3 MiL-Programmierung der Modelle A – C)

Wank- und Nickmodell

Die Grundlage für die Berechnung des Wank- und Nickwinkels ist das nichtlineare Zweispurmodell des Gesamtfahrzeugmodells C. Das Modell erlaubt die Berechnung des querbeschleunigungsabhängigen Wankwinkels nach Gl. (24):

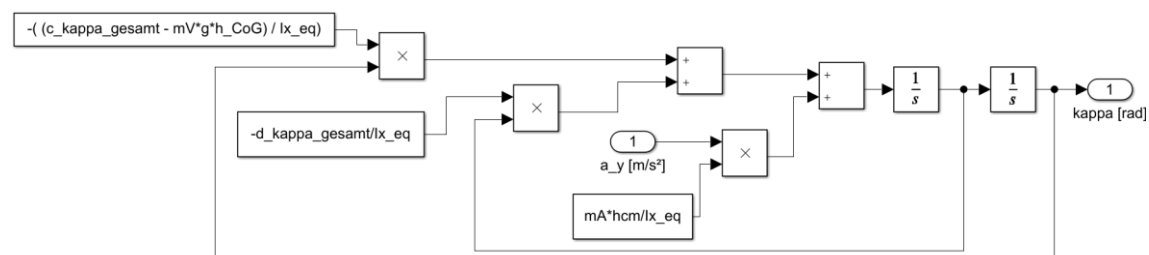


Abb. 43: Wankmodell [Eigene Darstellung] (vgl. A.4.3 MiL-Programmierung der Modelle A – C)

Ebenso erlaubt das Gesamtfahrzeugmodell C die Berechnung des längsbeschleunigungs-abhängigen Nickwinkels nach Gl. (26):

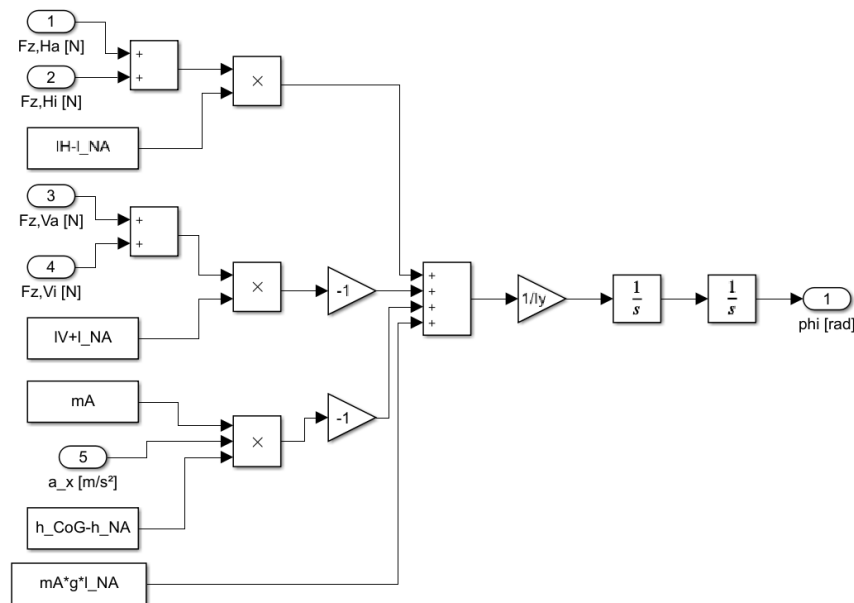


Abb. 44: Nickmodell [Eigene Darstellung] (vgl. A.4.3 MiL-Programmierung der Modelle A – C)

Fahrbahnrelative Position

Zur Transformation der fahrzeugfesten Koordinaten in das ortsfeste Koordinatensystem sind Fahrzeugmodellergebnisse wie der Schwimmwinkel $Beta$ und die zu integrierende Giergeschwindigkeit $Yaw Rate$ erforderlich. Die Addition von Schwimm- und Gierwinkel ergeben den Kurswinkel (vgl. Mitschke, 2014, S. 614). Aus der Multiplikation der Fahrzeuggeschwindigkeit und aus der anschließenden Integration resultieren, wie in den Gleichungen (27) und (28) beschrieben, die ortsfesten x - und y -Position des Fahrzeugs:

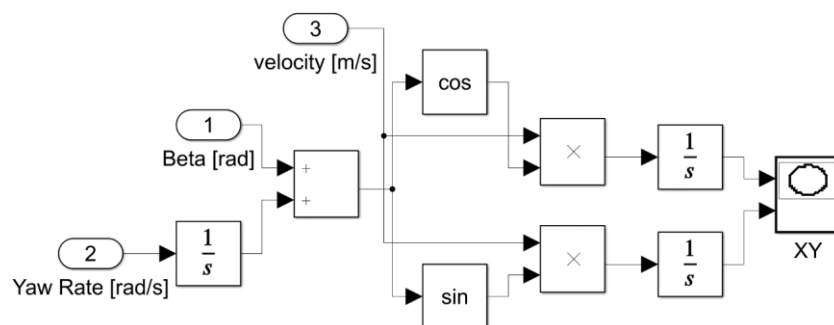


Abb. 45: Berechnung der x - y -Positionen im Weltkoordinatensystem [Eigene Darstellung] (vgl. A.4.3 MiL-Programmierung der Modelle A – C)

Ausgangsgrößen

Ausgangsgrößen werden für Visualisierungen der fahrdynamischen Größen in zweidimensionalen Diagrammen verwendet. Die Programmiersprache Python bietet unter Zuhilfenahme der Entwicklungsumgebung Geany eine Darstellung der Ergebnisse in Diagrammen an (vgl. A.5.1 PDD). Die fahrdynamischen Größen werden über die Zeitangabe oder anderen Größen in Form eines XY-Plots anschaulich dargestellt (vgl. Abbildung 45).

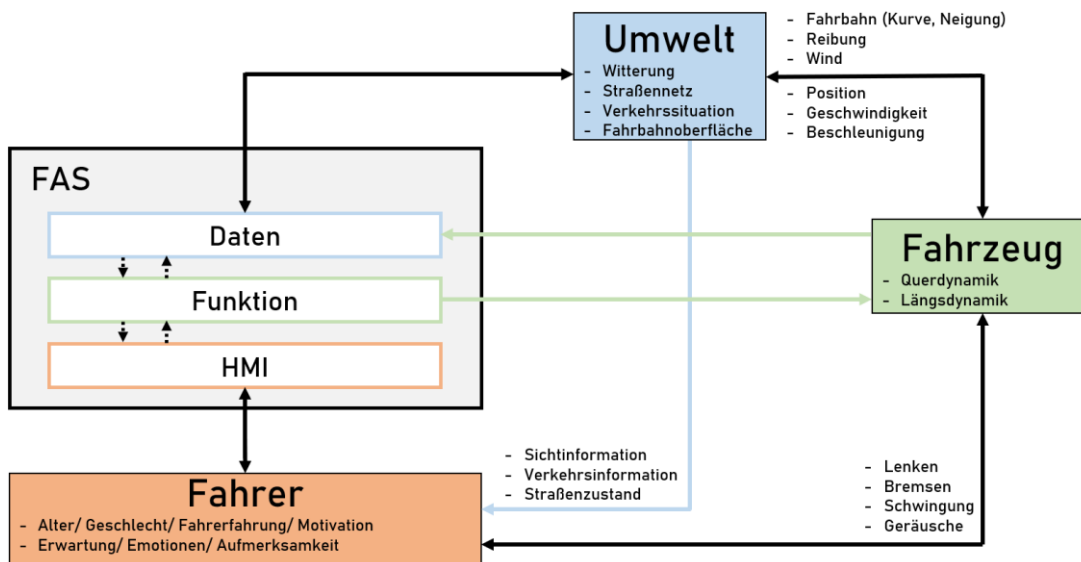
5 Integration der PPADD in CARLA

SiL-Simulationen mit idealen Kollisionsvermeidungstrajektorien können Rückschlüsse auf das Fahrverhalten beim Ausweichmanöver liefern. Sobald sich das Kraftfahrzeug mit autonomer Fahrfunktion in einer fahrsicherheitskritischen Lage befindet, die eigenständig nicht zu bewältigen ist, könnte der technische Assistent mit dem *PPADD*-Programm ein alternatives Fahrmanöver bewerten und freigeben. Vor dem Modelltest ist es möglich, kurzfristige Änderungen im FAS vorzunehmen, deren Funktionalität validierbar ist (vgl. Abbildung 11).

5.1 Vereinfachungen im Regelkreis

Zur Erfüllung der Fahraufgabe werden, wie Abbildung 46 aufzeigt, zuerst die Informationen aus Umwelt und Fahrzeug vom Fahrer wahrgenommen. Er kann die Quer- und Längsdynamik des Fahrzeugs verändern, indem er lenkt und bremst. Die Umwelt wirkt mit der Reifen-Fahrbahn-Kontaktfläche bei zunehmender Geschwindigkeit und Beschleunigung dem Fahrzeug entgegen und kann den Fahrer hinsichtlich des Straßenzustands beeinträchtigen. Umgekehrt beeinflusst der Fahrer mit seiner Fahrweise sein Fahrzeug und die Umwelt ebenso. Daher sind Vereinfachungen für eine idealisierte Betrachtung notwendig, um alle vier Elemente des Regelkreises zu bestimmen (vgl. Abbildung 46). Die Daten aus der Umwelt und dem Fahrzeug werden somit in der Simulation als gegeben angenommen und dienen als Eingangsparameter für die Ausführung einer Kollisionsvermeidung.

Das simulierte Gesamtfahrzeugmodell lässt Vereinfachungen im Ausweichmanöver-Regelkreis zu, sodass ohne den Besitz umfangreicher Datensätze einfache Fahrspurwechselfersuche bereits repräsentative Ergebnisse für die Analysen bieten. Fahrer, Fahrzeug und Umwelt stehen in gegenseitiger Wechselwirkung mit dem FAS, wie im Regelkreis der Abbildung 46 illustriert. Der Fahrer ist bei einem autonomen Ausweichmanöver nicht – wie bei ISO 3888-1 und 3888-2 – als aktiver Regler in das Verkehrsgeschehen involviert. Das *PPADD*-Programm kann mit der KI-Schnittstelle interagieren, welche die Zielposition an das Gesamtfahrzeugmodell überträgt. Jedoch führt der *PPADD*-Algorithmus den Prozess als Sicherheitssystem und nicht als Komfortsystem aus, sodass keine aktive Mensch-Maschine-Schnittstelle (HMI) berücksichtigt wird.



	Vereinfachung	Begründung
Fahrer	<ul style="list-style-type: none"> Keine Verletzungsgefahr des Fahrers Keine aktive Fahrsicherheit für den Fahrer in der Simulation 	Ein Eingriff des Fahrers in das System ist nicht Bestandteil des Regelkreises.
Fahrzeug	<ul style="list-style-type: none"> Gesamtfahrwiderstände $F_{W,ges} = 0$ gilt für die Modellierung der Längsdynamik: Beschleunigungs- F_a, Steigungs- F_{Steig}, Roll- F_{Roll} und Luftwiderstand F_{Luft} werden vernachlässigt Ego-Fahrzeug hat keine Sonderausstattung, wie z. B. Anhängerkopplung, Gepäckträger etc. Idealer Schlupf der Reifen-Fahrbahn-Kontaktzone: Kraftschlussbeiwert $\mu = 1,0 \rightarrow$ konstant Konstante Fahrzeugparameter: keine Änderung der Gesamtmasse, Schwerpunktlage, Gesamtreifenachlauf etc. Vernachlässigung von Sturz- und Spurwinkeländerungen Keine Berücksichtigung von Reifenalterung Fahrzeugschwerpunkt in der Fahrzeugmitte $\frac{s}{2}$ 	Es existiert ein Ego-Fahrzeug mit relevanten Fahrparametern, die ein reales Fahrzeug nachbilden. Fahrerassistenzsysteme können im SiL-Prozess validiert und verifiziert werden.
Umwelt	<ul style="list-style-type: none"> Von anderen Verkehrsteilnehmern ist das Bewegungsprofil bekannt Gleichbleibende, sonnige Wetterlage Andere Verkehrsteilnehmer sind auf derselben Fahrbahnspur platziert Keine Berücksichtigung der Höchst- und Mindestquerneigung (z. B. 2,5 deg – Straßenneigung aus entwässerungstechnischen Gründen) (vgl. Mitschke, 2014, S. 628) 	Datensätze aus realen Versuchen sind nicht vorhanden, weshalb die Testszenarien für eine hohe Reproduzierbarkeit mit identischen Bedingungen verwendet werden.
FAS	<ul style="list-style-type: none"> Objekterkennung zur Situationserfassung und Distanzabschätzung sind nicht Untersuchungsgegenstand der Arbeit Ideale Erfassung der Messgrößen und Ausführung identischer Trajektorienverläufe wie bei Open-Loop-Manövern 	Erfahrungsgrundlagen mit dem zu entwickelnden FAS existieren nicht, da das System noch keine realen Ausweichmanöver ausgeführt hat (vgl. Unterkapitel 1.3).

Abb. 46: Fahrer-Fahrzeug-Umwelt-Fahrerassistenz-Regelkreis (oben) (vgl. Hofauer, 2017, S. 141) und Vereinfachungen im Fahrer-Fahrzeug-Umwelt-FAS-Regelkreis (unten) [Eigene Darstellung] (vgl. A.5.1 PDD und A.5.2 PPADD)

5.2 Qualitative Beurteilung der Simulationsergebnisse des Gesamtfahrzeugmodells

Neben dem Simulationsprogramm CARLA wird der Systemsimulator CarMaker verwendet, um reale Testszenarien vollständig nachzubilden. Das zentrale Werkzeug CarMaker verfügt in einer offenen Vehicle-in-the-Loop-Simulationsumgebung über virtuelle, manöverbasierte Simulationsmodelle in Echtzeit. Dieses X-in-the-Loop-Framework bedient die zuvor erwähnte Interaktion im Fahrer-Fahrzeug-Umwelt-Regelkreis über verschiedene Schnittstellen, wie z. B. über den Fahrercharakter, die Fahrbahn- und Verkehrssituation sowie die Fahrzeugvarianten. Die Benutzeroberfläche bietet die Parametrisierung realer Gesamtfahrzeugmodelle mit Verkehrssituationen in einem realitätsnahen Umfeld. Somit bietet sich CarMaker als Referenzmodell für den Vergleich von den Ergebnissen mit dem Simulationsprogramm CARLA an.

In der folgenden Tabelle werden wesentliche Punkte von CARLA mit CarMaker qualitativ verglichen, Gemeinsamkeiten werden behandelt und Unterschiede erklärt:

Tab. 7: Qualitative Bewertung von CarMaker und CARLA (im Default-Modus [vgl. A.1.3 Weitere Definitionen])

	CARLA	CarMaker
Hardware	+ ohne zusätzliche Eingabegeräte	+ ohne zusätzliche Eingabegeräte
Steuerung	+ Trajektorie Festlegbar	0 VDA Ausweichtest, ISO Spurwechsel
Zugangsberechtigung	+ Open-Source-Software	- Proprietäre Software
Quellcode	+ einsehbar, erweiterbar	- nicht zugänglich
Python Programmierschnittstelle	+ Kompatibilität Python-API	- Keine Kompatibilität (vgl. CarMaker, 2021)
Fahrzeugbewegung	- Punktmassenmodell (vgl. Dosovitskiy, 2017, S. 3)	+ Gesamtfahrzeugmodell vorhanden
Plattformunabhängigkeit	+ Windows, Linux	+ Windows, Linux

Der direkte Vergleich der Gesamtfahrzeugmodelle von CarMaker und CARLA im Default-Modus zeigt, dass nichts gegen die Nutzung des Simulationsprogramms bei Ausweichmanövern spricht. CARLA beinhaltet im Default-Modus ein Punktmassenmodell, das komplexe, hochdynamische Ausweichmanöver nicht wie in der Realität nachbilden kann. Bei Erstellung eines Gesamtfahrzeugmodells ist zwar CarMaker positiv zu bewerten, jedoch ist der Quellcode – in Tabelle 7 als negatives Kriterium markiert – nicht für Veränderungen an der vorgegebenen Programmierung zugänglich. Standardisierte Fahrversuche, wie z. B. der ISO Wedeltest und der VDA-Ausweichtest, sind im CarMaker-Programm hinterlegt (vgl. Abbildung 47). Jedoch ist im Programmcode für den Eigengebrauch eine Anpassung oder Veränderung nicht möglich. Wie in der nachfolgenden Abbildung mit dem Audi R8 in einer Bilderserie illustriert, ist der VDA-Ausweichtest während eines Fahrspurwechsels in CarMaker erkennbar.

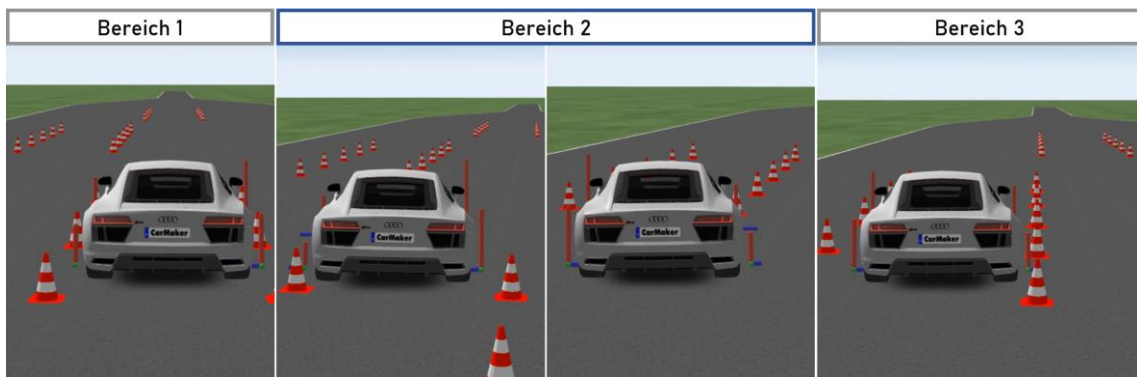


Abb. 47: Erster Fahrspurwechsel im VDA-Ausweichtest in CarMaker
[Eigene Darstellung] (vgl. A.3.2 VDA-Ausweichtest (ISO 3888-2))

Die Trajektorien und die Fahrsicherheit in der MiL-Simulation werden vor der Überführung der Ausweichmanöver in die SiL-Simulation und vor der Programmierung der erforderlichen Codierung geprüft (vgl. Abbildung 11). Mit der 2D-Darstellung mittels „Bird’s-Eye Scope“ (vgl. A.4.3 MiL-Programmierung der Modelle A – C) und anschließender 3D-Visualisierung (vgl. Abbildung 48) wird die Ausführung des selbstprogrammierten Ausweichmanövers in die SiL-Simulation überführt. Jedoch bietet MATLAB/Simulink keine Schnittstelle für das Simulationsprogramm CARLA an (vgl. MathWorks [2], 2021). Zur Prüfung des Programms von Ausweichmanövern auf Plausibilität wird im Rahmen dieser Masterarbeit die MiL-Simulation genutzt.



Abb. 48: 3D-Visualisierung in MATLAB/Simulink
[Eigene Darstellung] (vgl. A.4.3 MiL-Programmierung der Modelle A – C)

Die verwendete Entwicklungsumgebung Geany adaptiert mit der *PPADD* die Funktionalität der MiL-Programmierung und stellt die Richtigkeit der Ausweichmanöver sicher (vgl. Abbildung 11 und A.5 Python-Programmierung). Der Nachteil von CarMaker und MATLAB/Simulink wird somit durch die eingeschränkte Anpassung an die Codierung mit der Open-Source-Software kompensiert. Das Simulationsprogramm CARLA sichert über die einsehbare Programmcodierung durch mögliche Anpassungen der Fahrversuche deren Funktionalität. Somit entsteht bei der Erweiterung durch das *PPADD*-Programm eine schnelle Anwendungsentwicklung (Rapid Application Development), welches das Punktmassenmodell durch ein eigenständig programmiertes Gesamtfahrzeugmodell ersetzt (vgl. Tabelle 7).



Abb. 49: Bilderserie eines Ausweichmanövers in CARLA
[Eigene Darstellung] (vgl. A.5.5 Aufzeichnung vom Fahrsimulator CARLA)

Anhand der Bilderserie (vgl. Abbildung 49) ist ersichtlich, dass das Referenzfahrzeug Audi Q5 in CARLA, ähnlich wie bei CarMaker und MATLAB/Simulink, ein Ausweichmanöver auf die linke Fahrbahnspur ausführt. Um wie bei CarMaker einen zweiten Fahrspurwechsel auf die rechte Fahrbahnseite nachzustellen, ist beim Fahrsimulator CARLA eine gezielte Tasteneingabe notwendig. Ebenso ist die Platzierung von Hindernissen vom Nutzer frei wählbar, sodass verschiedene Szenarien flexibel und schnell abgearbeitet werden.

5.3 Verwendung von Python-APIs in CARLA

Das Punktmassenmodell ist aufgrund der geringen Modellkomplexität für die Auswertung hochdynamischer Ausweichmanöver nicht geeignet (vgl. Keller, 2017, S. 32 – 34), weshalb eine Erweiterung des implementierten Gesamtfahrzeugmodells notwendig ist (vgl. Tabelle 7). Um ein plausibles Fahrzeugverhalten im Simulationsprogramm CARLA zu realisieren, ist eine Ergänzung des bestehenden Grundaufbaus mit Bausteinen in Form von APIs (Application Programming Interface) erforderlich. Die vorhandenen Schnittstellen können mit der entsprechenden Programmcodierung der APIs an den Zugangspunkten mit dem CARLA Server kommunizieren. Diese Programmierschnittstelle dient dazu, das Punktmassenmodell durch das Gesamtfahrzeugmodell C zu ersetzen. Daher ist in folgendem Diagramm ein Überblick über die Softwaremodule und -funktionen des PPADD-Programms dargestellt:

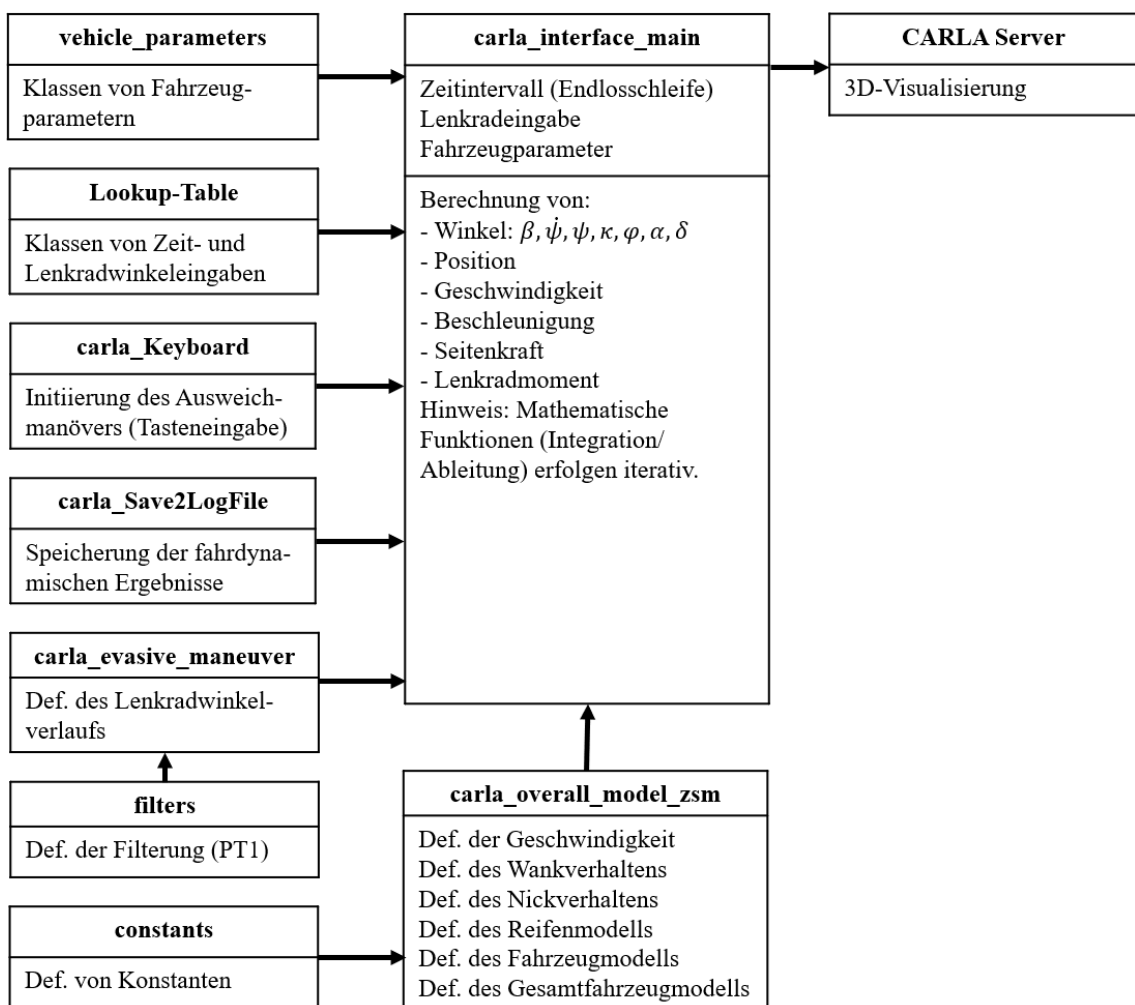


Abb. 50: Schematischer Überblick der PPADD-Softwaremodule und -funktionen
[Eigene Darstellung] (vgl. A.5.2 PPADD)

Der schematische Überblick über das *PPADD*-Programm mit der Schnittstelle zum Fahr-simulator CARLA weist Ähnlichkeiten mit dem *PDD*-Programm auf (vgl. Abbildung 37). Die Module **vehicle_parameters**, **Lookup-Table**, **carla_evasive_maneuver**, **filters** und **constants** wurden aus dem bestehenden Programm adaptiert. Allerdings unterscheidet sich das *PDD*-Programm mit dem Fahr-simulator CARLA hinsichtlich der unterschiedlichen Zeitdefinitionen. Die Softwaremodule **solvers** und **vehicle_position** sind für das vorgegebene Zeitintervall nicht integrierbar und ausschließlich für definierte Zeitbereiche programmiert (vgl. Abbildung 37 und 50).

