DLR Institut für Fahrzeugkonzepte

DHBW Duale Hochschule Baden-Württemberg

BACHELOR-THESIS

Konzeption und Implementierung einer streckenbasierten Fahrsimulationsumgebung zur Vorauslegung elektrischer Fahrzeugantriebe

Emre Sari
Matr.-Nr: 3624658
Deutsches Zentrum für Luft- und Raumfahrt e.V.
Institut für Fahrzeugkonzepte

Betreuer:
Prof. Dr.-Ing. Michael Kornhaas
M.Sc. Patrick Schwarz

Abgabedatum: 01.09.2026

Abstract

The electrification of the powertrain has transitioned from a strategic consideration to a mandatory requirement. The design of an electric traction machine necessitates a load profile that accurately reflects its intended operating conditions. However, standardized cycles such as the WLTC provide only an averaged, rather than a specific, usage profile.

This thesis develops a route-based driving simulation environment that generates a load cycle from any real-world route and derives the requirements for the preliminary design of electric machines. Starting from an origin and destination address, a routing service supplies the point sequence, elevation profile, road classes and speed limits, which are supplemented by traffic light positions from OpenStreetMap. A look-ahead driver model derives a target speed profile from these data, which a longitudinal vehicle model with limited tractive force then follows. The tool is complemented by a vehicle database, a driving cycle database and a graphical user interface.

Validation is conducted in two stages: first, against seven measured drives of an Opel Corsa Electric on a 76 km commuting route, and second, against the WLTC. The simulation accurately reproduces speed profiles, power, and speed requirements, while the net energy demand is underestimated by approximately 15%. The primary causes are identified and quantified as auxiliary loads not represented in the model and recuperation potential calculated without a power limit.

Peak and continuous power, torque and speed ranges as well as energy demand can be derived from the load cycle. The distribution of operating points shows that the machine works predominantly at low torque and high speed. The recuperation share amounts to 8 % on the real route but 23 % in the WLTC, demonstrating that characteristic values derived from standardised cycles cannot readily be transferred to real usage profiles.

Inhaltsverzeichnis

Abstract I

Abkürzungsverzeichnis IV

Formelverzeichnis VI

1 Einleitung 1

1.1 Motivation und Hintergrund 1

1.2 Problemstellung und Zielsetzung 2

1.3 Abgrenzung des Themas 3

1.4 Übersicht 4

2 Grundlagen 4

2.1 Fahrzyklen 5

2.1.1 Definition und Bedeutung in der Antriebsentwicklung 5

2.1.2 Normierte Fahrzyklen (WLTP, NEFZ, RDE) 5

2.1.3 Synthetische und reale Fahrzyklen 8

2.2 Fahrzeuglängsdynamik und Fahrwiderstände 9

2.2.1 Rollwiderstand 10

2.2.2 Luftwiderstand 10

2.2.3 Steigungswiderstand 11

2.2.4 Beschleunigungswiderstand 11

2.2.5 Vom Fahrwiderstand zum Lastzyklus 12

2.3 Elektrische Antriebe und Maschinen 13

2.3.1 Aufbau und Funktionsweise elektrischer Maschinen 13

2.3.2 Betriebsrelevante Kenngrößen 15

2.4 Datenhaltung 17

2.4.1 Grundlagen relationaler Datenmodelle 17

2.4.2 Datenmodellierung und Schemaentwurf 17

3 Stand der Technik 19

3.1 Bestehende Fahrsimulationsprogramme und -Werkzeuge 19

3.2 Routing-Dienste und Kartendaten 20

3.3 Wahl der Programmiersprache 22

3.4 Verfügbare Referenzdatenquellen für Fahrzeuge und Fahrzyklen 23

4 Konzeption 24

4.1 Anforderungsanalyse 24
4.2 Gesamtarchitektur und streckenbasierte Fahrsimulation 26
4.2.1 Datenbeschaffung und Streckenrepräsentation 28
4.2.2 Zweistufiges Fahrermodell 29
4.2.3 Stochastische Elemente 30
4.3 Konzept der Fahrzyklusdatenbank 30
4.4 Konzept der Fahrzeugdatenbank 31
4.5 Methodik der Lastzyklusanalyse auf Basis von Fahrwiderstandsmodellen 33

5 Implementierung 35
5.1 Entwicklungsumgebung 36
5.2 Aufbau der Fahrzyklusdatenbank mit Referenzdaten 37
5.3 Aufbau der Fahrzeugdatenbank mit Referenzdaten 38
5.4 Strecken- und zyklusbasierte Fahrweise 40
5.4.1 Routenerzeugung über die GraphHopper-API 41
5.4.2 Einbindung von Höhenprofilen und Streckendaten 41
5.4.3 Sollwertbildung und Regelkreis 43
5.4.4 Zyklusbasierte Fahrsimulation 46
5.5 Simulationskern: Fahrwiderstandsmodell und Längsdynamiksimulation 48
5.6 Benutzeroberfläche zur Parametrierung und Auswertung 55

6 Validierung 60
6.1 Reale Fahrerprobung 61
6.2 WLTP-Vergleich 65

7 Ergebnisse 66
7.1 Ableitung von Anforderungen zur Grobauslegung elektrischer Maschinen 66
7.2 Bewertung der Ergebnisse 67

8 Zusammenfassung und Ausblick 68

Literaturverzeichnis VIII

Abbildungsverzeichnis XI

Tabellenverzeichnis XIII

Anhang XIV

Erklärung XV

Abkürzungsverzeichnis

Abkürzung	Bedeutung
ADAC	Allgemeiner Deutscher Automobil-Club
ADVISOR	Advanced Vehicle Simulator
API	Application Programming Interface
ASM	Asynchronmaschine
BMS	Batteriemanagementsystem
CSV	Comma-Separated Values (Dateiformat)
DHBW	Duale Hochschule Baden-Württemberg
DLR	Deutsches Zentrum für Luft- und Raumfahrt
EMR3	Electronics Motor Reducer, dritte Generation
EPA	Environmental Protection Agency
ER	Entity-Relationship
EU	Europäische Union
FA	Funktionale Anforderung
FESM	Fremderregte Synchronmaschine
FTP	Federal Test Procedure
GPX	GPS Exchange Format
GUI	Graphical User Interface (grafische Benutzeroberfläche)
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
KIT	Karlsruher Institut für Technologie
NEFZ	Neuer Europäischer Fahrzyklus
NFA	Nichtfunktionale Anforderung
NREL	National Renewable Energy Laboratory
OBD	On-Board-Diagnose
OSRM	Open Source Routing Machine
PDF	Portable Document Format
PEMS	Portable Emissions Measurement System

PID	Proportional-Integral-Differential
PMSM	Permanentmagneterregte Synchronmaschine
RDE	Real Driving Emissions
RMSE	Root Mean Square Error
SRM	Switched Reluctance Machine
SRTM	Shuttle Radar Topography Mission
SUV	Sport Utility Vehicle
UNECE	United Nations Economic Commission for Europe
VECTO	Vehicle Energy Consumption Calculation Tool
WLTC	Worldwide Harmonized Light Vehicles Test Cycle
WLTP	Worldwide Harmonized Light Vehicles Test Procedure
XOR	Exklusiv-Oder

Formelverzeichnis

Formelzeichen	Bedeutung	Einheit
A	Stirnfläche (projizierte Querschnittsfläche)	m²
a	Beschleunigung	m/s²
a_{dec}	Komfortverzögerung des Fahrermodells	m/s²
$a_{dec,max}$	maximal stellbare Bremsverzögerung	m/s²
$a_{y,max}$	maximale Querbeschleunigung	m/s²
C	nutzbare Energiekapazität	kWh
c_w	Luftwiderstandsbeiwert	–
E_Z	Energiebedarf eines Zyklusdurchlaufs	kWh
e	Regelabweichung des Geschwindigkeitsreglers	m/s
F_a	Beschleunigungswiderstand	N
F_{cmd}	Kraftanforderung des Fahrermodells	N
F_{FF}	Vorsteuerungsanteil der Kraftanforderung	N
F_{ges}	Gesamtfahrwiderstand	N
F_L	Luftwiderstand	N
F_R	Rollwiderstand	N
F_{St}	Steigungswiderstand	N
f_r	Rollwiderstandsbeiwert	–
g	Erdbeschleunigung	m/s²
h	Höhe über Normalnull	m
i	Gesamtübersetzung des Antriebsstrangs	–
K_d	Differentialbeiwert des Reglers	N·s²/m
K_i	Integralbeiwert des Reglers	N/m
K_p	Proportionalbeiwert des Reglers	N·s/m
M	Drehmoment an der Antriebsmaschine	N·m
M_{max}	maximales Motormoment	N·m
m	Fahrzeugmasse	kg
n	Motordrehzahl	1/min
n_{eck}	Eckdrehzahl	1/min

n_{max}	maximale Motordrehzahl	1/min
P	Leistung	kW
P_{max}	maximale Motorleistung	kW
R	lokaler Kurvenradius	m
r_w	Reifenrollradius	m
s	zurückgelegter Weg	m
s_R	Reichweite	km
s_Z	Weglänge eines Zyklusdurchlaufs	km
T	Umgebungstemperatur	°C
t	Zeit	s
v	Fahrgeschwindigkeit	m/s bzw. km/h
v_{Soll}	Sollgeschwindigkeit	km/h
α	Steigungswinkel der Fahrbahn	°
ε	Drehmassenzuschlagsfaktor	–
η	Wirkungsgrad des Antriebsstrangs	–
λ, φ	geografische Länge und Breite	°
μ	Kraftschlussbeiwert Reifen/Fahrbahn	–
ρ	Luftdichte	kg/m³
φ	Kurvenwinkel am Streckenpunkt	°
ω	Winkelgeschwindigkeit	rad/s

1 Einleitung

1.1 Motivation und Hintergrund

Der Verkehrssektor hat sich in Deutschland zu einer der bedeutendsten Quellen von Treibhausgasemissionen entwickelt. Während 1990 noch etwa 13 Prozent der Emissionen auf diesen Bereich entfielen, lag der Anteil im Jahr 2024 bereits bei 22,3 Prozent (vgl. [1]). Diese Entwicklung ist weniger auf ein hohes Verkehrsaufkommen zurückzuführen als vielmehr darauf, dass andere Sektoren ihre Emissionen deutlich reduzieren konnten, während im Verkehrsbereich kaum Fortschritte erzielt wurden. Als Reaktion darauf wurden von der Gesetzgeberseite verbindliche Flottenzielwerte eingeführt: Ab 2035 dürfen neu zugelassene Personenkraftwagen in der Europäischen Union im Prüfverfahren keine CO_2-Emissionen mehr aufweisen (vgl. [2, Art. 1]). Für neue schwere Nutzfahrzeuge gelten im Vergleich zum Referenzzeitraum Emissionsminderungen von 45 Prozent ab 2030, 65 Prozent ab 2035 und 90 Prozent ab 2040 (vgl. [3, Art. 3a]). Diese Vorgaben sind mit Verbrennungsmotorantrieben praktisch nicht zu erfüllen. Die Elektrifizierung des Antriebsstrangs stellt somit keine strategische Option mehr dar, sondern eine verbindliche Rahmenbedingung für die Fahrzeugentwicklung.

Der Markthochlauf spiegelt diese Entwicklung wider: Im Juli 2026 waren 29,3 Prozent der in Deutschland neu zugelassenen Personenkraftwagen batterieelektrisch (vgl. [4]). Gleichzeitig erweitert sich das Anwendungsfeld des elektrischen Antriebs erheblich. Elektrische Antriebe sind nicht mehr ausschließlich auf kompakte Stadtfahrzeuge beschränkt, sondern finden Anwendung vom Kleinwagen über schwere Geländelimousinen bis hin zu Lastkraftwagen im Fernverkehr. Auch im Luft-, Bahn- und Schiffsverkehrssektor gewinnen elektrische Antriebe zunehmend an Bedeutung. Für die Antriebsentwicklung bedeutet dies einen grundlegenden Wandel gegenüber der bisherigen Praxis: Während Verbrennungsmotoren über Jahrzehnte hinweg in Baureihen weiterentwickelt und aus bewährten Vorgängermodellen abgeleitet werden konnten, erfordert die Entwicklung elektrischer Maschinen für einige Anwendungssektoren eine weitgehend neue, einsatzspezifische Auslegung.

Die vorliegende Arbeit entstand im Rahmen eines Forschungsprojekts des Deutschen Zentrums für Luft- und Raumfahrt e.V. (DLR). Ziel des Projekts ist es, einen Beitrag zur Verbesserung der Zuverlässigkeit und Qualität von Simulationen bei der Entwicklung elektrischer Antriebe zu leisten. Dazu wird eine Reihe aufeinander abgestimmter Programme entwickelt, die jeweils einzelne Schritte der Auslegungskette abdecken. Diese Werkzeuge werden Herstellern

und Hochschulen frei zur Verfügung gestellt und eröffnen damit insbesondere kleineren Entwicklungsteams und der Lehre einen Zugang zu Methoden, die bislang kommerziellen Werkzeugketten vorbehalten waren. Die in dieser Arbeit entwickelte Fahrsimulationsumgebung steht am Anfang dieser Kette: Sie erzeugt das Lastprofil, aus dem die nachgelagerten Auslegungsschritte ihre Anforderungen beziehen.

1.2 Problemstellung und Zielsetzung

Um Lastanforderungen in der Vorauslegung effizient und präzise, mit geringem Zeitaufwand und somit kostengünstig abzuleiten, ist eine Simulation erforderlich, die Topographie, reale Geschwindigkeitsbegrenzungen, Kurvenradien, Ampeln sowie die Verkehrsdichte entlang einer realen Strecke berücksichtigt. Eine Variabilität bei der Auswahl relevanter Fahrzeugparameter sowie bei der gefahrenen Route ist dabei unerlässlich.

Ziel dieser Arbeit ist daher die Konzeption und Implementierung einer streckenbasierten Fahrsimulationsumgebung, die aus einer frei wählbaren realen Strecke ein Fahrprofil erzeugt und daraus die Vorauslegungsgrößen für elektrische Fahrzeugantriebe ableitet. Dazu werden Routen-, Höhen- und Kartendaten öffentlich zugänglicher Dienste zu einer simulationsfähigen Streckenrepräsentation aufbereitet und mit einem vorausschauenden Fahrermodell sowie einem längsdynamischen Fahrzeugmodell zu einem synthetischen Fahrzyklus verarbeitet. Das Fahrzeug wird dabei bewusst anhand elementarer Eckdaten beschrieben, da ein detailliertes Kennfeld der elektrischen Maschine in dieser Entwicklungsphase das Ergebnis und nicht die Voraussetzung der Auslegung ist. Ergänzend werden eine Fahrzeugdatenbank mit repräsentativen Fahrzeugklassen sowie eine Fahrzyklusdatenbank erstellt. Diese sollen eine einfache Speicherung und einen einfachen Import gewünschter Fahrzyklen und Fahrzeuge ermöglichen, so dass diese für eine Fahrsimulation genutzt werden können. Die Auswertung soll Geschwindigkeits-, Drehmoment- und Leistungsverläufe, den Energiebedarf, das Rekuperationspotenzial sowie die Betriebspunkte im Kennfeld der Maschine umfassen. Dadurch wird eine nachvollziehbare und reproduzierbare Bewertung verschiedener Fahrzeug-Strecken-Kombinationen ermöglicht und die Vorauslegung elektrischer Fahrzeugantriebe unterstützt. Zusätzlich wird dieses Modell in einer realen Fahrerprobung sowie anhand offizieller Vergleichswerte eines realen Fahrzeugs validiert.

1.3 Abgrenzung des Themas

Die Arbeit betrachtet ausschließlich Straßenfahrzeuge. Schienen-, Luft- und Wasserfahrzeuge sowie mobile Arbeitsmaschinen bleiben ebenso außer Betracht wie Fahrzeuge, deren Betrieb sich nicht sinnvoll über eine Route beschreiben lässt. Im Vordergrund stehen Personenkraftwagen, da sie den größten Teil des Fahrzeugbestands ausmachen und dort häufige Neuentwicklungen markttypisch sind. Ergänzend werden schwere Nutzfahrzeuge einbezogen: Für sie gelten eigene gesetzliche Höchstgeschwindigkeiten, sie reagieren wegen ihrer Masse besonders empfindlich auf die Topographie und stellen damit erhöhte Anforderungen an das Modell. Motorräder sind im Datenbestand ergänzt, um die Bandbreite der Modellierung zu zeigen, sie sind jedoch nicht Gegenstand der Untersuchung.

Inhaltlich liegt der Schwerpunkt auf elektrischen Antrieben, die Parametrisierung und Auswertung sind auf die Kenngrößen elektrischer Maschinen ausgerichtet. Fahrzeuge mit stufenlosen oder leistungsverzweigten Getrieben sind ausgeschlossen, da sie sich prinzipbedingt nicht mit festen Übersetzungen abbilden lassen.

Gegenstand der Arbeit sind die Anforderungen an die elektrische Maschine, nicht deren konstruktiver Entwurf. Fragen der Wicklungs- und Magnetauslegung, der Bauraumgeometrie und der Kühlung bleiben ebenso außen vor wie die Auslegung von Energiespeichern und Leistungselektronik sowie Kosten- und Lebenszyklusbetrachtungen. Nicht behandelt wird ferner die gesamte Querdynamik als eigener Freiheitsgrad; sie geht ausschließlich als Grenzwert der Kurvengeschwindigkeit ein. Zudem werden eine fahrzeugindividuelle Abbildung des umgebenden Verkehrs sowie thermische Effekte im Antriebsstrang vernachlässigt. Diese Abgrenzungen sind bewusst gewählt: Sie halten den Parametrisierungsaufwand auf einem Niveau, das in der frühen Entwicklungsphase tatsächlich zu leisten ist, und beschränken das Modell auf jene Effekte, welche die gesuchten Auslegungsgrößen dominieren.

1.4 Übersicht

Die Arbeit gliedert sich in 8 Kapitel:

- Kapitel 2 erklärt die notwendigen Grundlagen für das Verständnis folgender Kapitel
- Kapitel 3 beschreibt den Stand der Technik und vergleicht verschiedene Werkzeuge und Datenquellen
- Kapitel 4 gibt eine Übersicht über das Konzept der Fahrsimulationsumgebung und wichtige Zusammenhänge
- Kapitel 5 beschreibt die technische Umsetzung des entwickelten Fahrzyklusgenerators, einschließlich Datenbanken, Routenerzeugung, Fahrzeugsimulation, Simulationskern und der Benutzeroberfläche
- Kapitel 6 stellt die Validierung, Analyse und Auswertung einer erzeugten Fahrsimulation mit realitätsnahen Fahrzeugparametern dar und vergleicht diese mit der realen Fahrerprobung desselben Fahrzeugs auf derselben Strecke
- Kapitel 7 leitet daraus Anforderungen zur Grobauslegung elektrischer Maschinen ab und bewertet die Ergebnisse der Simulation
- Kapitel 8 fasst die Arbeit zusammen und gibt einen Ausblick auf die weiteren Schritte dieses Projekts

2 Grundlagen

Dieses Kapitel führt die für das Verständnis der Arbeit erforderlichen Grundlagen ein. Zunächst werden Fahrzyklen als zentrales Beschreibungsmittel des Fahrbetriebs definiert und die relevanten Zyklusarten vorgestellt. Abschnitt 2.2 behandelt die Fahrzeuglängsdynamik unter Berücksichtigung der Fahrwiderstände, aus denen der Momenten- und Leistungsbedarf eines Fahrzeugs hervorgeht. Abschnitt 2.3 gibt einen Überblick über elektrische Antriebe und die für deren Auslegung maßgeblichen Kenngrößen, bevor Abschnitt 2.4 mit den Grundlagen relationaler Datenbankkonzepte und der Datenmodellierung schließt. Die Inhalte werden bewusst grundlegend gehalten; ihre anwendungsspezifische Vertiefung erfolgt in den Kapiteln 4 und 5.

2.1 Fahrzyklen

2.1.1 Definition und Bedeutung in der Antriebsentwicklung

Ein Fahrzyklus ist ein zeitlich definiertes Geschwindigkeitsprofil $v(t)$, das den Ablauf einer Fahrt unabhängig von einem konkreten Fahrzeug beschreibt; ergänzend können weitere Vorgaben wie Schaltpunkte oder ein Steigungsprofil hinterlegt sein. Der Fahrzyklus trennt damit die Beschreibung des Fahrbetriebs von der Beschreibung des Fahrzeugs: Dasselbe Profil kann von unterschiedlichen Fahrzeugen abgefahren werden und liefert jeweils fahrzeugspezifische Last- und Verbrauchsverläufe[5, S. 17–19]. Erst diese Trennung ermöglicht eine Vergleichbarkeit von Mess- und Simulationsergebnissen verschiedener Fahrzeuge untereinander.

In der Antriebsentwicklung erfüllen Fahrzyklen zwei Aufgaben: Zum einen bilden sie die Grundlage der gesetzlichen Typprüfung, in der Energieverbrauch, Emissionen und bei batterieelektrischen Fahrzeugen die Reichweite unter definierten Bedingungen ermittelt werden (vgl. [6, S. 33]). Zum anderen dienen sie als Lastvorgabe für die Auslegung von Antriebskomponenten: Aus dem Geschwindigkeitsverlauf lassen sich, wie Abschnitt 2.2 zeigt, die zeitlichen Verläufe von Zugkraft, Drehmoment und Leistung ableiten, aus denen Anforderungen an elektrische Maschinen, Getriebe und Energiespeicher folgen. Aktuell gibt es eine Vielzahl von Fahrzyklen. Ein Projektbericht des englischen Verkehrsministeriums zu Fahrzyklen zur Messung von Fahrzeugemissionen listet mehr als 200 entwickelte Fahrzyklen auf und vergleicht sie [7, S. 276].

2.1.2 Normierte Fahrzyklen (WLTP, NEFZ, RDE)

Normierte Fahrzyklen sind gesetzlich festgelegte Referenzprofile für die Typprüfung. In Europa war von 1992 bis 2017 der Neue Europäische Fahrzyklus (NEFZ) maßgeblich. Er setzt sich aus vier identischen innerstädtischen Teilzyklen und einem Überlandteil zusammen, dauert 1180 s, führt über rund 11 km und erreicht eine Höchstgeschwindigkeit von 120 km/h bei einer mittleren Geschwindigkeit von etwa 34 km/h. Charakteristisch ist der modale, stilisierte Aufbau aus Abschnitten konstanter Geschwindigkeit und konstanter Beschleunigung. Aufgrund dieser geringen Dynamik und des niedrigen Lastniveaus gilt der NEFZ als wenig repräsentativ für den realen Fahrbetrieb [6, S. 288].

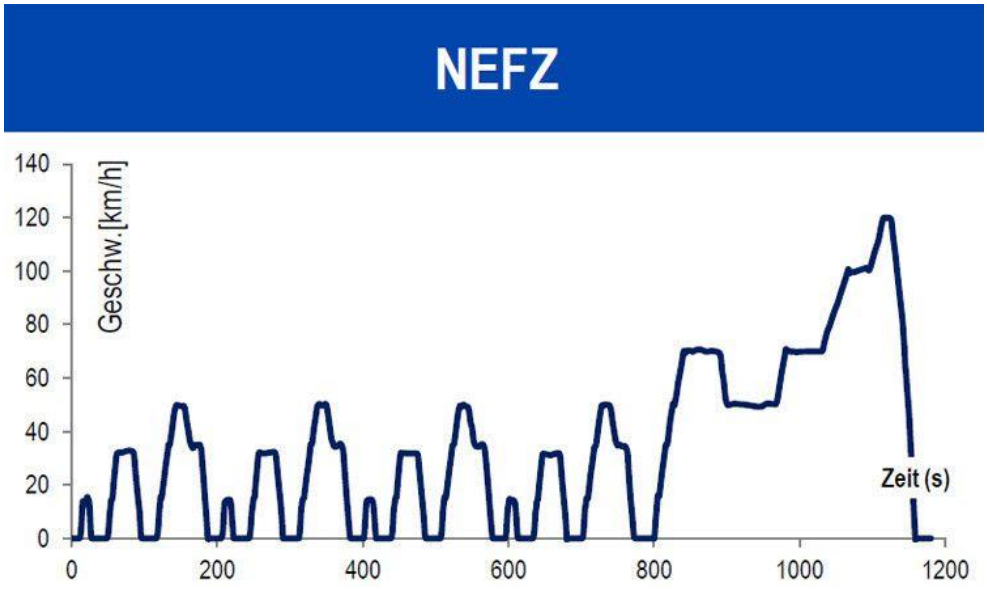


Abbildung 1: v(t)-Diagramm NEFZ, Quelle: Tüv Süd

Als Nachfolger wurde die Worldwide Harmonized Light Vehicles Test Procedure (WLTP) eingeführt, die in der Europäischen Union seit September 2017 für die Typgenehmigung von Personenkraftwagen verbindlich ist (vgl. [8, Art. 3 Abs. 1 i. V. m. Anhang XXI sowie Art. 18]). Die WLTP wurde auf Ebene der Wirtschaftskommission der Vereinten Nationen für Europa (UN-ECE) als Global Technical Regulation Nr. 15 entwickelt; ihr Fahrprofil basiert auf real aufgezeichneten Fahrdaten aus mehreren Weltregionen (vgl. [9]). Die Prozedur legt neben dem Fahrprofil auch die Randbedingungen der Prüfung fest. Dazu gehören die Ermittlung der Fahrwiderstände, der Prüfmasse und der Umgebungstemperatur. Das zugehörige Geschwindigkeitsprofil ist der Worldwide Harmonized Light Vehicles Test Cycle (WLTC). Fahrzeuge werden über das Verhältnis von Nennleistung zu Leermasse in drei Klassen eingeteilt, moderne Personenkraftwagen liegen mit mehr als 34 W/kg nahezu ausnahmslos in Klasse 3 und müssen somit den vollen Fahrzyklus durchfahren.

Der WLTC der Klasse 3 dauert 1800 s, führt über etwa 23,3 km und gliedert sich in die vier Phasen Low, Medium, High und Extra High mit steigendem Geschwindigkeitsniveau bis etwa 131 km/h (siehe Abbildung 2). Die Phasen repräsentieren innerstädtischen Verkehr, Überlandfahrten sowie Fahrten auf Schnellstraßen und Autobahnen. Gegenüber dem NEFZ ist der WLTC deutlich transienter: Konstantfahrtanteile sind seltener, Beschleunigungsvorgänge häufiger und kräftiger, wodurch das Profil dem realen Fahrbetrieb erheblich näherkommt. Allerdings wird der WLTC genauso wie der NEFZ auf dem Prüfstand von einem menschlichen Fahrer gefahren, leichte Abweichungen von den vorgegebenen Geschwindigkeiten sind

dadurch innerhalb des Toleranzfensters erlaubt. Für diese Arbeit ist der WLTC in zweifacher Hinsicht relevant: als Referenzlastvorgabe der zyklusbasierten Simulation und als Bezugsgröße, mit der Simulationsergebnisse mit veröffentlichten Prüfwerten realer Fahrzeuge verglichen werden können.

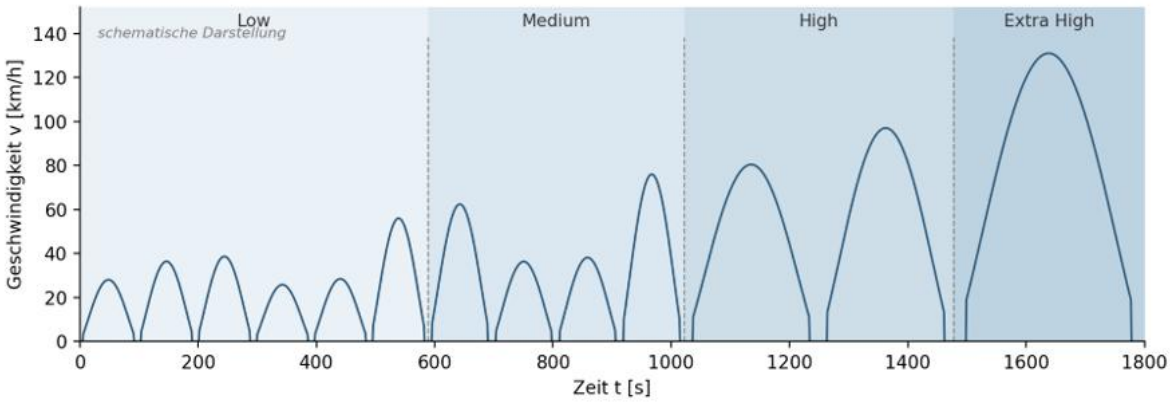


Abbildung 2: v(t)-Diagramm des WLTC der Klasse 3 mit den Phasen Low, Medium, High und Extra High. (eigene Darstellung)

Normierte Prüfstandszyklen können den realen Betrieb dennoch nur in Ausschnitten abbilden. Mit der Real-Driving-Emissions-Prüfung (RDE) wurde die Typprüfung deshalb um Emissionsmessungen im realen Straßenverkehr ergänzt (vgl. [10, Art. 1 Nr. 5 i. V. m. Anhang IIIA Nr. 1]). Die RDE-Prüfung gibt kein festes Geschwindigkeitsprofil vor; stattdessen wird eine reale Fahrt mit portablen Emissionsmesssystemen (PEMS) durchgeführt, wobei Randbedingungen wie die Streckenanteile von Stadt, Landstraße und Autobahn, zulässige Toleranzbereiche für das Fahrverhalten (positive und negative Beschleunigung) sowie Temperatur- und Höhenbereiche festgelegt werden. Für die vorliegende Arbeit ist an der RDE-Gesetzgebung vor allem der zugrunde liegende Gedanke von Bedeutung: Das Verhalten eines Fahrzeugs soll nicht nur nach einem einheitlichen Normprofil, sondern auch unter individuellen, realen Fahrbedingungen bewertet werden.

2.1.3 Synthetische und reale Fahrzyklen

Neben den normierten Zyklen werden reale und synthetische Fahrzyklen unterschieden. Reale Fahrzyklen entstehen durch die Aufzeichnung tatsächlicher Fahrten, beispielsweise mittels Satellitenortung oder des Fahrzeugdatenbusses. Sie erfassen sämtliche Einflüsse des realen Betriebs wie Verkehr, Ampelschaltungen, Topographie und individuelles Fahrerverhalten und bieten damit die realitätsnaheste Beschreibung eines Einsatzprofils. Allerdings sind mit dieser Methode erhebliche Nachteile verbunden: Messfahrten sind aufwendig, ihre Ergebnisse gelten ausschließlich für die gefahrene Strecke unter den jeweiligen Bedingungen, und einzelne Aufzeichnungen sind aufgrund des stochastischen Verkehrsgeschehens nur eingeschränkt reproduzierbar und übertragbar.

Synthetische Fahrzyklen werden hingegen rechnerisch erzeugt. Dabei sind zwei Ansätze verbreitet: Statistische Verfahren verdichten große Mengen aufgezeichneter Fahrdaten zu repräsentativen Profilen, indem beispielsweise Geschwindigkeits- und Beschleunigungsübergänge als Markow-Ketten modelliert werden. Eine Markow-Kette beschreibt einen zufälligen Prozess, bei dem der jeweils nächste Zustand ausschließlich vom aktuellen Zustand abhängt und nicht vom weiteren Verlauf der Vorgeschichte (vgl. [11, Kap. 8]). Übertragen auf einen Fahrzyklus bedeutet dies, dass aus den Messdaten die Wahrscheinlichkeiten ermittelt werden, mit denen ein Fahrzustand in einen bestimmten Folgezustand übergeht. Durch wiederholtes Ziehen aus dieser Übergangsmatrix entsteht ein neues Geschwindigkeitsprofil, das dieselben statistischen Eigenschaften wie die zugrunde liegenden Aufzeichnungen aufweist. Beim zweiten Ansatz erzeugen streckenbasierte Verfahren das Profil für eine spezifische Route, indem aus digitalen Karten-, Topographie- und Verkehrsdaten in Kombination mit einem Fahrer- und Fahrzeugmodell simuliert wird, wie die Strecke befahren würde. Der Hauptvorteil synthetischer Zyklen besteht darin, dass beliebige Strecken und Szenarien ohne Messfahrten untersucht werden können und die Erzeugung reproduzierbar ist.

2.2 Fahrzeuglängsdynamik und Fahrwiderstände

Die Längsdynamik beschreibt die Bewegung des Fahrzeugs in Fahrtrichtung. Damit das Fahrzeug seinen Bewegungszustand hält, muss ein Kräftegleichgewicht zwischen der vom Fahrzeugantrieb erzeugten Antriebskraft und den auftretenden Widerstandskräften herrschen, die als Reaktion der Fahrbahn und der Umgebung entstehen. Wenn die Widerstandskräfte überwiegen, erfährt das Fahrzeug eine negative Beschleunigung; im umgekehrten Fall eine positive Beschleunigung. Dieser Gesamtfahrwiderstand setzt sich aus den Anteilen Rollwiderstand F_{R}, Luftwiderstand F_{L}, Steigungswiderstand F_{St} sowie dem Beschleunigungswiderstand F_{a} zusammen.

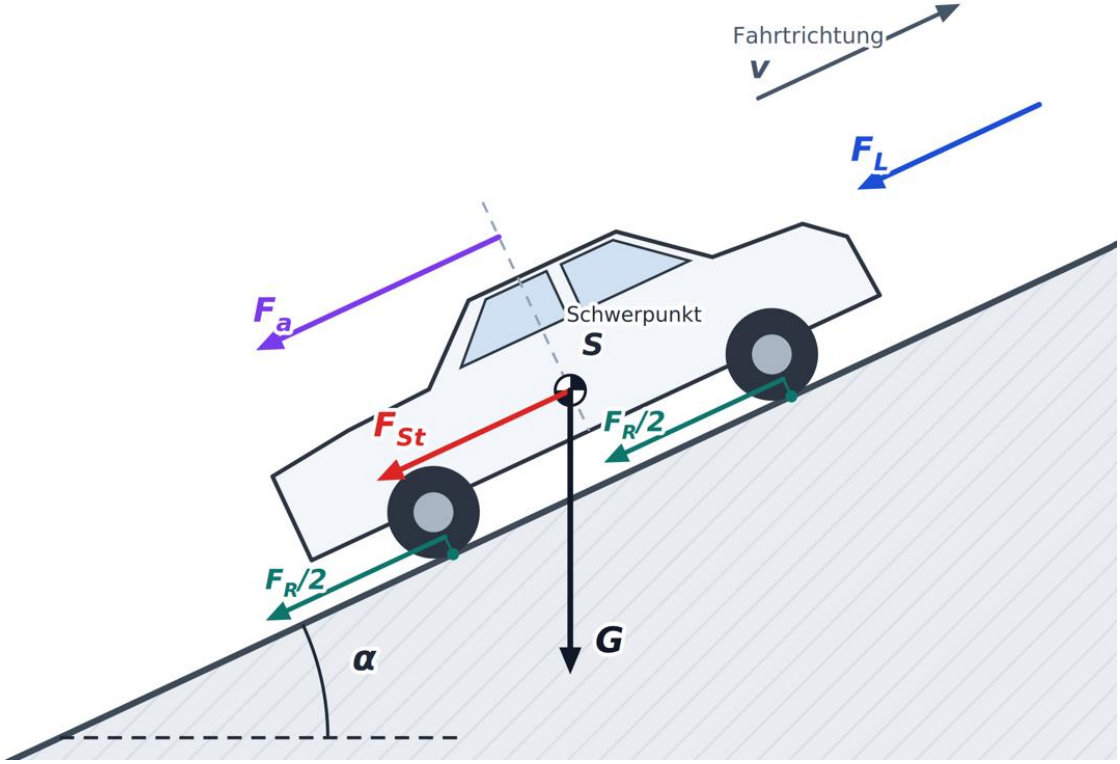


Abbildung 3: Fahrwiderstände am Fahrzeug, Quelle: eigene Darstellung

$$F_{\mathrm{ges}} = F_{\mathrm{R}} + F_{\mathrm{St}} + F_{\mathrm{L}} + F_{\mathrm{a}} \tag{2.1}$$

Abbildung 3 und Formel 2.1 stellen den Gesamtfahrwiderstand am Fahrzeug in der Steigung dar, die einzelnen Kräfte werden im Folgenden einzeln beschrieben [12].

2.2.1 Rollwiderstand

Der Rollwiderstand entsteht im Wesentlichen durch die Verformung des Reifens im Bereich der Aufstandsfläche. Beim Ein- und Auslaufen des Reifens in bzw. aus der Aufstandsfläche wird dieser elastisch verformt. Die dabei aufgewendete mechanische Verformungsarbeit wird aufgrund der Hystereseeigenschaften des Reifenmaterials nicht vollständig zurückgewonnen, sondern durch Reibung teilweise in Wärme umgewandelt. Durch diese Energiewandlung entsteht der Rollwiderstand. Zur Beschreibung des Rollwiderstands wird der dimensionslose Rollwiderstandsbeiwert f_R verwendet:

$$F_\mathrm{R} = f_\mathrm{R} \cdot m \cdot g \tag{2.2}$$

wobei m die Fahrzeugmasse und g die Erdbeschleunigung bezeichnet [6, S. 49]. Der Rollwiderstandsbeiwert f_R wird unter anderem durch die Reifenbauart, die Reifenverformung und den Fülldruck beeinflusst. Darüber hinaus besteht eine Abhängigkeit von der Fahrgeschwindigkeit. Für die Betrachtung der Fahrzeuglängsdynamik kann der Rollwiderstandsbeiwert innerhalb eines begrenzten Geschwindigkeitsbereichs näherungsweise als konstant angenommen werden. Eine konkrete Wertannahme wird in Abschnitt 5.5 beschrieben.

2.2.2 Luftwiderstand

Der Luftwiderstand entsteht durch die Umströmung des Fahrzeugs und wirkt der Bewegung in Fahrtrichtung entgegen. Er setzt sich im Wesentlichen aus Druck- und Reibungskräften zusammen, die durch die Strömung der Luft um die Fahrzeugoberfläche entstehen. Dabei beeinflussen insbesondere die Fahrzeugform und die Strömungsablösung den resultierenden Widerstand. Die Luftwiderstandskraft F_L lässt sich beschreiben durch:

$$F_\mathrm{L} = \frac{1}{2} \cdot \rho_\mathrm{L} \cdot c_\mathrm{W} \cdot A \cdot v^2 \tag{2.3}$$

Dabei bezeichnet die ρ_L die Luftdichte, c_W den dimensionslosen Luftwiderstandsbeiwert, A die in Fahrtrichtung projizierte Querschnitts- bzw. Stirnfläche des Fahrzeugs und v die Anströmgeschwindigkeit [6, S. 49]. Bei Windstille entspricht die Anströmgeschwindigkeit der Fahrgeschwindigkeit des Fahrzeugs.

Der Luftwiderstandsbeiwert charakterisiert dabei den aerodynamischen Einfluss der Fahrzeugform. Er hängt insbesondere von der Gestaltung der Fahrzeugfront, des Fahrzeughecks

und der übrigen Fahrzeugkontur ab. Abgerundete Stirnflächen und eine strömungsgünstige Heckgestaltung können den Luftwiderstandsbeiwert deutlich reduzieren, da dadurch die Strömungsablösung beeinflusst wird. Für Pkw werden je nach Fahrzeugbauform beispielsweise Werte zwischen etwa 0,2 und 0,5 angegeben ([13, S. 171]).

Die Luftdichte ist von den Umgebungsbedingungen, insbesondere von Luftdruck und Temperatur, abhängig. Aufgrund der quadratischen Abhängigkeit von der Anströmgeschwindigkeit nimmt der Luftwiderstand mit zunehmender Geschwindigkeit stark zu. Die hierfür erforderliche Antriebsleistung ergibt sich aus dem Produkt von Kraft und Geschwindigkeit,

$$P_{\mathrm{L}} = F_{\mathrm{L}} \cdot v \tag{2.4}$$

und steigt damit bei konstantem c_{W}, A und ρ_{L} proportional zur dritten Potenz der Geschwindigkeit. Der Luftwiderstand gewinnt daher insbesondere bei höheren Fahrgeschwindigkeiten gegenüber den übrigen Fahrwiderständen an Bedeutung.

2.2.3 Steigungswiderstand

Der Steigungswiderstand beschreibt die Komponente der Gewichtskraft, die bei einer geneigten Fahrbahn in Fahrbahnlängsrichtung wirkt, mit:

$$F_{\mathrm{St}} = m \cdot g \cdot \sin \alpha \tag{2.5}$$

Dabei bezeichnet m die Fahrzeugmasse, g die Erdbeschleunigung und α den Steigungswinkel der Fahrbahn [6, S. 50]. Bei einer Fahrt bergauf wirkt der Steigungswiderstand entgegen der Fahrtrichtung und muss durch die Antriebskraft überwunden werden. Im Gefälle wirkt die Gewichtskraft hingegen als Hangabtriebskraft in Fahrtrichtung und muss gegebenenfalls durch die Bremskraft kompensiert werden.

2.2.4 Beschleunigungswiderstand

Der Beschleunigungswiderstand beschreibt die Trägheitskraft, die bei einer Änderung der Fahrgeschwindigkeit zu überwinden ist. Neben der translatorisch beschleunigten Fahrzeugmasse müssen auch die rotierenden Komponenten des Antriebsstrangs, insbesondere Räder und Getriebe, beschleunigt werden. Diese zusätzlichen rotatorischen Trägheiten werden über den Drehmassenzuschlagsfaktor berücksichtigt:

$$F_a = \varepsilon \cdot m \cdot a \tag{2.6}$$

Dabei bezeichnet a die Längsbeschleunigung des Fahrzeugs und ε den Drehmassenzuschlagsfaktor [6, S. 50]. Dieser berücksichtigt die Trägheitsmomente der rotierenden Komponenten und hängt unter anderem von der Gesamtübersetzung des Antriebsstrangs ab; typische Werte liegen für lang übersetzte Gänge bei etwa 1,05 bis 1,4 [14, S. 82]. Bei Verzögerung wird der Beschleunigungswiderstand negativ und wirkt als treibende Kraft, die von der Bremse oder der generatorisch arbeitenden elektrischen Maschine aufgenommen werden muss.

2.2.5 Vom Fahrwiderstand zum Lastzyklus

Aus den einzelnen Fahrwiderständen ergibt sich die Gesamtkraft F_{ges}, die der Antrieb am Rad aufbringen muss. Mit dem Rollradius r_{w} lässt sich daraus das am Rad benötigte Drehmoment M_{Rad} bestimmen:

$$M_{\mathrm{Rad}} = F_{\mathrm{ges}} \cdot r_{\mathrm{w}} \tag{2.7}$$

Die benötigte Antriebsleistung P_{Bed} ergibt sich aus der Gesamtkraft und der jeweiligen Fahrgeschwindigkeit:

$$P_{\mathrm{Bed}} = F_{\mathrm{ges}} \cdot v \tag{2.8}$$

Für die Auslegung des Antriebs sind neben Kraft und Leistung insbesondere Drehmoment und Drehzahl an der Antriebsmaschine relevant. Diese Größen lassen sich über die Gesamtübersetzung des Antriebsstrangs aus den entsprechenden Radgrößen bestimmen. Dabei wird das Drehmoment durch die Übersetzung geteilt, während sich die Drehzahl entsprechend erhöht. Aufbauend auf diesen Grundlagen wird der Prozess der Lastzyklusanalyse in Abschnitt 4.6 ausführlicher erläutert.

2.3 Elektrische Antriebe und Maschinen

2.3.1 Aufbau und Funktionsweise elektrischer Maschinen

Ein batterieelektrischer Antriebsstrang besteht grundsätzlich aus den Hauptbestandteilen: Traktionsbatterie als Energiespeicher, Leistungselektronik, elektrische Maschine und einem Getriebe (siehe Abbildung 4). Die Leistungselektronik wandelt die Gleichspannung der Batterie in ein Drehstromsystem mit einstellbarer Frequenz und Amplitude um und stellt damit Drehzahl und Drehmoment der Maschine ein.

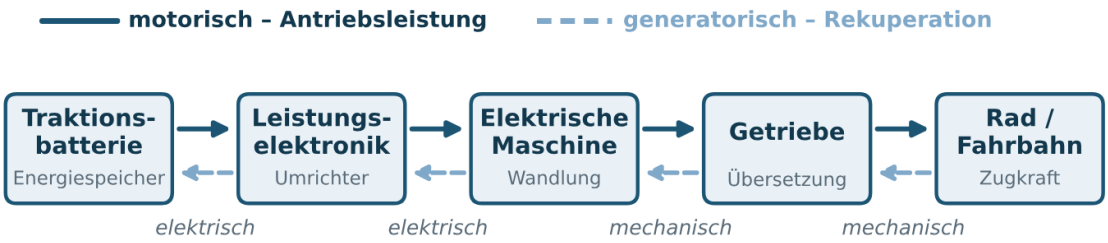


Abbildung 4: Grundlegender Aufbau eines elektrischen Antriebsstrangs, mit Energiefluss (eigene Darstellung)

Elektrische Maschinen bestehen grundsätzlich aus einem feststehenden Stator und einem drehbaren Rotor, zur Veranschaulichung wird in Abbildung 5 eine elektrische Maschine schematisch dargestellt. Die Drehstromwicklung des Stators erzeugt ein rotierendes Magnetfeld. Durch die Wechselwirkung zwischen dem Magnetfeld des Stators und dem Permanentmagnetfeld des Rotors entsteht im Luftspalt der Maschine ein elektromagnetisches Drehmoment. Die verschiedenen Bauarten elektrischer Maschinen unterscheiden sich vor allem darin, wie das Rotorfeld erzeugt wird (siehe Tabelle 1).

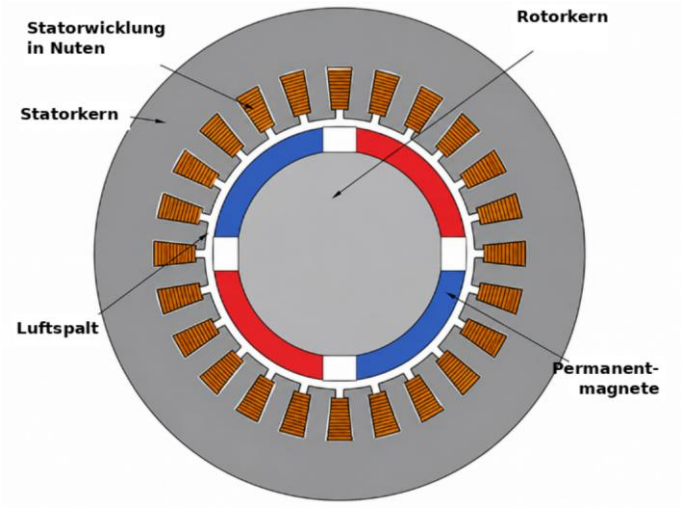


Abbildung 5: Prinzipieller Aufbau einer Permanentmagneterregten Synchronmaschine im Querschnitt (Quelle: [15], Bezeichnungen auf Deutsch geändert)

Bauart	Erzeugung des Rotorfelds	Merkmale
Permanentmagneterregte Synchronmaschine (PMSM)	Permanentmagnete im Rotor	hohe Leistungsdichte und Wirkungsgrade; im Fahrzeugbereich am weitesten verbreitet; benötigt Magnetwerkstoffe mit Seltenerdmetallen
Asynchronmaschine (ASM)	Induktion im Käfigläufer oder Schleifringläufer	robust und kostengünstig, magnetfrei; geringfügig niedrigere Wirkungsgrade i.A.
Fremderregte Synchronmaschine (FESM)	bestromte Rotorwicklungen	Rotorfeld einstellbar, dadurch günstiges Teillastverhalten; benötigt eine Übertragung des Erregerstroms auf den Rotor
Reluktanzmaschine (SRM)	Kein eigenes Rotorfeld vorhanden; Ausrichtung entsteht an magnetischem Widerstand	einfacher, magnetfreier Aufbau, rotorseitig verlustarm; jedoch erhöhte Eisenverluste und ungünstiger Leistungsfaktor, dadurch geringerer Systemwirkungsgrad; erhöhte Drehmomentwelligkeit und Geräuschentwicklung

Tabelle 1: Bauarten elektrischer Traktionsmaschinen (nach [16, Kap. 7.4, 7.5])

Für den Fahrzeugantrieb wesentlich ist der Vier-Quadranten-Betrieb (Abbildung 6): Dieselbe Maschine kann in beiden Drehrichtungen sowohl motorisch antreiben als auch generatorisch bremsen. Beim generatorischen Bremsen wird ein Teil der kinetischen Energie in elektrische Energie gewandelt und in die Batterie zurückgespeist. Dies versteht man unter Rekuperation. Diese Eigenschaft unterscheidet den elektrischen Antrieb grundlegend vom Verbrennungsmotor (vgl. [16]).