Im Folgenden werden weitere grundlegende Unterschiede zwischen dem *PDD*- und dem *PPADD*-Programm definiert.

5.3.1 Numerische Integrationsverfahren

Der ganzheitliche Kurvenverlauf der fahrdynamischen Wertepaare ist im impliziten Euler- bzw. Runge-Kutta-Verfahren in den *PDD*-Programmen mit einem festgelegten Zeitbereich definiert (vgl. Abbildung 37). Die impliziten Integrationsverfahren erlauben es, bekannte Werte für die Integration der Differenzialgleichungen des Gesamtfahrzeugmodells einzusetzen (vgl. Dahmen, 2008, S. 386). Der umfangreiche Rechenaufwand im Integrationsschritt ist für eine Konvergenzkontrolle der unbekannt fahrdynamischen Größen Schwimmwinkel, Giergeschwindigkeit, Gierwinkel, Wankgeschwindigkeit, Wankwinkel, Nickgeschwindigkeit und Nickwinkel von Vorteil. Somit ist die ideale Kollisionsvermeidungstrajektorie für die Ausweichmanöver des Ego-Fahrzeugs bei der Verwendung des impliziten Verfahrens exakt bestimmbar.

Die mathematischen Berechnungsmethoden mit implizitem Euler- oder Runge-Kutta-Verfahren werden beim Fahr-simulationsmodell nicht verwendet. Wegen der Zeitintervallberechnung $h (= \Delta t)$ unterscheidet sich der nicht definierbare Endpunkt t_{\max} bei Game-Loops vom **vehicle_zsm** (vgl. Abbildung 37), da dieser keinen festgelegten Zeitraum t_{\min} bis t_{\max} besitzt. Die Vorwärtsrechnung der Integrationsverfahren ist die einzige Alternative, die eine iterative Berechnung der fahrdynamischen Größen in CARLA zulässt. Diese einzelnen Parameter werden in der Simulation in einem Zeitintervall h bei jedem Durchlauf der Schleife berechnet. Das Ziel besteht dann darin, ausgehend vom Wert (t_k, x_k) den nachfolgenden Wert (t_{k+1}, x_{k+1}) mittels Einschrittverfahren zu kalkulieren (vgl. Abbildung 51).

Anwendung in der PPADD-Programmierung

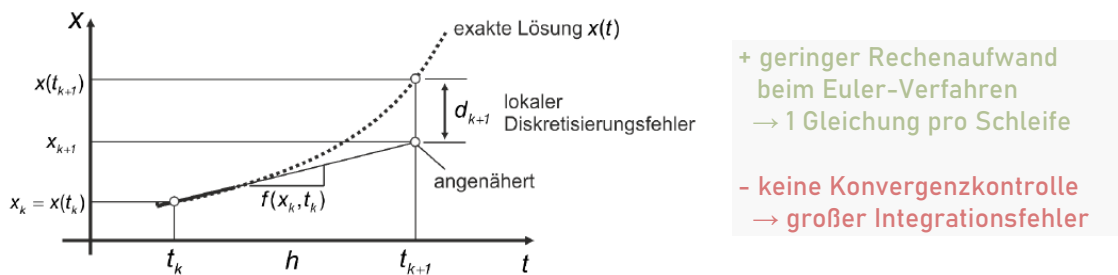


Abb. 51: Vorwärtsrechnung mit explizitem Integrationsverfahren
[Eigene Darstellung] (vgl. Dahmen, 2008, S. 386)

Mit der Zeitschrittweite h ergeben sich die Näherungswerte

$$x_{k+1} = x_k + h \cdot f(t_k, x_k), k = 0, 1, 2, \dots \quad (29)$$

Das explizite Euler-Verfahren approximiert in einem Schleifendurchlauf die unbekannte Lösung $x(t)$ im Zeitintervall $[t_k, t_{k+1}]$ linear mit der Steigung zum Zeitpunkt t_k . Wie aus der Abbildung 51 hervorgeht, unterscheidet sich der Näherungswert beim expliziten Integrationsverfahren von der exakten Lösung $x(t)$ durch den Wert d_{k+1} . Dieser lokale Diskretisierungsfehler addiert sich in jedem weiteren Schleifendurchlauf zu einem globalen Integrationsfehler auf (vgl. Dahmen, 2008, S. 393 – 404). Die Schrittweiten h sind mit weniger Rechenaufwand je Integrationsschritt auszuführen, daher besteht keine Konvergenzkontrolle mit der exakten Lösung $x(t)$.

Aus dem Vergleich beider Integrationsmethoden geht hervor, dass die Konvergenzkontrolle im impliziten Integrationsverfahren für die *PDD*-Programmierung exakte Parameter ergibt, während das explizite Verfahren dies nicht berücksichtigt. Nach der Datenaufzeichnung der berechneten Vorwärtsrechnungen kann die Abweichung der Integrationsfehler – anders als beim Rückwärtsverfahren – in dem *PPP*-Programm dargestellt werden (vgl. A.5.4 PPP).

5.3.2 Datengenerierung für die Validierung

Anhand der CARLA-Rohdatengenerierung wird die Divergenz zwischen dem expliziten und impliziten Integrationsverfahren für die Validierung untersucht. Das Modul `carla_Save2LogFile` ist für die Aufzeichnung der expliziten fahrdynamischen Parameter während der Testfahrt, welche mit den impliziten Zahlenwertpaaren verglichen werden, verantwortlich. Im anschließenden *PPP*-Programm (vgl. A.5.4 PPP) findet die Überprüfung statt, während der globale Integrationsfehler sowie die Ergebnisse des Vorwärtsverfahrens außerhalb des zulässigen Toleranzbereichs kontrolliert werden.

time [s]	Steering wheel angle [rad]	Slip angle [rad]	Yaw Rate [rad/s]	Yaw angle [rad]
...				
13.186928110197186	-1.0842494930482196	-0.06409651434598507	-0.047626447373219354	0.2708182045214
13.196928109973669	-1.1165810505591682	-0.06225981195347776	-0.07822798702392743	0.2700359246687
13.206928109750152	-1.1459733756288462	-0.06026096749515029	-0.10786554297830414	0.2689572692630
13.216928109526634	-1.17269367120103	-0.05811556876037707	-0.1364798308120518	0.2675924709854
13.226928109303117	-1.1969848490432837	-0.05583947680025765	-0.16402199884698884	0.2659522510336
13.2369281090796	-1.219067738035659	-0.05344862676627134	-0.19045325350072817	0.2640477185411
13.246928108856082	-1.2391430917058837	-0.05095884803667566	-0.21574448045371938	0.2618902737848
13.256928108632565	-1.257393413261354	-0.048385703900498726	-0.23987583521145187	0.2594915154863
13.266928108409047	-1.273984614709131	-0.045744351021130586	-0.26283628450281604	0.2568631527001
13.27692810818553	-1.2890675251468489	-0.043049418741664416	-0.28462308790257185	0.2540169218846
...				

Abb. 52: Struktur einer txt-Datei mit dem entwickelten Datenformat
[Eigene Darstellung] (vgl. A.5.4 PPP)

Dieser Ausschnitt eines Ausweichmanövers zeigt abhängig vom Zeitintervall die fahrdynamischen Größen Lenkrad-, Schwimm- und Gierwinkel sowie die Giergeschwindigkeit zeilenweise an (vgl. Abbildung 52). Die Entwicklungsumgebung Geany kann damit die einzelnen fahrdynamischen Parameter bei der Datenauswertung entnehmen und Analysen effektiver ausführen.

5.3.3 Zustandsautomat

Bevorstehende Kollisionsgefahren lassen sich als realistisches Ereignis bspw. in CARLA mit einem Zustandsautomaten realisieren, der ein virtuelles, hochdynamisches Ausweichmanöver einleitet (vgl. Rupp, 2012, S. 329 – 400). Der Zustandsautomat beschreibt die Zustandsfunktion des Fahrerassistenzsystems, dessen Verhalten vom Ereignis auf der Ego-Fahrbahn und vom Zustand des Ego-Fahrzeugs abhängt. Des Weiteren setzt die Zustandsmaschine die Programmlogik der *PPADD* um, die das Implementieren der Einzelheiten eines Zustands abhängig vom erfassten Eingangssignal, das Umschalten von einem Zustand zum nächsten sowie Ausgabefunktionen gewährleistet. Ausführbare Handlungen von Zustandsautomaten sind somit das Wechseln des Systems in einem in einen anderen Zustand, das Ausführen von Aktionen oder das Verändern von Variablen:

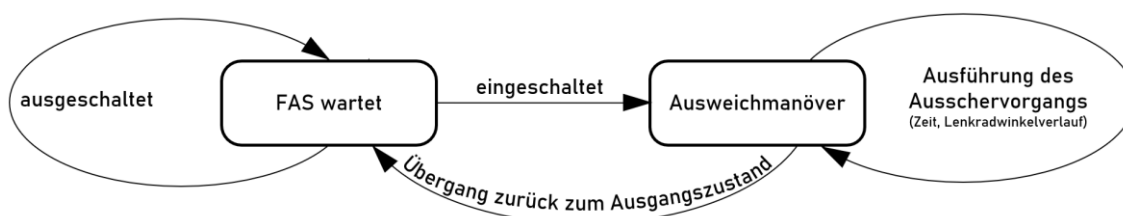


Abb. 53: Zustandsautomat des Fahrerassistenzsystems
[Eigene Darstellung] (vgl. Rupp, 2012, S. 341 – 349)

Der Simulator besitzt einen „FAS wartet“-Startzustand während der Normalfahrt des Fahrers, der ohne kritische Situationen zu provozieren das virtuelle Fahrzeug steuert. Registriert das Assistenzsystem eine bevorstehende Kollision, wechselt, wie in Abbildung 53 dargestellt, der Zustandsautomat beim **Einschalten** in den vorbereitenden Zustand **Ausweichmanöver**. Falls ein Hindernis auf der gleichen Fahrbahnspur ist und die bevorstehende Kollision nicht durch Bremsen vermeidbar ist, erfolgt vor dem Zustandswechsel eine Überprüfung. Es wird geprüft, ob die situationsabhängigen Randbedingungen Δx , Δy und v_{rel} zur Vermeidung der bevorstehenden Kollision mit einem hochdynamischen Ausweichmanöver realisierbar sind.

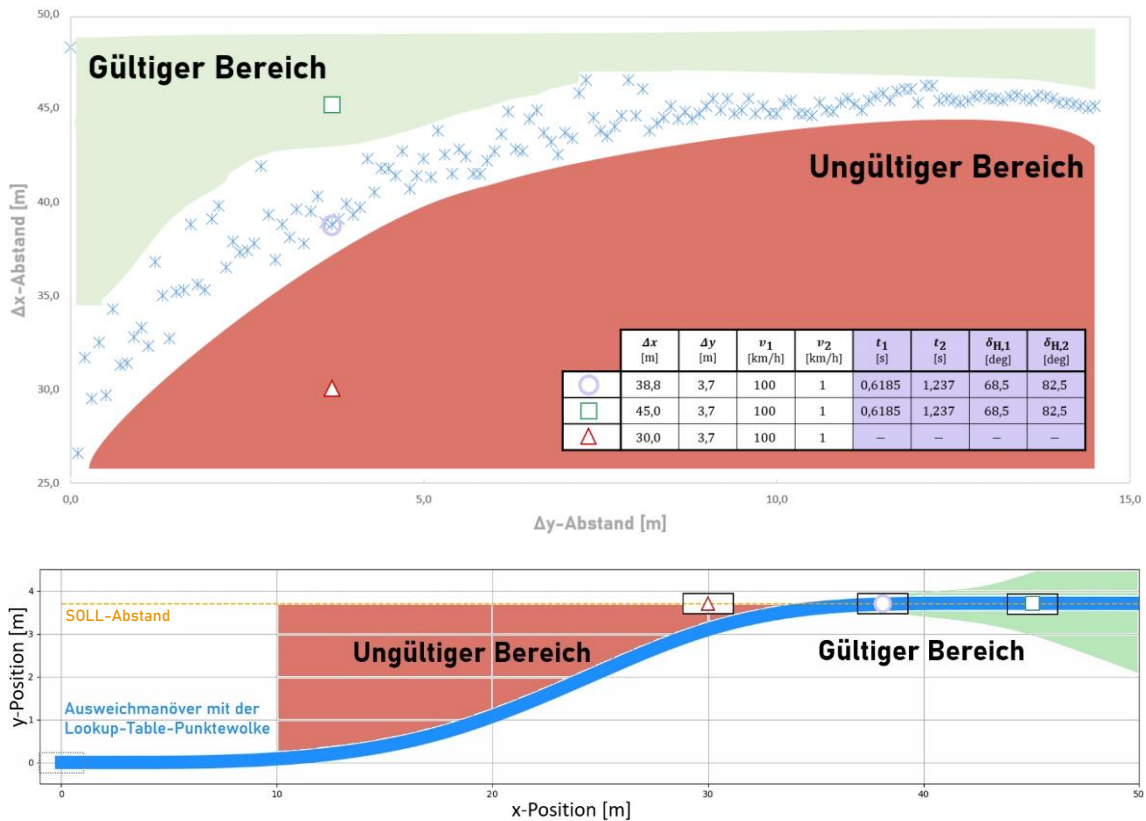


Abb. 54: Unterscheidung zwischen gültigem und ungültigem Bereich (skizziert) anhand dreier Beispielwerte aus dem PPADD-Lookup-Table bei einer Relativgeschwindigkeit von $100 \frac{\text{km}}{\text{h}}$ für ein Ausweichmanöver [Eigene Darstellung] (vgl. A.5.3 PPADD-Lookup-Table)






Für die Generierung der Lookup-Table wird zuerst die standardisierte Relativgeschwindigkeit $v_{\text{rel}} = v_1 = 100 \frac{\text{km}}{\text{h}}$ nach ISO 7401 gewählt, um erste Ergebnisse für ein hochdynamisches Ausweichmanöver zu erhalten. Wie die blaue Punktewolke in Abbildung 54 visualisiert, können die Eingangsparameter ober- oder unterhalb der Δx - Δy -Lookup-Table-Paare liegen. Befindet sich die vorgegebene Bedingung im roten, ungültigen Bereich (vgl. rot-weißes Dreieck in Abbildung 54), wird die Vorbereitung zur Ausführung des Ausweichmanövers im Fahrsimulator abgebrochen und in den Zustand **FAS wartet** überführt (vgl. A.5.2 PPADD). Dies tritt z. B. dann auf, wenn die vom Ego-Fahrzeug aufzubringende Querschleunigung außerhalb des gültigen physikalischen Bereichs der Kippgrenze liegt und folglich die Lookup-Table-Parameter nicht generiert werden. Sind hingegen die Randbedingungen erfüllt und befindet sich ein Δx - Δy -Paar im gültigen Bereich (vgl. grün-weißes Quadrat oder violetter Kreis in Abbildung 54), erfolgt das Ausweichmanöver in einer Schleife, bis die Zeit $t_{\text{Ausweichmanöver}}$ mit t_1 und t_2 sowie dem geglätteten Lenkradwinkelverlauf $\delta_{\text{Ausweichmanöver}}(t)$, bestehend aus $\delta_{H,1}$ und $\delta_{H,2}$, ausgeführt ist. Am Ende liegt bei der Zielkoordinate des grün-weißen Quadrates eine Δx -Differenz zum violetten Kreis von 6,2 m vor, die nach dem Ausweichmanöver zur

Geradausfahrt genutzt wird. Der Lenkradwinkelverlauf sowie die Reduzierung der Fahrzeuggeschwindigkeit sind in der Wertetabelle als voreingestellte Konfiguration vorgegeben und abgespeichert (vgl. A.5.3 PPADD-Lookup-Table). Das autonome System agiert in diesem Zustand ohne den aktiven Eingriff des Fahrers und führt die Kollisionsvermeidung selbstständig aus. Somit kann der Zustandsautomat nach Ausführung der Fahraufgabe unter Berücksichtigung der Fahrsicherheit wieder den Ausgangszustand **FAS wartet** einnehmen und der Fahrer bzw. ein anderes FAS übernimmt die Fahraufgabe.

Initiierung des Ausweichmanövers

Eine API-Schnittstelle für das *PPADD*-Programm existierte vor der Validierungsphase des autonomen Ausweichmanövers nicht. Die Evaluierung der Objekterkennung und die Entscheidungsfindung zur Kollisionsvermeidung übernimmt alternativ der Fahrer manuell (vgl. Abbildung 13). Eine aktive Interaktion mit externen Eingriffsmöglichkeiten in die CARLA-Game-Loop wurde mit einer Python-API erweitert, um eine prädestinierte virtuelle Darstellung hochdynamischer Ausweichmanöver auszuführen. Die Interaktion zwischen dem Fahrer und CARLA erfolgt per Tastaturdruck (vgl. Tabelle 8) mit dem Modul `carla_keyboard` (vgl. Abbildung 50).

Tab. 8: Übersicht über die auszuführenden Aktionen bei Tastendruck

FAS		Ausweichmanöver nach links
		Ausweichmanöver nach rechts
CARLA		Reset in den Ausgangszustand
		Fahrsimulator im Zustand Pause, Unterbrechung der Simulation
		Überleitung vom Zustand Pause in den Zustand Run

6 Validierung des Modells

Die Voraussetzung für die Entwicklung von Programmen, welche Fahrmanöver aus der Realität nachbilden, erfordern ein robustes Referenzmodell mit veränderbaren Schnittstellen. Folglich wird ein bestimmtes Modell, wie bspw. das Referenzfahrzeug Audi Q5, mit allen relevanten Fahrzeugparametern und seinen fahrdynamischen Größen generiert. Dieses Gesamtfahrzeugmodell besitzt Grenzen, die im Validierungsprozess analysiert und bewertet werden. Ein solcher Prozess erlaubt Rückschlüsse zur Stabilität und zur Funktionalität der Fahrerassistenzsysteme. Die Mindestanforderung an dieses FAS beim Ausweichmanöver ist die Einhaltung der physikalischen Grenzen.

6.1 Grenzen der Programmierung

Die API-Programmierung in dieser Arbeit benötigt Eingabeparameter für die Berechnung der Zielwerte, um die erwünschten Ergebnisse zu erhalten. Die Wahl der geeigneten Parameter erfordert numerische und fahrdynamische Kenntnisse, damit die autonome Kollisionsvermeidung gelingt. Zu Beginn wird MATLAB/Simulink im MiL für die Bestimmung der Referenzwerte gewählt, da hier bereits ein vorinstalliertes lineares Reifen- und Fahrzeugmodell vorhanden ist (vgl. Abbildung 11). Jedoch sind die Standard-Toolbox-Datensätze für die Nutzer nicht veränderbar. Die Schwierigkeit bei der universellen Nutzung ist die Limitierung infolge der festgelegten Definitionen im Programm und zu wenig veröffentlichten Fahrzeugdaten in der Forschung. Daher ist die Überführung von Teilmodellen in die SiL-Programmierung und die Ergänzung um weitere Modelle Voraussetzung, um ein geeignetes Gesamtfahrzeugmodell zur Verfügung zu haben, das im Post-Processing validiert wird.

Eine physikalische Grenze stellt beim Ausweichmanöver die Kippgrenze dar, deren Überschreitung eine Fehlermeldung im Programm ausgeben muss. Beim Abheben eines Rades besteht eine mögliche Unfallgefahr, was die sofortige Unterbrechung des Fahrmanövers erfordert. Zuerst berechnet das Programm $a_{y,v/h,kipp}$ aus Gl. (22) und prüft in einer if-Anweisung, ob zum Zeitpunkt des Fahrzustandes der vorhandene Querschleunigungsbetrag überschritten ist. Ein mögliches Ergebnis: Die Verwendung des vorhandenen Programms des Referenzfahrzeugs Audi Q5 mit den Fahrzeugdaten eines anderen Fahrzeugs, wie z. B. eines BMW (vgl. A.5.1 PDD und A.5.2 PPADD), weist Fehler auf. Folgende Fehlermeldung erscheint auf dem Terminal des `main`-files im *PDD*- und *PPADD*-Programm, wenn das kurveninnere Rad droht, abzuheben: „Querschleunigung

Alat_nonlin 'aktuelle Querbesehleunigung' ist größer als ay_max 'a_{y,v/h,kip} aus Gl. (22)'. Simulation ist nicht plausibel.“ (s. A.5.1 PDD und A.5.2 PPADD). Die Behebung dieses Fehlercodes ist mit einer angepassten Lookup-Table realisierbar, deren Ausgabedaten sich im Modul **Lookup-Table** (vgl. Abbildung 37 und 50) einfügen lassen.

6.1.1 Funktionsauswertung des Post-Processing

Die Nachbearbeitung des programmierten Ausweichmanövers setzt die Gegenüberstellung der Datenaufzeichnung mit dem impliziten Integrationsverfahren als Referenz voraus. Mittels unterschiedlicher Integrationsverfahren wird also geprüft, ob beim einfachen Fahrspurwechsel **carla_evasive_maneuver** (violetter Graph) eine Abweichung vom Vergleichswert vorliegt. Bei der Betrachtung der Kurvenverläufe des impliziten und expliziten Integrationsverfahrens ist ersichtlich, dass eine Abweichung von $\Delta y_{\max} \approx 0,07$ m im *PPP*-Programm (grüner Graph) vernachlässigbar ist:

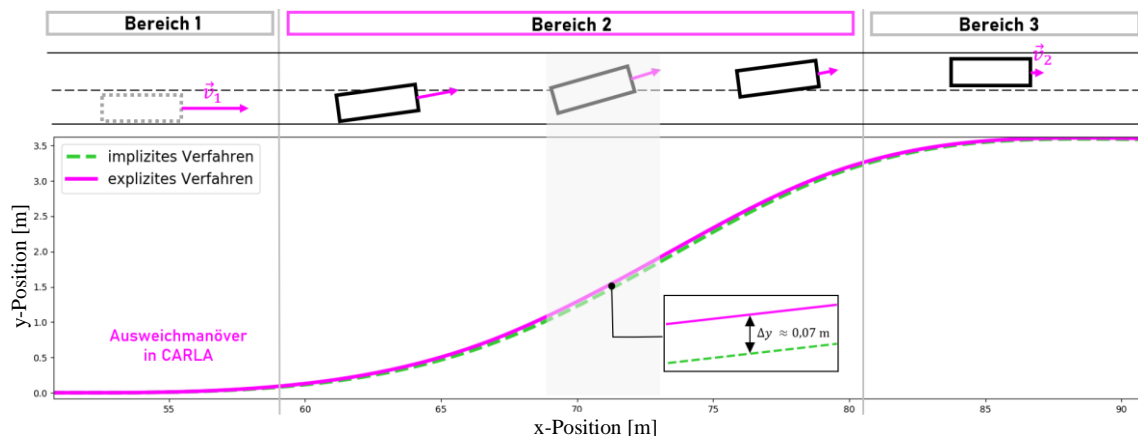


Abb. 55: Vergleich zwischen implizitem und explizitem Integrationsverfahren
[Eigene Darstellung] (vgl. A.5.4 PPP)

Die maximale Abweichung im Gesamtfahrzeugmodell ergibt sich im grau hinterlegten Abschnitt des Bereichs 2 (größter Integrationsfehler mit dem explizitem Euler-Verfahren). Kleinere Abweichungen können durch kleinere Schrittweiten erzielt werden, da so die Fehlerfortpflanzung reduziert wird. Im Rahmen dieser Arbeit wurde die Schrittweite $h \approx 10$ ms gewählt, die bei der x_0 -Koordinate $x \approx 72$ m einem relativen Fehler von ca. 4 % im anwendungsbezogenen Toleranzbereich entspricht. Aus den Ergebnissen im Post-Processing geht hervor, dass das programmierte Gesamtfahrzeugmodell mit dem explizitem Euler-Verfahren eine Abweichung im vernachlässigbaren Toleranzbereich besitzt.