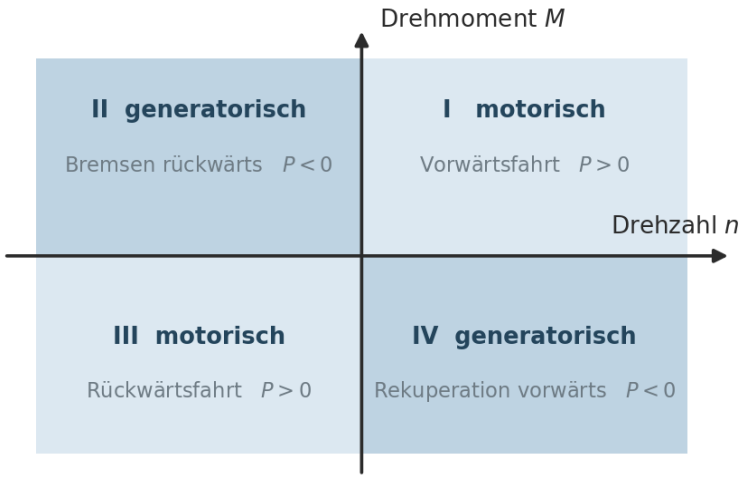


Abbildung 6: Vier-Quadranten-Betrieb einer elektrischen Maschine (eigene Darstellung)

2.3.2 Betriebsrelevante Kenngrößen

Das Betriebsverhalten einer elektrischen Maschine wird durch das Drehmoment-Drehzahl-Kennfeld beschrieben, das in Abbildung 7 dargestellt ist. Im Grunddrehzahlbereich begrenzt der zulässige Strom das Drehmoment: Die Maschine stellt ihr maximales Moment $M_{\max}$ konstant, die Leistung wächst proportional zur Drehzahl. Oberhalb der Eckdrehzahl n_{eck} reicht die Spannung nicht mehr aus, um den Strom gegen die wachsende induzierte Gegenspannung zu treiben. Durch Feldschwächung bleibt die Leistung näherungsweise bei $P_{\max}$ konstant, während das Moment hyperbolisch abfällt. Die Maximaldrehzahl n_{ma} begrenzt den Betriebsbereich mechanisch[17, S. 7/3, 7/29, 7/30].

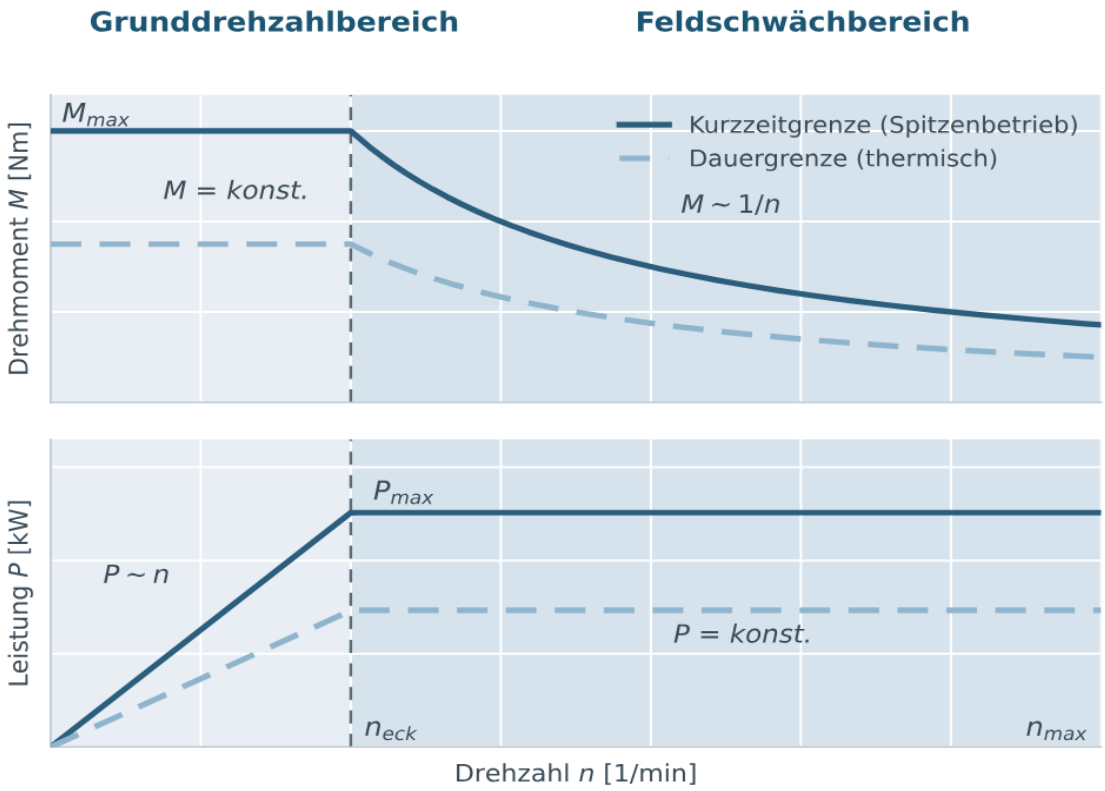


Abbildung 7: Drehmoment- und Leistungsverlauf einer elektrischen Maschine über der Drehzahl (eigene Darstellung, Prinzipdarstellung)

Maßgeblich für die Höhe dieser Grenzen ist die Erwärmung von Wicklungen, Magneten und der Leistungselektronik. Kurzzeitig, im Bereich von Sekunden bis wenigen Minuten, kann eine Maschine deutlich über ihrer Dauerleistung betrieben werden, auf Dauer ist dies jedoch schädlich. Dadurch ist nur die durch die Kühlung bestimmte Dauerleistung dauerhaft abrufbar.

Eine weitere auslegungsrelevante Kenngröße ist der Wirkungsgrad η, definiert als das Verhältnis der abgegebenen zur aufgenommenen Leistung; im Motorbetrieb also $\eta = \frac{P_{\text{Mechanisch}}}{P_{\text{Elektrisch}}}$,

im generatorischen Betrieb entsprechend umgekehrt [16, S. 260]. Die Verluste setzen sich im Wesentlichen aus Stromwärmeverlusten in den Wicklungen, Ummagnetisierungsverlusten im Blechpaket sowie Reibungs- und lastabhängigen Zusatzverlusten zusammen und sind zugleich die Ursache der zuvor beschriebenen Erwärmung. Der Wirkungsgrad ist damit keine Konstante, sondern eine Funktion des Betriebspunktes über Moment und Drehzahl. Je nach Betriebspunkt erreichen elektrische Maschinen einen Wirkungsgrad zwischen 80 und 95 %, wobei niedrige Teillastbereiche darin nicht berücksichtigt sind. Der zugehörige Wechselrichter liegt bei rund 90 bis 95 % [16, S. 73].

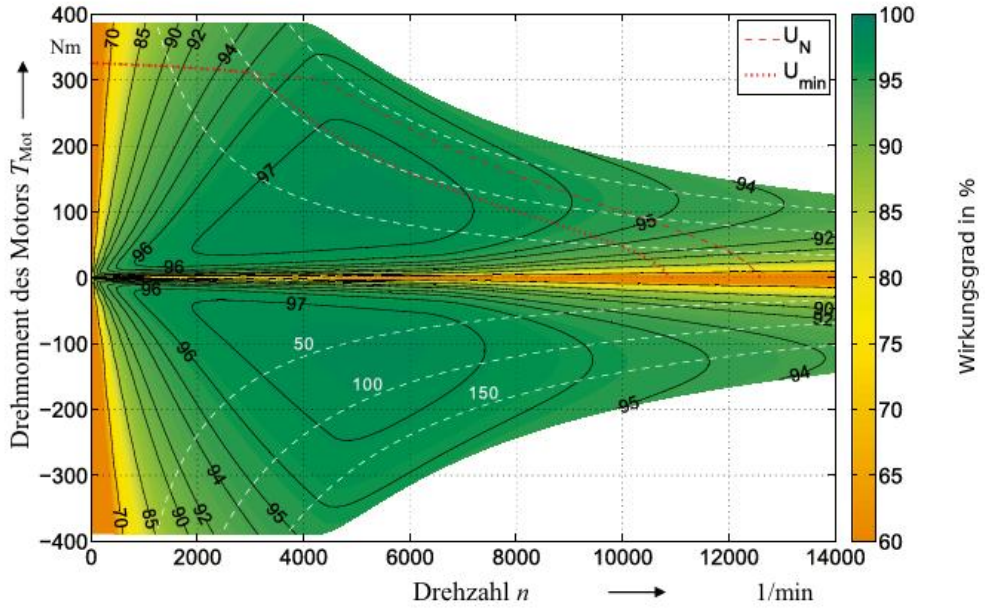


Abbildung 8: Simuliertes Kennfeld einer PMSM zum Antrieb eines Elektroautos (Quelle:[16, S. 29])

Abbildung 8 zeigt ein Wirkungsgradkennfeld einer permanentmagneterregten Synchronmaschine als Fahrzeugantrieb (innere Motorverluste), die obere Hälfte über der horizontalen Nulllinie stellt dabei den Motorbetrieb und die untere Hälfte den Generatorbetrieb dar. Für die Grobauslegung wird in dieser Arbeit ein konstanter, betriebspunktunabhängiger Wirkungsgrad angesetzt, da im Auslegungsstadium noch kein Motorkennfeld der zu dimensionierenden Maschine vorliegt.

2.4 Datenhaltung

2.4.1 Grundlagen relationaler Datenmodelle

Eine Datenbank ist eine strukturierte, dauerhaft gespeicherte Sammlung logisch zusammengehöriger Daten. Ihre Verwaltung übernimmt ein Datenbankmanagementsystem, das Konsistenzsicherung, Zugriffskontrolle, Mehrbenutzerbetrieb und eine Abfragesprache implementiert. Das heute dominierende Datenmodell ist das relationale Modell. Es organisiert alle Daten in Relationen, anschaulich in Tabellen (Abbildung 8): Jede Zeile (Tupel) beschreibt ein Objekt, jede Spalte (Attribut) eine Eigenschaft mit definiertem Wertebereich.

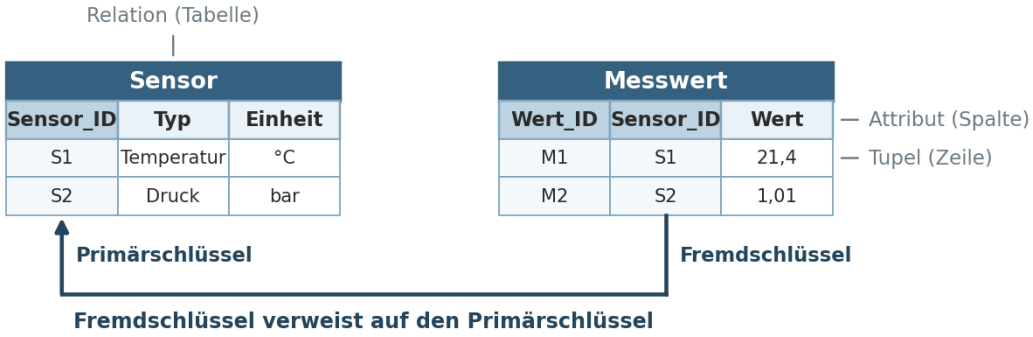


Abbildung 9: Beispielhaftes relationales Datenmodell mit zwei Relationen (eigene Darstellung)

Ein Primärschlüssel identifiziert jede Zeile eindeutig, ein Fremdschlüssel verweist auf den Primärschlüssel einer anderen Relation und stellt so Beziehungen zwischen den Tabellen her. Die Forderung, dass jeder Fremdschlüsselwert einen existierenden Primärschlüssel bezeichnet, wird als referentielle Integrität bezeichnet (vgl.[18]).

2.4.2 Datenmodellierung und Schemaentwurf

Der Entwurf einer Datenbank erfolgt über mehrere Abstraktionsebenen: Ein konzeptionelles Modell beschreibt implementierungsunabhängig, welche Objekte mit welchen Eigenschaften und Beziehungen zu speichern sind. Daraus wird das logische Schema, das im relationalen Fall die Tabellenstruktur darstellt, abgeleitet. Dessen physische Ablage bestimmt schließlich das Speichersystem. Für die konzeptionelle Modellierung hat sich das Entity-Relationship-Modell (ER-Modell) etabliert. Es beschreibt die Anwendungswelt durch Entitätstypen mit ihren Attributen sowie durch Beziehungstypen, die mit Kardinalitäten versehen werden (Abbildung 9).

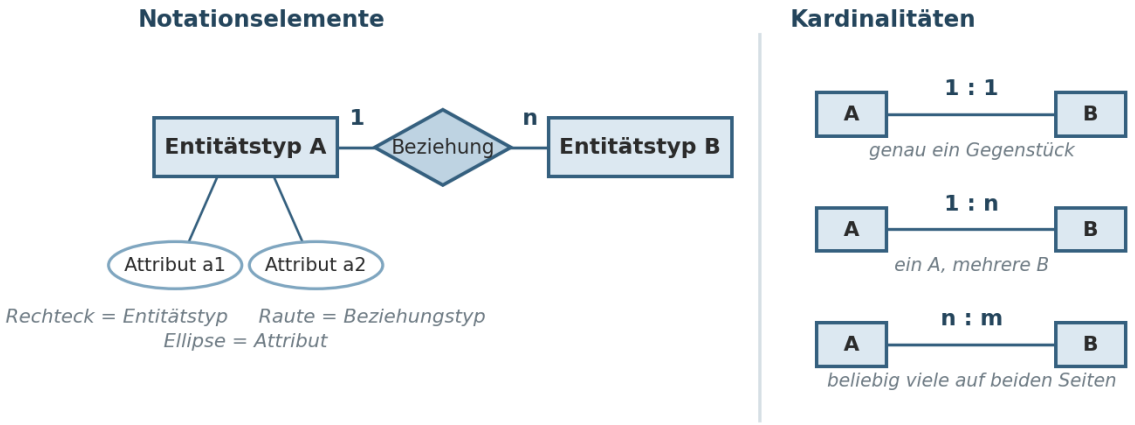


Abbildung 10: Notationselemente und Kardinalitäten des Entity-Relationship-Modells (eigene Darstellung)

Die Überführung in ein relationales Schema erfolgt nach klar definierten Regeln (vgl. [19]): Jeder Entitätstyp wird als eigene Tabelle modelliert, und die zugehörigen Attribute werden als Spalten abgebildet. 1:n-Beziehungen werden durch einen Fremdschlüssel auf der n-Seite dargestellt, während n:m-Beziehungen mithilfe einer separaten Beziehungstabelle realisiert werden. Beispiel: Ein Studierender kann mehrere Lehrveranstaltungen belegen, während eine Lehrveranstaltung von mehreren Studierenden besucht werden kann. Im ER-Modell wird dies als n:m-Beziehung zwischen den Entitätstypen *Studierender* und *Lehrveranstaltung* modelliert. Im relationalen Schema entstehen daraus die Tabellen *STUDIERENDER* und *LEHRVERANSTALTUNG* sowie die Beziehungstabelle *BELEGUNG*. Diese enthält die jeweiligen Schlüssel der beiden Entitätstabellen als Fremdschlüssel, beispielsweise *Matrikelnummer* und *Veranstaltungsnummer*. Besitzt die Beziehung zusätzliche Eigenschaften, etwa ein Anmeldedatum oder eine erzielte Note, werden auch diese in der Tabelle *BELEGUNG* gespeichert.

Das Ziel des Schemaentwurfs besteht darin, ein redundanzarmes Schema zu schaffen, in dem jede Information nur einmal gespeichert wird. Die Normalisierung systematisiert diesen Prozess durch aufeinander aufbauende Normalformen, wobei in der Praxis häufig die dritte Normalform angestrebt wird. Die Reduktion von Datenredundanzen trägt dazu bei, Einfüge-, Änderungs- und Löschanomalien zu vermeiden, die durch mehrfach gespeicherte Informationen entstehen können (vgl.[20, Kap. 6])

3 Stand der Technik

Dieses Kapitel positioniert die vorliegende Arbeit im Kontext des aktuellen Stands der Technik. Der erste Abschnitt vergleicht bestehende Fahrsimulationsprogramme mit der Zielsetzung dieser Arbeit. Im Folgenden werden verschiedene Routing-Dienste und Datenquellen gegenübergestellt, die Auswahl der Programmiersprache begründet sowie die für Datenbanken und die Validierung erforderlichen Referenzdatenquellen vorgestellt.

3.1 Bestehende Fahrsimulationsprogramme und -Werkzeuge

Fahrsimulationswerkzeuge sind in der Fahrzeug- und Antriebsentwicklung fest etabliert. Hersteller und Zulieferer setzen eigene Simulationsumgebungen auch für die Grobauslegung von Antrieben ein (vgl.[21]). Diese internen Werkzeuge sind nicht frei zugänglich, da die darin gebundene Methodik als Wettbewerbsvorteil gilt. Diese Abschottung hat strukturelle Nachteile: In isolierten Insellösungen werden gleichartige Modelle und Fehler mehrfach entwickelt, und bei unzureichender Dokumentation und personellen Wechseln veralten die Programme schnell. Gerade in der gegenwärtigen Krisensituation, in der der Wettbewerb nicht zwischen inländischen Unternehmen ausgetragen werden sollte, da auf internationaler Ebene eine erhebliche Konkurrenz besteht, ist ein intensiverer Austausch von entscheidender Bedeutung.

Kommerzielle Umgebungen wie AVL CRUISE oder IPG CarMaker sind kostenpflichtig und auf eine Parametrisierungstiefe ausgelegt, die in der Vorauslegungsphase noch nicht vorliegt, zudem ist der Quellcode nicht zugänglich. Für einen Vergleich verbleiben damit im Wesentlichen frei zugängliche Werkzeuge, die in Tabelle 2 dargestellt sind.

Werkzeug (Herausgeber), Jahr	Lizenz / Sprache	Ansatz und Fokus	Bewertung bezüglich der Zielsetzung
ADVISOR (NREL, USA), 2002	Quellcode verfügbar; MATLAB/Simulink	zyklusbasierte, rückwärtsrechnende Antriebsstrang- und Verbrauchssimulation	seit Jahren nicht weiterentwickelt; MATLAB-Lizenz erforderlich; keine Routensimulation [22]
VECTO (Europäische Kommission), 2018 (aktuellster Stand 2025)	Quellcode verfügbar; C# / .NET	amtliche CO2-Zertifizierungssimulation schwerer Nutzfahrzeuge	auf regulatorisch festgelegte Missionsprofile beschränkt; keine Maschinenvorauslegung [23]
emobpy (DIW Berlin), 2021	Quellcode verfügbar; Python tool	statistische Erzeugung von Zeitreihen batterieelektrischer Fahrzeuge	auf Mobilitätsstatistiken gestützt, nicht streckenindividuell; keine Momenten- und Drehzahlanalyse [24]
RoadHopper (A. Wolf, KIT), 2015	Quellcode verfügbar; Scala/JVM	streckenbasierte Fahrzyklusgenerierung auf Basis von GraphHopper	konzeptionell am nächsten verwandt; seit 2015 unverändert, liefert nur ein v(t)-Profil ohne Lastanalyse [25]

Tabelle 2: Frei zugängliche Fahrsimulations- und Fahrzykluswerkzeuge im Vergleich

Dem Ansatz dieser Arbeit am nächsten kommt die Diplomarbeit „Map-Based Driving Cycle Generation“ von Andreas Wolf. Er erweitert den frei zugänglichen Routenplaner GraphHopper um ein Fahrer- und Fahrzeugmodell zum Simulator RoadHopper und erzeugt damit Fahrzyklen für frei wählbare Routen aus OpenStreetMap-Daten [25]. Die Arbeit belegt die Tragfähigkeit des streckenbasierten Ansatzes, unterscheidet sich jedoch in Zweck und Umfang wesentlich von der vorliegenden: RoadHopper liefert ausschließlich ein Zeit-Geschwindigkeits-Profil, dessen Auswertung externen Werkzeugen überlassen bleibt. Eine Lastanalyse für die Vorauslegung elektrischer Maschinen ist nicht Gegenstand, der Antriebsstrang wurde bewusst nicht modelliert (vgl. [25, S. 29]), und den erreichten Modellstand stuft der Autor selbst als unzureichend ein, um reale Fahrverläufe eng nachzubilden (vgl. [25, S. 78]). Zudem ist die Software seit 2015 unverändert und daher veraltet. Als Scala-/JVM-Anwendung erfordert sie eine lokal modifizierte GraphHopper-Instanz samt Import von OpenStreetMap-Rohdaten, was die Einstiegshürde gegenüber einer Python-Anwendung mit Cloud-Schnittstelle erheblich erhöht.

Damit zeigt der Vergleich eine Lücke: Kein frei zugängliches Werkzeug verbindet die Erzeugung streckenindividueller Fahrzyklen aus realen Routen mit einer auslegungsorientierten Lastanalyse für elektrische Maschinen in einer leicht installierbaren, quelloffenen Umgebung, die etablierten Werkzeuge sind rein zyklusbasiert (ADVISOR, VECTO), verfolgen andere Fragestellungen (emobpy) oder sind veraltet und ohne Lastauswertung (RoadHopper). Genau diese Lücke adressiert die vorliegende Arbeit.

3.2 Routing-Dienste und Kartendaten

Für die streckenbasierte Simulation werden für jede Route die Punktfolge mit Koordinaten und Höhenangaben sowie abschnittsweise Tempolimits und Straßenklassen benötigt (vgl. Abschnitt 4.2.1). Daraus ergeben sich mehrere Auswahlkriterien für Routing-Dienste: die Bereitstellung sämtlicher erforderlicher Informationen in einer einzigen Anfrage, ein kostenfreies Nutzungskontingent sowie eine dauerhaft frei verfügbare Datenbasis. Tabelle 3 bietet einen Vergleich verbreiteter Routing-Dienste anhand dieser Kriterien.

Dienst	Datenbasis	Höhenprofil in der Antwort	Tempolimit / Straßenklasse	Nutzungsbedingungen und Kosten
GraphHopper Directions API	OpenStreetMap	ja (SRTM, in die Routenantwort integriert)	ja (Path Details max_speed und road_class)	Speicherung zulässig; kostenfreies Kontingent; Quellcode öffentlich zugänglich (vgl. [26])
Project OSRM (Open Source Routing Machine)	OpenStreetMap	nein	nein	Quellcode öffentlich; Demo-Server ohne Verfügbarkeitszusagen, produktiv nur im Eigenbetrieb (vgl. [27])
openrouteservice	OpenStreetMap	optional zuschaltbar	nein (nur Wegtyp- und Steigungsklassen)	kostenfreies Kontingent (vgl. [28])
Google Routes API	proprietär	nein (separate Elevation API)	nein (nur über zugangsbeschränkte Zusatzdienste)	Speicherung der Inhalte untersagt; kostenpflichtig (vgl. [29])
HERE Routing API	proprietär	nein	ja (Tempolimit je Streckenabschnitt)	kommerzielles Lizenzmodell; Weiterverwendung eingeschränkt (vgl. [30])

Tabelle 3: Vergleich verbreiteter Routing-Dienste

Die Auswahl fällt auf die GraphHopper Directions API, da sie als einziger untersuchter Dienst die Routengeometrie einschließlich Höhenprofil, Tempolimits und Straßenklassen in einer einzigen Anfrage bereitstellt und zudem die dauerhafte Weiterverwendung der Antworten gestattet. Dies ermöglicht eine offline reproduzierbare Simulationsbasis. Das kostenfreie Nutzungskontingent ist ausreichend, da jede Strecke nur einmal abgefragt wird. Die Möglichkeit, den Dienst bei Bedarf lokal auf eigener Hardware zu betreiben, gewährleistet zusätzlich die langfristige Verfügbarkeit. Bereits Wolf identifizierte GraphHopper als die geeignetste Grundlage, damals noch als lokal installationspflichtigen Dienst (vgl. [25]). Die heutige Cloud-Schnittstelle macht diesen Installationsaufwand jedoch überflüssig.

Mit dieser Entscheidung sind die Datenquellen für Höhen- und Streckeninformationen festgelegt. Die Höhenangaben je Routenpunkt bezieht GraphHopper aus dem Datensatz der Shuttle Radar Topography Mission (SRTM), einer elftägigen Space-Shuttle-Mission der NASA in Zusammenarbeit mit internationalen Partnern wie dem DLR und der italienischen Raumfahrtagentur ASI. Während der Mission vom 11. bis 22. Februar 2000 wurden mithilfe von Single-Pass-Radarinterferometrie nahezu flächendeckend Radardaten der Landoberfläche zwischen 60° nördlicher und 56° südlicher Breite erfasst. Daraus wurde ein nahezu globales digitales Höhenmodell mit einer räumlichen Auflösung von etwa 30 m (1 Bogensekunde) erstellt [31]. Der Datensatz ist frei verfügbar und wird von GraphHopper serverseitig integriert, sodass keine separate Abfrage, Georeferenzierung oder Interpolation erforderlich ist. Die Höhengenauigkeit im einstelligen Meterbereich ist für die Bestimmung des Steigungswiderstands

ausreichend, da die Steigung segmentweise über Distanzen von mehreren zehn Metern hinweg gemittelt wird (vgl. Abschnitt 5.4.2). Höher aufgelöste amtliche Geländemodelle der Landesvermessung bieten zwar Rasterweiten von einem Meter, sind jedoch nur regional verfügbar und würden die freie überregionale Nutzbarkeit einschränken.

Die Streckendaten werden aus OpenStreetMap bezogen, einer weltweit gemeinschaftlich gepflegten und unter freier Lizenz verfügbaren Geodatenbank. Diese Daten gelangen auf zwei Wegen in die Simulation: Zum einen mittelbar über das darauf aufbauende GraphHopper-Routing, das abschnittsweise Straßenklassen und Tempolimits bereitstellt, zum anderen unmittelbar über die Overpass-Schnittstelle, mit der gezielt Ampelpositionen abgefragt werden, die von GraphHopper nicht oder fehlerhaft erfasst werden.

3.3 Wahl der Programmiersprache

Die Wahl der Programmiersprache folgt unmittelbar aus der Zielsetzung dieser Arbeit (vgl. Abschnitt 1.2). Da das entstehende Werkzeug Unternehmen und Hochschulen zur Erprobung und Weiterentwicklung bereitgestellt werden soll, muss es ohne Lizenzkosten weitergegeben, von Dritten nachvollzogen und erweitert sowie mit vertretbarem Aufwand installiert werden können. Hinzu kommt die Forderung nach kurzen Rechenzeiten: Die angestrebte Variation von Fahrzeugparametern und Strecken ist erst dann sinnvoll möglich, wenn eine Simulation im Dialog wiederholt werden kann.

MATLAB ist in der Fahrzeugtechnik verbreitet und bietet mit Simulink eine etablierte Simulationsumgebung, erfordert jedoch je Anwender eine kostenpflichtige Lizenz – ein Widerspruch zur angestrebten freien Bereitstellung an Unternehmen und Hochschulen. Java und das darauf aufbauende Scala erreichen eine Ausführungsgeschwindigkeit nahe der von C++: In einer sprachvergleichenden Messreihe über zehn Standardalgorithmen benötigt Java das 1,89-Fache, Python dagegen das 71,9-Fache der Rechenzeit von C++ [32]. Dem steht ein deutlich höherer Aufwand bei Installation und Weitergabe gegenüber, da zum Übersetzen und Ausführen eines Scala-Projekts neben einer Java-Laufzeitumgebung ein eigenes Build-Werkzeug erforderlich ist (vgl. [33]), die Installationshürde von RoadHopper veranschaulicht dies (vgl. Abschnitt 3.1). C++ erzielt die höchste Rechenleistung, verursacht aber den größten Entwicklungsaufwand und bietet die benötigten Fachbibliotheken nicht von Haus aus.

Python ist demgegenüber für Entwickler wie Anwender kostenfrei: Der Referenzinterpreter CPython steht mit seiner Standardbibliothek unter einer gebührenfreien Lizenz, die Nutzung,

Veränderung und Weitergabe ausdrücklich einschließt (vgl. [34]); für die eingesetzten Fachbibliotheken gilt Entsprechendes. Wer das Werkzeug erhält, muss dafür keine Lizenz erwerben. Hinzu kommt die weite Verbreitung der Sprache: Im jährlichen Sprachvergleich der IEEE Spectrum, der unter anderem Suchanfragen, Beiträge in Entwicklerforen, Nennungen in Fachveröffentlichungen und Aktivität auf Softwareplattformen auswertet, belegt Python den ersten Platz (vgl. [35]). Für die vorgesehene Weitergabe ist das wesentlich, da sich das Programm nur dann von Dritten weiterentwickeln lässt, wenn diese die Sprache bereits beherrschen oder sich ohne Einarbeitung in eine Nischensprache einlesen können.

Fachlich entscheidend ist schließlich der Bibliotheksbestand: *NumPy* bildet die Grundlage des wissenschaftlichen Python-Ökosystems und stellt zusammen mit den darauf aufbauenden Bibliotheken *pandas* und *Matplotlib* eine in der Forschung breit eingesetzte Werkzeugkette für numerische Berechnungen, tabellarische Datenhaltung und Visualisierung bereit (vgl. [36]). Damit werden sämtliche in dieser Arbeit benötigten Aufgaben abgedeckt. Die Bedienoberfläche lässt sich mit dem in der Standardbibliothek enthaltenen *Tkinter* ohne zusätzliche Abhängigkeiten umsetzen. Der genannte Geschwindigkeitsnachteil der Interpretersprache wird durch Vektorisierung der numerisch kritischen Teile mit *NumPy* weitgehend ausgeglichen, da die eigentlichen Rechenoperationen dabei in übersetztem Maschinencode ausgeführt werden (vgl. [36]). Die Rechenzeiten bleiben dadurch im Sekundenbereich, sodass die geforderten iterativen Parameterstudien möglich sind. Die Implementierung erfolgt daher in Python 3, da dies die neueste Python-Instanz ist und daher zukunftsfähig.

3.4 Verfügbare Referenzdatenquellen für Fahrzeuge und Fahrzyklen

Für die Datenablage in den Datenbanken und die Validierung sind drei Arten von Referenzdaten erforderlich: Fahrzeugparameter, Fahrzyklen sowie Prüf- und Messwerte realer Fahrzeuge als Vergleichsmaßstab.

Fahrzeugdaten: Verbindliche Herstellerangaben zu Leistung, Masse, Verbrauch und Reichweite werden im Rahmen der Typgenehmigung nach WLTP festgelegt und sind für Neufahrzeuge zu veröffentlichen [8]. Diese Angaben dienen zudem der späteren Validierung von Simulationen anhand realer Prüfwerte. Eine aggregierte, redaktionell geprüfte Sammlung dieser Daten stellt der Allgemeine Deutsche Automobil-Club e.V. (ADAC) bereit [37]. Die European Environment Agency (EEA) bietet ebenfalls eine öffentliche Datenbank, in der jedes in der EU registrierte Fahrzeug und dessen meldepflichtige Angaben gelistet sind [38]. Darüber hinaus

sind Datenquellen der United States Environmental Protection Agency (US EPA) öffentlich verfügbar, umfassen jedoch im Vergleich zu den EU-Daten einen deutlich geringeren Umfang an Fahrzeugdaten [39]. Auslegungsrelevante Detailparameter wie Stirnfläche, Luftwiderstandsbeiwert oder genaue Übersetzungen werden von den Herstellern kaum veröffentlicht. Die Fachliteratur liefert hierfür klassentypische Wertebereiche, beispielsweise für Stirnfläche, Rollwiderstandsbeiwert, Übersetzung und Drehmassenzuschlag (vgl. [22, Tab. 3.1, Abb. 2.5, Abb. 5.4]).

Fahrzyklusdaten: Der WLTC ist tabellarisch dokumentiert und frei zugänglich (vgl. Abschnitt 2.1.2). Weitere Fahrzyklen, wie die US-amerikanischen Prüfzyklen FTP-75 und US06, sind über die US-Umweltbehörde EPA frei verfügbar [40].

4 Konzeption

Aufbauend auf den Grundlagen und der Analyse des Stands der Technik entwickelt dieses Kapitel das Konzept der Fahrsimulationsumgebung. Ausgehend von der Zielsetzung der Arbeit – der Vorauslegung elektrischer Fahrzeugantriebe – werden zunächst die Anforderungen an das System erhoben. Daraus wird die Gesamtarchitektur abgeleitet, bevor die Teilsysteme im Einzelnen entworfen werden: die Fahrzyklusdatenbank, die Fahrzeugdatenbank und die streckenbasierte Szenariengenerierung. Zuletzt wird festgelegt, welche Methodik zur Ableitung der Anforderungen an eine elektrische Maschine aus den erzeugten Lastzyklen verwendet werden soll.

4.1 Anforderungsanalyse

Der Ausgangspunkt der Konzeption ist der in Kapitel 1 beschriebene Anwendungsfall: In der frühen Phase der Antriebsentwicklung soll abgeschätzt werden, welche Anforderungen ein geplantes Fahrzeug im realen Fahrbetrieb an seine elektrische Maschine stellt – zu einem Zeitpunkt, zu dem weder ein Prototyp noch ein gemessenes Motorkennfeld existiert. Normierte Fahrzyklen wie der WLTC bilden hierfür lediglich einen Teil der Realität ab: Sie sind streckenneutral definiert und ermöglichen einfache Vergleiche, enthalten jedoch keine Topographie und berücksichtigen keine individuellen Einsatzprofile, wie beispielsweise eine Pendelstrecke mit langen Steigungsabschnitten. Reale Fahrerprobungen sind in der Vorauslegungsphase ebenfalls nicht verfügbar. Aus dieser Lücke ergibt sich die zentrale funktionale Anforderung

an das System: die synthetische Erzeugung realitätsnaher, streckenindividueller Fahrsimulationen aus frei wählbaren Strecken, ergänzt durch die Simulation zyklusbasierter Fahrzyklen. Tabelle 4 fasst die funktionalen Anforderungen (FA) zusammen.

ID	Anforderung	Beschreibung
FA1	Streckenbasierte Lastzyklusgenerierung	Erzeugung eines Geschwindigkeits- und Lastprofils aus einer frei wählbaren Strecke (Eingabe von Start und Ziel als Adresse oder Koordinaten)
FA2	Berücksichtigung realer Streckeneigenschaften	Topographie, Tempolimits, Straßenklassen, Kurven, Ampeln und Verkehr beeinflussen das erzeugte Profil
FA3	Zyklusbasierte Fahrsimulation	Abfahren normierter Referenzzyklen (insbesondere WLTC) sowie beliebiger nutzerdefinierter Profile mit demselben Fahrzeugmodell
FA4	Parametrierbares Fahrzeugmodell	Masse, Aerodynamik, Rollwiderstand, Kraftschluss sowie Antriebsstrang (Leistung, Drehmoment) mit variabler Übersetzung und Motordrehzahlgrenze
FA5	Fahrzeugdatenbank	Hinterlegte Parametersätze typischer Fahrzeugklassen als Beispiel, einfaches hinzufügen neuer Fahrzeuge soll ermöglicht werden
FA6	Fahrervariation	Unterschiedliche Fahrertypen erlauben aus derselben Strecke systematisch unterschiedlich aggressive Lastzyklen zu erzeugen
FA7	Auslegungsorientierte Auswertung	Zeitreihen, Energie und Verbrauch, Maximal- und Mittelwerte von Leistung und Moment sowie ein Motorkennfeld mit Betriebspunkten einschließlich Rekuperationspotenzial
FA8	Reichweitenabschätzung	Ermittlung der Reichweite aus einer nutzbaren Energiekapazität
FA9	Dokumentation und Wiederverwendung	Speichern und Laden von Routen, Export der Ergebnisse als Dokument

Tabelle 4: Funktionale Anforderungen an die Fahrsimulationsumgebung

Neben den funktionalen Anforderungen bestimmen die nichtfunktionellen Anforderungen (NFA) den Entwurf maßgeblich. Diese sind orientierend an der ISO 25010:2023 festgelegt und in Tabelle 5 dargestellt [41]. Als Zielgruppen werden im Rahmen dieser Arbeit Entwicklungsingenieure in der frühen Auslegungsphase, die das Werkzeug ohne spezielle Programmierkenntnisse bedienen sollen, sowie Studierende angenommen.

ID	Anforderung	Beschreibung
NFA1	Interaktivität	Rechenzeiten im Sekundenbereich, damit iterative Parameterstudien im Dialog möglich sind.
NFA2	Bedienbarkeit	Vollständige Bedienung über eine grafische Oberfläche ohne Programmierkenntnisse; verständliche Parametererläuterungen.
NFA3	Erweiterbarkeit der Datenbestände	Fahrzyklus- und Fahrzeugdatenbank sind mit einer Tabellenkalkulation, also ohne Programmierung, pfleg- und erweiterbar.
NFA4	Reproduzierbarkeit	Identische Eingaben liefern identische Ergebnisse; Datenbestände sind versionierbar; gespeicherte Routen sind offline wiederholbar simulierbar.
NFA5	Robustheit	Fehlertolerante Datenimporte; nicht fahrbare Parameterkombinationen werden erkannt und verständlich diagnostiziert statt als Programmfehler zu erscheinen.
NFA6	Geringe Einstiegshürden	Frei zugängliche Datenquellen, geringer Abhängigkeitsumfang der Software, Schutz von Zugangsschlüsseln bei Weitergabe.

Tabelle 5: Nichtfunktionale Anforderungen an die Fahrsimulationsumgebung

Die Abgrenzungen des Konzepts sind ebenso bewusst gewählt wie die Anforderungen. Die Querdynamik wird nicht simuliert (vgl. Abschnitt 1.3), ein gemessenes Motorkennfeld wird nicht als Eingabe gefordert (vgl. Abschnitt 2.3.2); die Beschreibung über Eckdaten stellt hier die angemessene Modellierungstiefe dar. Der umgebende Verkehr wird nicht mikroskopisch, also nicht fahrzeugindividuell, sondern als stochastische Begrenzung abgebildet, um schnelle Rechenzeiten zu ermöglichen. Diese Abgrenzungen minimieren den Parametrisierungsaufwand und sind auf den Einsatzzweck abgestimmt.

4.2 Gesamtarchitektur und streckenbasierte Fahrsimulation

Die Architektur gliedert sich basierend auf den Anforderungen in vier Schichten (siehe Abbildung 10). Die oberste Schicht besteht aus externen Datenquellen: einem Routing-Dienst, der bei Start- und Zielangaben die Route inklusive Höhenprofil, Tempolimits und Straßenklassifizierungen liefert, sowie OpenStreetMap, CARTO, Esri für Ampelpositionen und Kartendarstellung, und die WLTC-Fahrzyklen als Referenzprofile. Die Datenhaltungsschicht soll das System von diesen Quellen entkoppeln: Einmal abgerufene Routen werden lokal gespeichert, während Fahrzeug- und Fahrzyklusdaten in dateibasierten Datenbanken verwaltet werden. Der Verarbeitungskern umfasst die Fachlogik – Streckenaufbereitung, Fahrer-, Fahrzeug- und Ver-

kehrsmodelle sowie den Simulationskern für die Längsdynamik. Die Auswertungsschicht enthält die grafische Oberfläche zur Parametrierung, Kennzahlenberechnung und Ergebnisbereitstellung.

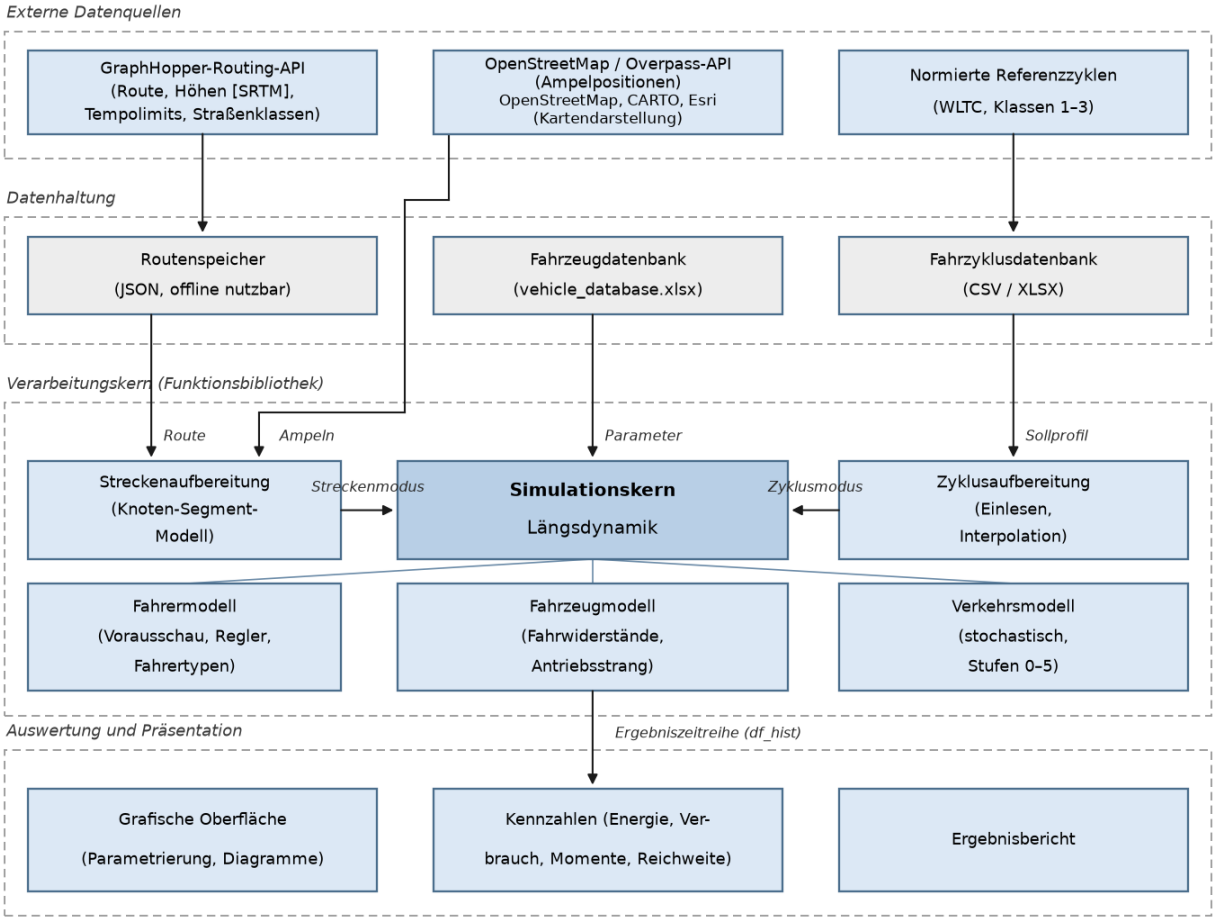


Abbildung 11: Gesamtarchitektur der Fahrsimulationsumgebung (eigene Darstellung)

Diese Architektur wird durch vier zentrale Entwurfsentscheidungen geprägt. Die erste Entscheidung betrifft das Prinzip zweier Betriebsarten innerhalb eines gemeinsamen Simulationskerns: Strecken- und Zyklusmodus unterscheiden sich ausschließlich in der Herkunft der Sollgeschwindigkeit. Im Streckenmodus generiert ein vorausschauendes Fahrermodell den Sollwert basierend auf den Streckeneigenschaften, während im Zyklusmodus das Referenzprofil den Sollwert direkt vorgibt. Fahrzeugmodell, Geschwindigkeitsregler und Zeitintegration bleiben in beiden Modi identisch. Dadurch sind die Ergebnisse beider Betriebsarten unmittelbar vergleichbar, da Unterschiede ausschließlich aus dem Lastprofil und nicht aus abweichender Fahrzeugphysik resultieren. Zudem wirken sich Modellverbesserungen automatisch auf beide Betriebsarten aus.

Die zweite Entwurfsentscheidung betrifft die Trennung von Fachlogik und Benutzeroberfläche. Alle Modell- und Simulationsfunktionen werden in einer eigenständigen Funktionsbibliothek implementiert, während die Oberfläche auf Parametrierung und Darstellung beschränkt bleibt. Dadurch ist die Bibliothek automatisiert testbar und für Stapelverarbeitung einsetzbar, ohne dass eine Benutzeroberfläche erforderlich ist. Die dritte Entscheidung trennt die Datenbeschaffung von der Simulation: Die vergleichsweise langsame und kontingentierte Abfrage des Routing-Dienstes erfolgt pro Strecke nur einmal. Das Ergebnis wird als Route gespeichert und kann anschließend, bei Bedarf auch offline, beliebig oft mit unterschiedlichen Fahrzeugen, Fahrertypen und Verkehrsstufen simuliert werden. Diese Vorgehensweise erfüllt sowohl die Interaktivitätsanforderung NFA1 als auch die Reproduzierbarkeitsanforderung NFA4. Die vierte Entscheidung sieht eine dateibasierte Datenhaltung ohne Datenbankserver vor; die Begründung hierfür erfolgt in den Abschnitten 4.3 und 4.4.

Die streckenbasierte Fahrsimulation bildet das zentrale Element dieser Architektur. Ihr Entwurf wird in den folgenden drei Abschnitten vorgestellt. Im Vergleich zum Zyklusmodus ist der Entwicklungsaufwand deutlich höher: Eine Strecke definiert kein festes Geschwindigkeitsprofil. Kartendaten liefern lediglich Randbedingungen, während die tatsächlich gefahrene Geschwindigkeit erst durch das Zusammenspiel von Fahrer- und Fahrzeugverhalten entsteht. Eine identische Strecke wird von einem zurückhaltenden Fahrer anders befahren als von einem dynamischen Fahrer, und ein schweres Nutzfahrzeug erreicht an Steigungen nicht die gleiche Geschwindigkeit wie ein Personenkraftwagen. Strecke, Fahrer und Fahrzeug sind daher im Konzept über die Fahrbarkeit miteinander gekoppelt. Eine aus den Streckenbeschränkungen abgeleitete Sollgeschwindigkeit ist nur erreichbar, wenn Zugkraft und Kraftschluss des Fahrzeugs ausreichen. Der Entwurf gliedert sich in drei Bausteine, die im Folgenden erläutert werden.