Vor und nach einem Ausweichmanöver ist ein geringer Lenkradwinkeleinschlag für die Normalfahrt auszuführen, weil das Ego-Fahrzeug bereits parallel zur Fahrbahn ausgerichtet ist. Eine Überdeckung der Kurven in Bereich 1 und 3 zeigt, dass das explizite und implizite Verfahren für die programmierten Trajektorie näherungsweise kongruent sind. Der Vergleich der beiden Graphen verdeutlicht, dass das explizite Integrationsverfahren für die Anwendung in der Simulationssoftware CARLA ein reales Ausweichmanöver nachbildet. Mit Blick auf die Trajektorie überzeugt die Programmierung des Gesamtfahrzeugmodells C mit nichtlinearem Reifen- und Zweispurmodell im SiL-Prozess.

6.1.2 Post-Processing-Ergebnisse unterschiedlicher Fahrzeugtypen

Um die zweite eingangs gestellte Hypothese zu prüfen – dass die Trajektorien bei der Eingabe unterschiedlicher Fahrzeugdaten variieren – werden die Grenzen der Programmierung unter diesen Voraussetzungen behandelt. So wird das bisherige Referenzmodell, der Audi Q5, mit den eigenständig programmierten Fahrzeugen BMW, Volvo und Audi R8, deren jeweilige Fahrzeugdaten aus CarMaker (vgl. A.5.1 PDD und A.5.2 PPADD) und dem Passenger Car von Pauwelussen (vgl. Pauwelussen, 2015, S. 285f.) entnommen sind, verglichen. Die unterschiedlichen Parameter und geometrischen Daten des jeweiligen Fahrzeugs, dessen Fahrzeugmasse sowie dessen Massenträgheitsmoment sind aus den zuvor genannten Quellen entnehmbar. Einige nicht angegebene Größen – etwa die Lenkungs- und Reifeneigenschaften sowie die Spurweite – werden für die Versuche vom Modell des Audi Q5 als Ersatzwerte übernommen. Diese Annahmen beruhen auf Erfahrungswerten aus der Fahrzeugtechnik, die hinsichtlich der Plausibilitätsprüfung in dieser Masterarbeit als geeignet bewertet werden.

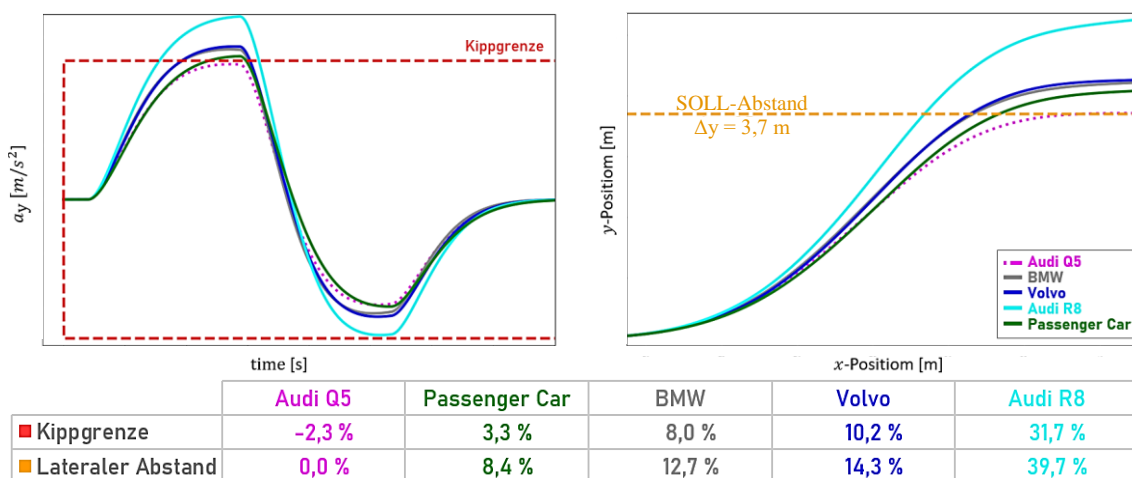


Abb. 56: Relative Abweichung der Fahrzeuge vom Referenzfahrzeug Audi Q5
[Eigene Darstellung] (vgl. A.5.4 PPP)

Das Gesamtfahrzeugmodell C und die entworfene Lookup-Table für das Referenzfahrzeug Audi Q5 sind bei allen Untersuchungen integriert, um herauszufinden, wie sich die Funktionalität des FAS bei den in Abbildung 56 aufgelisteten Fahrzeugen auswirkt. Die Betrachtung der relativen Abweichungen im Vergleich zum Audi Q5 verdeutlicht (vgl. Abbildung 56 links), dass die Kippgrenzen aller Vergleichsfahrzeuge überschritten werden. Alle relativen Abweichungen von Passenger Car, BMW, Volvo und Audi R8 liegen über 0 % (vgl. Abbildung 56: Kippgrenze, Lateraler Abstand) und können für die weiteren Schleifendurchläufe demnach nicht verwendet werden. Die Simulation muss beim Überschreiten der Kippgrenze immer abgebrochen und die Lookup-Table-Parameter für jedes Fahrzeug neu generiert werden. Falls es nicht zum Abbruch der Simulation kommt, können die x - y -Positionen der Trajektorien aufgrund fehlerhafter Berechnungsergebnisse nicht weiterverwendet werden (vgl. Abbildung 56 rechts). Infolgedessen darf die relative Abweichung des lateralen Abstandes nicht über 0 % betragen, soll die erwünschte Zielposition erreicht werden.

6.2 Gegenüberstellung der Gesamtfahrzeugmodelle

Die zu Beginn aufgestellte Hypothese – ob eine Einspurmodell-Simulation im realitätsnahen Versuch ein unzureichendes Fahrverhalten aufweist – lässt sich mit der nachfolgenden Tabelle belegen:

Tab. 9: Gegenüberstellung der Gesamtfahrzeugmodelle (vgl. Abbildung 3, A.5.1 PDD und A.5.2 PPADD)

		Ausgangszustand (Stand 01.05.2021)	Modell A (Stand 01.11.2021)	Modell B (Stand 01.11.2021)	Modell C (Stand 01.11.2021)
Ego-Fahrzeug	Gesamtfahrwiderstände	+	0 erweiterbar	0 erweiterbar	0 erweiterbar
	Sturz- und Spurwinkeländerung	-	-	-	0 erweiterbar
	Fahrzeugschwerpunkt	0	0	0	+ Schwerpunktdefinitor
	Reifenmodell	0 lin. Reifenmodell	0 lin. Reifenmodell	+ nichtlin. Reifenmodell	+ nichtlin. Reifenmodell
	Lenkungsmodell	- nicht vorhanden	+	+	+
	Fahrzeugmodell	- lin. ESM	- lin. ESM	0 nichtlin. ESM	+ nichtlin. ZSM, Schwingungen
	Brems- und Antriebsmodell	0 erweiterbar	0 erweiterbar	0 erweiterbar	0 erweiterbar
	ESP- und ABS-Eingriff	-	0 Annahme: aktiver Modus	0 Annahme: aktiver Modus	0 Annahme: aktiver Modus
	Physikalische Grenzen	0 $ a_{y,max} = 4 \text{ m/s}^2$	0 $ a_{y,max} = 4 \text{ m/s}^2$	+ $ a_{y,max} = a_{y,v/h,kipp}$	+ $ a_{y,max} = a_{y,v/h,kipp}$
Umwelt	Hindernisse auf der Fahrbahn (Gegenstand)	0 unterschiedliche geometrische Formen	+	+	+
	Parkende Autos	-	+	+	+
	Fahrradfahrer	-	+	+	+
	Fußgänger, Tiere	-	0 erweiterbar	0 erweiterbar	0 erweiterbar
	Reifen-Fahrzeug-Kontakt (Kraftschlussbeiwert)	- nicht definiert	0 vereinfacht	0 vereinfacht	0 vereinfacht
	Fahrbahnoberfläche	- keine vorhanden, Qt-Oberfläche	+	+	+
	Wetterverhältnis	- nicht definiert	0 vereinfacht	0 vereinfacht	0 vereinfacht
	Sichtinformation (Objekterkennung)	- Sichtweite 180 deg	0 plausible Annahmen	0 plausible Annahmen	0 plausible Annahmen
Simulation mit CARLA	Trajektorienverlauf	+ Kreisbogen	+ einfacher Fahrspurwechsel	+ einfacher Fahrspurwechsel	+ einfacher Fahrspurwechsel
	Fahrzeugvisualisierung	0 2D-Visualisierung	+ 3D-Visualisierung	+ 3D-Visualisierung	+ 3D-Visualisierung
	Nachbildung eines realen Verkehrsgeschehens	- nicht realitätsnah	+	+	+
	Fahrzeugnachbildung	- 2D-Darstellung, rechteckförmige Skizze	+	+	+
	Anpassungsfähigkeit der Programmierung	0 C++	+ Python und C++	+ Python und C++	+ Python und C++
	Schnittstelle mit Fahrsimulator	-	+	+	+
	Bedienbarkeit, Eingriffsmöglichkeit des Probanden	+	+	+	+
	Reproduzierbarkeit der Testfahrten	+	+	+	+
	Fehler- bzw. Abbruchmeldungen	- keine Grenzen oder Abbrüche	+ Fehlermeldung	+ Fehlermeldung	+ Fehlermeldung
FAS	Manöver zur Kollisionsvermeidung	+	+	+	+
	Erkennungsverfahren zur Objekterkennung	0 nicht technisch realisierbar	+	+	+
	Fahrsicherheit	-	+	+	+
	Ausfallsicherheit	0	0	0	0

Die visuelle Darstellung von Verkehrsteilnehmern und -umgebung bietet nur mit den geometrischen Formen als Hindernis keine plausiblen Abbildungen an. Der Nutzer besitzt ohne das *PPADD*-Programm keine Schnittstelle, um ein nachvollziehbares Ausweichmanöver im dreidimensionalen Raum illustriert wahrzunehmen. Wahrnehmungsphysiologische Einschränkungen entstehen durch die begrenzte 2D-Visualisierung (vgl. Abbildung 3 und Tabelle 9), welche durch die Verwendung des Fahrstimulators bei den Gesamtfahrzeugmodellen A – C nicht mehr vorliegt. Zusätzlich erschwert die zweidimensionale Ansicht somit die subjektive Validation in der SiL-Anwendung. Demnach ist das zu untersuchende Szenario zum Validieren der Gesamtfahrzeugmodellsimulation mit exakter Parametrier- und Reproduzierbarkeit ungenügend.

Das im Ausgangszustand verwendete Modell schneidet im Vergleich mit den Gesamtfahrzeugmodellen A – C am schlechtesten ab, welches der Umwelt und dem Ego-Fahrzeug geschuldet ist. Ein Einspurmodell im Ausgangszustand ist für ein hochdynamisches Ausweichmanöver nicht geeignet. In der Simulation wird nach der Beendigung des Ausweichmanövers (Kreisbogen) die Fahrsicherheit durch die ausgeführte Trajektorie nicht eingehalten. Die ursprüngliche visuelle Abbildung der Trajektorien basiert ausschließlich auf einer zweidimensionalen Ebene (vgl. Abbildung 3) und ist mithilfe der *PPADD*-Anwendung auf die dreidimensionale Ebene erweitert worden (vgl. A.5.2 *PPADD*). Somit ist der Ausgangszustand durch das *PPADD*-Gesamtfahrzeugmodell C zu ersetzen (vgl. Tabelle 9).

6.2.1 Analogie der Kollisionsvermeidungsmaßnahmen

Ein aktives Lenkmanöver kann in einem unübersichtlichen Verkehrsgeschehen einen potenziellen Unfall mit anderen Verkehrsteilnehmern verhindern. Das Startup hat im Ausgangszustand ein Ausweichmanöver mit einer konstanten Lenkradwinkelstellung simuliert, welche durch ein Ego-Fahrzeug in einem Abbiegevorgang in 2D ausgeführt wird (vgl. Abbildung 3). Dieses Konzept (Stand: 01.05.2021) mit der Verwendung eines linearen Reifen- und Einspurmodells erfüllt auf den ersten Blick das Kriterium, den Insassen vor einer Kollision zu bewahren. Die Skizze über den Verlauf der Trajektorie, bei der das Ego-Fahrzeug ein Ausweichmanöver nach links bzw. rechts ausführen kann, ist vereinfacht in der Abbildung 57 visualisiert.

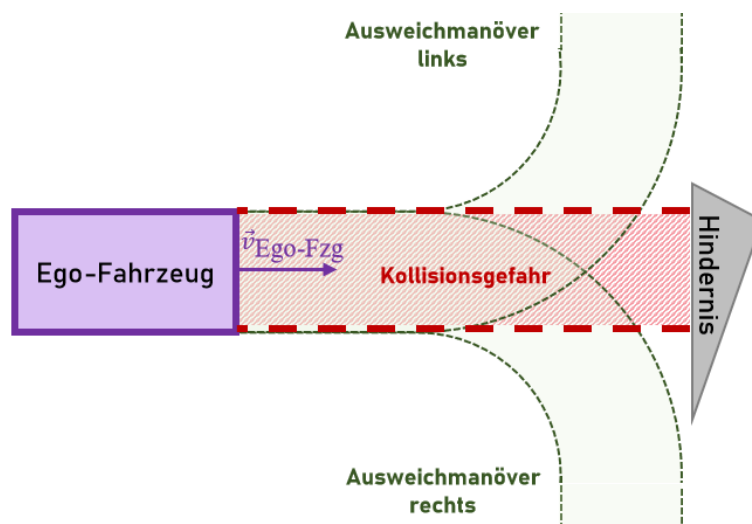


Abb. 57: Skizze der Trajektorienplanung mit Ausweichmanöver im Ausgangszustand (vereinfacht) (Stand: 01.05.2021) [Eigene Darstellung] (vgl. Startup TrajectoriX, 2021)

Nach dem prototypischen Ausweichmanöver ist infolge der fehlenden Fahrspurdefinitionen und undefinierten Fahrzeugausrichtung keine Gewährleistung der Fahrsicherheit vorhanden (vgl. Startup TrajectoriX, 2021). Die unterschiedlichen geometrischen Figuren als Hindernis, wie z. B. das graue Dreieck, und die Annahme der 180 deg-Objekterkennung sind außerhalb der Simulationsumgebung nicht auf den realen Straßenverkehr übertragbar (vgl. Abbildung 3 und 57). Eine realitätsnahe Simulation setzt voraus, dass die physikalischen Grenzwerte eingehalten und die Maßstäbe in 3D als Straßenbegrenzungen und Verkehrsteilnehmer (Fahrradfahrer, parkendes Auto etc.) illustriert werden. Somit ist im ersten Ansatz vor allem die Fahrbahn zu berücksichtigen, um die Reifen-Fahrbahn-Kontaktzonen für die propriozeptiven Assistenzsysteme nicht weiterhin zu vernachlässigen.

Die physikalischen Grenzen des ESM im Ausgangszustand sind nur bis zu einer Querbesehleunigung von maximal $4 \frac{\text{m}}{\text{s}^2}$ definiert und nicht auf hochdynamische Ausweichmanöver anwendbar (vgl. Tabelle 4). Folglich wurden im Rahmen dieser Abschlussarbeit ein nichtlineares ESM und ZSM im PDD-Programm entwickelt (vgl. A.5.1 PDD), die – aufgrund der Komplexität des Gesamtfahrzeugmodells – unter Berücksichtigung der Kippgrenze bei hohen Querbesehleunigungen ein Ausweichmanöver ermöglichen. Ebenso ist die Fahrsicherheit sowohl zu Beginn als bei Beendigung des einfachen Fahrspurwechsels stets durch das FAS gegeben. Der Zustandsautomat erhält das Eingangssignal und führt in einer Schleife ein autonomes, hochdynamisches Ausweichmanöver aus (vgl. Unterkapitel 5.3.3 und Abbildung 13). In Abbildung 58 bspw. messen die Umfeldsensoren einen

kurzen Bremsweg, um eine Kollision mit dem Fahrradfahrer zu vermeiden, und starten einen einfachen Fahrspurwechsel.

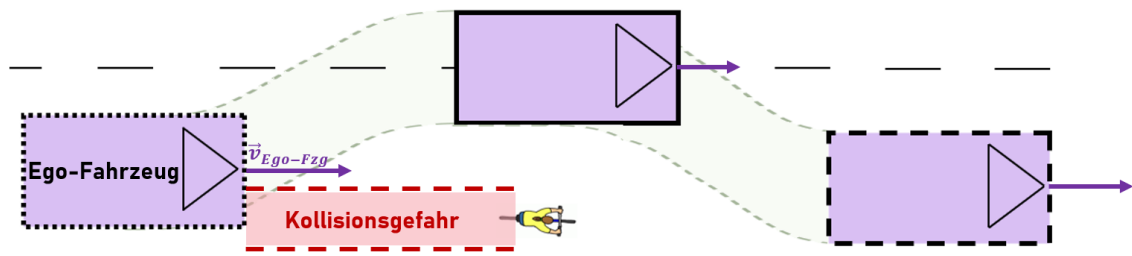


Abb. 58: Hochdynamisches Ausweichmanöver durch *PPADD*-Programmierung (Stand: 01.11.2021)
[Eigene Darstellung]

Die Lookup-Table im *PPADD*-Programm wählt in einer Schleife anhand von der Zielposition ein Lenkradwinkel-Zeit-Paar aus (vgl. Abbildung 35). Alle so übermittelten Datensätze werden ins Gesamtfahrzeugmodell eingelesen und anschließend von der Python-API an das Simulationsprogramm übertragen. Zum Ausführen eines zweiten Fahrspurwechsels, z. B. nach rechts, besitzt das Programm bereits die Vorlage zum Erstellen der fehlenden Datensätze (vgl. beschriebene Vorgehensweise in Unterkapitel 4.2.2). Diese *PPADD*-Lösungsmethode entspricht einem Open-Loop-Manöver mit einer Steuerung, das unabhängig vom Fahrereingriff funktioniert. Ein Konzept mit einer Regelung wurde nicht ausgearbeitet, da bereits viele derartige Ansätze auf dem Markt angeboten werden (vgl. Tabelle 1). Das Regelungskonzept ESA von ZF TRW Automotive Inc. (vgl. Abbildung 8) unterscheidet sich von *PPADD* darin, dass das hochdynamische Ausweichmanöver auf einem Punktmassenmodell und nicht auf einem nichtlinearen Zweispurmodell (vgl. A.5.2 *PPADD*) beruht. Der große Vorteil des Gesamtfahrzeugmodells C dieser Abschlussarbeit liegt in der geschwindigkeitsunabhängigen Nutzungsmöglichkeit des erweiterbaren Lookup-Tables (vgl. A.5.3 *PPADD*-Lookup-Table). Etwaige Zielpositionierungstools können die erstellten Datensätze für weiterführende Untersuchungen verarbeiten und ein hochdynamisches, autonomes Ausweichmanöver für jede Geschwindigkeit ausführen. Das Programm besitzt eine gegliederte Software-Architektur und ist mit Python-APIs erweiterbar.

6.2.2 Gesamtmodellvalidierung für die Simulation

Bei der Validierung der Gesamtfahrzeugmodelle gilt es, die Lenkradwinkel-Zeit-Kombinationen der *PPADD*-Lookup-Table sowie die physikalischen Grenzen zu prüfen. Beim Modell A gilt für alle Berechnungen die Kleinwinkelnäherung, da die Winkel aufgrund der maximalen Querbeschleunigung von $4 \frac{\text{m}}{\text{s}^2}$ linear angenommen werden. Dementgegen gilt bei nichtlinearen Reifen- und Einspurmodellen nicht $|a_{y,\text{max}}| \leq 4 \frac{\text{m}}{\text{s}^2}$, weil bei hochdynamischen Ausweichmanövern große Fahrzeugwinkel auftreten und die Querbeschleunigung $|a_{y,\text{max}}| \leq a_{y,v/h,\text{kipp}}$ als physikalische Grenze agiert. Beide Modelle können somit die Realität im hochdynamischen Anwendungsfall nicht nachbilden, weil diese Einspurmodelle kein Wanken und kein eventuelles Abheben eines kurveninneren Rades berücksichtigen (vgl. Abbildung 25). Im nachfolgenden wird ein exemplarisches Beispiel aus zuvor verglichenen Parametern für das Verhalten der Gesamtfahrzeugsimulationsmodelle dargestellt (vgl. A.5.1 PDD):

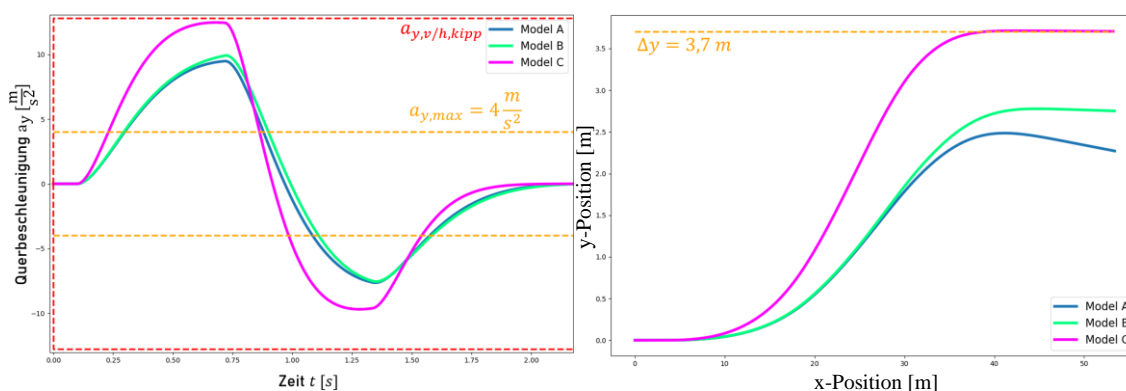


Abb. 59: Unterschied zwischen den Modellen – Übersicht der Querbeschleunigungen (links) und x-y-Positionen im Weltkoordinatensystem (rechts) [Eigene Darstellung] (vgl. A.5.1 PDD)

Im linken Diagramm ist die Überschreitung der ersten Querbeschleunigungsgrenze (gestrichelte orange Linie, $a_{y,\text{max}} = 4 \frac{\text{m}}{\text{s}^2}$) durch den blauen Graphen (Modell A) ersichtlich, woraus sich schließen lässt, dass das erste Modell für hochdynamische Ausweichmanöver ungeeignet ist (vgl. Abbildung 59 links und Tabelle 4). Modelle B und C liegen im Gegensatz dazu zwar unterhalb der Kippgrenze (gestrichelte rote Linie, $a_{y,v/h,\text{kipp}}$), jedoch muss weiter überprüft werden, ob die gewünschte Zielposition ebenfalls erreicht wird. Erst die Auswertung der rechten Grafik gibt Rückschlüsse darauf, dass der grüne Graph (Modell B) ein Ausweichmanöver mit $\Delta y_{\text{IST}} = 2,7 \text{ m}$ anstelle von $\Delta y_{\text{SOLL}} = 3,7 \text{ m}$ ausführt (vgl. Abbildung 59 rechts). Nur Modell C weist einen lateralen SOLL-Abstand auf und erfüllt das Kriterium für eine Trajektorie zur Kollisionsvermeidung. Für die Praxis würde die Nutzung des Gesamtfahrzeugmodells C mit der *PPADD*-Anwendung dazu

führen, dass die Fahrstabilität gewährleistet ist und die gewünschte Zielposition erreicht wird. Aus diesem Grund ist diese Applikation in Fahrversuchen zu empfehlen, um die simulierten Ergebnisse aus dem *PPADD*-Programm mit realen Messwerten zu vergleichen und das Fahrerassistenzsystem zu validieren.

Die simulierten Ausweichmanöver weisen im Simulationsprogramm CARLA messbare Unterschiede mit $\Delta y_{IST} = 2,7$ m statt $\Delta y_{SOLL} = 3,7$ m zwischen dem ESM und dem ZSM auf (vgl. Abbildung 59). In der Visualisierung mit dem Gesamtfahrzeugmodell B ist deutlich zu erkennen, dass der Fahrradfahrer vom Ego-Fahrzeug erfasst wird (vgl. Abbildung 60). Der Grund für die entstehende Kollision liegt in der *PPADD*-Lookup-Table, die für das nichtlineare ESM nur Informationen für ein Zweispurfahrzeug ausgibt. Der laterale Abstand bleibt von Bereich 2 bis 3 konstant und führt unweigerlich zu einem Unfall, da die Zielposition für einen Trajektorienverlauf nicht im *PPADD*-Programm definiert ist. Daraus lässt sich schließen, dass der Ausgangszustand und Modell A/ B mit einem (nicht-)linearen Einspurmodell für hochdynamische Ausweichmanöver nicht geeignet ist (vgl. Tabelle 9).

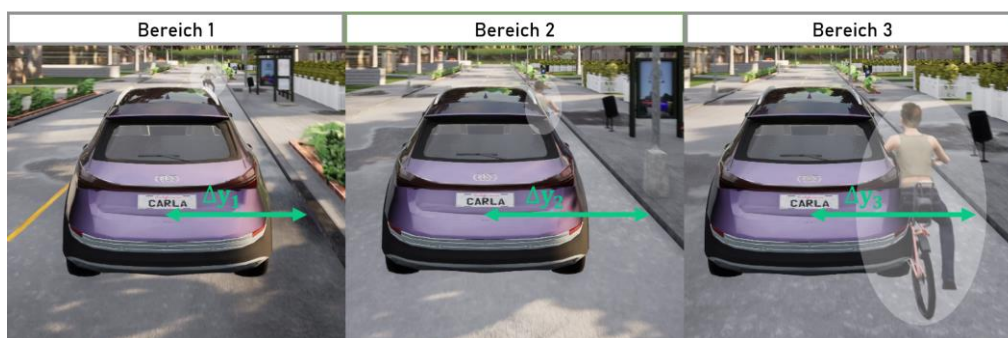


Abb. 60: Fahrradfahrer- Fahrzeug-Kollision mit Gesamtfahrzeugmodell B in CARLA
[Eigene Darstellung] (vgl. A.5.5 Aufzeichnung vom Fahrsimulator CARLA)

Im Gegensatz zu Modell B aus Abbildung 60 kann das Gesamtfahrzeugmodell C mit dem gleichen *PPADD*-Programm Trajektorien ohne Kollision ausführen. Die dazugehörige Lookup-Table ist in der Lage, das Ego-Fahrzeug in die gewünschte Zielposition in Bereich 3 zu navigieren. In der Abbildung 61 ist zu sehen, wie sich der laterale Abstand mit zunehmender Annäherung an den Radfahrer verändert. Mit dieser Vergrößerung des Abstandes vermeidet das Fahrzeug mit Gesamtfahrzeugmodell C aktiv eine Kollision.

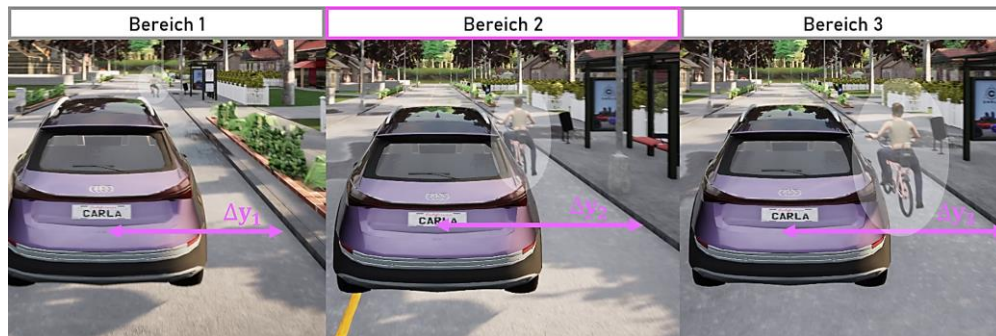


Abb. 61: Kollisionsvermeidung durch Ausweichen mit Gesamtfahrzeugmodell C [Eigene Darstellung] (vgl. A.5.5 Aufzeichnung vom Fahrsimulator CARLA)

6.3 Qualitative Bewertung der Gesamtfahrzeugmodelle

Die Forschung und Entwicklung setzt zunehmend auf funktionsbildende SiL-Systeme, die das Fahrerassistenzsystem mit seinen Teilkomponenten testen. Gesamtfahrzeugmodellierungen, Definitionen physikalischer Grenzen etc. tragen zum Verständnis der Ausweichmanöver und zur Analyse der Ursache-Wirkungskette bei Fahrsimulationen bei. Je nach Anforderung wird in der Simulation ein geeignetes Gesamtfahrzeugmodell verwendet, dessen Komplexitätsgrad von linearen und nichtlinearen Einspurmodellen bis hin zu Zweispurmodellen reicht. Die Vorgehensweise mit numerischen Berechnungsverfahren und erstellten *PPADD*-Lookup-Tables für Zweispurmodelle besitzt fünf Freiheitsgrade (vgl. Unterkapitel 3.3.3), die eine ausreichende Näherung zur Realität bieten. In der Vorentwicklung, der Produktentstehungsphase, ermöglicht die SiL die größte Beeinflussbarkeit der FAS-Eigenschaften (vgl. Abbildung 11). Voraussetzung für die Anwendung der universell veränderbaren Gesamtfahrzeugmodelle ist die Anpassbarkeit der physikalischen Größen an den jeweiligen Fahrzeugtyp.

Schließlich sind bei der Auswahl des Gesamtfahrzeugmodells das Ego-Fahrzeug, die Umwelt, die Simulation und das Fahrerassistenzsystem zu unterscheiden, die in Abbildung 62 qualitativ gegenübergestellt sind. Der Oberpunkt „Ego-Fahrzeug“ beinhaltet die Eigenschaften „Gesamtfahrzeugmodell“ (Lenkungs-, Reifen- und Fahrzeugmodell), „Physikalische Grenzen“ (Querbefleunigung: $|a_{y,\max}| = 4 \frac{\text{m}}{\text{s}^2}$, Kippgrenze $a_{y,v/h,\text{kipp}}$) und „Gesamtfahrwiderstände“. Im nächsten Oberpunkt „Umwelt“ sind die im Fahrsimulator vorhandenen Fahroberflächen, Wetterverhältnisse und Verkehrsteilnehmer enthalten. Des Weiteren wird im dritten Oberpunkt „Simulation“ die Programmierungsauswertung realitätsnaher Trajektorien unter Berücksichtigung des im System hinterlegten Fehlermeldungen und illustrierten Fahrzeugnachbildung evaluiert. Die Fahrsicherheit beim Ausweichmanöver, die Verwendung von ABS und ESP mit dem Zustandsautomaten werden unter dem letzten Punkt „Fahrerassistenzsystem“ bewertet (vgl. Abbildung 62). Je nach Bewertungskriterium kann die Evaluierung einer Eigenschaft nach subjektivem Empfinden in die positive oder in die negative Richtung neigen – vom Zahlenwert +2 bis -2 – wodurch auf einfachstem Wege die Abweichung vom zu erzielenden Optimum festgestellt werden kann. Eine qualitative Leistungsbeurteilung mit +2 bedeutet „sehr gut“, +1 „gut“, 0 „befriedigend“, -1 „ausreichend“ sowie -2 „ungenügend“ (vgl. Mintzlauff, 2016, S. 4 und Heißing, 2002, S. 11 – 16 und 27f.).

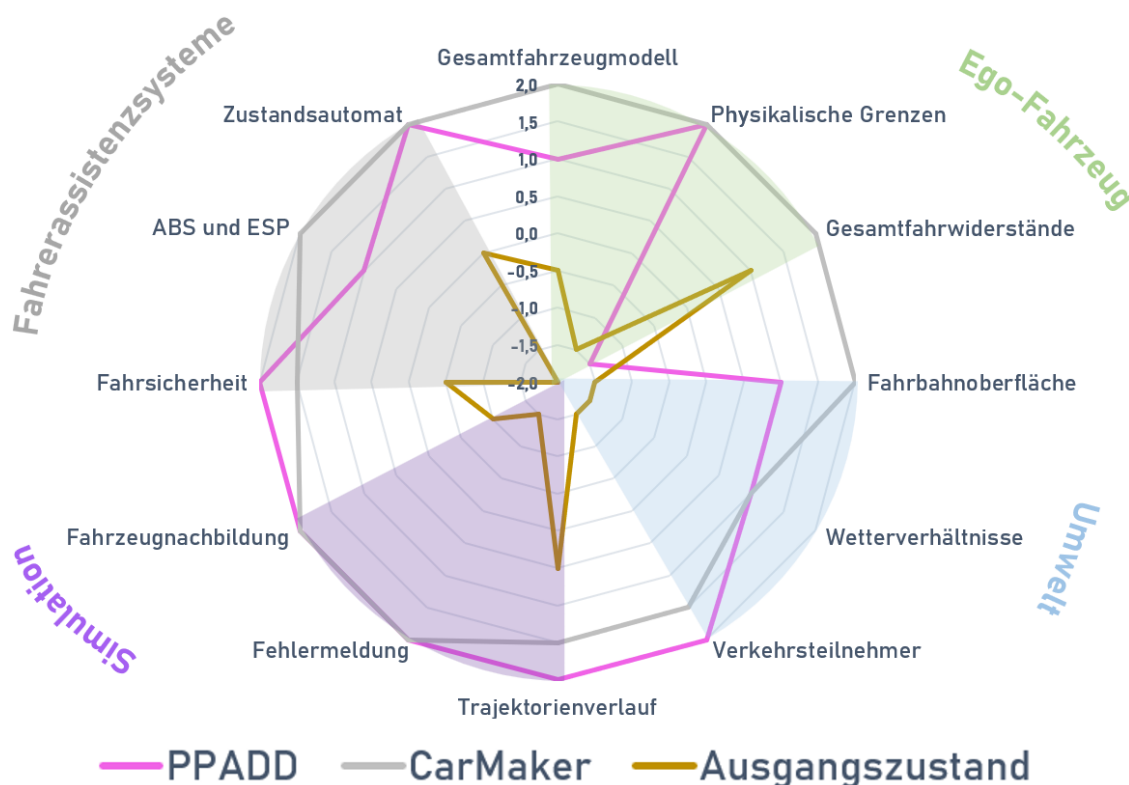


Abb. 62: Spinnennetzdiagramm zur Beurteilung der Fahrsimulationen im Vergleich [Eigene Darstellung] (in Anlehnung an Heißing, 2002, S. 11 – 16, 23 – 26, 34f. und 143 – 151)

Aufgrund der fehlenden Sicherheitsmaßnahmen beim Ausweichmanöver und der nicht vorhandenen Kippgrenze kann die Ausgangszustandsprogrammierung nicht überzeugen. Eine Verbesserung ist mit dem *PPADD*-Programm zu erzielen, die bezüglich Fahrerassistenzsystem, Simulation und Umwelt Parallelen zum Systemsimulator CarMaker aufweist. Die Abweichung vom Optimum im Fahrerassistenzsystem beim Unterpunkt ABS und ESP entsteht aufgrund der Vereinfachung der Kraftschlussbeiwerte auf $\mu = 1,0$ (vgl. Abbildung 46), wodurch die Umwelt in der Simulation keine Fahrbahnoberfläche mit einer Reifen-Fahrbahn-Kontaktzone besitzt. Der größte Unterschied ist unter dem Oberpunkt „Ego-Fahrzeug“ zwischen *PPADD*-Programm und CarMaker ersichtlich, da das Gesamtfahrzeugmodell C auf Lenkungs-, Reifen- und Fahrzeugmodell beschränkt ist und die Gesamtfahrwiderstände vernachlässigt. Dieses Modell kann jedoch trotz seiner einfachen Modellierung für einen großen Teil an Fahraufgaben am Fahrsimulator empfohlen werden, da es einen guten und weitgehend realistischen Fahrerimmersionsgrad wiedergibt (vgl. Tabelle 9 und Abbildung 62). Auch wenn hier einige Punkte vereinfacht und vernachlässigt werden (vgl. Abbildung 46), stellt das *PPADD*-Gesamtfahrzeugmodell C aufgrund seiner vergleichsweisen einfachen Parametrisierung ein vielseitig einsetzbares Modell dar.

7 Zusammenfassung, Diskussion und Ausblick

Das Ziel ist die Entwicklung eines verbesserten Gesamtfahrzeugmodells im Vergleich zum Ausgangszustand, das für hochdynamische, autonome Ausweichmanöver geeignet ist. Dieses *PPADD*-Modell bietet eine realitätsnahe Darstellung der Umwelt und des Gesamtfahrzeugmodells. Vor allem muss bei der Durchführung von Ausweichmanövern die Fahrzeugsicherheit für die Insassen gewährleistet sein, um Gefahr für Leib und Leben auszuschließen. Die Validierung von Gesamtfahrzeugmodell C ist vielversprechend und kann in weiteren Arbeiten mit ergänzenden Teilmodellen erweitert werden.

7.1 Zusammenfassung mit Schlussfolgerung

Das Einspurmodell im Ausgangszustand würde einfache anfängliche Ausweichmanöver zur Kollisionsvermeidung erfüllen. Die Auswahl eines geeigneten Lenkungs-, Reifen- und Fahrzeugmodells ist Voraussetzung für die Erstellung der Zielpositions-Koordinaten mithilfe der fahrzeugabhängigen Lookup-Table. Lenkradwinkel-Zeit-Kombinationen werden zunächst heuristisch und können bei Bedarf gradientenbasiert interpoliert werden. Schnittstellen zwischen der Python-API und dem Simulationsprogramm ermöglichen es, derartige Fahrmanöver mit den anfänglich gesetzten Anforderungen umzusetzen.

Eine Rekonstruktion der SiL-Simulation mit dem neuen *PPADD*-Programm bietet die Möglichkeit, das bisherige Gesamtfahrzeugmodell zu erweitern. Für die Integration des Modells mit dem *PPADD*-Programm in CARLA sind explizite Integrationsverfahren in der Game-Loop zu wählen. Die erste anfangs gestellte Hypothese, dass ein lineares Einspurmodell im realen Versuch ein unzureichendes Fahrverhalten ergibt, kann sowohl mit der Überschreitung der maximalen Querbeschleunigung $|a_{y,\max}| = 4 \frac{\text{m}}{\text{s}^2}$, als mit damit einhergehenden Kollisionen mit anderen Verkehrsteilnehmern belegt werden. Der *PPADD*-Trajektorienverlauf basiert auf einem einfachen Fahrspurwechsel, vergleichbar mit dem Open-Loop-Manöver Single Sine. Dieses nichtlineare Zweispurmodell berücksichtigt stets die Kippgrenze $a_{y,v/h,\text{kipp}}$, welche immer größer als $|a_{y,\max}|$ sein muss. Die Validierung des Gesamtfahrzeugmodells C zeigt, dass ein Einspurmodell für hochdynamische Ausweichmanöver, bestehend aus Brems- und Lenkmanöver, ungenügend ist. Im Gegensatz dazu überzeugt die Lenkungsmodellierung mit PT1-Glied und einem nichtlinearen Reifen- und Zweispurmodell im Modell C.

Unter Einhaltung der physikalischen Randbedingungen ist ein sicherheitskonformes Ausweichmanöver unabhängig von der jeweiligen Verkehrssituation und -umgebung realisierbar. Komplexe Rückwärtsberechnungen für die Ermittlung der Lenkradwinkel-Zeit-

Kombinationen ist ohne vereinfachende Annahmen mit einer Invertierung des Gesamtfahrzeugmodells C nicht lösbar. Das analytisch nicht lösbare Gleichungssystem zur Berechnung der fahrstreifenrelativen Fahrdynamik, bestehend aus Gleichungen (27) und (28), ergibt eine unendliche Lösungsmenge, weshalb eine Rückwärtsrechnung zur Ermittlung der Parameter mathematisch nicht definiert ist. Alternativ werden für rechenintensive Kollisionsvermeidungsgleichungssysteme Regelungskonzepte eingesetzt, die eine größere Freiheit bei der Trajektorienplanung mit einem Punktmassenmodell bieten. Sobald die Regelung mit einem Zustandsautomaten unter Anwendung der *PPADD*-Lookup-Table verwendet wird, ist die Nutzung des Gesamtfahrzeugmodells C auf Basis eines nichtlinearen Zweispurmodells für hochdynamische Ausweichmanöver möglich. Hierbei ist zu beachten, dass jede Wertetabelle nur für einen Fahrzeugtyp angelegt wird. Folglich sind bei anderen Fahrzeugen neue fahrzeugspezifische Lookup-Tables erforderlich. Bei der Verwendung der Datensätze des Referenzfahrzeugs für andere Fahrzeugtypen entstehen größere Abweichungen, bspw. liegt beim Versuchsfahrzeug BMW die relative Abweichung von der Kippgrenze bei 8,0 % und die des lateralen Abstandes bei 12,7 % (vgl. Abbildung 56). Eine Überschreitung dieser beiden physikalischen Grenzen führt zu einer falschen Simulation und zu einem in der realen Anwendung unplausiblen Fahrverhalten. Die zweite These, ob ein universeller Einsatz des *PPADD*-Programms mit verschiedenen Fahrzeugtypen möglich ist, wird somit widerlegt, da die Auswertungsergebnisse bei unterschiedlichen Fahrzeugdaten variieren.

Das Simulationsprogramm CARLA ist nicht darauf ausgelegt, die Raddrehzahlen der Fahrzeugreifen während der Versuchsfahrt aufzuzeichnen. Obwohl die Fahrbahnoberfläche anschaulich visualisiert wird, fehlen die physikalischen Parameter an den Reifen-Fahrbahn-Kontaktzonen, wie z. B. der Kraftschlussbeiwert. Eine Berücksichtigung der wechselnden Straßenbedingungen für die Reibwertermittlung ist im Simulationsprogramm CARLA daher nicht gegeben. Nicht vorhandene Werte erfordern es, Annahmen über die ABS- und ESP-Nutzung zu treffen, die – als konstant angenommene Größen – nicht mit realen Messergebnissen gleichzusetzen sind. Zum Ausführen der Ausweichmanöver sind die angenommenen Parameter jedoch plausibel und in der Visualisierung validierbar, ein Vergleich der vom Kraftschlussbeiwert abhängigen Reifenseitenkräfte mit dem Referenzmodell im Systemsimulator CarMaker weist nahezu kongruenten Verlauf nach. Der Trajektorienverlauf gleicht einem realen Ausweichmanöver, verursacht keine Kollision mit anderen Verkehrsteilnehmern und gewährleistet somit die Sicherheit für Leib und Leben im Straßenverkehr.

7.2 Diskussion des PPADD-Reifenmodells

Die letzte Hypothese, ob sich eine schlechte Fahrbahnoberfläche negativ auf das Fahrverhalten während der Kollisionsvermeidung auswirkt, ist im verwendeten Simulationsprogramm CARLA wegen der fehlenden Datengenerierung des Kraftschlussbeiwertes während der Testfahrt nicht belegbar. Der Reibungskoeffizient ist eine sich verändernde Variable, die je nach Beschaffenheit der Straßen- und Reifenoberfläche sowie abhängig von der Radausrichtung im Wertebereich zwischen 0,0 und $\approx 1,1$ liegt. Die Auswirkungen der Fahrbahnoberfläche auf die Reifenseitenkräfte des *PPADD*-Gesamtfahrzeugmodells C werden durch die Annahme einer ideal trockenen Fahrbahn $\mu = 1,0$ kompensiert (vgl. Abbildung 46). Sobald sich der Kraftschlussbeiwert verändert, muss entsprechend des Reibungskoeffizienten ein neues Lookup-Table erstellt werden. Anstelle dieser Vereinfachung wäre eine Zuordnung von definierten Parametern bei unterschiedlichen Fahrbahnoberflächen, abhängig von der Wetterlage, gleichfalls realisierbar. Eine trockene Asphaltstraße könnte weiterhin mit $\mu = 1,0$ und eine nasse mit $\mu = 0,6$ definiert werden, sodass die Berechnung der Reifenseitenkräfte die Kontaktzone berücksichtigen. Somit ist eine umweltrealistischere Plausibilisierung an ein Ausweichmanöver als Teststrecke mit dem *PPADD*-Programm umsetzbar.

Trotz definierter Kraftschlussbeiwerte ist eine Erweiterung des *PPADD*-Programms mit einem Antriebsmodell, bestehend aus Motor- und Getriebemodell, vorzuziehen. Dieser Ansatz gewährleistet, die vier Raddrehzahlen und folglich den Kraftschlussbeiwert unabhängig vom Simulationsprogramm CARLA und vom Trajektorienverlauf zu berechnen. Der Nachteil ist, dass – wie bei der *PPADD*-Lookup-Table – eine fahrzeugspezifische Modellierung des Antriebs vorausgesetzt wird. Eine weitere Modellierung zu umgehen, ist mit der Nutzung des Referenzsimulators CarMaker oder mit Messdaten vom Versuchsfahrzeug möglich. Das *PPADD*-Programm kann mit wenig Rechenzeit die Raddrehzahlen standardisierter Fahrmanöver verwenden. Ähnlich wie beim Zustandsautomaten besteht die Option, einen Kraftschlussbeiwert-Datensatz aufzurufen und in die Programmierung einzubinden. Daraus wird ersichtlich, dass durch die Einbeziehung der Kraftschlussbeiwerte die Ausweichmanöver in realitätsnahen Simulationen visualisiert werden können.

7.3 Ausblick

Die Ergebnisse der Validierung bestätigen die Richtigkeit der zu Beginn aufgestellten Hypothesen. Der auf einem Einspurmodell basierende Ausgangszustand wird mit dem neu entwickelten *PPADD*-Programm abgelöst, um kollisionsfreie hochdynamische Ausweichmanöver im Simulationsprogramm CARLA auszuführen. Eine Erweiterung des *PPADD*-Gesamtfahrzeugmodells mit einem Antriebsmodell könnte die derzeit nicht vorhandene Raddrehzahlberechnung möglich machen. Diese Berücksichtigung würde die Bestimmung des Kraftschlussbeiwertes und folglich die propriozeptiven Assistenzsysteme für Simulation bei unterschiedlichen Wetterverhältnissen und Fahrbahnoberflächen miteinbeziehen. Die zunehmende Steigerung des Komplexitätsgrades des Reifen- und Fahrzeugmodells sowie ein zusätzliches Antriebsmodell können z. B. die Vernachlässigungen der Sturz- und Spurwinkeländerung und Vereinfachungen wie $\mu = 1,0$ vermeiden. Des Weiteren ist die Erhöhung des Detaillierungsgrades mithilfe eines komplexeren Reifenmodells, wie bspw. dem FTire Modell, zu empfehlen (vgl. Gisper, 2001, S. 3).

Erst die Einbeziehung eines Mehrkörpersystems lässt während des Ausweichmanövers eine Visualisierung plausibler Fahrzeugpositionierungen im Weltkoordinatensystem zu (vgl. ebd.). Mehrkörpersysteme sind bauteilorientierte Simulationsmodelle, die nach der Erstellung und Validierung einer einmaligen Modellierung eine realitätsnähere Simulation nachstellen. Die komplexeren Modellvarianten mit unterschiedlichen Teilmodellen und Fahrndynamikbeschreibungen lassen detailliertere Fahrzeugbewegungen und Fahrverhalten im Fahr Simulator zu. Die erforderlichen Eingangsparameter für die Simulationen liegen den Fahrzeugherstellern vollständig vor, sind aber für Außenstehende kaum zugänglich. Die angenommenen Eingangsgrößen können aus Referenzmodellen und Fachliteratur simulativ bestimmt werden. Somit kann das für reale Testfahrten nicht vorhandene Fahrzeug auf der Entwicklungsumgebung als API validiert und im nächsten Entwicklungsprozess, im HiL-Prüfstand, integriert werden. Finalisiert wird ein Fahrerassistenzsystem durch reale Fahrversuche mit einer Messtechnik, die die Funktionalität des hochdynamischen, autonomen Ausweichmanövers unter allen umwelttechnischen Einflüssen garantiert.

Literaturverzeichnis

Ackermann C., Isermann R., Min S., Kim C. (2014): *Collision avoidance with automatic braking and swerving*, Proceedings of the 19th World Congress The International Federation of Automatic Control Cape Town, IFAC Proceedings Volumes, Volume 47, Issue 3.

Ammon D. (1997): *Modellbildung und Systementwicklung in der Fahrzeugdynamik*, B. G. Teubner, Stuttgart.

Audi AG (2011): *Audi pre sense*. Abgerufen am 20.08.2021 von <https://www.audi-technology-portal.de/de/elektrik-elektronik/sicherheitssysteme/audi-pre-sense>.

Audi AG (2016): *Ausweichassistent*. Abgerufen am 20.08.2021 von <https://www.audi-mediacycenter.com/de/fotos/detail/ausweichassistent-42335>.

Audi AG (2017): *Ausweichassistent*. Abgerufen am 20.08.2021 von <https://www.audi-mediacycenter.com/de/technik-lexikon-7180/fahrerassistenzsysteme-7184>.

Bastani O., Pu Y., Solar-Lezama A. (2018): *Verifiable Reinforcement Learning via Policy Extraction*, 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

Bengler K., Dietmayer K., Farber B., Maurer M., Stiller C., Winner H. (2014): *Three Decades of Driver Assistance Systems: Review and Future Perspectives*, in IEEE Intelligent Transportation Systems Magazine, vol. 6, no. 4.

Bevan G. P. (2008): *Development of a vehicle dynamics controller for obstacle avoidance*, Dissertation, University of Glasgow. Abgerufen am 20.08.2021 von <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.368.7212&rep=rep1&type=pdf>.

Bosch GmbH (2021): *Ausweichmanöver*, Deckblatt. Abgerufen am 20.08.2021 von <https://www.bosch-mobility-solutions.com/de/loesungen/assistenzsysteme/ausweichassistent/>.

Bundesgerichtshof (2010): *Urteil vom 21. September 2010 - VI ZR 263/09*, OLG Brandenburg.