4.2.1 Datenbeschaffung und Streckenrepräsentation

Das Konzept fordert den Routing-Dienst auf, sämtliche erforderlichen Streckeninformationen in einer einzigen Anfrage bereitzustellen: die Punktfolge der Route mit Angaben zu Länge, Breite und Höhe sowie abschnittsweise Straßenklassifikationen und Tempolimits. Der Vergleich in Abschnitt 3.2 zeigt GraphHopper als den einzigen Dienst auf, der diese Anforderungen erfüllt und gleichzeitig die Wiederverwendung gespeicherter Ergebnisse ermöglicht. Das Konzept muss zwei Lücken schließen: Da Tempolimits in OpenStreetMap nicht flächende-

ckend gepflegt sind, ist eine Rückfallkaskade vorgesehen, die fehlende Werte ersetzt und fahrzeugtypabhängig einschränkt; beispielsweise dürfen LKW außerhalb geschlossener Ortschaften nur mit einer Geschwindigkeit von 60 km/h statt 100 km/h wie PKW fahren[42]. Außerdem sind Ampelpositionen durch eine separate Abfrage bei OpenStreetMap zu ergänzen, da diese von GraphHopper nicht erfasst werden.

Für die Streckenrepräsentation wird ein Knoten-Segment-Modell verwendet: Knoten erfassen die punktbezogenen Eigenschaften, während die Segmente die abschnittsbezogenen Größen wie Länge, Steigungswinkel und Richtung darstellen. Kurvengeschwindigkeiten sollen ausschließlich anhand der geometrischen Richtungsvektoren aufeinanderfolgender Segmente sowie der zulässigen Querbeschleunigung des Fahrzeugs abgeleitet werden. Diese Entscheidung ist bewusst getroffen: Enge Kurven, Kreisverkehre und Abbiegevorgänge führen so zu physikalisch begründeten Verlangsamungen, ohne dass ein Manöverkatalog notwendig ist – jede Streckengeometrie wird auf einheitliche Weise behandelt.

4.2.2 Zweistufiges Fahrermodell

Das Fahrermodell wird in zwei Stufen aufgebaut, um – dem menschlichen Fahrverhalten nachempfunden – die Planungsphase von der Ausführungsphase zu trennen. Die erste Stufe, die vorausschauende Sollwertbildung, legt die Zielgeschwindigkeit als Minimum aller Beschränkungen innerhalb der Sichtweite des Fahrers fest (Abbildung 11): Tempolimit, Kurvengrenze und Bremsvorgaben für Haltepunkte. Die Sichtweite wird dabei nicht als fixe Größe betrachtet, sondern leitet sich aus dem Komfortbremsweg bei der aktuellen Geschwindigkeit ab. Dadurch blickt der Fahrer so weit voraus, wie er für ein komfortables Anhalten benötigt.

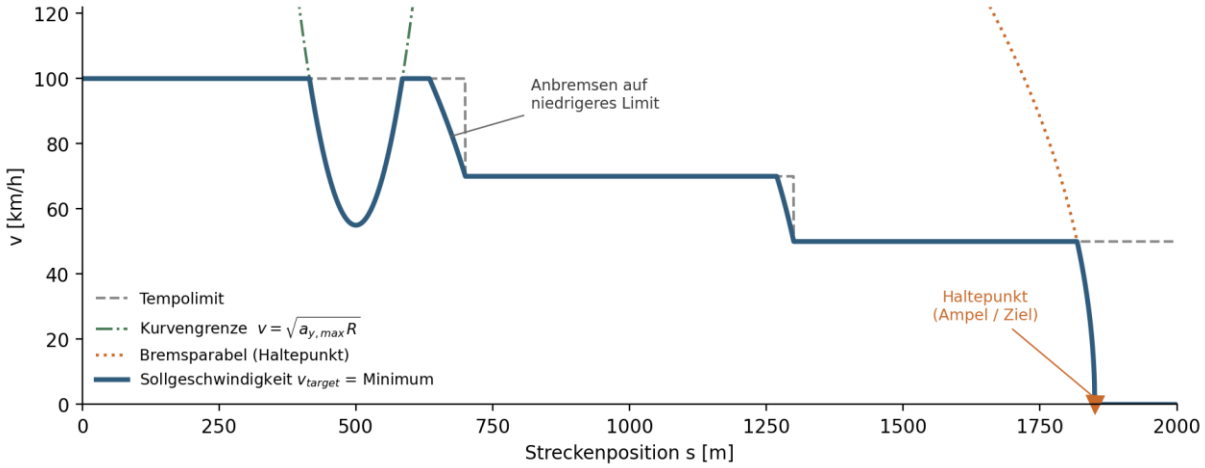


Abbildung 12: Minimum-Prinzip der vorausschauenden Sollwertbildung (schematisch, eigene Darstellung)

Die zweite Stufe, der Geschwindigkeitsregler, wandelt die Abweichung zwischen Ziel- und Ist-Geschwindigkeit in eine Kraftanforderung um. Es ist ein Regler mit physikalischer Vorsteuerung konzipiert, der die bekannten Fahrwiderstände und den Kraftbedarf der Sollbeschleunigung kompensiert, sodass lediglich Modell- und Störabweichungen ausgeglichen werden müssen. Die Reglerbeiwerte sind massennormiert festgelegt, um sicherzustellen, dass derselbe Fahrertyp bei einem Personenkraftwagen sowie bei einem schweren Nutzfahrzeug ein vergleichbares Fahrverhalten erzeugt. Drei parametrierbare Fahrertypen (langsam, normal, schnell) dienen der Skalierung von Reglerdynamik und Anbremsverhalten und decken somit die Bandbreite realer Fahrweisen ab (FA6).

4.2.3 Stochastische Elemente

Reale Fahrten unterliegen Einflüssen, die sich nicht aus der Streckengeometrie ergeben. Das Konzept modelliert diese durch zwei stochastische Elemente: den Zustand einer in Sichtweite kommenden Ampel sowie den umgebenden Verkehr, der die erreichbare Geschwindigkeit schrittweise auf einen Anteil des lokalen Tempolimits begrenzt. Für die Reproduzierbarkeit (NFA4) ist entscheidend, dass die niedrigste Verkehrsstufe den Zufallsprozess überhaupt nicht initialisiert: Die eigentliche Fahrsimulation bleibt komplett gleich, wobei stochastische Effekte nur dann wirksam werden, wenn sie ausdrücklich gewünscht sind. Im Zyklusmodus entfallen diese stochastischen Elemente, da ein vorgegebenes Geschwindigkeitsprofil vorliegt.

4.3 Konzept der Fahrzyklusdatenbank

Die Fahrzyklusdatenbank soll gewählte Referenzzyklen als Zeit-Geschwindigkeits-Profile bereitstellen und erfüllt im Gesamtsystem zwei Aufgaben: Sie dient sowohl als Lastvorgabe des Zyklusmodus als auch als Vergleichsmaßstab – vor allem über einheitlich definierte Zyklen lassen sich Simulationsergebnisse mit veröffentlichten Prüfwerten realer Fahrzeuge gut vergleichen.

Das Datenmodell eines Zyklus soll bewusst minimal gehalten werden: Ein Zyklus ist eine benannte Folge von Wertepaaren, die aus Zeit und Sollgeschwindigkeit besteht; bei normierten Zyklen ist dabei das Zeitraster implizit durch eine Frequenz festgelegt, sodass bereits eine einspaltige Geschwindigkeitsliste ausreicht. Dies stellt eine Stärke dar: Jede Quelle, die ein Geschwindigkeitsprofil exportieren kann, kann unmittelbar für einen Zyklus genutzt werden.

Für die Ablage wird entschieden, einfache Dateien im .xlsx- oder .csv-Format anstelle eines Datenbanksystems zu verwenden. Entscheidende Gründe hierfür sind das geringe Datenvolumen, die unmittelbare Lesbarkeit für den Menschen sowie die Anforderung NFA3, wonach die Datenbank mit einer Tabellenkalkulation erweiterbar sein soll, ohne dass Programmierkenntnisse erforderlich sind. Ein Datenbankserver würde bei diesen Datenmengen keinen funktionalen Mehrwert bieten, sondern den Installations- und Wartungsaufwand erhöhen und somit der Anforderung NFA6 widersprechen. Aus dieser Formatentscheidung resultiert jedoch eine Anforderung an die Einlesefunktionen: Da die Dateien aus Quellen wie Normdokumenten stammen, ist eine fehlertolerante Verarbeitung notwendig, was in Abschnitt 5.2 näher erläutert wird.

4.4 Konzept der Fahrzeugdatenbank

Die Fahrzeugdatenbank wurde konzipiert, um eine unkomplizierte Speicherung verschiedener Fahrzeugtypen mit unterschiedlichen Fahrzeugparametern zu ermöglichen. Daher ist sie im .xlsx-Format umgesetzt, wodurch der einfache Import oder die Modifikation mehrerer Fahrzeugdaten gewährleistet ist. Das Schema entspricht dem Fahrzeug- und Fahrermodell dieser Arbeit: Es werden exakt die vom Modell benötigten Parameter erfasst. Jede Spalte hat damit einen eindeutigen Abnehmer in der Simulation (Tabelle 6).

Modellbaustein	Parameter	Wirkung im Modell
Fahrwiderstände	Masse m, Stirnfläche A, c_w-Wert	Masse geht in Roll-, Steigungs- und Beschleunigungswiderstand ein; Stirnfläche und Luftwiderstandsbeiwert bestimmen den Luftwiderstand.
Rad und Kraftschluss	Rollradius r_w, Reibbeiwert μ	Der Rollradius verbindet Motormoment und Zugkraft bzw. Geschwindigkeit und Drehzahl $F_{Zug} = \frac{M_{Rad}}{rw}$; μ begrenzt die übertragbare Antriebs- und Bremskraft.
Trägheiten	Drehmassenzuschlagsfaktor ε	Der Drehmassenzuschlagsfaktor berücksichtigt Rotationsträgheiten des Antriebs pauschal in der Kräftebilanz.
Antriebsstrang	M_{max}, P_{max}, n_{max} (maximales Motormoment, max. Leistung, max. Drehzahl), Gangzahl, Übersetzungen	Eckdaten der Maschine und Übersetzungen spannen die Zugkraft-Hüllkurve auf; die Drehzahlgrenze legt die Höchstgeschwindigkeit je Gang fest.
Fahrermodell	Komfortverzögerung a_{dec}, Maximalverzögerung $a_{dec,max}$, max. Querbeschleunigung $a_{y,max}$	Komfortverzögerung (Vorausschau und Anbremsen), maximal stellbare Bremsverzögerung (Notbremsung) und zulässige Querbeschleunigung (Kurvengeschwindigkeit).

Tabelle 6: Parametergruppen der Fahrzeugdatenbank und ihre Modellwirkung

Abbildung 12 ordnet die Fahrzeugdatenbank in das logische Datenmodell des Gesamtsystems ein. Ein Simulationslauf verknüpft genau ein Fahrzeug mit genau einer Lastvorgabe, die entweder ein Fahrzyklus aus der Fahrzyklusdatenbank oder eine berechnete Route ist (XOR). Er trägt die Laufparameter Fahrertyp, Verkehrsstufe und Umgebungsbedingungen und protokolliert je Zeitschritt die Zustands- und Stellgrößen. Das Datenmodell macht damit die Schnittstellen zwischen den Teilsystemen der Architektur explizit: Die Fahrzeug- und Fahrzyklusdatenbank sind über den Simulationslauf entkoppelt und unabhängig voneinander erweiterbar.

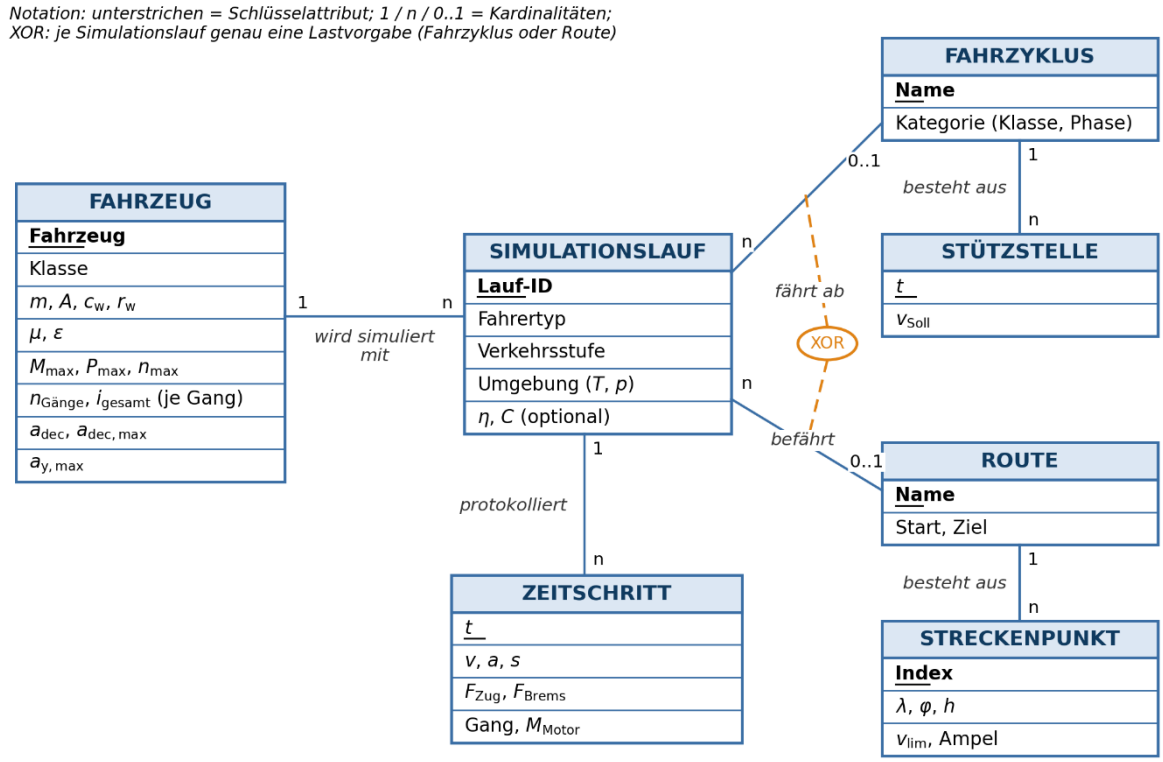


Abbildung 13: Logisches Datenmodell des Gesamtsystems als ER-Diagramm (eigene Darstellung)

Das Format soll einfach gehalten werden: In der obersten Zeile befinden sich die Parameterbezeichnungen, danach folgt jeweils eine Zeile mit einem Fahrzeug, gefolgt von den Parametern in den jeweiligen Spalten. Für die Datenablage wurde bewusst eine Entscheidung gegen reale Fahrzeugmodelle getroffen, da vollständige und belastbare Parametersätze von den Herstellern nicht veröffentlicht werden, wodurch eine modellgenaue Parametrierung nicht nachweisbar wäre. Stattdessen sind Beispielfahrzeuge mit angenommenen, klassentypischen Werten vorgesehen.

Zur Sicherstellung der Nachvollziehbarkeit (NFA4) soll die Arbeitsmappe, insbesondere bei der Erstanwendung, aus einem Skript generiert werden können, welches sowohl die Einheit und Bedeutung jeder Spalte dokumentiert als auch die Einheitlichkeit gewährleistet.

4.5 Methodik der Lastzyklusanalyse auf Basis von Fahrwiderstandsmodellen

Der erzeugte Lastzyklus bildet die Ausgangsbasis für die Auslegungsanalyse. Die Methodik folgt der in Abbildung 13 dargestellten Sequenz: Ausgehend vom Geschwindigkeits- und Steigungsverlauf werden mittels des Fahrwiderstandsmodells die Bedarfsgrößen am Rad ermittelt, auf den Antriebsstrang übertragen und dort als Betriebspunkte im Momenten-Drehzahl-Kennfeld analysiert. Diese Analyse liefert abschließend die Anforderungen an die elektrische Maschine.

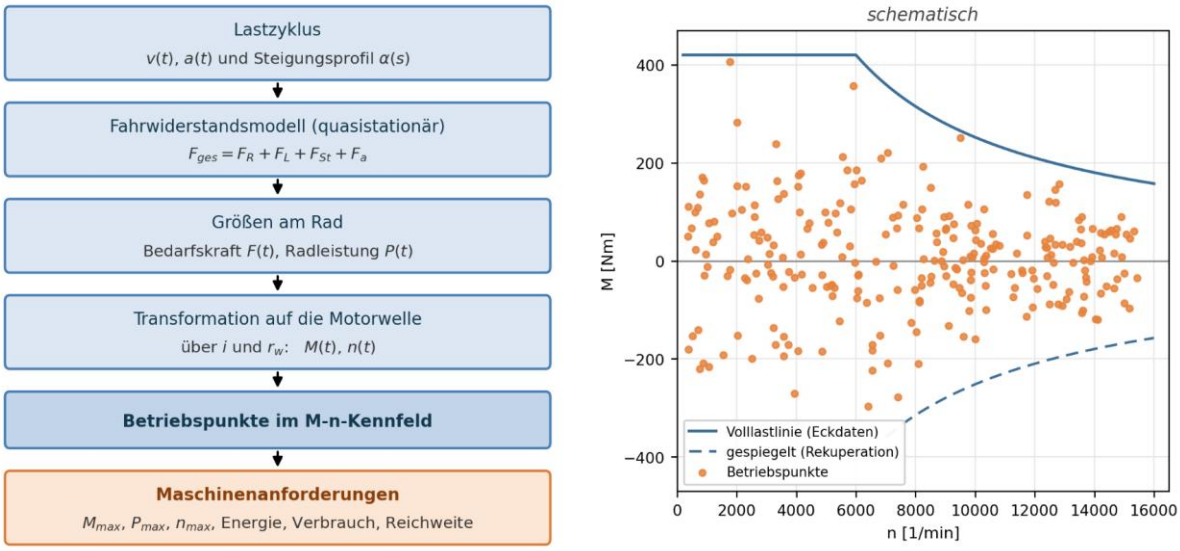


Abbildung 14: Methodik der Lastzyklusanalyse (schematisch, eigene Darstellung)

Grundlage soll ein quasistationäres Punktmassenmodell der Fahrzeuglängsdynamik sein. In jedem Zeitschritt setzt sich die Bedarfskraft aus den in Abschnitt 2.2 hergeleiteten Fahrwiderständen und dem Beschleunigungswiderstand zusammen. Die Radleistung ergibt sich als Produkt aus Bedarfskraft und Geschwindigkeit.

Diese Modelltiefe ist bewusst für die Vorauslegung gewählt: Energiebedarf, Momenten- und Leistungsanforderungen werden vor allem durch die Längsdynamik bestimmt. Effekte, die ein detailliertes Mehrkörper- oder Kennfeldmodell erfordern würden, sind in dieser Phase weder parametrierbar noch entscheidungsrelevant.

Für die Betriebspunktanalyse sollen die Radgrößen über die aktive Gesamtübersetzung i und den Rollradius r_w auf die Motorwelle übersetzt werden:

$$M(t) = \frac{F_{\mathrm{ges}} \cdot r_{\mathrm{w}}}{i} \tag{4.1}$$

$$n(t) = \frac{v(t) \cdot i}{2\pi \cdot r_{\mathrm{w}}} \cdot 60 \tag{4.2}$$

So ergeben sich das Drehmoment an der Motorwelle $M(t)$ in Nm und die Drehzahl der Motorwelle $n(t)$ in $\frac{1}{min}$.

Bei Mehrganggetrieben wird für jeden Zeitschritt die Übersetzung des aktiven Gangs berücksichtigt. Die Simulation basiert konzeptionell auf der Zugkraft-Hüllkurve, wodurch der jeweils zugkräftigste Gang simuliert wird, was einem optimal schaltenden Fahrer entspricht. Das Ergebnis ist eine Punktwolke im Momenten-Drehzahl-Diagramm (Abbildung 13, rechts), aus der die Eckdaten abgelesen werden: Das maximale Drehmoment wird aus dem größten Momentbedarf ermittelt, der typischerweise beim Anfahren an Steigungen auftritt; die maximale Leistung ergibt sich aus dem Bedarf bei hoher Geschwindigkeit; die Maximaldrehzahl wird anhand der Höchstgeschwindigkeit und Übersetzung bestimmt. Bremsphasen sollen mit negativem Vorzeichen rückwirkend berechnet werden; die an der Nulllinie gespiegelt dargestellte Grenzkennlinie dient als Näherung für das aufnehmbare Generatormoment und ermöglicht somit die Abschätzung des Rekuperationspotenzials. Für die Energiebilanz wird die Antriebsleistung über die Fahrstrecke integriert und daraus der streckenbezogene Energieverbrauch ermittelt.

5 Implementierung

Aufbauend auf der in Kapitel 4 entwickelten Konzeption erläutert dieses Kapitel die softwaretechnische Umsetzung der Fahrsimulationsumgebung. Zunächst werden die Entwicklungsumgebung sowie die verwendeten Bibliotheken vorgestellt (Abschnitt 5.1), anschließend erfolgt die Darstellung des Aufbaus der beiden Datenbanken für Referenzfahrzyklen (Abschnitt 5.2) und Fahrzeuge (Abschnitt 5.3). Den zentralen Bestandteil bildet die streckenbasierte Fahrsimulation (Abschnitt 5.4) sowie der Simulationskern mit dem Fahrwiderstandsmodell (Abschnitt 5.5). Abschnitt 5.6 beschreibt die Schnittstelle zwischen Streckendaten und Fahrzeugdatenbank, während Abschnitt 5.7 die grafische Benutzeroberfläche für die Parametrierung und Auswertung erläutert. Abbildung 14 zeigt den vollständigen Programmablauf des implementierten Systems, von der Eingabe über die Benutzeroberfläche bis zur Ergebnisausgabe. Dabei ist die in Abschnitt 4.2 festgelegte Grundstruktur erkennbar: Strecken- und Zyklusmodus unterscheiden sich ausschließlich hinsichtlich der Herkunft der Sollgeschwindigkeit und münden in denselben Simulationskern, der sie mit dem parametrisierten Fahrzeugmodell verknüpft und die Ergebnisse als einheitlichen Datensatz an die Auswertung übergibt. Die einzelnen Verarbeitungsschritte werden in den nachfolgenden Abschnitten in der Reihenfolge des Ablaufs dargestellt.