Bootz A. (2004): *Konzept eines energiesparenden elektrohydraulischen Closed-Center-Lenkensystems für Pkw mit hoher Lenkleistung*, Dissertation Maschinenbau, Technische Universität Darmstadt. Abgerufen am 20.08.2021 von https://tuprints.ulb.tu-darmstadt.de/485/1/Dissertation_Bootz.pdf.

- Braess H. H., Seiffert U. (2013): *Vieweg Handbuch Kraftfahrzeugtechnik*, Springer Vieweg, ATZ/MTZ-Fachbuch, 7., aktualisierte Auflage.
- Brand W., Brenner P. (2012): *Functional Development under Conditions Similar to Series Production to Optimize the Steering Feel*, 21st Aachen Colloquium Automobile and Engine Technology.
- Breuer S., Rohrbach-Kerl A. (2015): *Fahrzeugdynamik Mechanik des bewegten Fahrzeugs*, ATZ/MTZ-Fachbuch, Springer Vieweg.
- CARLA (2021): *CARLA Dokumentation*. Abgerufen am 20.08.2021 von https://carla.readthedocs.io/en/latest/python_api/.
- CarMaker (2021): *CarMaker: Pkw und leichte Nutzfahrzeuge virtuell testen*. Abgerufen am 20.08.2021 von <https://ipg-automotive.com/de/produkte-services/simulation-software/carmaker/>.
- Carwow (2021): *Volkswagen Golf 7*. Abgerufen am 20.08.2021 von <https://www.carwow.de/volkswagen/golf/2013#gref>.
- Choi C., Kang Y., Lee S. (2012): *Emergency collision avoidance maneuver based on nonlinear model predictive control*, IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012).
- Dahmen W., Reusken A. (2008): *Numerik für Ingenieure und Naturwissenschaftler*, Springer, 2., korrigierte Auflage.
- D'Ambrosi D. (2020): *ESC sulle auto: 500 mila incidenti evitati in 25 anni*, LaGazzetta dello Sport. Abgerufen am 20.08.2021 von <https://www.sicurauto.it/news/sistemi-di-sicurezza/esc-sulle-auto-500-mila-incidenti-evitati-in-25-anni/>.
- DEKRA (2016): *Unfälle können vermieden werden im Verkehr, bei der Arbeit, zu Hause*, Zeitschrift. Abgerufen am 20.08.2021 von <https://www.dekra-roadsafety.com/media/dekra-verkehrssicherheitsreport-2016-de.pdf>.
- Deutscher Bundestag (2009): *Beantwortung eines Fragenkataloges zum Straßenbau, zur technischen Konstruktion, zu Lärmschutzansprüchen und zu den Auswirkungen eines Tempolimits*, Aktenzeichen: WD 5 – 3000 – 162/09.
- DIN ISO 8855 (2013): *Straßenfahrzeuge – Fahrzeugdynamik und Fahrverhalten – Begriffe*, International Organization for Standardization, ICS 01.040.43; 43.020.

Dosovitskiy A., Ros G., Codevilla F., Lopez A., Koltun V. (2017): *CARLA: An Open Urban Driving Simulator*, arXiv:1711.03938. Abgerufen am 20.08.2021 von <https://arxiv.org/pdf/1711.03938.pdf>.

Dudenredaktion [1] (2021): *Manöver auf Duden online*. Abgerufen am 20.08.2021 von <https://www.duden.de/rechtschreibung/Manoever#Bedeutung-2>.

Dudenredaktion [2] (2021): *Ausweichmanöver auf Duden online*. Abgerufen am 20.08.2021 von <https://www.duden.de/rechtschreibung/Ausweichmanoever>.

Eckert A., Hartmann B., Sevenich M., Rieth P. (2011): *Emergency steer & brake assist – a systematic approach for system integration of two complementary driver assistance systems*, 22nd International Technical Conference on the Enhanced Safety of Vehicles (ESV), Washington DC.

Eskandarian A., Soudbakhsh D. (2008): *Enhanced Active Steering System for Collision Avoidance Maneuvers*, 11th International IEEE Conference on Intelligent Transportation Systems.

Falcone P., Borrelli F., Asgari J., Tseng H. E., Hrovat D. (2007): *Predictive Active Steering Control for Autonomous Vehicle Systems*, IEEE Transactions on Control Systems Technology, vol. 15, no. 3.

Fernández-Llorca D., Milanés V., Alonso I. P., Gavilán M., Daza I. G., Pérez J., Sotelo M. Á. (2011): *Autonomous Pedestrian Collision Avoidance Using a Fuzzy Steering Controller*, IEEE Transactions on Intelligent Transportation Systems, vol. 12, Nr. 2.

Freenet (2021): *ISO-Wedeltest*. Abgerufen am 20.08.2021 von https://www.freenet.de/auto/tests/pictureshow/audi-q3-bmw-x1-vw-tiguan-ford-kuga--co-im-test_3193436_4718656_3193396_58.html.

Gallenbacher J. (2017): *Abenteuer Informatik: IT zum Anfassen für alle von 9 bis 99–vom Navi bis Social Media*. Springer-Verlag, 4. Auflage.

Gao Y., Gray A., Frasc J. V., Lin T., Tseng E., Hedrick J. K., Borrelli F. (2012): *Spatial Predictive Control for Agile Semi-Autonomous Ground Vehicles*, Spatial predictive control for agile semi-autonomous ground vehicles. In Proceedings of the 11th international symposium on advanced vehicle control.

Geany (2021): *Geany Dokumentation*. Abgerufen am 22.11.2021 von <https://www.geany.org/manual/dev/index.html>.

- Gipser M. (2001): *Reifenmodelle in der Fahrzeugdynamik: eine einfache Formel genügt nicht mehr, auch wenn sie magisch ist*, Tagungsband Seminar Fahrzeugtechnik und Simulation, SFT Graz, 2001.
- Götte C., Keller M., Rösmann C., Nattermann T., Haß C., Glander K.-H., Seewald A., Bertram T. (2016): *A Real-Time Capable Model Predictive Approach to Lateral Vehicle Guidance*, IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Windsor Oceanico Hotel, Rio de Janeiro, Brazil.
- Graf M. (2014): *Methode zur Erstellung und Absicherung einer modellbasierten Sollvorgabe für Fahrdynamikregelsysteme*, Dissertation Maschinenwesen, Technische Universität München. Abgerufen am 20.08.2021 von https://www.researchgate.net/publication/280559313_Methode_zur_Erstellung_und_Absicherung_einer_modellbasierter_Sollvorgabe_fur_Fahrdynamikregelsysteme.
- Günnel T. (2014): *TRW zeigt Notausweichassistenten*, Automobil Industrie. Abgerufen am 01.06.2021 von <https://www.automobil-industrie.vogel.de/trw-zeigt-notausweichassistenten-a-449589/>.
- Halfmann C., Holzmann H. (2003): *Adaptive Modelle für die Kraftfahrzeugdynamik*, Springer-Verlag.
- Hansen T., Schulz M., Knopp M., Konigorski U. (2016): *Trajektorienplanung für automatisierte Fahrstreifenwechsel*, ATZ - Automobiltechnische Zeitschrift. Abgerufen am 20.08.2021 von <https://link.springer.com/content/pdf/10.1007/s35148-016-0071-8.pdf>.
- Heißing B., Brandl H. J. (2002): *Subjektive Beurteilung des Fahrverhaltens*, Vogel Buchverlag, 1. Auflage, Würzburg.
- Heißing B., Ersoy M., Gies S. (2013): *Fahrwerkhandbuch: Grundlagen Fahrdynamik Komponenten Systeme Mechatronik Perspektiven*, Springer Vieweg, 4., überarbeitete und ergänzte Auflage.
- Hochschule München (2021): *EXIST-GRÜNDERSTIPENDIUM TRAJECTORIX*. Abgerufen am 23.11.2021 von https://www.hm.edu/allgemein/forschung_entwicklung/forschungsprojekte/projektetails/krug/exist_trajectorix.de.html.
- Hofauer S. A. (2017): *Fahrer-Fahrzeug-Interaktion einer automatisierten, kraftstoffeffizienten Fahrzeuglängsführung*, Dissertation, Technische Universität München. Abgerufen am 01.06.2021 von <https://mediatum.ub.tum.de/doc/1353704/1353704.pdf>.

- Isermann R. (2006): *Fahrdynamik-Regelung Modellbildung, Fahrerassistenzsysteme, Mechatronik*, ATZ/MTZ-Fachbuch, Friedr. Vieweg & Sohn Verlag, 2006.
- ISO 3888-1 (2018): *Passenger cars — Test track for a severe lane-change manoeuvre — Part 1: Double lane-change*, International Organization for Standardization.
- ISO 3888-2 (2011): *Passenger cars — Test track for a severe lane-change manoeuvre — Part 2: Obstacle avoidance*, International Organization for Standardization.
- ISO 7401 (2011): *Road vehicle — Lateral transient response test methods — Open-loop test methods*, International Organization for Standardization.
- Iwano K., Raksincharoensak P., Nagai M. (2014): *A Study on Shared Control between the Driver and an Active Steering Control System in Emergency Obstacle Avoidance Situations*, Proceedings of the 19th World Congress The International Federation of Automatic Control Cape Town, IFAC Proceedings Volumes, Volume 47, Issue 3.
- Kallmeyer F. (2019): *Autonomes Fahren – Chancen und Herausforderungen, Zukunft Mobilität*. Abgerufen am 20.08.2021 von <https://www.zukunft-mobilitaet.net/170765/strassenverkehr/autonomes-fahren-chancen-und-herausforderungen-sae-level5/>.
- Katriniok A., Maschuw J. P., Christen F., Eckstein L., Abel D. (2013): *Optimal vehicle dynamics control for combined longitudinal and lateral autonomous vehicle guidance*, 2013 European Control Conference (ECC).
- Kehl S. [1] (2007): *Querregelung eines Versuchsfahrzeugs entlang vorgegebener Bahnen*, Dissertation, Universität Stuttgart. Abgerufen am 20.08.2021 von <https://d-nb.info/997159340/34>.
- Kehl S., Pölsler W.-D., Zeitz M. [2] (2007): *Querregelung eines Versuchsfahrzeugs entlang vorgegebener Bahnen (Lateral Control of a Test Vehicle along Predefined Paths)*, Zeitschrift at – Automatisierungstechnik, Jg. 55, Nr. 6.
- Keller M., Haß C., Seewald A., Bertram T. [1] (2015): *A Model Predictive Approach to Emergency Maneuvers in Critical Traffic Situations*, IEEE 18th International Conference on Intelligent Transportation Systems.
- Keller M., Haß C., Seewald A., Bertram T. [2] (2015): *A vehicle lateral control approach for collision avoidance by emergency steering maneuvers*, Springer Fachmedien Wiesbaden, 6th International Munich Chassis Symposium, 2015.

- Keller M., Haß C., Seewald A., Bertram T. (2014): *Driving Simulator Study on an Emergency Steering Assist*, IEEE International Conference on Systems, Man, and Cybernetics, San Diego, CA, USA.
- Keller M. (2017): *Trajektorienplanung zur Kollisionsvermeidung im Straßenverkehr*, VDI Verlag, Dissertation TU Dortmund.
- König L., Neubeck J., Wiedemann J. (2007): *Nichtlineare Lenkregler für den querdynamischen Grenzbereich (Nonlinear Steering Controllers for the Lateral Dynamics Stability Limit)*, Zeitschrift at – Automatisierungstechnik.
- König L. (2009): *Ein virtueller Testfahrer für den querdynamischen Grenzbereich*, Schriftenreihe des Instituts für Verbrennungsmotoren und Kraftfahrwesen der Universität Stuttgart, Band 43, expert-Verlag.
- Kontes G. D., Scherer D. D., Nisslbeck T., Fischer J., Mutschler C. (2020): *High-Speed Collision Avoidance using Deep Reinforcement Learning and Domain Randomization for Autonomous Vehicles*, IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC).
- Kramer U. (2008): *Kraftfahrzeugführung Modelle - Simulation - Regelung*, Hanser.
- Leemann T. (2017): *Effiziente Kollisionserkennung für konvexe Körper in parallelen Simulationen*, Lehrstuhl für Informatik 10 (Systemsimulation), Bachelorarbeit, Friedrich-Alexander-Universität Erlangen-Nürnberg. Abgerufen am 15.11.2021 von https://www10.cs.fau.de/publications/theses/2017/Leemann_BT_2017.pdf.
- Lipp P., Sadeghi A.-R., Koch K.-M. (2008): *Trusted Computing - Challenges and Applications*. First International Conference on Trusted Computing and Trust in Information Technologies, Trust 2008, LNCS 4968, Lecture Notes in Computer Science, vol 4968. Springer, Berlin, Heidelberg.
- Lippold C. (2006): *RAA: Ganzheitliche Richtlinie für den Entwurf von Autobahnen, Kolloquium "Neue Richtlinie für den Straßenentwurf"*, Lehrstuhl Gestaltung von Straßenverkehrsanlagen. Abgerufen am 20.08.2021 von https://www.ise.kit.edu/rd_download/SEB/Kolloquium_SEB_06-12_C._Lippold.pdf.
- MathWorks (1996): *MATLAB The Language of Technical Computing, Getting Started with MATLAB, Version 5*. Abgerufen am 20.08.2021 von <http://www-eio.upc.es/lceio/manuals/matlab/TECHDOC/PDFDOCS/GETSTART.PDF>.

MathWorks (2015): *Optimization Toolbox User's Guide*. Abgerufen am 20.08.2021 von <https://pdfroom.com/books/optimization-toolbox-users-guide/E1d4DjlLdOb/download>.

MathWorks [1] (2021): *Vehicle Dynamics Blockset*. Abgerufen am 22.11.2021 von <https://de.mathworks.com/help/vdynblks/>.

MathWorks [2] (2021): *MATLAB mit anderen Programmiersprachen nutzen*. Abgerufen am 13.11.2021 von <https://de.mathworks.com/products/matlab/matlab-and-other-programming-languages.html>.

Mayr R. (2001): *Regelungsstrategien für die automatische Fahrzeugführung Längs- und Querregelung, Spurwechsel- und Überholmanöver*, Springer-Verlag.

Menhour L., d'Andréa-Novel B., Fliess M., Mounier H. (2014): *Coupled nonlinear vehicle control: Flatness-based setting with algebraic estimation techniques*, Control Engineering Practice, Elsevier. Abgerufen am 20.08.2021 von <https://hal-polytechnique.archives-ouvertes.fr/hal-00869053/document>.

Meywerk M. (2007): *CAE-Methoden in der Fahrzeugtechnik*, Springer-Verlag Berlin Heidelberg.

Mintzlaff J. (2016): *Ablauf des VTP Subjektive Fahrdynamikbeurteilung*, Dokumentation eines Versuchs Technischen Praktikums (VTP) an der Hochschule München. Abgerufen am 16.11.2021 von https://w3-mediapool.hm.edu/mediapool/media/fk03/fk03_lokal/verbundlabore/messundfahrzeugtechnik/kraftfahrzeugtechnik/sonstiges_6/vtp_sf/VTP_subjektive_Fahrdynamikbeurteilung_Ablauf1.pdf.

Mitschke M., Wallentowitz H. (2014): *Dynamik der Kraftfahrzeuge*, Springer Vieweg, 5., überarbeitete und ergänzte Auflage.

Modellboard (2013): *Reifenschablone*. Abgerufen am 20.08.2021 von <https://www.modellboard.net/index.php?topic=45525.0>.

Pacejka H. B. (2012): *Tyre and Vehicle Dynamics*, Butterworth-Heinemann, 3. Auflage.

Pauwellussen J. P. (2015): *Essentials of Vehicle Dynamics*, Butterworth-Heinemann.

Pfeffer P. E. (2015): *6th International Munich Chassis Symposium 2015*, Springer Vieweg.

Pfeffer P. E., Harrer M. (2011): *Lenkungshandbuch Lenksysteme, Lenkgefühl, Fahrdynamik von Kraftfahrzeugen*, Vieweg+Teubner Verlag, 1. Auflage.

Pütz R., Serné T. (2019): *Rennwagentechnik – Praxislehrgang Fahrdynamik Eine praktische Anleitung für Amateure und Profis*, Springer Vieweg, 2., aktualisierte Auflage.

Qt Creator (2021): *Code Editor Example*. Abgerufen am 22.11.2021 von <https://doc.qt.io/qt-6/qtwidgets-widgets-codeeditor-example.html>.

Reif K. (2010): *Fahrstabilisierungssysteme und Fahrerassistenzsysteme*, Wiesbaden: Vieweg + Teubner Verlag, 1. Auflage.

Rupp C., Queins S., SOPHISTen (2012): *UML 2 glasklar*, Carl Hanser Verlag GmbH & Co. KG.

Rupp C., SOPHISTen (2021): *Requirements-Engineering und -Management Das Handbuch für Anforderungen in jeder Situation*, Hanser Verlag, 7., aktualisierte und erweiterte Auflage.

Russel S., Norvig P. (2012): *Künstliche Intelligenz – ein moderner Ansatz*, Pearson Verlag, 3., aktualisierte Auflage.

SAE J670E (1976): *Vehicle Dynamics Terminology*. Abgerufen am 20.08.2021 von <https://fddocuments.in/reader/full/sae-j670e-1976-vehicle-dynamics-terminology>.

Schindler E. (2019): *Fahrdynamik Grundlagen des Lenkverhaltens und ihre Anwendung für Fahrzeugregelsysteme*, expert, 3., aktualisierte Auflage.

Schmidt A. G., Weisz G., French M. (2017): *Evaluating Rapid Application Development with Python for Heterogeneous Processor-Based FPGAs*, 2017 IEEE 25th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM).

Schönemann B., Henze R., Küçükay F., Kudritzki D. (2013): *Auswirkungen der Rekupe-ration auf die Fahrdynamik*, ATZ PEER Review. Abgerufen am 20.08.2021 von <https://link.springer.com/content/pdf/10.1007/s35148-013-0149-5.pdf>.

Schöttle M. (2011): *Zukunft der Fahrerassistenz Mit neuen E/E-Architekturen*, ATZ Elektron 6. Abgerufen am 22.11.2021 von <https://link.springer.com/content/pdf/10.1365/s35658-011-0055-0.pdf>.

Sippl M. (2018): *Technische Voraussetzungen beim „autonomen Fahren“*, Verkehrsreferenten-Tagung 2018, ADAC Ostwestfalen-Lippe e.V. Abgerufen am 15.11.2021 von https://www.adac-owl.de/wp-content/uploads/2018/03/Fahrerassistenzsysteme_Automatisiertes-Fahren_Sippl_OWL-1.pdf.

- Schorn M., Isermann R. [1] (2006): *Automatic steering and braking for a collision avoiding vehicle*, Institute of Automatic Control, TU Darmstadt, IFAC Proceedings Volumes, Volume 39, Issue 16.
- Schorn M., Stahlin U., Khanafer A., Isermann R. [2] (2006): *Nonlinear trajectory following control for automatic steering of a collision avoiding vehicle*, 2006 American Control Conference.
- Schramm D., Hiller M., Bardini R. (2018): *Modellierung und Simulation von der Dynamik von Kraftfahrzeugen*, Springer Vieweg, 3., aktualisierte und ergänzte Auflage.
- Seewald A., Haß C., Keller M., Bertram T. (2015): *Notausweichassistent zur Vermeidung von Kollisionen*, ATZ - Automobiltechnische Zeitschrift. Abgerufen am 20.08.2021 von <https://link.springer.com/content/pdf/10.1007/s35148-015-0352-7.pdf>.
- Serarslan B. (2015): *Robuste Querdynamikregelung von Fahrzeugen mit aktiven Lenk-systemen*, Dissertation, Universität Duisburg-Essen. Abgerufen am 01.06.2021 von https://duepublico2.uni-due.de/servlets/MCRFileNodeServlet/duepublico_derivate_00040720/Serarslan_Diss.pdf.
- Sievers G., Graf S., Peperhowe M., Seiger C. (2018): *Simulation von Sensoren im SiL- und HiL-Umfeld*, ATZ Extra. Abgerufen am 20.08.2021 von <https://link.springer.com/content/pdf/10.1007/s35778-018-0038-3.pdf>.
- Smith D. E., Starkey M., Benton R. E. (1995): *Nonlinear-gain-optimized controller development and evaluation for automated emergency vehicle steering*, Proceedings of 1995 American Control Conference.
- Söhnitz I. (2001): *Querregelung eines autonomen Straßenfahrzeugs*, VDI Fortschritt-Berichte, Bd. 882, Reihe 8, Technische Universität Braunschweig, Dissertation.
- Sokolova (2011): *Lenkung*. Abgerufen am 20.08.2021 von http://www.sokolova.de/projekte_lenkung.html.
- Solmaz S., Akar M., Shorten R. (2008): *Adaptive Rollover Prevention for Automotive Vehicles with Differential Braking*, The International Federation of Automatic Control.
- Startup TrajectoriX (2021): *Ausweichmanöver*. Abgerufen am 02.07.2021 von <http://trajectorix.de/>.
- Talvala K. L. R., Kritayakirana K., Gerdes J. C. (2011): *Pushing the limits: From lane-keeping to autonomous racing*, Annual Reviews in Control, Volume 35, Issue 1.

- Trzesniowski M. (2019): *Fahrwerk*, Springer Vieweg, 2., aktualisierte Auflage.
- Trzesniowski M. (2014): *Rennwagentechnik*, Springer Vieweg, 4., überarbeitete und erweiterte Auflage.
- Unreal Engine (2021): Plugins Module Index. Abgerufen am 22.11.2021 von <https://docs.unrealengine.com/4.26/en-US/API/Plugins/>.
- Verordnung [EU] der Kommission (2012): *Verordnung (EU) zur Durchführung der Verordnung (EG) Nr. 661/2009 des Europäischen Parlaments und des Rates über die Typgenehmigung von Notbremsassistenzsystemen für bestimmte Kraftfahrzeugklassen*.
- Volkswagen (2021): *Volkswagen Konfigurator*. Abgerufen am 20.08.2021 von https://www.volkswagen.de/de/konfigurator.html/__app/der-neue-golf/der-neue-golf/life.app?
- Wang Y. (2011): *Aspekte der Simulation und Regelung der Querdynamik von Kraftfahrzeugen Regelungsstrategie für die Lenkungstechnik bei Fahrzeugen*, Diplomica Verlag, 1. Auflage.
- Werling R., Gröll L., Bretthauer G. (2008): *Ein Multiregler zur Erprobung vollautonomen Fahrens A Multi Controller for Testing Full-autonomous Driving*, Zeitschrift at - Automatisierungstechnik. Abgerufen am 20.08.2021 von <https://www.researchgate.net/publication/277429815>.
- Wikipedia (2021): *Bremseingriffe bei μ -split-Bedingungen*. Abgerufen am 20.08.2021 von <https://de.wikipedia.org/wiki/Datei:M%C3%BC-Split-Conditions.svg>.
- Winner H., Hakuli S., Lotz F., Singer C. (2015): *Handbuch Fahrerassistenzsysteme Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort*, Springer Vieweg, 3. Auflage.
- Wissing C., Nattermann T., Glander K.-H., Seewald A., Bertram T. (2016): *Environment Simulation for the Development, Evaluation and Verification of Underlying Algorithms for Automated Driving*, VDE Verlag GmbH, Berlin.
- Woyand H.-B. (2019): *Python für Ingenieure und Naturwissenschaftler Einführung in die Programmierung, mathematische Anwendungen und Visualisierungen*, Hanser Verlag, 3., überarbeitete und erweiterte Auflage, 2019.
- Zomotor A. (1991): *Fahrwerktechnik: Fahrverhalten*, Vogel Buchverlag, 2. Auflage.