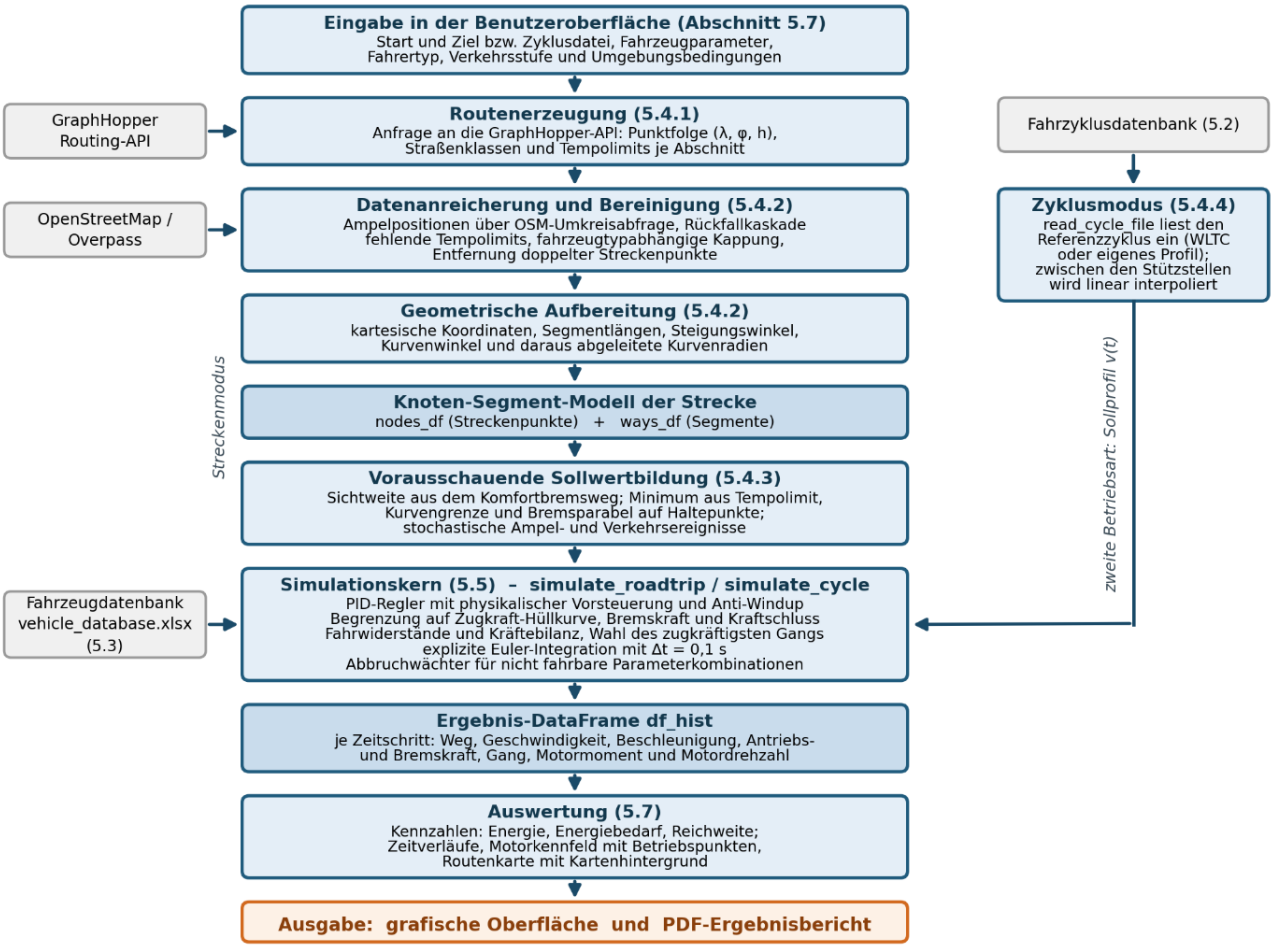


Abbildung 15: Programmablauf des implementierten Gesamtsystems (eigene Darstellung)

5.1 Entwicklungsumgebung

Die gesamte Implementierung erfolgt in Python 3, wobei die Begründung für diese Wahl in Abschnitt 3.3 dargelegt wird. Das System ist in vier Programmdateien unterteilt: Die Datei *DrivingCycleGenFunctions.py* enthält als zentrale Funktionsbibliothek sämtliche Modell- und Simulationsfunktionen. *DrivingCycleDesktop.py* implementiert die grafische Benutzeroberfläche, *DrivingCycleGen.py* bietet eine Kommandozeilenvariante für die Stapelverarbeitung, und *CreateVehicleDatabase.py* erzeugt die Fahrzeugdatenbank reproduzierbar (vgl. Abschnitt 5.3).

Die Nutzung der GraphHopper-API erfordert einen personengebundenen Zugangsschlüssel, der nach kostenfreier Registrierung im Kundenportal des Anbieters generiert wird. Da ein solcher Schlüssel nicht übertragbar ist, hinterlegt jeder Nutzer nach der Installation seinen eigenen Schlüssel in einer lokalen .env-Datei, die mit der Bibliothek python-dotenv gelesen wird. Der Quelltext selbst enthält keinen Schlüssel und ist somit ohne Weiteres weiterverwendbar, was zur Erfüllung der nichtfunktionalen Anforderung NFA6 beiträgt (Abschnitt 4.1).

Die verwendeten Bibliotheken und deren Einsatzzweck sind in Anlage 1 dargestellt. Für die Benutzeroberfläche wird das in der Python-Standardbibliothek enthaltene *Tkinter* verwendet, sodass keine zusätzliche GUI-Abhängigkeit entsteht. Die Erzeugung des PDF-Ergebnisberichts erfolgt bewusst über die bereits vorhandene *matplotlib*-Funktion *PdfPages*, um den Abhängigkeitsumfang der Anwendung klein zu halten.

5.2 Aufbau der Fahrzyklusdatenbank mit Referenzdaten

Die Fahrzyklusdatenbank stellt Referenzzyklen in Form von Zeit-Geschwindigkeits-Profilen bereit. Als primäre Referenz dienen die Phasenprofile des WLTC, welche in Abschnitt 3.4 erläutert werden. Dies gewährleistet eine Vergleichbarkeit des simulierten Fahrzeugs mit realen Fahrzeugen. Die Daten werden als CSV-Dateien abgelegt, da dieses Format sowohl von Tabellenkalkulationsprogrammen als auch von Messdatenexporten unmittelbar unterstützt wird und die Datenbank dadurch ohne Programmierkenntnisse erweitert werden kann. Innerhalb der Datenbank sind die Profile phasenweise sowie als Gesamtzyklus gespeichert, sodass sowohl vollständige Homologationszyklen als auch einzelne Phasen (beispielsweise der städtische Anteil) als Lastvorgabe genutzt werden können.

Die Einlesefunktion *read_cycle_csv* wurde absichtlich fehlertolerant gestaltet: Unterstützt werden einspaltige Dateien (Geschwindigkeitswerte mit implizitem 1-Hz-Zeitgitter) sowie zwei- und mehrspaltige Dateien mit expliziter Zeitspalte. Kopf-, Leer- und Kommentarzeilen werden erkannt und übersprungen; die Trennzeichen Semikolon, Tabulator und Komma werden automatisch erkannt, Dezimalkommas werden umgewandelt. Die Geschwindigkeitseinheit ist standardmäßig km/h und wird intern in m/s konvertiert. Das Ergebnis ist ein DataFrame mit den Spalten *time* und *velocity*, das dem Simulationskern im Zyklusmodus (Funktion *simulate_cycle*, vgl. Abschnitt 5.5) als Sollwertvorgabe dient. Als Gütemaß für die Zyklusfolge wird die Wurzel des mittleren quadratischen Folgefehlers (RMSE) zwischen Soll- und Ist-Geschwindigkeit berechnet; diese Größe wird in Kapitel 6 für die Validierung herangezogen.

Der Folgefehler wird über alle N Zeitschritte der Simulation gebildet:

$$RMSE = \sqrt{\frac{1}{N} \cdot \sum_{i=1}^{N} (v_{\text{Soll},i} - v_{\text{ist},i})^2} \tag{5.1}$$

Er wird ausschließlich im Zyklusmodus ausgewiesen, da dort nur ein extern vorgegebenes Sollprofil existiert. Im Streckenmodus bildet das Fahrermodell den Sollwert eigenständig, wodurch ein Folgefehler keine Aussagekraft bezüglich der Fahrzeugeignung hätte. Zusätzlich wird die größte auftretende Einzelabweichung protokolliert, da Homologationsvorschriften neben dem mittleren Fehler auch punktuelle Toleranzbänder vorsehen. Neben CSV-Dateien können Fahrzyklen direkt als Excel-Arbeitsmappe importiert werden. Die Funktion *read_cycle_file* verzweigt dazu anhand der Dateiendung an *read_cycle_xlsx* oder *read_cycle_csv*, sodass beide Formate denselben Aufbau aufweisen und identisch weiterverarbeitet werden können. Die Erkennung der Datenspalten erfolgt spaltenweise: Es werden nur diejenigen Spalten analysiert, die in allen Datenzeilen numerische Werte enthalten; beschreibende Textspalten, wie beispielsweise eine Phasenkennzeichnung des WLTC, werden dabei übersprungen. Führende Titel- und Kopfzeilen sowie abschließende Summenzeilen werden entfernt. Falls eine Arbeitsmappe mehrere Tabellenblätter oder mehr als zwei Zahlenspalten enthält, wird ein Auswahldialog angezeigt, der die Auswahl von Tabellenblatt und Geschwindigkeitsspalte ermöglicht. Die getroffene Auswahl wird in der Oberfläche angezeigt und im Ergebnisbericht dokumentiert.

5.3 Aufbau der Fahrzeugdatenbank mit Referenzdaten

Die Fahrzeugdatenbank ist als Excel-Arbeitsmappe (*vehicle_database.xlsx*, Tabellenblatt „Fahrzeuge“) implementiert; die in Abschnitt 5.2 genannten Gründe für ein tabellenbasiertes Ablageformat gelten hier gleichermaßen. Jede Zeile repräsentiert ein Fahrzeug, jede Spalte einen Parameter des Fahrzeugmodells aus Abschnitt 5.5. Tabelle 8 dokumentiert das Datenbankschema.

Die Datenbank ist bewusst, aufgrund der genannten Gründe in Abschnitt 3.4, nicht mit realen Fahrzeugmodellen befüllt. Daher sind generische Beispielfahrzeuge mit angenommenen Werten hinterlegt, die typische Größenordnungen ihrer Fahrzeugklasse abbilden. Für die Zielsetzung der vorliegenden Arbeit ist dies ausreichend, da die Datenbank nicht der Nachbildung eines spezifischen Fahrzeugs dient, sondern das Modellverhalten über ein breites Spektrum

an Masse, Aerodynamik und Antriebsstrangauslegung demonstrieren und die Parametrierung der Simulation vereinfachen soll. Für eine konkrete Auslegung werden die Fahrzeugparameter ohnehin direkt in der Benutzeroberfläche eingegeben.

Feld	Einheit	Bedeutung
Fahrzeug, Klasse	–	Bezeichnung und Fahrzeugklasse
m_kg	kg	Fahrzeugmasse inklusive Beladung
A_m2, cw	m², –	Stirnfläche und Luftwiderstandsbeiwert
rw_m	m	Reifenrollradius
mu	–	Kraftschlussbeiwert Reifen–Fahrbahn
ε	–	Drehmassenzuschlagsfaktor
i	–	Gesamtübersetzung (Rückfalloption bei einem Gang)
M_eng_max_Nm, P_eng_max_kW	Nm, kW	maximales Motormoment und maximale Motorleistung
a_dec_ms2	m/s²	Komfortverzögerung des Fahrermodells
a_dec_max_ms2	m/s²	maximal stellbare Bremsverzögerung (Notbremsung)
a_y_max_ms2	m/s²	maximale Querbeschleunigung (Kurvenfahrt)
n_gaenge	–	Anzahl der Gänge (1–20)
uebersetzungen_gesamt	–	Gesamtübersetzungen je Gang als Semikolonliste, absteigend ab Gang 1
n_eng_max_1min	1/min	maximale Motordrehzahl (leer: keine Begrenzung)

Tabelle 8: Schema der Fahrzeugdatenbank (vehicle_database.xlsx)

Die Datenbank ist mit fünf Beispielfahrzeugen befüllt, die das für die Modellprüfung relevante Spektrum aufspannen (Tabelle 9): einem leichten Stadtfahrzeug mit Eingang-Elektroantrieb, einer Mittelklasse-Limousine mit Achtganggetriebe, einem schweren SUV mit Elektroantrieb, einem sehr leichten, hochdrehenden Motorrad mit eng gestuftem Getriebe und einem schweren Nutzfahrzeug mit zwölf Gängen und niedriger Nenndrehzahl. Damit sind Massen von 200 kg bis 18 t, Stirnflächen von 0,6 m² bis 6,5 m² sowie Ein- und Mehrganggetriebe abgedeckt.

Beispielfahrzeug	m [kg]	A [m²] / cw	M / P	Gänge	n_max [1/min]
Kleinwagen (elektrisch)	1400	2,10 / 0,29	220 Nm / 100 kW	1	12.000
Mittelklasse-Limousine	1800	2,30 / 0,26	400 Nm / 190 kW	8	5.000
SUV (elektrisch)	2200	2,80 / 0,32	400 Nm / 220 kW	1	15.000
Motorrad	200	0,60 / 0,35	70 Nm / 90 kW	6	13.000
Schwer-LKW	18000	6,50 / 0,45	2400 Nm / 360 kW	12	2.300

Tabelle 9: Beispielfahrzeuge der Fahrzeugdatenbank (angenommene Werte)

Zwei Festlegungen sind für die Auswertung wesentlich: Die Übersetzungen sind ausdrücklich als Gesamtübersetzungen hinterlegt, also als Produkt aus Getriebegang und Achsantrieb, da das Fahrzeugmodell keinen separaten Achsantrieb kennt (vgl. Abschnitt 5.5). Ebenso unterscheiden die beiden Verzögerungsfelder die geplante Komfortbremsung von der maximal möglichen Bremskraft. Ihre Wirkung wird ebenfalls in Abschnitt 5.5 erläutert.

Die Werte wurden anhand des Modells auf Plausibilität geprüft. Die simulierten Beschleunigungen von 0 auf 100 km/h reichen von 3,9 s für das Motorrad über 6,6 s für die Limousine und 7,9 s für den SUV bis zu 7,7 s für den Kleinwagen und 25,0 s für das Nutzfahrzeug. Die maximal erreichbaren Fahrzeuggeschwindigkeiten liegen dabei zwischen 141 km/h und 283 km/h. Damit ordnen sich alle Beispielfahrzeuge etwa in den erwarteten Bereich ihrer jeweiligen Klasse ein.

Zur Sicherstellung der Nachvollziehbarkeit wird die Arbeitsmappe nicht manuell gepflegt, sondern vollständig durch das Skript *create_vehicle_database.py* aus dem Quelltext generiert. Änderungen am Datenbestand sind dadurch versionierbar, und die Datei kann jederzeit neu erstellt werden. Das Skript dokumentiert zugleich die Einheit und Bedeutung jeder Spalte sowie den Hinweis auf den Beispielcharakter der Werte als Kommentar in der Kopfzeile, sodass die Dokumentation unmittelbar an die Daten gebunden ist und beim Neugenerieren erhalten bleibt.

5.4 Strecken- und zyklusbasierte Fahrweise

Die Umgebung bietet zwei Betriebsarten, die dasselbe Fahrzeug- und Fahrermodell verwenden und sich ausschließlich in der Herkunft der Sollgeschwindigkeit unterscheiden. Im Streckenmodus wird eine frei wählbare reale Strecke in ein synthetisches Fahrszenario überführt. Dieser Prozess umfasst die Routenerzeugung (5.4.1), die Aufbereitung der Strecken- und Höhendaten zu einer simulationsfähigen Repräsentation (5.4.2) sowie die Generierung des Lastzyklus durch ein vorausschauendes Fahrermodell (5.4.3). Im Zyklusmodus (5.4.4) hingegen wird ein extern vorgegebenes Geschwindigkeitsprofil abgefahren, beispielsweise ein WLTC-Profil aus der Fahrzyklusdatenbank.

5.4.1 Routenerzeugung über die GraphHopper-API

Ausgangspunkt ist die Eingabe der Start- und Zieladressen. Die Anfrage an die GraphHopper-Routing-API liefert eine JSON-Antwort [43]. Unter einer API (Application Programming Interface) versteht man eine Programmierschnittstelle, die es zwei Programmen ermöglicht, miteinander Daten auszutauschen[44]. JSON (JavaScript Object Notation) ist ein kompaktes Datenformat, welches den Datenaustausch zwischen diversen Plattformen und Programmiersprachen vereinfacht[45]. Die JSON-Antwort umfasst neben der Punktfolge der Route (mit geographischer Länge, Breite und Höhe pro Punkt) zwei abschnittsweise detaillierte Informationen: die Straßenklasse (*road_class*) und das zulässige Tempolimit (*max_speed*). Beide Angaben liegen als Bereichsangaben über Punktindizes vor und werden bei der Datenaufbereitung auf die einzelnen Streckenpunkte ausgeweitet. Doppelte Koordinatenpunkte in der API-Antwort werden dabei konsequent aus allen Datenreihen entfernt. Start und Ziel können entweder als Adresse oder als Koordinatenpaar angegeben werden. Die Adressauflösung erfolgt durch den Routing-Dienst. Falls der Dienst nicht erreichbar ist oder die Anfrage fehlerhaft ist, wird der Fehler abgefangen und dem Nutzer als Statusmeldung angezeigt, ohne dass die Anwendung beendet wird.

Da Tempolimits in OpenStreetMap nicht flächendeckend gepflegt sind, wird es aus einer Zuordnungstabelle der Straßenklasse abgeleitet (motorway 130 km/h; trunk 100 km/h; primary 80 km/h; secondary 70 km/h; tertiary 50 km/h; residential 30 km/h). Falls auch dies fehlschlägt, wird der Wert des vorherigen Punktes übernommen; im letzten Rückfall gilt ein Wert von 50 km/h. Der von GraphHopper für unbegrenzte Autobahnabschnitte gelieferte Wert von 150 km/h wird auf die Richtgeschwindigkeit von 130 km/h angepasst. Für Fahrzeugklassen mit gesetzlich abweichenden Höchstgeschwindigkeiten (wie Transporter und Lastkraftwagen) werden die Limits anschließend fahrzeugtypspezifisch angepasst. So wird beispielsweise für Lastkraftwagen ab 7,5 Tonnen automatisch das Tempolimit von 80 km/h auf Autobahnen und 60 km/h auf Landstraßen festgelegt.

5.4.2 Einbindung von Höhenprofilen und Streckendaten

Die Höheninformation je Streckenpunkt stammt aus dem von GraphHopper eingebundenen SRTM-Datensatz (vgl. Abschnitt 3.2.2). Für die geometrische Weiterverarbeitung werden die geographischen Koordinaten in kartesische Koordinaten überführt; damit lassen sich Abstände, Richtungen und Winkel mit gewöhnlicher Vektorrechnung bestimmen. Das Ergebnis der Aufbereitung sind zwei Tabellen: *nodes_df* beschreibt die Streckenpunkte (Tempolimit,

Höhe, kumulierte Strecke, Ampelinformation), *ways_df* die verbindenden Segmente mit ihren Richtungsvektoren.

Aus je zwei aufeinanderfolgenden Richtungsvektoren $\overrightarrow{r_1}$ und $\overrightarrow{r_2}$ wird über das Skalarprodukt der Kurvenwinkel bestimmt:

$$\cos \varphi_j = \frac{(\overrightarrow{r_1} \cdot \overrightarrow{r_2})}{(|\overrightarrow{r_1}| \cdot |\overrightarrow{r_2}|)} \tag{5.2}$$

Aus Kurvenwinkel und Segmentlänge wird ein lokaler Kurvenradius R_j abgeschätzt:

$$R_j = \frac{\min(s_j, s_{j+1})}{|\varphi_j|} \tag{5.3}$$

Dabei ist φ_j der Richtungsänderungswinkel am Streckenpunkt j, also der Winkel zwischen dem einlaufenden und dem auslaufenden Segment, und s_j beziehungsweise s_{j+1} sind die Längen dieser beiden Segmente. Verwendet wird die kürzere der beiden.

Aus diesem Kurvenradius R_j leitet das Fahrermodell über die parametrisierbare zulässige Querbeschleunigung die Kurvengeschwindigkeit ab:

$$v_{\text{Kurve}} = \sqrt{(a_{\text{y,max}} \cdot R_j)} \tag{5.4}$$

Enge Kurven, Kreisverkehre und Abbiegevorgänge führen so ohne gesonderte Manöverlogik zu physikalisch begründeten Verlangsamungen. Die Steigung α eines Segments ergibt sich aus Höhendifferenz Δh und Segmentlänge Δs zu

$$\alpha = \tan^{-1}\left(\frac{\Delta h}{\Delta s}\right) \tag{5.5}$$

und geht als konstanter Winkel des Segments in die Fahrwiderstände ein. Ampelpositionen werden über eine Umkreisabfrage gegen OpenStreetMap ermittelt und den betroffenen Streckenpunkten zugeordnet. Abbildung 15 veranschaulicht die entstehende Streckenrepräsentation, dabei sind k_i die einzelnen Knoten.

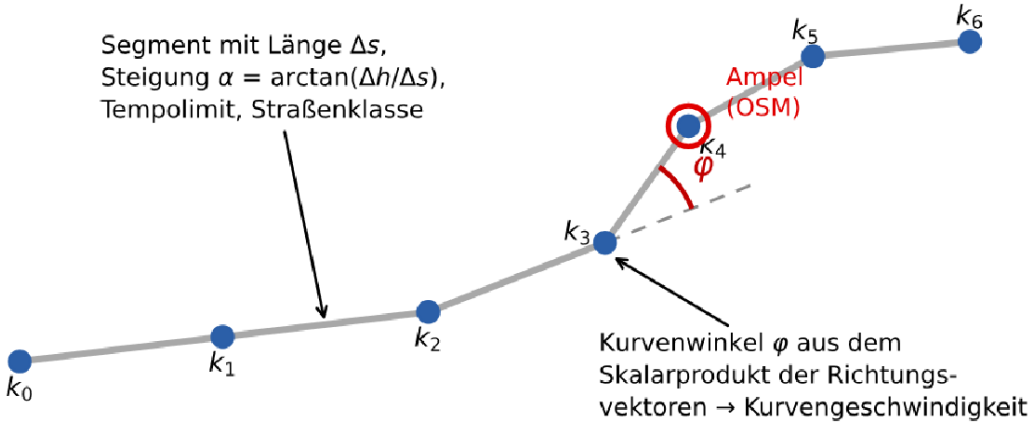


Abbildung 16: Streckenrepräsentation als Knoten-Segment-Modell

5.4.3 Sollwertbildung und Regelkreis

Die Sollwertbildung bestimmt in jedem Zeitschritt die Zielgeschwindigkeit v_{Soll} als Minimum mehrerer Beschränkungen innerhalb der Sichtweite des Fahrers. Die Sichtweite entspricht dem Komfortbremsweg aus der aktuellen Geschwindigkeit zuzüglich eines Puffers von 10 m. Berücksichtigt werden die Geschwindigkeitsbegrenzung des aktuellen und der einsehbaren folgenden Segmente, die aus dem Kurvenwinkel abgeleitete Kurvengeschwindigkeit sowie Haltepunkte: Vor einer als rot erkannten Ampel und vor dem Routenende wird die Zielgeschwindigkeit entlang einer Bremsparabel geführt:

$$v_{Soll}(d_{\mathrm{Rest}}) = \sqrt{2 \cdot a_{\mathrm{B}} \cdot d_{\mathrm{Rest}}} \tag{5.6}$$

Dadurch wird erreicht, dass das Fahrzeug mit konstanter Komfortverzögerung a_{B} exakt an der Haltelinie zum Stehen kommt (siehe Abbildung 16). Ob eine in Sichtweite kommende Ampel rot ist, wird stochastisch entschieden; im Stand wird die Rotphasen-Wartezeit pauschal aufgeschlagen.