Abbildungsverzeichnis

Deckblatt: Ausweichmanöver (von Bosch)	
Abb. 1: Ablauf für die Programmierung eines geeigneten Gesamtfahrzeugmodells.....	2
Abb. 2: Kollisionsvermeidungsmanöver.....	3
Abb. 3: Skizze zur Illustration möglicher Ausweichmanöver (Ausgangszustand) (Stand: 01.05.2021).....	5
Abb. 4: Typischer Kraftschlussbeiwert-Schlupf-Verlauf eines Fahrzeugreifens (links) und Bremsengriffe bei μ -Split-Bedingungen (rechts)	8
Abb. 5: Bremsengriff bei Fahrdynamikregelung bei a) untersteuerndem und b) übersteuerndem Fahrzeug	9
Abb. 6: Bremsassistent <i>Audi pre sense plus</i>	11
Abb. 7: Ausweichassistent <i>Audi pre sense front</i>	12
Abb. 8: Modellprädiktives Regelungskonzept: Fahrzeugquer- und Lenkradwinkelregelung	13
Abb. 9: Interaktionsschleife zwischen Agent und Umgebung.....	15
Abb. 10: Generierung eines Entscheidungsbaumes durch neuronale Netze.....	16
Abb. 11: V-Modell	17
Abb. 12: Implementierung aus verschiedenen Entwicklungsumgebungen	20
Abb. 13: Darstellung des Fahrtrainers CARLA	21
Abb. 14: Fahrzeugfestes und ortsfestes Achsensystem.....	24
Abb. 15: Doppelspurwechsel im Trajektorienverlauf	25
Abb. 16: Abmaße des ISO Wedeltests nach ISO 3888-1.....	26
Abb. 17: Pylonengasse und Fahrzeugtrajektorie des VDA-Ausweichtests nach ISO 3888-2.....	27
Abb. 18: Einzelsinus nach ISO 7401	29
Abb. 19: Planung in vier Ebenen für die Modellierung eines virtuellen Gesamtfahrzeugs	30
Abb. 20: Reifenkräfte im Reifenlatsch und kinematische Größen am Reifen	34
Abb. 21: Interpretation der Magic Formula-Koeffizienten.....	35
Abb. 22: Struktur des Gesamtfahrzeugmodells	36
Abb. 23: Mathematische Beschreibung des Einspurmodells	37
Abb. 24: Mathematische Beschreibung des Zweispurmodells	40
Abb. 25: Annahmen und Vereinfachungen des Zweispurmodells in der y_A - z_A - Ebene.....	41
Abb. 26: Annahmen und Vereinfachungen des Zweispurmodells auf der x_A - z_A - Ebene.....	44
Abb. 27: Qualitativer Lenkradwinkelverlauf eines Single Sine Tests	46
Abb. 28: Qualitative Ergebnisse des Einzelsinus: Schwimmwinkelverlauf (links) und Giergeschwindigkeitsverlauf (rechts) zur Überprüfung der Plausibilität der MiL-Simulation.....	46
Abb. 29: Abstraktion eines Ausweichmanövers mit Start- und Zielzustand	47
Abb. 30: Darstellung relevanter Elemente zur Kollisionsvermeidung	49
Abb. 31: Vergleich zweier verschiedener Lenkradwinkelverläufe (links) und die daraus resultierenden Positionen im Weltkoordinatensystem (rechts)	50

Abb. 32: Unterteilung eines Ausweichmanövers in drei Bereiche	51
Abb. 33: Optimierungsverfahren zur Ermittlung der Koeffizienten der Magic Formula nach Pacejka	54
Abb. 34: Optimierungsverfahren zur Ermittlung der Koeffizienten der Magic Formula – Optimierungsschleifen mit der SOLL-Seitenkraftkurve und Anzahl der Schleifen	56
Abb. 35: Überblick über das Konzept für ein hochdynamisches Ausweichmanöver	57
Abb. 36: Fahrdynamische Ergebnisse aus drei verschiedenen Ausweichmanövern mit den lateralen Abständen $\Delta y_{min} = 0,0$ m, $\Delta y_{spur} = 3,7$ m und $\Delta y_{max} = 14,5$ m: Querb beschleunigung (links) und x - y -Position im ortsfesten Koordinatensystem (rechts).....	59
Abb. 37: Übersicht über die <i>PDD</i> -Softwaremodule und -funktionen.....	60
Abb. 38: Lenkradwinkelverlauf eines Ausweichmanövers mit den Lenkradwinkel- Zeit-Paaren aus der Lookup-Table.....	62
Abb. 39: Geschwindigkeitsreduzierungsprozess bei einem Ausweichmanöver.....	62
Abb. 40: Lenkungsmodell	63
Abb. 41: Reifenmodellierung mittels Magic Formula nach Pacejka	64
Abb. 42: Nichtlineares Zweispurmodell	65
Abb. 43: Wankmodell	65
Abb. 44: Nickmodell.....	66
Abb. 45: Berechnung der x - y -Positionen im Weltkoordinatensystem	66
Abb. 46: Fahrer-Fahrzeug-Umwelt-Fahrerassistenz-Regelkreis (oben) und Vereinfachungen im Fahrer-Fahrzeug-Umwelt-FAS-Regelkreis (unten)	69
Abb. 47: Erster Fahrspurwechsel im VDA-Ausweichtest in CarMaker	71
Abb. 48: 3D-Visualisierung in MATLAB/Simulink	72
Abb. 49: Bilderserie eines Ausweichmanövers in CARLA.....	72
Abb. 50: Schematischer Überblick der <i>PPADD</i> -Softwaremodule und -funktionen.....	73
Abb. 51: Vorwärtsrechnung mit explizitem Integrationsverfahren	75
Abb. 52: Struktur einer txt-Datei mit dem entwickelten Datenformat.....	76
Abb. 53: Zustandsautomat des Fahrerassistenzsystems.....	77
Abb. 54: Unterscheidung zwischen gültigem und ungültigem Bereich (skizziert) anhand dreier Beispielwerte aus dem <i>PPADD</i> -Lookup-Table bei einer Relativgeschwindigkeit von 100 km/h für ein Ausweichmanöver	78
Abb. 55: Vergleich zwischen implizitem und explizitem Integrationsverfahren	81
Abb. 56: Relative Abweichung der Fahrzeuge vom Referenzfahrzeug Audi Q5.....	82
Abb. 57: Skizze der Trajektorienplanung mit Ausweichmanöver im Ausgangszustand (vereinfacht) (Stand: 01.05.2021)	86
Abb. 58: Hochdynamisches Ausweichmanöver durch <i>PPADD</i> -Programmierung (Stand: 01.11.2021).....	87
Abb. 59: Unterschied zwischen den Modellen – Übersicht der Querb beschleunigungen (links) und x - y -Positionen im Weltkoordinatensystem (rechts)	88
Abb. 60: Fahrradfahrer- Fahrzeug-Kollision mit Gesamtfahrzeugmodell B in CARLA	89
Abb. 61: Kollisionsvermeidung durch Ausweichen mit Gesamtfahrzeugmodell C	90

Abb. 62: Spinnennetzdiagramm zur Beurteilung der Fahrsimulationen im Vergleich	92
Abb. 63: Generierung eines Entscheidungsbaum durch neuronale Netze	112
Abb. 64: Überblick der Simulink-Datei, Schritt 1	121
Abb. 65: Überblick der Simulink-Datei, Schritt 2	121
Abb. 66: Übersicht der Simulink-Datei, Schritt 3	122
Abb. 67: Auswahl der <i>Fahrbahn.mat</i> -Datei für Bird's-Eye-Scope	122
Abb. 68: Bird's-Eye Scope	122
Abb. 69: Schematischer Übersicht der <i>PDD</i> -Softwaremodule und -funktionen	123
Abb. 70: Schematischer Überblick der <i>PPADD</i> -Softwaremodule und -funktionen [Eigene Darstellung]	124
Abb. 71: Unterscheidung zwischen gültigem und ungültigem Bereich für ein Ausweichmanöver bei konstanter Relativgeschwindigkeit	126
Abb. 72: Unterscheidung des gültigen und ungültigen Bereichs für ein Ausweichmanöver bei konstantem lateralem Abstand	126
Abb. 73: Schematischer Überblick der <i>PPP</i> -Softwaremodule und -funktionen	128

Tabellenverzeichnis

Tab. 1: Überblick über unterschiedliche Regelungsverfahren für Ausweichmanöver ..	14
Tab. 2: CARLA Attribute	22
Tab. 3: Übersicht der modellierten Gesamtfahrzeugmodelle	45
Tab. 4: Qualitative Bewertung der Modelle A – C nach fahrsicherheitsrelevanten Kriterien	48
Tab. 5: Fehlende Parameter für die Erstellung eines Gesamtfahrzeugmodells	52
Tab. 6: Verwendete Fahrzeugparameter	53
Tab. 7: Qualitative Bewertung von CarMaker und CARLA (im Default-Modus)	70
Tab. 8: Übersicht über die auszuführenden Aktionen bei Tastendruck	79
Tab. 9: Gegenüberstellung der Gesamtfahrzeugmodelle	84

A.1 Definition

Im Folgenden werden einige Begriffe der Fahrzeugführung, der Kritikalität von Verkehrssituationen und weitere allgemeine Begriffe definiert.

A.1.1 Begriffe der Fahrzeugführung

Manöver

Ein Manöver ist eine „geschickt ausgeführte Wendung, taktische Bewegung“ (s. Dudenredaktion [1], 2021) (z. B. eines Autos).

Ausweichmanöver

Nach Duden Online ist ein Ausweichmanöver eine taktische Bewegung, „durch das jemand mit seinem Fahrzeug einem Hindernis ausweicht“ (s. Dudenredaktion [2], 2021). Im Allgemeinen besteht ein Ausweichmanöver aus kombinierten Lenk- und Bremsingriffen, bei der es sich weder um eine Komfortfunktion noch um ein Überholmanöver handelt. Als Sicherheitsfunktion ist das hochdynamische Ausweichmanöver mit einer hohen Querschleunigungskraft verbunden. Nach der Straßenverkehrsordnung dient ein Fahrstreifenwechsel nach links meist dem Ziel, ein langsames Vorderfahrzeug zu überholen und schneller zu fahren. Im Falle des Notausweichassistenten geht mit dem Notausweichmanöver im Allgemeinen eine Geschwindigkeitsreduzierung einher, um das Hindernis auszuweichen.

Trajektorie

„Eine Trajektorie ist die Beschreibung der Bewegung eines Punktes $(X(t), Y(t))$ des Fahrzeugs (meist Schwerpunkt) in Abhängigkeit der Zeit. Alle kinematischen Größen sind eindeutig bestimmt.“ (s. Keller, 2017, S. 102)

A.1.2 Kritikalität von Verkehrssituationen

Ideale Kollisionsvermeidungstrajektorie

„Die ideale Kollisionsvermeidungstrajektorie [...] ist eine fiktive Trajektorie, welche in jeder Situation von allen Betrachtern als die bestmögliche Lösung einer kritischen Verkehrssituation zu bezeichnen ist.“ (s. Keller, 2017, S. 103)

A.1.3 Weitere Definitionen

Brute-Force-Methode

Die Brute-Force-Methode ist ein trivialer Ansatz zur Lösung eines Problems. Es werden so lange verschiedene Möglichkeiten überprüft, bis eine erfüllende Lösung gefunden wird, oder der Selbstversuch wird nach der Überprüfung aller möglichen Parameterbelegungen beendet (vgl. Gallenberger, 2017, S. 1 und Leemann, 2017, S. 36).

Default-Modus

Der Default-Modus beschreibt den Standardmodus bzw. die Voreinstellung eines Programms, falls der Benutzer selbst keine eigenen Änderungen vornimmt (vgl. Lipp, 2008, S. 36).

Rapid Application Development

Das Rapid Application Development ist eine schnelle Anwendungsentwicklung von bspw. der Programmiersprache Python für Softwareprojekte mit einem prototypischen Vorgehensmodell (vgl. Schmidt, 2017, S. 121 und Woyand, 2019, S. 1).

Open-Loop

Ein Open-Loop (Fahrverhaltenstest) ist ein offener Regelkreis, der den „Fahrversuch mit standardisierten Eingaben für die Betätigungselemente (Lenkrad, Fahrpedal, Bremspedal) und Messung und Analyse der Bewegungsgrößen“ (s. Heißing, 2013, S. 127) beschreibt.

Closed-Loop

Ein Closed-Loop (Fahrleistungstest) ist ein geschlossener Regelkreis, der das „Fahrzeug durch den Fahrer im Normalfahrbereich oder im Grenzbereich [...] führt, die fahrdynamischen Eigenschaften [...] subjektiv beurteilt [...] [und das] Fahrzeug [...] durch den Fahrer in standardisierten Fahrmanövern [...] führt mit Messung und Analyse der Bewegungsgrößen.“ (s. Heißing, 2013, S. 127)

A.2 Ausweichassistent der Fraunhofer IIS

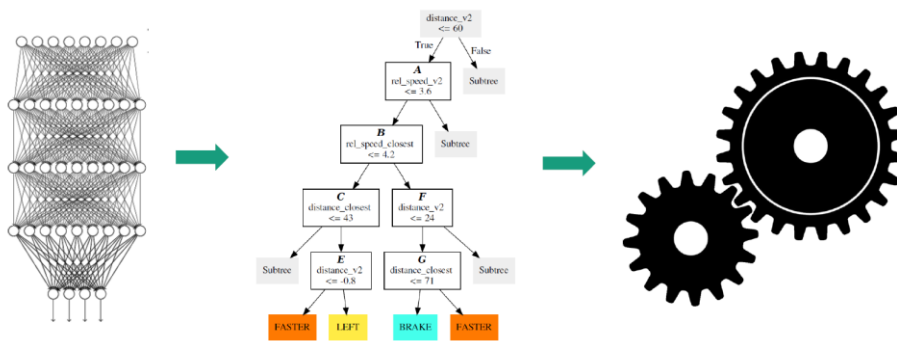
Vortragsfolien von Herrn Dr.-Ing. Axel Plinge

Zur Übersicht der Funktionsweise wird der Vortrag „HOW TO TEACH A MACHINE TO DRIVE IN DIFFICULT SITUATIONS AND BE ABLE TO RELY ON IT“ von Herrn Dr.-Ing. Axel Plinge am 21.04.2021 digital angehängt:

- Vortragsfolien HOW TO TEACH A MACHINE TO DRIVE IN DIFFICULT SITUATIONS AND BE ABLE TO RELY ON IT (s. „*HOW TO TEACH A MACHINE TO DRIVE IN DIFFICULT SITUATIONS AND BE ABLE TO RELY ON IT von Plinge.pdf*“).

Auf der 31. Folie des Vortrags ist eine Übersicht des Assistenzsystems durch neuronale Netze mit einem Entscheidungsbaum visualisiert.

Safe and Verifiable RL Rule Extraction and Verification



Bastani, O., Pu, Y., & Solar-Lezama, A. (2018). Verifiable reinforcement learning via policy extraction. In *Advances in neural information processing systems* (pp. 2494-2504).

Abb. 63: Generierung eines Entscheidungsbaum durch neuronale Netze (s. Vortragsfolien im digitalen Anhang)

- Vortrag als Video: Vortrag des Autors „*HOW TO TEACH A MACHINE TO DRIVE IN DIFFICULT SITUATIONS AND BE ABLE TO RELY ON IT von Plinge.mp4*“

A.3 Referenzmodell CarMaker

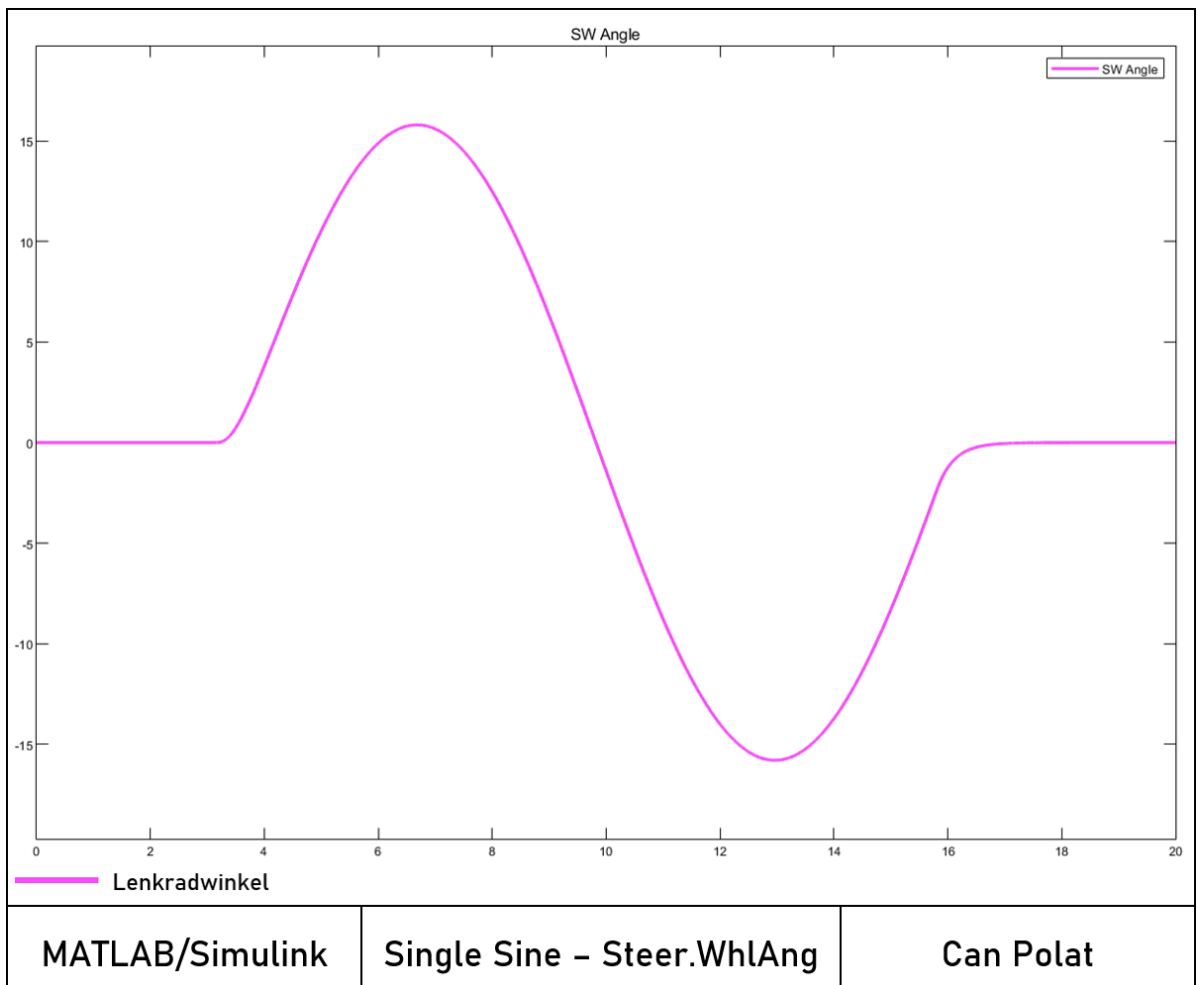
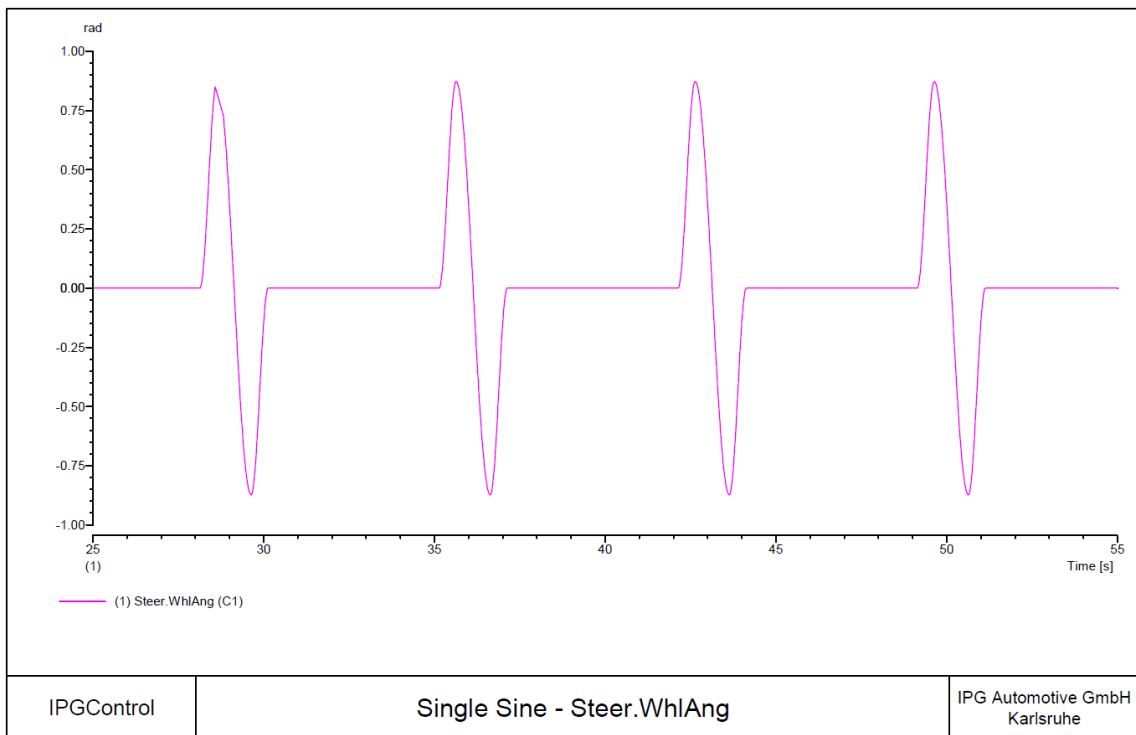
Im Rahmen dieser Abschlussarbeit dient die Software IPG CarMaker als Referenzmodell für die Modellierung, Simulierung und Validierung des Gesamtfahrzeugmodells C (vgl. Unterkapitel 4.1 und 5.2).

A.3.1 Fahrdynamische Größen des Einzelsinus-Tests (ISO 7401)

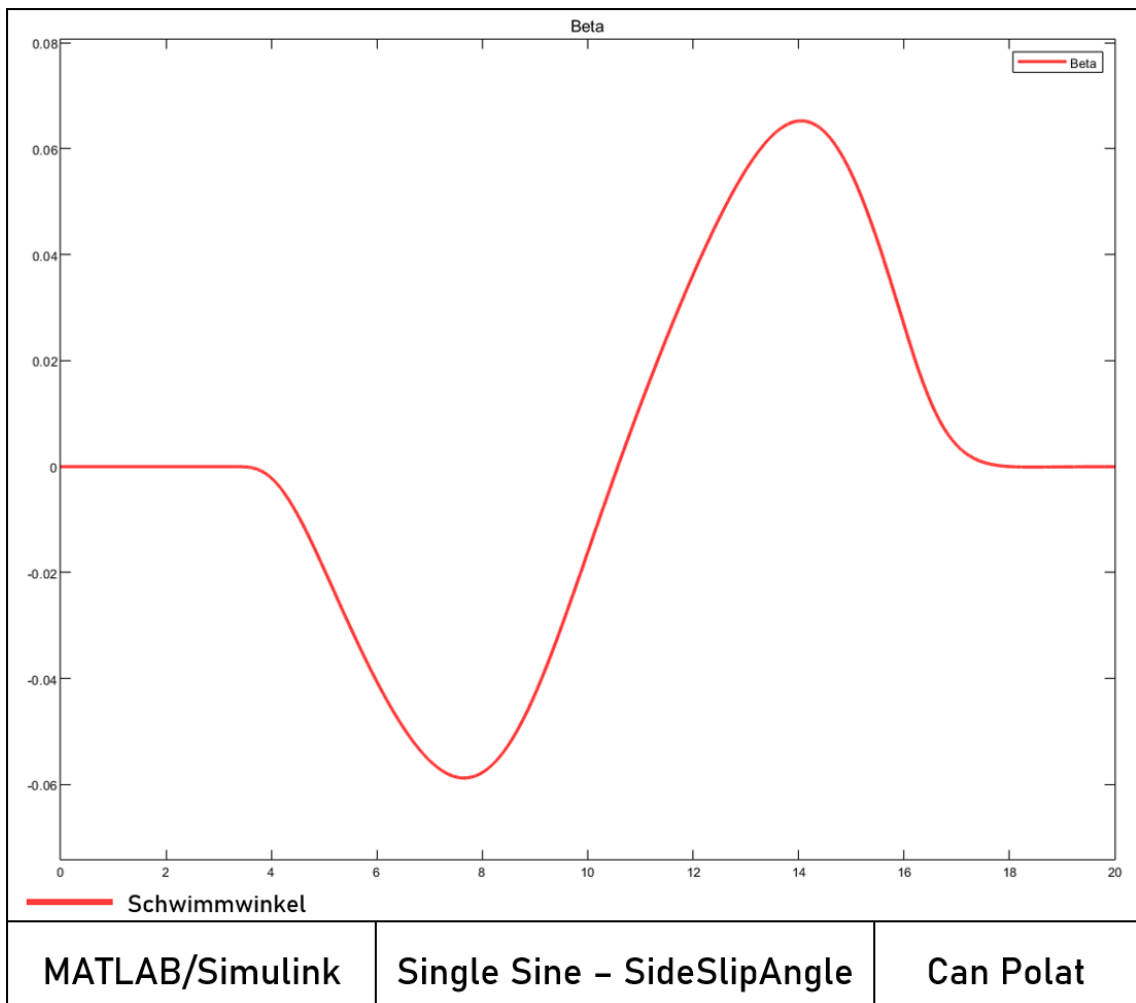
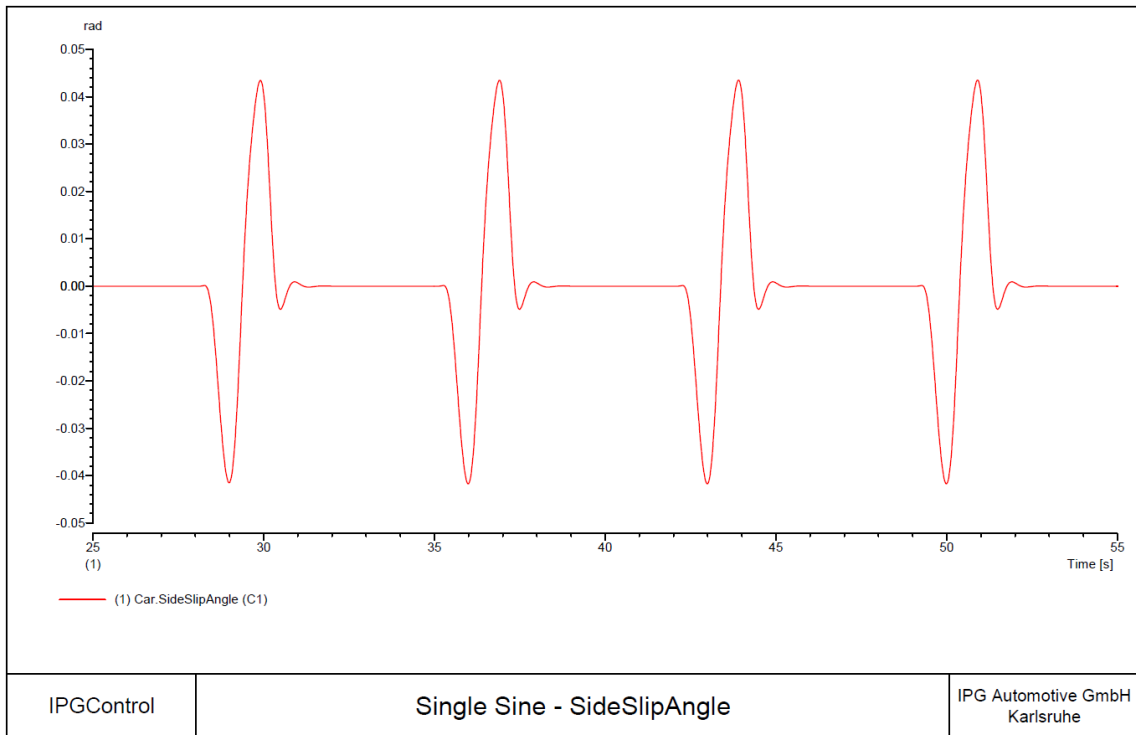
Für die erste Überprüfung der MiL-Simulation ist der Einzelsinus-Test nach ISO 7401 mit den Ergebnissen der Entwicklungsumgebung CarMaker verglichen worden. Hierbei ist zu beachten, dass ein nahezu qualitativ identischer Kurvenverlauf die Richtigkeit der MiL-Simulation bestätigt.