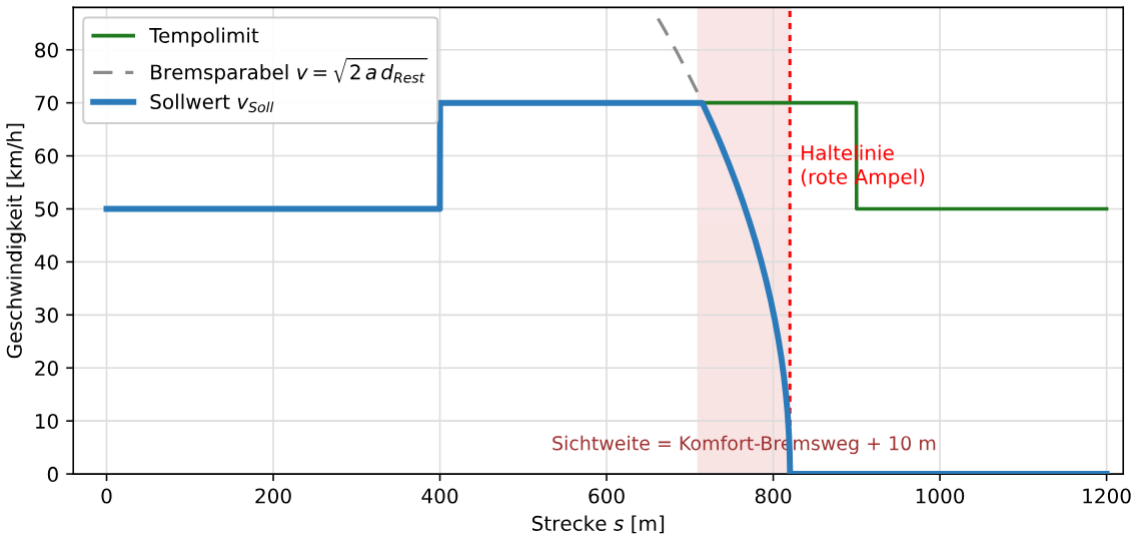


Abbildung 17: v-s-Diagramm einer Bremsung an einer roten Ampel (vorausschauende Sollwertbildung)

Implementiert ist ein PID-Regler mit physikalischer Vorsteuerung, dies wird in Abbildung 17 grafisch dargestellt. Gleichung (5.6) beschreibt den Regler im Zeitbereich, es wird die Kraftanforderung des Fahrermodells $F_{\mathrm{cmd}}(t)$ berechnet. Dabei ist $F_{FF}(t) = F_{\mathrm{R}} + F_{\mathrm{L}} + F_{\mathrm{St}} + m \cdot \varepsilon \cdot a_{\mathrm{Soll}}$ die Vorsteuerung im Regler, $e(t) = v_{Soll}(t) - v_{ist}(t)$ die Regelabweichung in m/s mit dem Proportionalbeiwert K_{p} in Ns/m. K_{i} ist der Integralbeiwert in N/m und K_{d} der Differentialbeiwert in Ns^2/m.

$$F_{\mathrm{cmd}}(t) = F_{\mathrm{FF}}(t) + K_{\mathrm{p}} \cdot e(t) + K_{\mathrm{i}} \cdot \int_0^t e(\tau)\,d\tau + K_{\mathrm{d}} \frac{de(t)}{dt} \qquad [6, \mathrm{S.\,347\text{-}352}](5.7)$$

Implementiert ist er zeitdiskret mit der Schrittweite Δt der Simulation (vgl. Abschnitt 5.5): Der Integralanteil wird als Rechtecksumme fortgeschrieben, $e_{\mathrm{int},k} = e_{\mathrm{int},k-1} + e_k \cdot \Delta t$, der Differentialanteil als Rückwärtsdifferenz gebildet. Letzterer wird bewusst nicht auf den Regelfehler, sondern auf die Messgröße angewendet, also $-\frac{v_{\mathrm{ist},k} - v_{\mathrm{ist},k-1}}{\Delta t}$. Damit führt ein Sprung der Sollgeschwindigkeit, etwa beim Wechsel des Tempolimits, nicht zu einem Stellgrößensprung, sondern nur zu einer Änderung des Proportional- und Integralanteils.

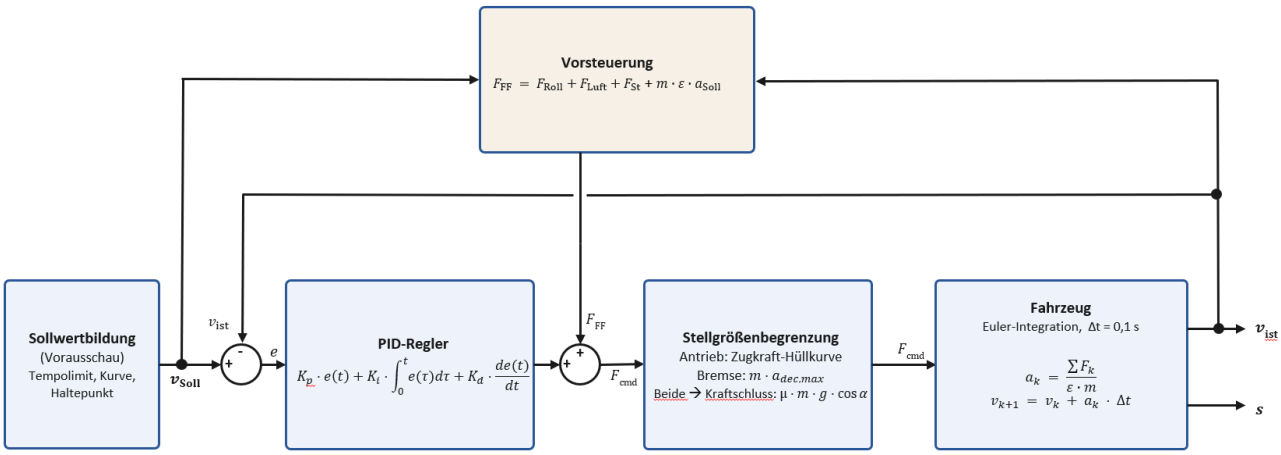


Abbildung 18: Struktur des Geschwindigkeitsregelkreises

Die Vorsteuerung F_{FF} kompensiert die bekannten Fahrwiderstände sowie den Kraftbedarf der Sollbeschleunigung ($m \cdot \varepsilon \cdot a_{Soll}$). Der Regler muss damit lediglich Modell- und Störabweichungen ausregeln. Ein bedingtes Integrationsverfahren (Anti-Windup) hält den Integralanteil an, wenn die Stellgröße außerhalb der Begrenzung liegt und der Regelfehler weiter in die Sättigung führt. Wechselt der Regelfehler das Vorzeichen, wird die Integration fortgesetzt, damit sich der Integrator wieder abbauen kann. Dies geschieht in genau zwei Fällen: Antriebsseitig fordert der Regler mehr Kraft als verfügbar ($F_{\mathrm{cmd}} > F_{\mathrm{eng,max}}$) und das Fahrzeug ist zu langsam ($e > 0$), oder wenn bremsseitig mehr Bremskraft als möglich gefordert wird und das Fahrzeug zu schnell ist ($e < 0$). Drei parametrierte Fahrertypen bilden unterschiedliche Fahrweisen ab (siehe Tabelle 10), die Werte für k_{p} wurden dabei erprobt und angenommen.

Fahrertyp	**k_{p}[1/s]**	**Charakteristik**
langsam	0,12	träges Folgeverhalten, frühe und sanfte Komfortbremsung
normal	0,25	ausgewogenes Folgeverhalten (Standard)
schnell	0,50	straffes Folgeverhalten, späteres und kräftigeres Anbremsen

Tabelle 10: Parametrierung der Fahrertypen

Der Fahrertyp wirkt an zwei Stellen: Er skaliert die Reglerbeiwerte und bestimmt über den Komfortbremsfaktor, wie früh und wie kräftig auf Haltepunkte und Geschwindigkeitsbegrenzungswechsel angebremst wird. Der Typ „langsam“ leitet Bremsungen entsprechend früher ein als der Typ „schnell“. Damit lassen sich aus derselben Strecke systematisch unterschiedlich aggressive Lastzyklen erzeugen, was für die spätere Sensitivitätsbetrachtung der Maschinenauslegung genutzt wird.

Zur Abbildung realer Verkehrsbedingungen wurde ein stochastisches Verkehrsmodell mit sechs wählbaren Stufen implementiert. Der umgebende Verkehr begrenzt die erreichbare Geschwindigkeit auf einen Anteil des lokalen Tempolimits. Modelliert ist dies als Zufallsprozess mit zwei Zuständen (freier Fluss und Verlangsamungsereignis), dessen Parameter stufenabhängig hinterlegt sind (Tabelle 11). In Stufe 5 entstehen dadurch ausgeprägte “Stop-and-go“-Phasen (Abbildung 18). Die Stufe 0 instanziiert das Modell bewusst nicht, sodass der Programmpfad ohne Verkehr unverändert und reproduzierbar bleibt.

Stufe	Grundfaktor	Ereignisrate	Stauanteil je Ereignis
0 (kein Verkehr)	1,00	–	–
1	0,97	1/300 s	0 %
2	0,93	1/210 s	3 %
3	0,88	1/120 s	5 %
4	0,78	1/80 s	15 %
5 (Stau/stockend)	0,65	1/50 s	35 %

Tabelle 11: Parametrierung der Verkehrsstufen

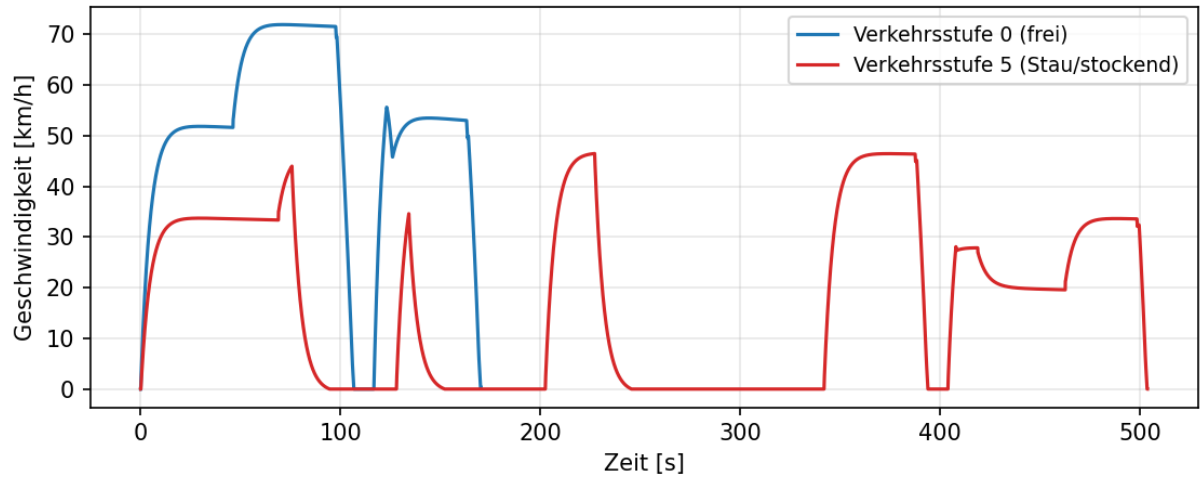


Abbildung 19: Auswirkung des Verkehrsmodells auf das Geschwindigkeitsprofil anhand einer simulierten Beispielstrecke (eigene Darstellung)

5.4.4 Zyklusbasierte Fahrsimulation

Neben der streckenbasierten Erzeugung kann ein extern vorgegebenes Geschwindigkeitsprofil abgefahren werden. Als Vorgabe dienen die Referenzzyklen der Fahrzyklusdatenbank, insbesondere der WLTC als Fahrprofil der WLTP-Prüfprozedur.

Der Zyklusmodus ist in der Funktion *simulate_cycle* implementiert. Diese nutzt dasselbe Fahrzeugmodell, denselben Antriebsstrang sowie denselben Geschwindigkeitsregler wie der Streckenmodus; einzig die Quelle der Sollgeschwindigkeit wird ausgetauscht. Anstelle der Sollwertbildung, die aus Geschwindigkeitsbegrenzungen, Kurven und Ampeln resultiert, tritt der zeitdiskrete Verlauf $v(t)$ aus der Zyklusdatei hervor. Damit sind beide Betriebsarten unmittelbar vergleichbar, da die Differenzen in den Ergebnissen ausschließlich auf das Lastprofil und nicht auf abweichende Fahrzeugphysik zurückzuführen sind.

Für die Implementation eines Normzyklus ergeben sich gemäß der definierten Vorgaben folgende Vereinfachungen: Die Fahrbahn wird als eben angenommen, so dass der Steigungswiderstand entfällt. Ampeln, Kurven sowie das Verkehrsmodell sind nicht wirksam, da die Haltephasen bereits im Profil enthalten sind. Anders als im Streckenmodus enthält der DataFrame keine Knoten, Straßenobjekte und Koordinaten.

Als Gütemaß dafür, wie genau das Fahrzeug dem Sollprofil folgt dient der Folgefehler nach (5.1), ergänzt um die größte auftretende Einzelabweichung. Während der Folgefehler die mittlere Abweichung beschreibt, erfasst

$$\Delta v_{\max} = \max_{i=1,\dots,N} |v_{\mathrm{Soll},i} - v_{\mathrm{Ist},i}| \tag{5.8}$$

die größte auftretende Abweichung. Dies ist relevant, weil Homologationsvorschriften ein zu keinem Zeitpunkt zu verlassendes Toleranzband vorgeben. Ein kleiner Wert des Folgefehlers belegt, dass das parametrierte Fahrzeug dem Sollprofil folgen kann. Wächst der Fehler, weist dies auf ein untermotorisiertes Fahrzeug mit zu geringer Zugkraft hin. Abbildung 19 zeigt dies exemplarisch für ein Kleinwagen-Beispielfahrzeug auf dem WLTC der Klasse 1: Die Ist-Geschwindigkeit folgt dem Sollprofil über den gesamten Zyklus, der Folgefehler bleibt unter 1,5 km/h und im quadratischen Mittel bei 0,4 km/h.

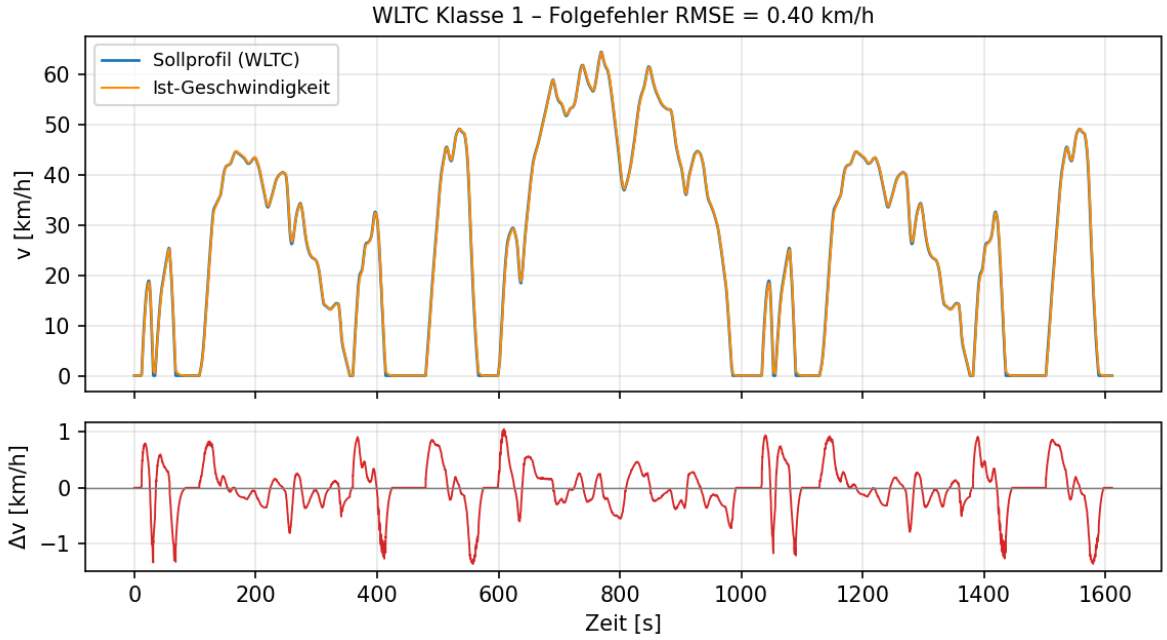


Abbildung 20: Zyklusbasierte Fahrsimulation (WLTC Class 1)

5.5 Simulationskern: Fahrwiderstandsmodell und Längsdynamiksimulation

Der Simulationskern (Funktion *simulate_roadtrip*) bildet die Fahrzeuglängsdynamik als Punktmassenmodell ab. Grundlage ist die in Abschnitt 2.2 hergeleitete Fahrwiderstandsgleichung (Formel 2.1). Je Zeitschritt werden Rollwiderstand, Luftwiderstand und Steigungswiderstand des aktuellen Segments berechnet. Der Rollwiderstandsbeiwert ist mit f_R = 0,01 als Standardwert für Fahrzeugreifen im Straßenobjekt hinterlegt. Dieser Wert wird zur Vereinfachung angenommen, übliche Werte variieren je nach Luftdruck im Bereich von 0,008-0,017 [47] (siehe Abbildung 20).

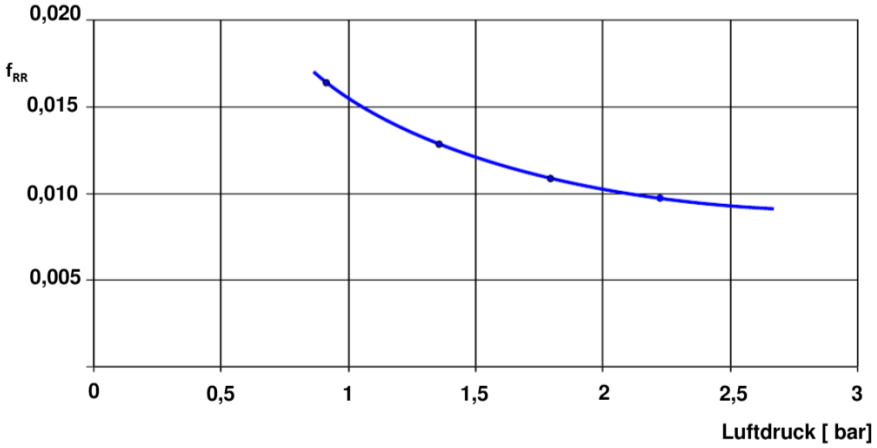


Abbildung 21: Rollwiderstandsbeiwert (grafisch, Quelle:[47] S.2-7)

Die Luftdichte φ wird aus den in der Oberfläche einstellbaren Umgebungsgrößen Temperatur T, Luftdruck p und der spezifischen Gaskonstante für Luft R_L über die ideale Gasgleichung bestimmt.

$$\rho = \frac{p}{R_\mathrm{L} \cdot T} \qquad [48](5.9)$$

Dabei ist T die absolute Temperatur in Kelvin; die Oberfläche erwartet die Eingabe in Grad Celsius und rechnet intern um. Bei 20 °C und 1,013 bar ergibt sich $\rho \approx 1{,}204$ kg/m^3. Die vom Antrieb wie von der Bremse übertragbare Kraft ist durch den Kraftschluss begrenzt,

$$F_{\mathrm{T,max}} = \mu \cdot m \cdot g \cdot \cos\alpha \qquad (5.10)$$

wobei vereinfachend die gesamte Fahrzeugmasse als kraftschlussrelevant angenommen wird und pro Simulationsgang nur ein fester Reibwert einstellbar ist. Werte des Reibwerts μ, welcher den Kraftschluss zwischen Fahrbahn und Reifen darstellt, variieren unter mehreren Faktoren. Darunter insbesondere die Streckenbeschaffenheit, die Witterungsbedingungen und die Reifenmischung. In Abbildung 21 ist eine Bandbreite dargestellt.

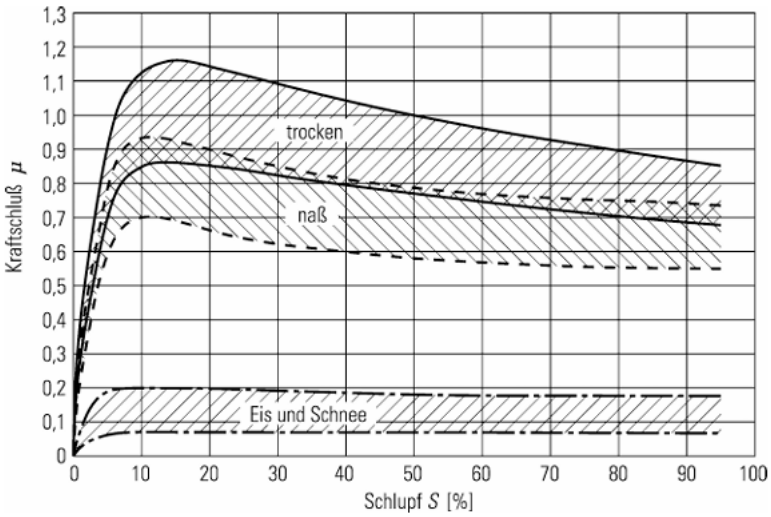


Abbildung 22: Bandbreite der auf ebenen, homogenen Fahrbahnen im öffentlichen Straßenverkehr gemessenen Reibwert-Schlupf-Kurven (Quelle: [49])

Die Beschleunigung folgt aus der Kräftebilanz:

$$a = \frac{F_{\mathrm{Zug}} - F_\mathrm{a} - F_\mathrm{R} - F_\mathrm{L} - F_{\mathrm{St}}}{\varepsilon \cdot m} \qquad (5.11)$$

Bremsseitig unterscheidet das Modell zwei Grenzen. Die Komfortverzögerung a_{dec} beschreibt das geplante Anbremsen und bestimmt damit Sichtweite und Bremsparabel des Fahrermodells nach (5.4). Die Maximalverzögerung $a_{\mathrm{dec,max}}$ legt die tatsächlich stellbare Bremskraft

$F_{\text{Brems}} = F_{\text{a}} = m \cdot a_{\text{dec,max}}$ fest. Diese Reserve wird nur abgerufen, wenn die Komfortkraft nicht ausreicht – etwa beim Anhalten im Gefälle, wo die Hangabtriebskraft der Bremskraft entgegenwirkt. Beide Grenzen unterliegen zusätzlich dem Kraftschluss nach (5.10), sodass eine Notbremsung auf glatter Fahrbahn korrekt begrenzt bleibt.

Der Antriebsstrang ist über die Motoreckdaten und die Gesamtübersetzung modelliert. Für den Gang k mit der Gesamtübersetzung i_k liefert der Motor die geschwindigkeitsabhängige Zugkraft

$$F_{\text{Zug},k}(v) = \min\left(\frac{i_k \cdot M_{\text{max}}}{r_{\text{w}}}, \frac{P_{\text{max}}}{v}\right) \tag{5.12}$$

solange die Motordrehzahl $n = \frac{v \cdot i_k}{r_{\text{w}}}$ unterhalb der Maximaldrehzahl bleibt; oberhalb liefert der Gang keine Kraft („überdrehter" Gang). Die Beziehung gilt für $v > 0$. im Stillstand ist die Leistungshyperbel nicht definiert, sodass dort allein das Moment die Zugkraft begrenzt. Daraus ergibt sich die Höchstgeschwindigkeit je Gang zu

$$v_{\text{max},k} = \frac{\omega_{\text{max}} \cdot r_{\text{w}}}{i_k} \tag{5.13}$$

Die Simulation fährt in jedem Zeitschritt auf der Zugkraft-Hüllkurve, das heißt mit dem jeweils zugkräftigsten fahrbaren Gang. Dies entspricht einem ideal schaltenden Fahrer. Für die Ergebnisdarstellung wird zusätzlich der bei Teillast plausible Gang bestimmt, also der höchste Gang, der die momentan genutzte Kraft noch aufbringt. Abbildung 21 zeigt die Hüllkurve eines Fahrzeugs mit Sechsganggetriebe sowie 250 Nm Drehmoment und 96 kW Leistung zusammen mit den Fahrwiderstandskurven für verschiedene Steigungen. Die Schnittpunkte markieren die jeweils erreichbare stationäre Höchstgeschwindigkeit.