Die einzelnen fahrdynamischen Größen wurden in CarMaker aufgezeichnet und in die Excel-Datei „*Single_Sine_ISO_7401_fahrdynamische_Größen-CarMaker.xlsx*“ exportiert.

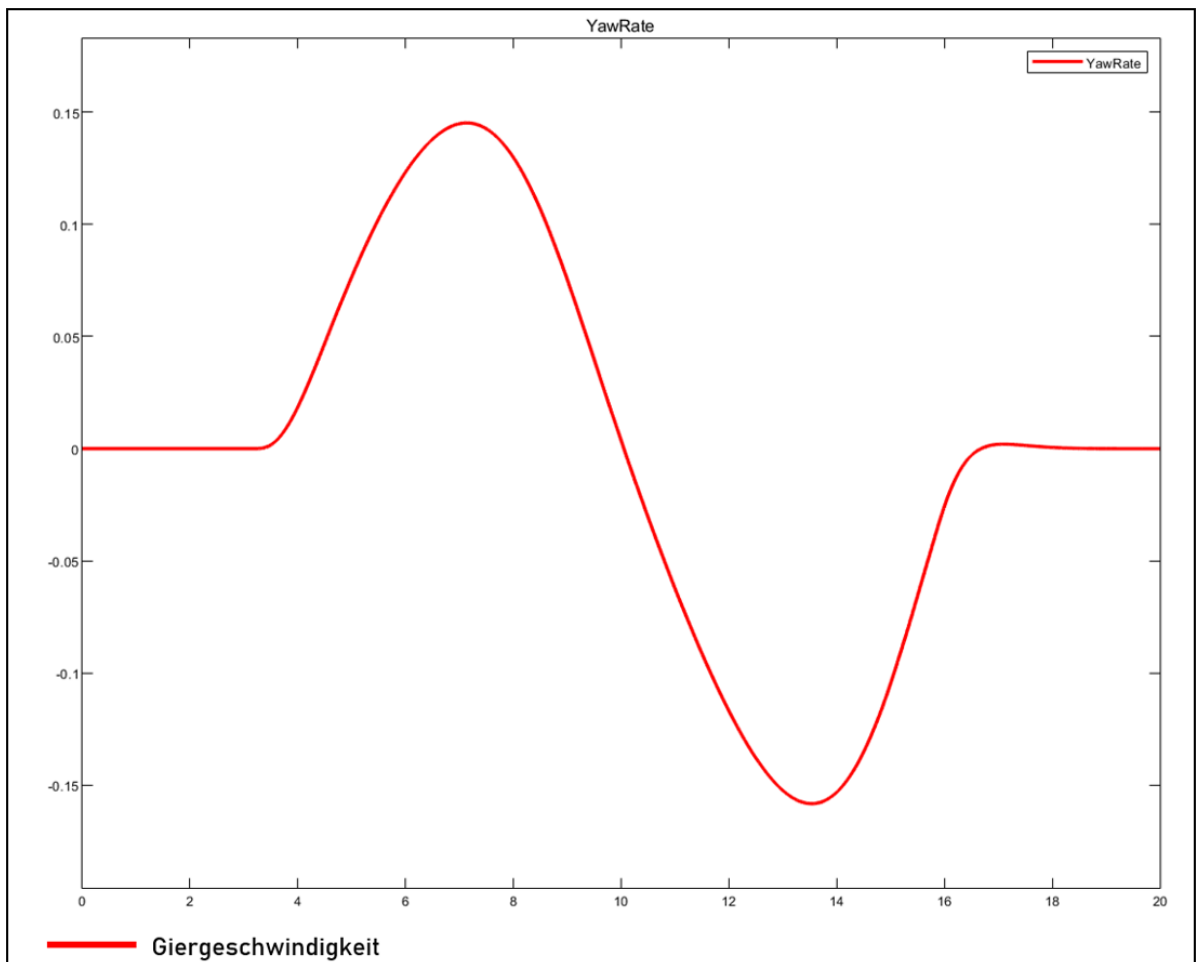
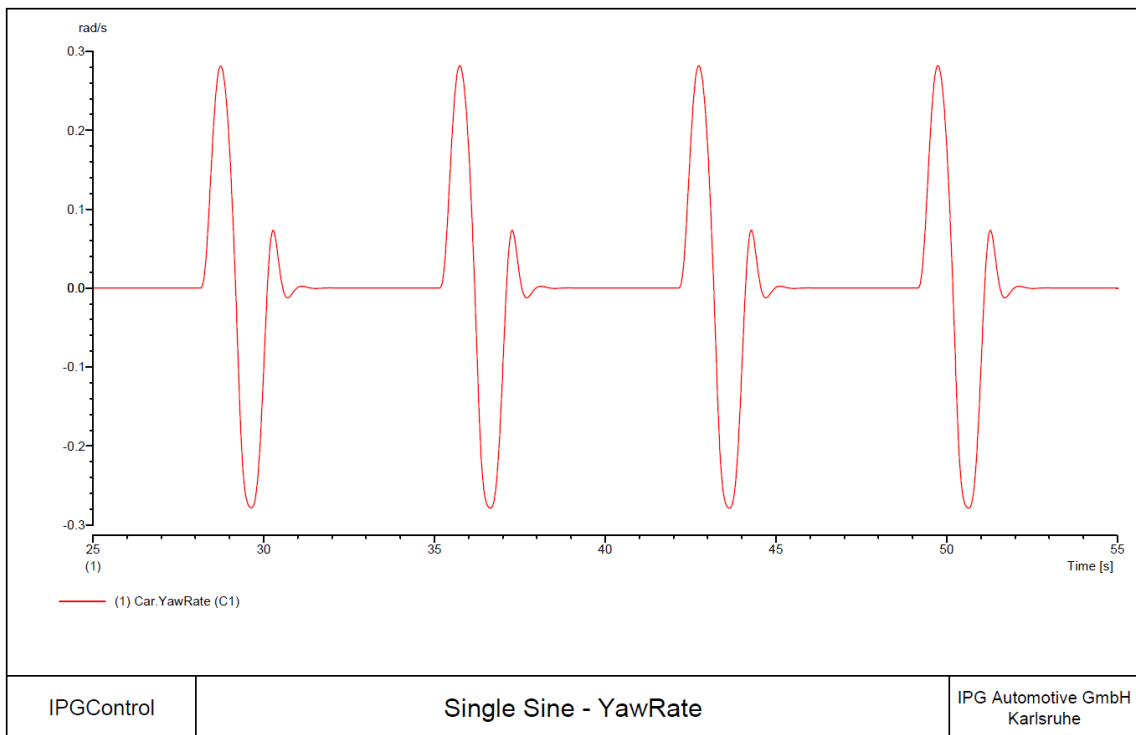
Lenkradwinkelverlauf



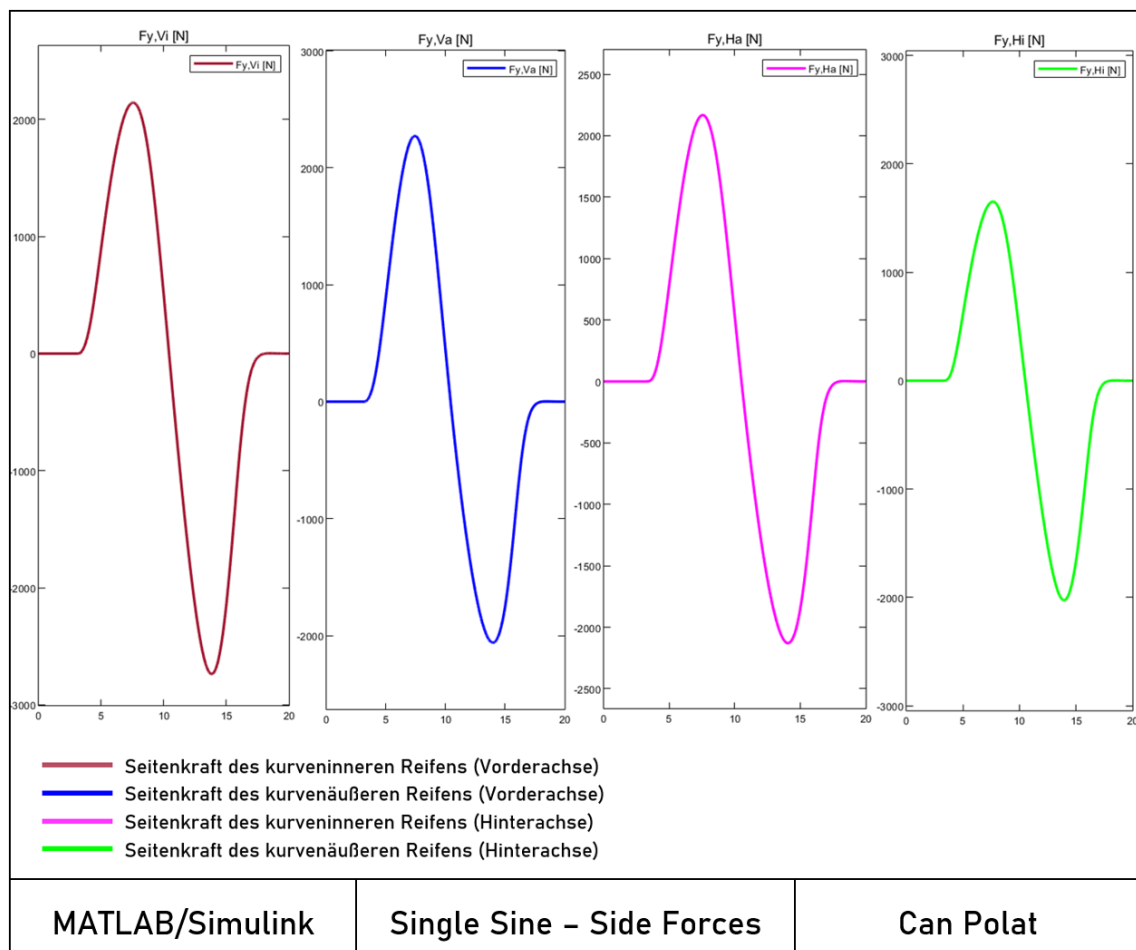
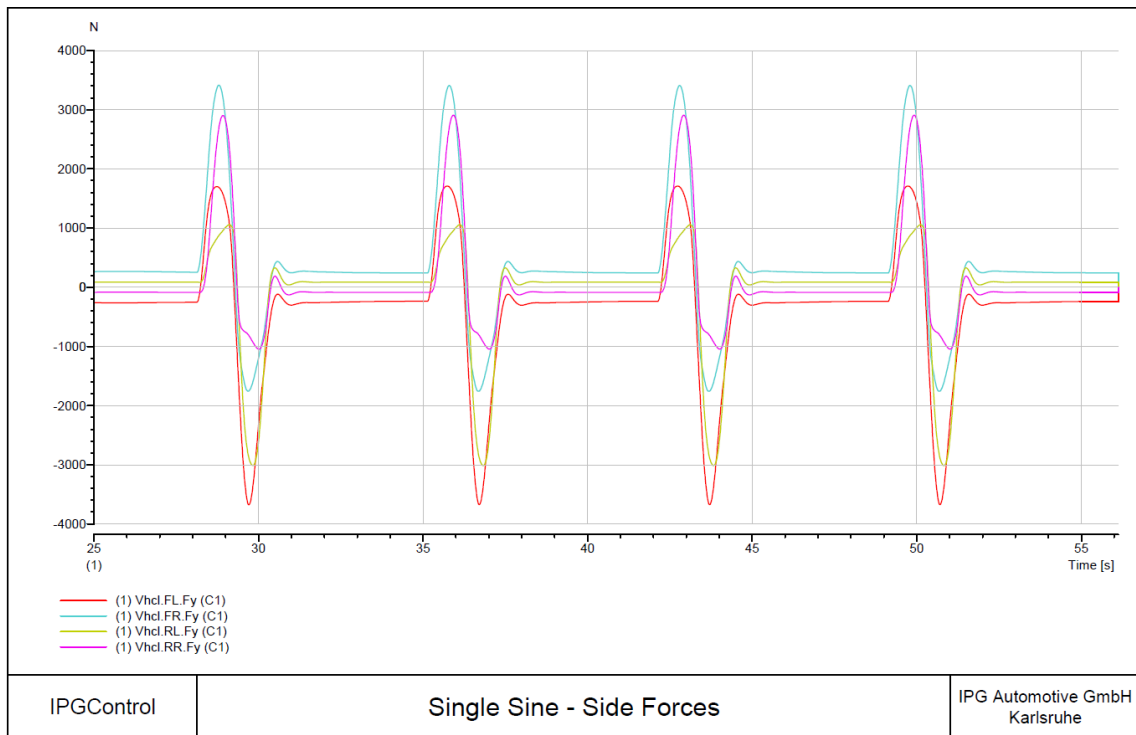
Schwimmwinkel



Giergeschwindigkeit



Seitenkräfte



A.3.2 VDA-Ausweichtest (ISO 3888-2)

Für die Simulation eines Ausweichmanövers dient der VDA-Ausweichtest von CarMaker. Dieser standardisierter Fahrmanöver wurde mit einem Audi R8 ausgeführt und ist als Video in „*VDA-Ausweichtest_Audi_R8_CarMaker.mp4*“ abgespeichert.

A.4 MATLAB/Simulink Programmierungen

MATLAB wird als „[the] language for technical computing“ (s. MathWorks, 1996, S. vi) vermarktet und ist heute zusammen mit Simulink („Simulation and model-based design“) sowie den vielen Toolboxen ein wichtiges Werkzeug für Ingenieure. MATLAB/Simulink ist eine fortschrittliche Umgebung zum Modellieren, Simulieren und Analysieren dynamischer Mehrdomänensysteme (vgl. MathWorks, 1996, S. 2 – 86). Die Simulink-Erweiterung für MATLAB gewährleistet Benutzern, grafische Kausalmodelle zu erstellen, ohne einen anwendungsspezifischen Programmiercode schreiben zu müssen. Heute ist Simulink in viele Richtungen gewachsen: Möglich sind mittlerweile das Hinzufügen weiterer Blöcke, das integrierte Lösen algebraischer Schleifen und eine Vielzahl von Toolboxen.

A.4.1 Kollisionsvermeidungsmanöver

Für die Darstellung der Kollisionsvermeidungsmanöver die Datei *Kollisionsvermeidung_-_Lenk-_und_Bremsmanöver_im_Unterschied.mat* erstellt (s. digitaler Anhang).

A.4.2 Ermittlung der Koeffizienten der Magic Formula nach Pacejka

Für die Darstellung der Richtigkeit für die Berechnung der Seitenkräfte an den Fahrzeugreifen wurde die Datei *Magic_Formula_Pacejka.mat* erstellt (s. digitaler Anhang). Die einzelnen fahrdynamischen Gleichungen stammen aus Unterkapitel 3.3.2 und werden nicht in der Programmierung erwähnt.

A.4.3 MiL-Programmierung der Modelle A – C

Für die MiL-Simulation dient die *Matlab_non_lin_ESM_and_nonlin_ZSM.mat*-Datei (s. digitaler Anhang). In dieser Datei wurden im MiL-Prozess die Modelle A, B und C programmiert. Die einzelnen fahrdynamischen Gleichungen stammen aus Unterkapitel 3.3 und werden nicht in der Programmierung erwähnt. Sie enthalten sämtliche fahrdynamische Größen, die für die Plausibilisierung des SiL-Modells notwendig sind. Es kann der Quelltext angepasst und die angehängten .xlsx- oder .txt-Dateien geändert werden.

Als Ergebnis der Programmierung resultiert folgende Abfrage:

Masterarbeit von Can Polat (54693016)

Entwicklung eines Fahrzeugmodells für hochdynamische Ausweichmanöver

In dieser Masterarbeit wurden verschiedene Fahrzeuge als Fahrzeugmodell nachmodelliert.

Für eine einfache Auswahl des Fahrzeugs wird an dieser Stelle gebeten, das passende Fahrzeug mit den entsprechenden Fahrzeugdaten auszuwählen.

Bestätigen Sie mit Ihrer Eingabe die Auswahl des Fahrzeugs:

- (1) Passenger Car (Pauwelussen)
- (2) Audi R8
- (3) BMW 3.30i Coupé
- (4) Volvo XC 90
- (5) Audi Q5

Ihre Eingabe: - **An dieser Stelle hat der Benutzer die Möglichkeit, sich ein Fahrzeugtyp für die Ausführung eines Manövers hochzuladen -**

In dieser Masterarbeit wurden Ausweichmanöver simuliert.

Für eine einfache Auswahl der Manöver wird an dieser Stelle gebeten, das passende Manöver einzustellen.

HINWEIS:

Bitte wählen Sie in SIMULINK SELBSTSTÄNDIG die ausgewählten Fahrmanöver aus!!!

Mit der Eingabe werden KEINE Veränderungen in Simulink vorgenommen.

Bestätigen Sie mit Ihrer Eingabe die Auswahl des Manövers:

- (1) Single Sine mit Frequenz $f = 0,5$ Hz (ISO 7401)
- (2) Single Sine mit Frequenz $f = 1,0$ Hz (ISO 7401)
- (3) Double Lane Change Maneuver (ISO-Wedeltest; ISO 3888-1)
- (4) Double Lane Change Maneuver (VDA/ Elchtest; ISO 3888-2)
- (5) Virtual test driving with data from CARLA Simulator (TrajectoriX)
- (6) Single Lane Change Maneuver (according to Can Polat)

Ihre Eingabe: - **Nach der passenden Manöverauswahl sind die eingestellten Parameter im Workspace abgelegt -**

Nach der Ausführung des *Matlab_non_lin_ESM_and_nonlin_ZSM.mat*-Datei wird automatisch *SimulinkModell_n_lin_ESM.slx*-Datei geladen. Das Aufrufen der Simulink-Datei kann einige Minuten benötigen.

Nachdem die Simulink-Datei geöffnet ist und das passende Manöver ausgewählt wurde, muss vor dem Klicken des Run-Buttons die *Fahrbahn.m*-Datei in den Scenario Reader abgelegt werden.

Hierfür sind folgende Schritte erforderlich:

1. Klicken Sie auf den blau markierten Block unten rechts namens „Scenario Reader“.

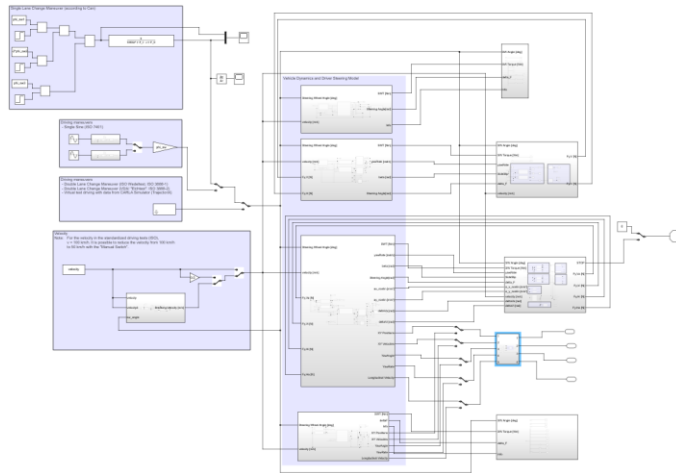


Abb. 64: Überblick der Simulink-Datei, Schritt 1 [Eigene Darstellung]

2. Klicken Sie auf den blau markierten Block „Subsystem“.

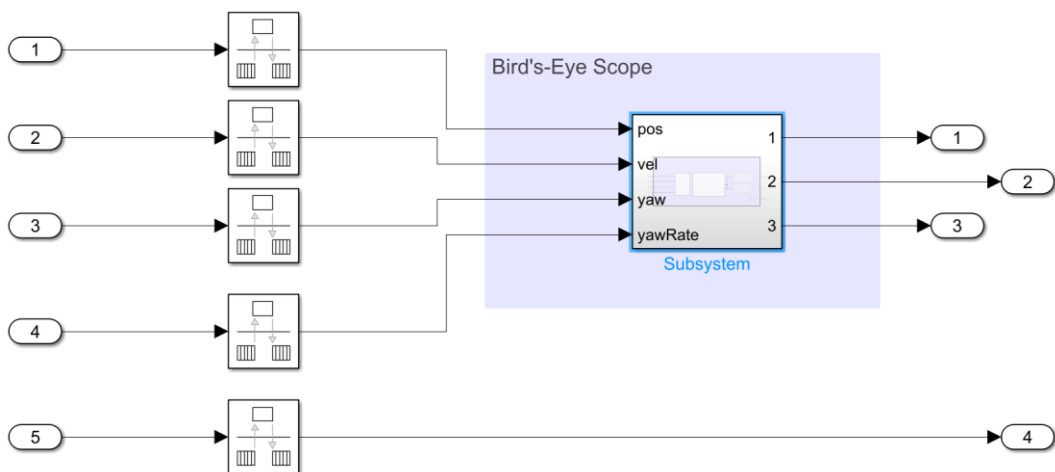


Abb. 65: Überblick der Simulink-Datei, Schritt 2 [Eigene Darstellung]

3. Klicken Sie auf den Block „Scenario Reader“

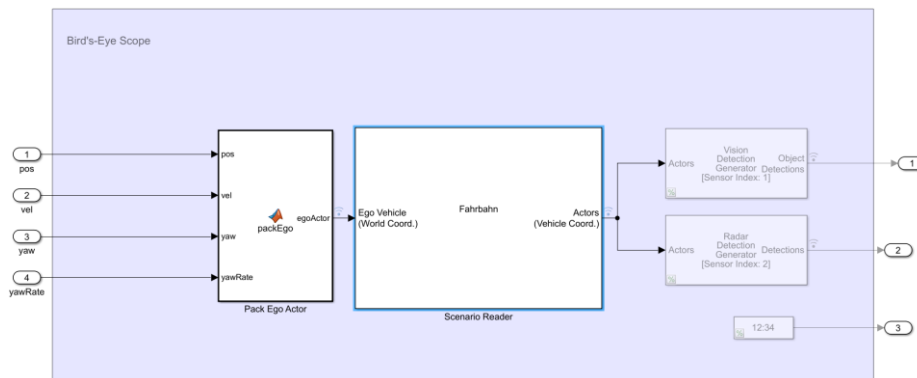


Abb. 66: Übersicht der Simulink-Datei, Schritt 3 [Eigene Darstellung]

4. Klicken Sie im Fenster „Block Parameters: Scenario Reader“ auf „Browse“ und wählen Sie die Datei „Fahrbahn“ aus.

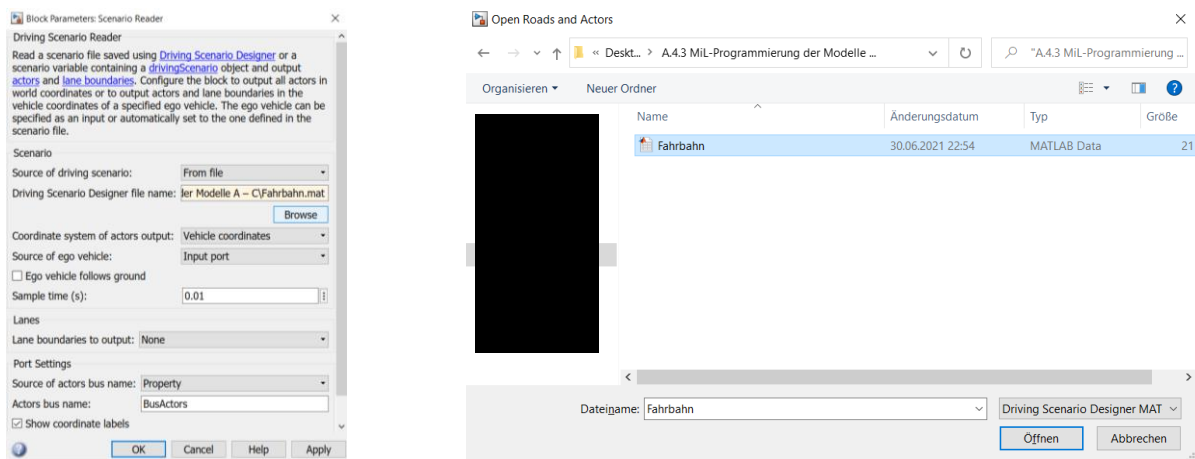


Abb. 67: Auswahl der Fahrbahn.mat-Datei für Bird's-Eye-Scope [Eigene Darstellung]

Als Ergebnis ist die 2D-Darstellung im Bird's-Eye Scope zu visualisieren.

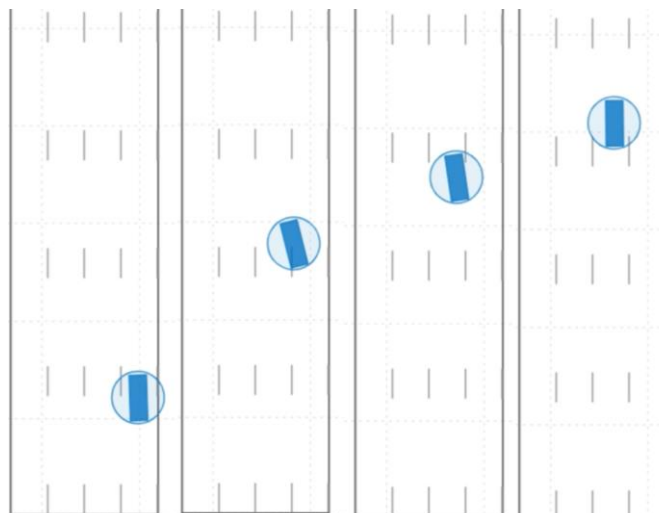


Abb. 68: Bird's-Eye Scope [Eigene Darstellung]

A.5 Python-Programmierungen

Mithilfe der Entwicklungsumgebung Geany wurde in der Programmiersprache Python (vgl. Geany, 2021) die Gesamtfahrzeugmodelle A – C programmiert und das Fahrerassistenzsystem (Zustandsautomat) erstellt. Die einzelnen fahrdynamischen Gleichungen stammen aus Unterkapitel 3.1 – 3.3 und werden nicht in der Programmierung erwähnt. Es wurden insgesamt drei Programme in der Programmiersprache Python entwickelt, die im Folgenden aufgelistet und zusammengefasst werden.

A.5.1 PDD

Im Rahmen der Dokumentation wird ausschließlich der Programmcode des Gesamtfahrzeugmodells C digital angehängt.

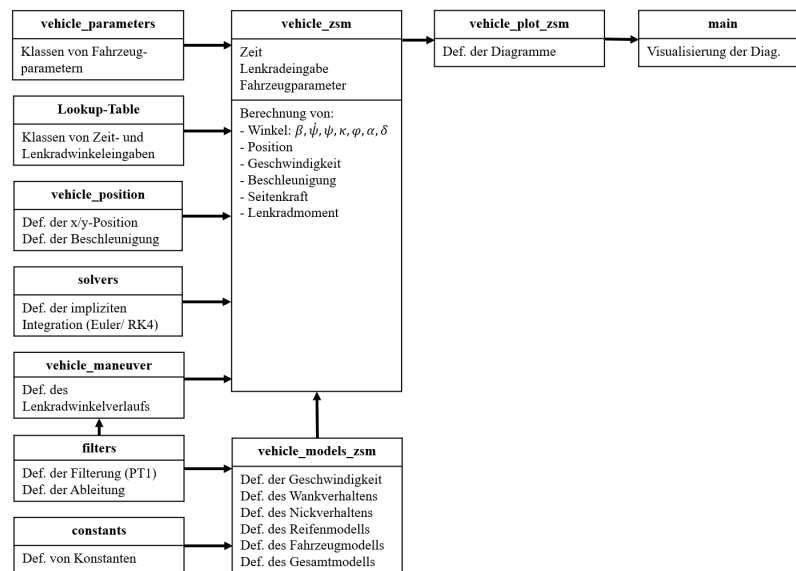


Abb. 69: Schematische Übersicht der PDD-Softwaremodule und -funktionen [Eigene Darstellung]

Diese setzt sich aus den folgenden Programmierungen zusammen (s. digitaler Anhang):

- *main.py*
- *vehicle_plot_zsm.py*
- *vehicle_zsm.py*
- *vehicle_models_zsm.py*
- *vehicle_parameters.py*
- *LookUpTable.py*
- *vehicle_position.py*
- *vehicle_maneuver.py*
- *filters.py*
- *constants.py*

A.5.2 PPADD

Im Rahmen der Dokumentation wird ausschließlich der Programmcode des Gesamtfahrzeugmodells C als Python-API digital angehängt.

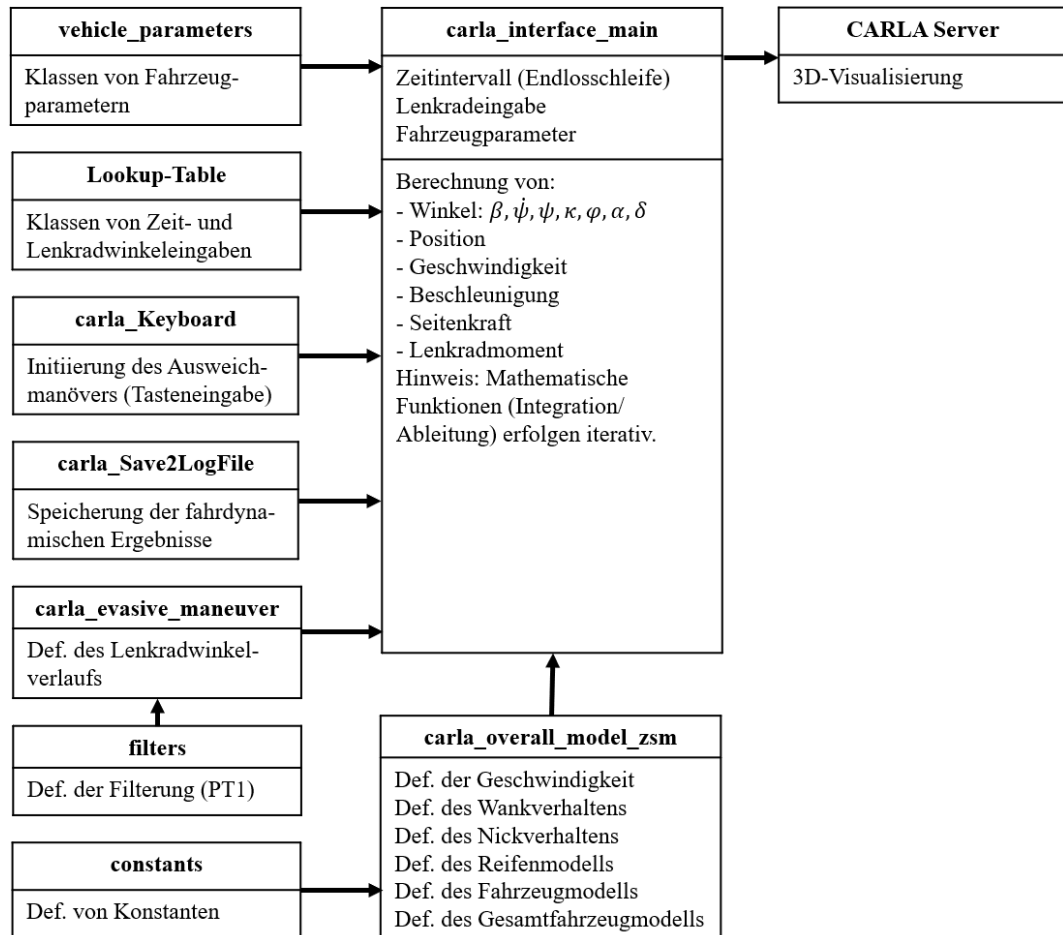


Abb. 70: Schematischer Überblick der PPADD-Softwaremodule und -funktionen [Eigene Darstellung]

Voraussetzung für die Benutzung der *PPADD*-Programmierung ist ein funktionsfähiger CARLA Server.

Die bereits vorgestellten Programmierungen von A.5.1 PDD wird in A.5.2 PPADD nicht erwähnt.

Die zusätzlichen Programme lauten (s. digitaler Anhang):

- *carla_interface_main.py*
- *carla_overall_model_zsm.py*
- *carla_models_zsm.py*
- *carla_Keyboard.py*
- *carla_Save2LogFile.py*
- *carla_pygame.py*

Modellierung eines Ausweichmanövers mittels Tastendruckes: carla_Keyboard

Das Ausweichmanöver kann durch Betätigung der Taste **4** nach links und mit der Taste **6** nach rechts navigiert werden (vgl. Tabelle 8). Nach dem Tastendruck des Fahrers führt das autonome Ego-Fahrzeug ohne Lenk- oder Bremsengriff des Fahrers ein Ausweichmanöver durch. Die Datenpaare werden der Lookup-Table entnommen und anschließend zur stetigen Reduzierung der Fahrzeuggeschwindigkeit verwendet sowie der vorgegebene Lenkradwinkelverlauf ausgeführt. Dabei hat der Fahrer während der Ausführung der Befehlskette die Option, das System mit den Tasten **Space**, **Backspace** und **Escape** zu beeinflussen (vgl. Abbildung 13). Die Betätigung der Pause-Taste (**Space**) bewirkt, dass die Simulation angehalten und ein Abbruch mit **Escape** möglich ist. Die Reset-Taste (**Backspace**) hingegen führt das System in den Ausgangszustand zurück.

Dokumentation der Datenaufzeichnung: carla_Save2LogFile

Entscheidend für die weitere Nutzung der *PPADD*-Daten ist zunächst die gewählte Formatierung, die mit verschiedenen Betriebssystemen kompatibel sein muss. Textdateien im txt-Format lassen eine Speicherung der relevanten Ergebnisse zu und können ebenso beim Protokollieren der Datensätze verwendet werden. In der .txt-Datei findet bei jedem Schleifendurchlauf des *PPADD*-Programms eine Zeilenerweiterung statt, die mit einem Tabulator je Spalte die fahrdynamischen Ergebnisse vom expliziten Integrationsverfahren trennt (vgl. Abbildung 52).

A.5.3 PPADD-Lookup-Table

Die in A.5.1 PDD und A.5.2 PPADD enthaltene Lookup-Table wird an dieser Stelle nicht als Python-Datei angehängt. Stattdessen wird auf die Excel-Datei „Lookup-Table.xlsx“ hingewiesen. Für die Python-Programme dient ausschließlich die Python-Datei, die mit der Bezeichnung „*LookUpTable.py*“ abgespeichert wurde. Diese Datei ist im digitalen Anhang hinzugefügt.

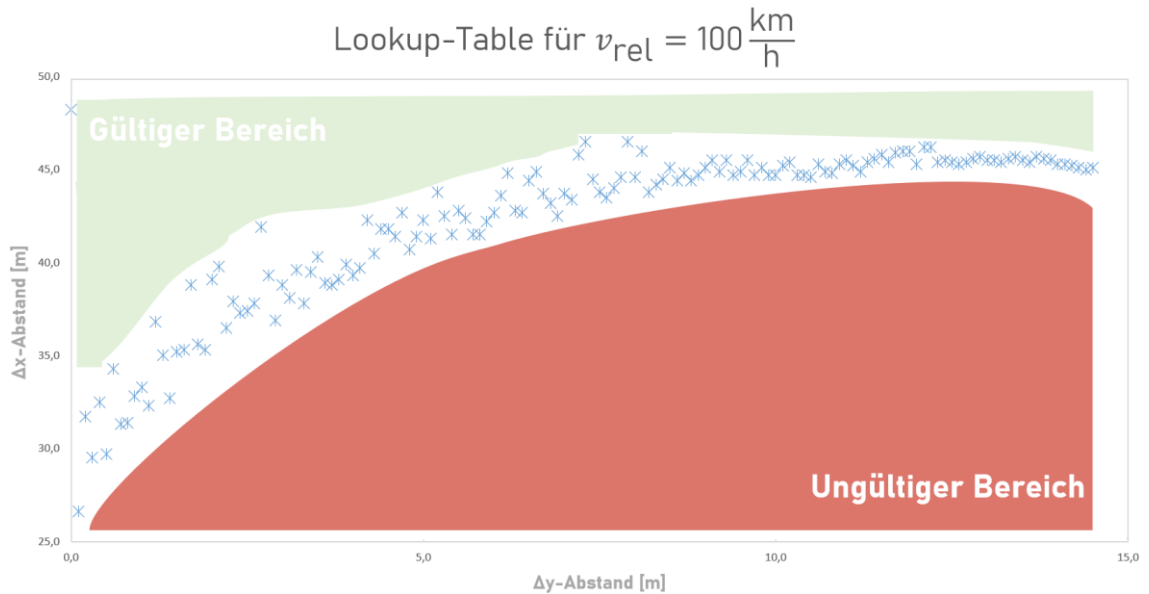


Abb. 71: Unterscheidung zwischen gültigem und ungültigem Bereich für ein Ausweichmanöver bei konstanter Relativgeschwindigkeit [Eigene Darstellung]

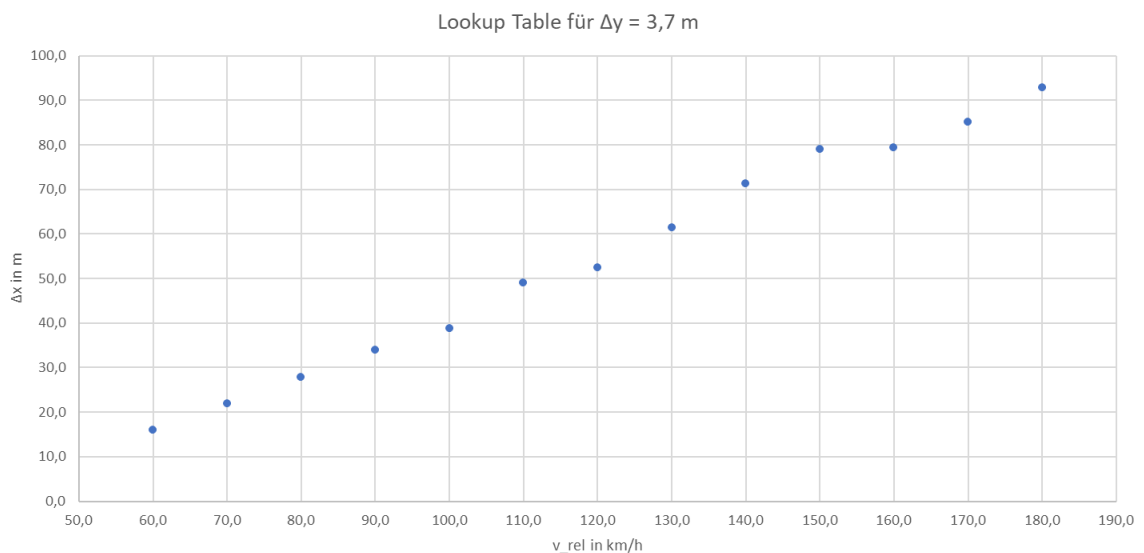


Abb. 72: Unterscheidung des gültigen und ungültigen Bereichs für ein Ausweichmanöver bei konstantem lateralem Abstand [Eigene Darstellung]

Die Daten der Lookup-Tables wurden für Programmierungen ebenfalls als .txt-file abgespeichert:

- LookUpTable_File_v_kleiner_50_km_h
- LookUpTable_File_v100_km_h
- LookUpTable_File_y3_7_m

In den dazugehörigen _README Dokumenten ist eine Erläuterung der einzelnen Spalten hinzugefügt.

An dieser Stelle sei bemerkt, dass für das Importieren der Lenkradwinkel-Zeit-Kombinationen bei Anwendung verschiedener Rechner unterschiedlich viel Zeit vergeht. Als Betriebsoberfläche wurde einerseits ein HP Pavilion Laptop 15-cs2xxx mit einem Windows 10 Home Betriebssystem (Prozessor: Intel® Core™ i5-8265U CPU @ 1.60 GHz x 4 und Grafikkarte: NVIDIA GeForce MX250) und ein Simulationsrechner mit einem Ubuntu 20.04 focal (x86-64) Betriebssystem (Prozessor: Intel® Core™ i7-3770K CPU @ 3.50 GHz x 4 und Grafikkarte: NVIDIA Corporation GM204 [GeForce GTX 970]) verwendet.

- verbrauchte Zeit für die Importierung der Daten beim HP: zwischen ca. 9,0 und 15,0 μ s
- verbrauchte Zeit für die Importierung der Daten beim Simulationsrechner: zwischen ca. 4,0 und 7,0 μ s

Somit wird ersichtlich, dass für eine echtzeitfähige Simulation ein HiL-Prüfstand mit einem einsetzbaren Steuergerät notwendig ist, um unterschiedliche Zeitmessungen zu vermeiden (vgl. Abbildung 11).

A.5.4 PPP

Im Rahmen der Dokumentation wird der Programmcode des Post-Processings angehängt.

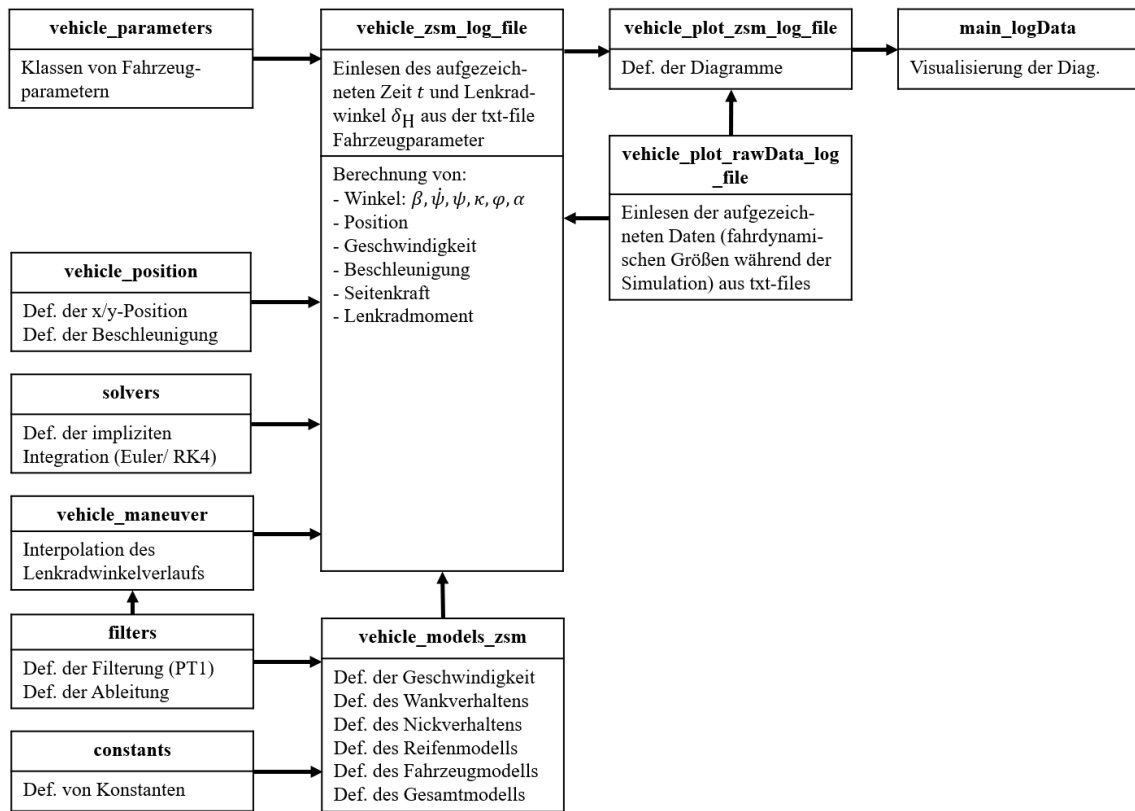


Abb. 73: Schematischer Überblick der PPP-Softwaremodule und -funktionen [Eigene Darstellung]

Der schematische Überblick des *PPP*-Programms dient zum Vergleich der explizit und implizit berechneten fahrdynamischen Größen (vgl. Unterkapitel 5.3.1 und 5.3.2). Die explizit berechneten Größen stammen, wie in der Abbildung 73 dargestellt, aus der Datenaufzeichnung während der CARLA-Simulation und werden in `vehicle_plot_rawData_log_file` ausgelesen. Die restlichen *PPP*-Module ähneln der *PPD*-Teilprogramme aus A.5.1 PDD mit den Softwaremodulen `vehicle_parameters`, `vehicle_position`, `solvers`, `vehicle_maneuver`, `filters`, `constants` und `vehicle_models_zsm`. Diese rechnen unter einem definierten Zeitbereich aus der Aufzeichnung „`steering4-8.txt`“ mit den aufgezeichneten CARLA-Lenkwinkelverläufen die impliziten fahrdynamischen Größen. Zusammenfassend können die Ergebnisse aus den CARLA-Rohdaten und den Vergleichsdaten in ein Diagramm visualisiert werden. Das Ergebnis des Vergleichs dient zur Beurteilung des Vorwärtsverfahrens als geeignetes Integrationsverfahren und somit für den Einsatz im Fahrsimulator CARLA.

Voraussetzung für die Benutzung der *PPP*-Programmierung ist die Python Bibliothek Pandas.

Die bereits vorgestellten Programmierungen von A.5.1 PDD wird in A.5.4 PPP nicht erwähnt.

Die zusätzlichen Softwaremodule lauten (s. digitaler Anhang):

- *main_logData.py*
- *vehicle_plot_esm_and_zsm_log_file.py*
- *vehicle_zsm_log_file.py*
- *vehicle_plot_rawData_log_file.py*

A.5.5 Aufzeichnung vom Fahrsimulator CARLA

Im Rahmen der Masterarbeit wurde die 3D-Simulation vom Fahrsimulator CARLA aufgezeichnet. Es handelt sich um ein einfachen Fahrspurwechsel mit den verschiedenen Gesamtfahrzeugmodellen:

1. Kollision des Ego-Fahrzeugs mit einem Fahrradfahrer – „*Kollision_mit_Fahrradfahrer-Gesamtfahrzeugmodell_B_CARLA.mp4*“
2. Kollisionsvermeidung des Ego-Fahrzeugs mit einem Fahrradfahrer – „*Kollisionsvermeidung_mit_Fahrradfahrer-Gesamtfahrzeugmodell_C_CARLA.mp4*“
3. Kollisionsvermeidung des Ego-Fahrzeugs mit einem Verkehrsteilnehmer – „*Kollisionsvermeidung_mit_Fahrzeug-Gesamtfahrzeugmodell_C_CARLA.mp4*“

A.6 C++-Programmierungen

Mit der Entwicklungsumgebung Qt Creator (Programmiersprache C++) (vgl. Qt Creator, 2021) wurde ausschließlich die Gesamtfahrzeugmodelle A – C inkl. die Fahrzeugparameter programmiert. Ein Zustandsautomat inkl. Einlesen des Lookup-Tables wurde in Qt Creator nicht programmiert.

Es ist an dieser Stelle angemerkt, dass das Programm vordefiniert und vom Verfasser dieser Masterarbeit fortgesetzt wurde. Eine eindeutige Zuweisung der Programmierung erfolgt im Rahmen dieser Masterarbeit nicht. Aus diesem Grund werden ausschließlich die Programme angehängt, die vom Verfasser dieser Arbeit fortgesetzt/ erstellt wurden.

Diese setzt sich aus den folgenden Softwaremodulen zusammen (s. digitaler Anhang):

- *VehicleState.hpp*
- *VehicleParameters.hpp*
- *VehicleModelSingleTrackLinear.cpp*
- *VehicleModelSingleTrackNonlinear.cpp*
- *VehicleModelDualTrackNonlinear.cpp*