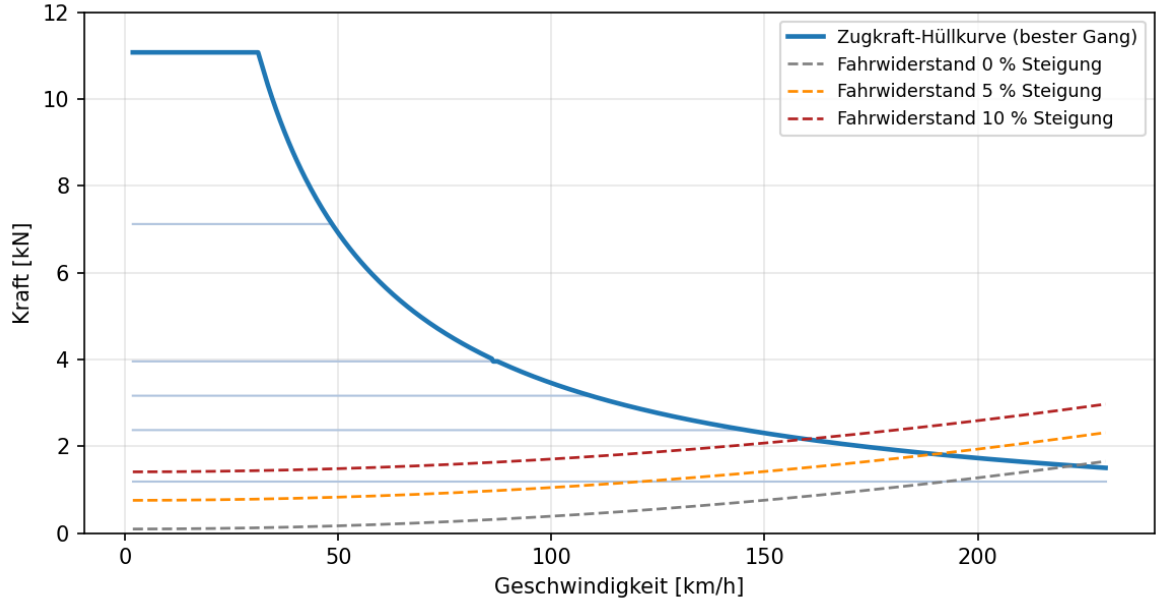


Abbildung 23: Zugkraftdiagramm des implementierten Antriebsstrangmodells (Beispielfahrzeug: 6 Gang, 250Nm, 96kW)

Für die Zielanwendung, die Vorauslegung elektrischer Fahrzeugantriebe, wurde die Motorbeschreibung bewusst auf die vier Eckdaten $M_{\max}, P_{\max}, n_{\max}$ und i beschränkt. Zusätzlich zeigt das ideale Eckdaten-Kennfeld, das mit konstantem Moment im Grunddrehzahlbereich und konstanter Leistung im Feldschwächbereich arbeitet, strukturell das Verhalten realer elektrischer Maschinen (vgl. Abschnitt 2.3.2).

Abbildung 22 zeigt das aus den Eckdaten aufgespannte Kennfeld mit den in einer Simulation auftretenden Betriebspunkten. Diese Darstellung wird in Kapitel 6 zur Ableitung der Maschinenanforderungen herangezogen. Neben den Antriebspunkten oberhalb der Nulllinie sind auch die Bremspunkte eingetragen, indem die Bremskraft mit der aktiven Übersetzung auf die Motorwelle zurückgerechnet und mit negativem Vorzeichen aufgetragen wird. Die an der Nulllinie gespiegelte Grenzkennlinie dient dabei als Orientierung für den generatorischen Betrieb: Bremspunkte innerhalb dieser Grenze könnte die Maschine theoretisch aufnehmen und damit rekuperieren, darüber hinausgehende Anteile müssten von der Reibbremse abgeführt werden. Die Spiegelung ist eine Näherung, da das mögliche Generatormoment realer Maschinen abweichen kann. Für die Vorauslegung genügt sie zur Abschätzung des Rekuperationspotenzials.

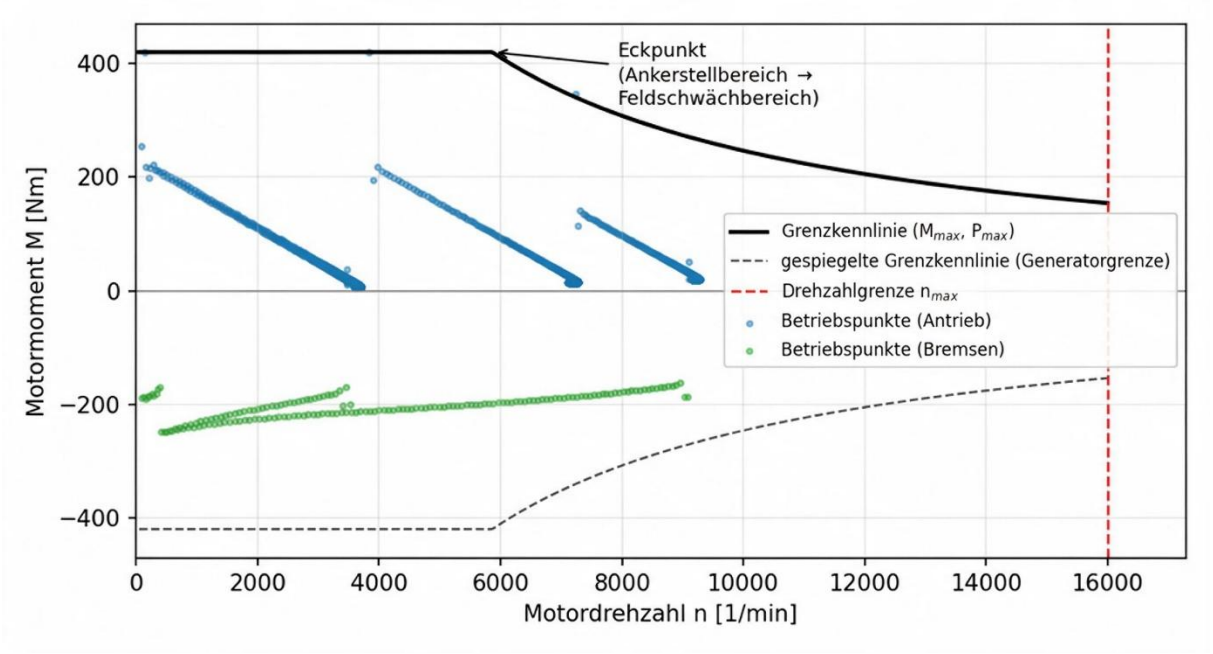


Abbildung 24: Eckdaten-Kennfeld einer elektrischen Maschine (420Nm, 258kW, 16000 1/min) mit den Betriebspunkten einer Simulation auf einer Stadt-Landstraßen-Autobahn-Strecke

Die zeitliche Integration erfolgt mit dem expliziten Euler-Verfahren[50]. Diese Wahl stellt einen Kompromiss aus Rechenzeit und Genauigkeit dar. An harten Begrenzungen, zum Beispiel der Drehzahlgrenze, ist ein kurzzeitiges Überschwingen um maximal einen Zeitschritt möglich, was bei den auftretenden Beschleunigungen einer Geschwindigkeitsabweichung in der Größenordnung von 1-5 km/h entspricht (vgl. Kapitel 6). Der Ablauf eines Zeitschritts gliedert sich wie folgt:

1. Sollwertbildung: Zielgeschwindigkeit aus der Vorausschau, anschließend Begrenzung durch das Verkehrsmodell.
2. Regelung: Kraftanforderung aus Vorsteuerung und PID-Regler nach Gleichung (5.6).
3. Stellgrößenbegrenzung: Aufteilung in Antriebs- oder Bremskraft und Begrenzung auf Zugkraft-Hüllkurve, Bremskraft und Traktionsgrenze. Daraus werden die Pedalstellungen gewählt.
4. Längsdynamik: Fahrwiderstände nach (2.1), Beschleunigung nach (5.11), Integration von Geschwindigkeit und Position.
5. Streckenfortschritt: Segmentwechsel bei Erreichen des nächsten Knotens (neues Limit, neue Steigung).

6. Ereignisbehandlung: Ampellogik (stochastische Rotphase, Anhalten an der Haltelinie, pauschale Wartezeit) und Erkennung des Routenendes.
7. Protokollierung aller Zustands- und Stellgrößen des Zeitschritts.

Die Ampellogik behandelt mehrere Randfälle: Eine Ampel wird erst berücksichtigt, wenn sie in die Sichtweite des Fahrers eintritt, erst dann wird stochastisch über die Rotphase entschieden. Bei roter Ampel wird der Haltepunkt zwei Meter vor dem Ampelknoten gesetzt und die Zielgeschwindigkeit über die Bremsparabel (5.5) geführt. Der Stopp gilt erst als abgefertigt, wenn das Fahrzeug tatsächlich an der Haltelinie steht. Für diese Prüfung wird der verkehrsfreie Sollwert herangezogen: Steht das Fahrzeug verkehrsbedingt bereits weit vor der Haltelinie (Stau), darf die rote Ampel nicht fälschlich als abgefertigt gelten. Das Routenende wird analog als Haltepunkt behandelt, sodass jede Simulation mit einem definierten Stillstand am Ziel endet. Die Rechenzeit des Kerns liegt für Überlandstrecken von mehreren zehn Kilometern im Sekundenbereich, sodass iterative Parameterstudien interaktiv möglich sind.

Das Ergebnis ist ein DataFrame *df_hist* mit einer Zeile je Zeitschritt (Tabelle 12). Er bildet die einheitliche Schnittstelle zu allen Auswertungen. Abbildung 23 zeigt exemplarisch das Ergebnis einer Simulation mit unterschiedlichen Tempolimits und Ampelstopp: Die Ist-Geschwindigkeit folgt dem vorausschauend gebildeten Sollwert, das Anbremsen auf die Haltelinie verläuft entlang der Bremsparabel.

Wert	Einheit	Bedeutung
time, distance	s, m	Zeit und zurückgelegte Strecke
velocity, acceleration	m/s, m/s²	Ist-Geschwindigkeit und -Beschleunigung
target_velocity	m/s	Sollwert des Fahrermodells
engine_force, brake_force	N	gestellte Antriebs- und Bremskraft
throttle, brake_pedal	–	normierte Pedalstellungen (0…1)
gear, gear_ratio	–	aktiver Gang und Gesamtübersetzung
engine_torque	Nm	ganggenau rückgerechnetes Motormoment
traction_force_max	N	momentane Kraftschlussgrenze
traffic_factor	–	Begrenzungsfaktor des Verkehrsmodells

Tabelle 12: Wichtige Werte des Ergebnis-DataFrames df_hist

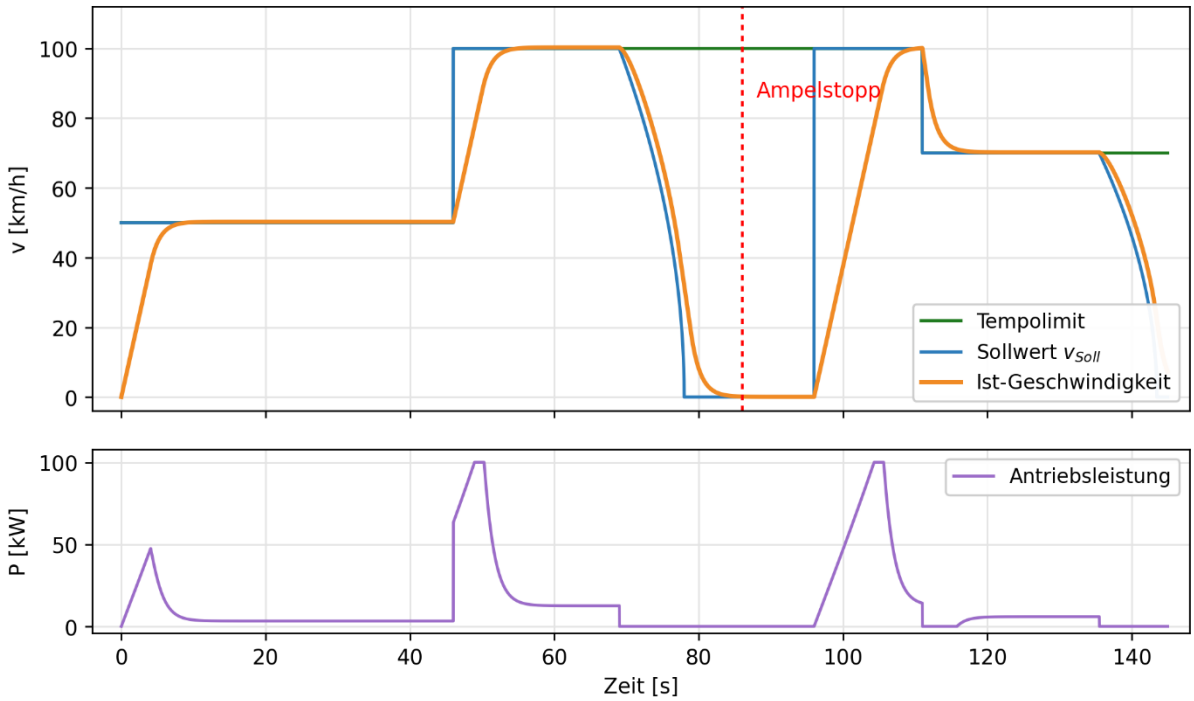


Abbildung 25: Exemplarisches Simulationsergebnis mit Ampelstopp

Zwei Schutzmechanismen sichern die Robustheit des Kerns. Ein Stillstandswächter bricht die Simulation ab, wenn das Fahrzeug trotz Fahrwunsch länger als 30 s keine Bewegung erreicht – dieser Fall tritt physikalisch korrekt auf, wenn die Anfahr-Zugkraft den Hangabtrieb nicht übersteigt. Die zugehörige Grenzbedingung folgt aus (2.2), (2.5) und (5.12) für $v \to 0$. Da der Rollwiderstand nach (2.2) selbst vom gesuchten Winkel abhängt, ergibt sich eine implizite Bestimmungsgleichung, die numerisch iterativ gelöst wird:

$$\sin \alpha_{\max} + f_{\mathrm{R}} \cdot \cos \alpha_{\max} = \frac{i \cdot M_{\max}}{r_{\mathrm{w}} \cdot m \cdot g} \tag{5.14}$$

Zusätzlich erkennt der Kern den Fall eines aussichtslosen Kriechens: Wird die Zugkraft voll ausgenutzt und übersteigt sie im Stand, also ohne den mit sinkender Geschwindigkeit verschwindenden Luftwiderstand, den Roll- und Steigungswiderstand nicht. Dann läuft die Geschwindigkeit zwangsläufig gegen null, allerdings nur asymptotisch. Ohne diese Überprüfung erreicht das Fahrzeug die Stillstandsschwelle praktisch nie, was dazu führt, dass die Simulation unbegrenzt fortgesetzt wird. Fahrzeuge, die eine Steigung noch bei konstanter, wenn auch niedrigerer Geschwindigkeit bewältigen können, sind hiervon nicht betroffen, da bei ihnen die Standzugkraft ausreicht.

Beim Abbruch werden Position, lokale Steigung sowie benötigte, verfügbare und kraftschlussbegrenzte Kraft als Diagnosedatensatz an das Ergebnis angehängt. Die Benutzeroberfläche kennzeichnet die Stelle in der Routenkarte. Damit wird eine nicht fahrbare Parameterkombination, etwa eine zu lange Gesamtübersetzung an einer Steigung mit Anfahrsituation, unmittelbar diagnostizierbar statt als scheinbarer Programmfehler zu erscheinen.

5.6 Benutzeroberfläche zur Parametrierung und Auswertung

Die Benutzeroberfläche (Abbildung 24) folgt einer Zweiteilung, links liegen die Eingaben:

- Streckenwahl mit Speicher- und Ladefunktion für Routen bzw. Auswahl einer Zyklusdatei
- Fahrzeugparametrierung (manuell oder aus der Datenbank)
- Getriebekonfiguration mit dynamisch erzeugten Eingabefeldern je Gang
- Umgebungsgrößen
- Fahrertyp
- Verkehrsstufe
- Kartenhintergrund
- Aktionselemente (Simulation, Statusanzeige, Berichterstellung)
- Verkleinerte Routenkarte

Rechts stellt ein 3×2-Kachelraster die Ergebnisdiagramme dar. Eine zweite, über Pfeilschaltflächen erreichbare Rasterseite nimmt das Motorkennfeld und die Kennzahlentabelle auf (Abbildung 25). Die Anordnung der Diagramme ist über einen Tauschmodus frei konfigurierbar; jede Kachel lässt sich per Mausklick vergrößern.

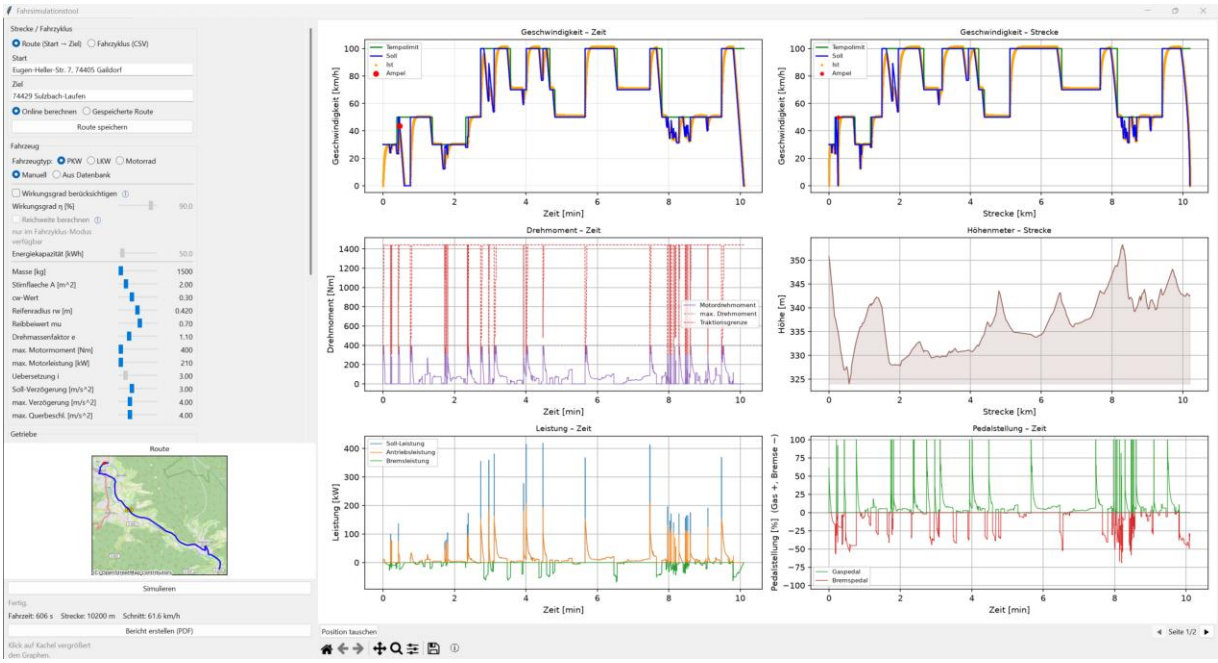


Abbildung 26: Grafische Benutzeroberfläche Fahrsimulationstool

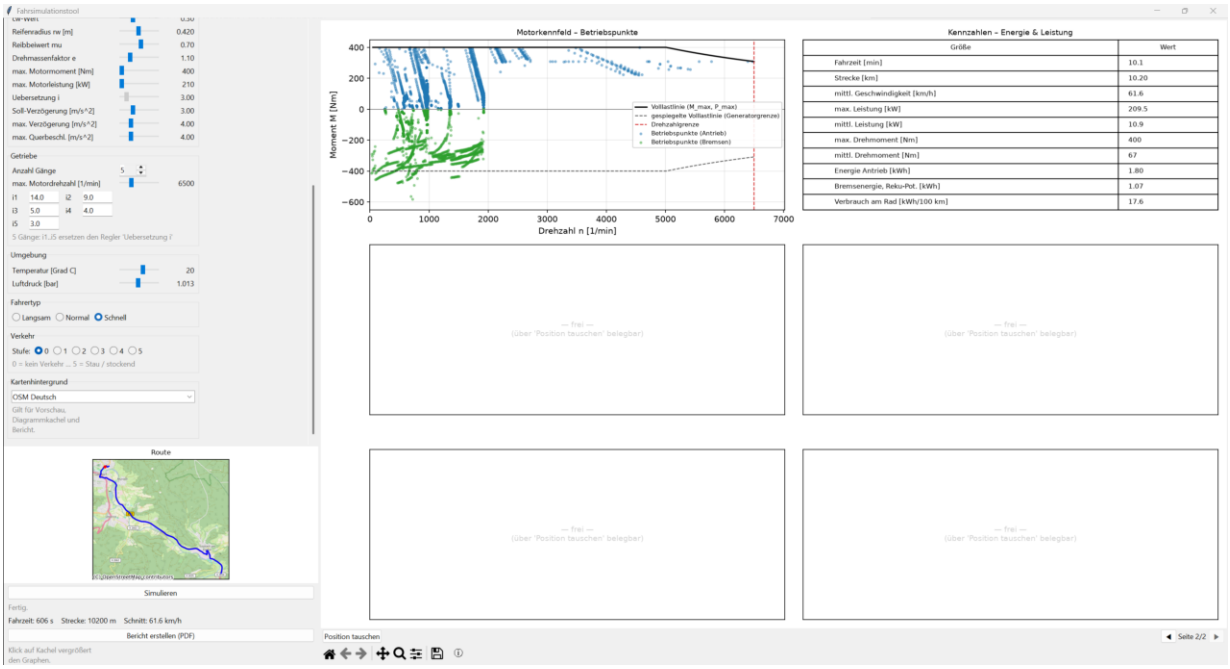


Abbildung 27: Grafische Benutzeroberfläche Fahrsimulationstool

Die erste Rasterseite enthält die acht Standarddiagramme des Lastzyklus:

- Geschwindigkeitsverlauf über der Zeit und über der Strecke (mit Tempolimit, Sollwert und Ampelmarkierungen)
- Drehmoment über Zeit
- Höhenprofil der Strecke
- Leistungsverläufe von Antrieb und Bremse
- Pedalstellungen
- Motorkennfeld welches unmittelbar der Auslegungsbeurteilung dient: Drängen sich die Betriebspunkte an die Drehzahlgrenze, ist die Übersetzung zu kurz oder die Maximaldrehzahl zu niedrig gewählt. Verbleiben alle Punkte weit unterhalb der Grenzkennlinie, ist die Maschine für das betrachtete Szenario überdimensioniert
- Ermittelte Ergebniskennzahlen

Bei der Umsetzung wurde besonderer Wert auf Robustheit gegenüber der Anzeigeumgebung gelegt: Die Anwendung meldet sich unter Windows als DPI-bewusst an, erkennt also die Pixeldichte des verwendeten Bildschirms und passt die Skalierung der Benutzeroberfläche automatisch an. Jeder Parameterwert ist per Mausklick auf das entsprechende Eingabefeld anwählbar und veränderbar (mit Bereichsprüfung und Unterstützung des Dezimalkommas). Ein Informationsdialog erläutert sämtliche Parameter mit typischen Wertebereichen und deren Quelle. Während der Rechnung meldet die Statuszeile den Streckenfortschritt, wodurch die Oberfläche auch bei längerer Rechenzeit bedienbar bleibt.

Nach jeder Simulation werden Kennzahlen der gesamten Fahrt berechnet und ausgewertet. Die Antriebsenergie ergibt sich durch numerische Integration der Antriebsleistung,

$$E_{\text{Antrieb}} = \int_0^{t_{\text{Ende}}} F_{\text{Zug}}(t) \cdot v(t)\, dt \tag{5.15}$$

der streckenbezogene Energiebedarf als

$$b = \frac{E_{\text{Antrieb}}}{s \cdot 100} \tag{5.16}$$

in kWh je 100 km. Beide Größen werden zunächst „am Rad“, also ohne Annahme eines Antriebsstrangwirkungsgrads ausgewiesen. Ergänzend werden Maximalwerte und Fahrt-Mittelwerte von Antriebsleistung und Motormoment bestimmt – unmittelbare Eingangsgrößen für die

Maschinenauslegung in Kapitel 7. Im Zyklusmodus werden zudem der Folgefehler nach (5.1) und die größte Einzelabweichung berechnet. Bei aktivierter Reichweitenrechnung zusätzlich Reichweite und Zyklenzahl. Alle Kennzahlen erscheinen in der Tabellenkachel der zweiten Rasterseite.

Optional lässt sich ein Wirkungsgrad η des Antriebsstrangs zuschalten, der die Verluste in Maschine, Leistungselektronik und Getriebe zusammenfasst. Die Auswertung weist dann zusätzlich die Größen am Eingang des Antriebsstrangs aus

$$E_{\mathrm{Eingang}} = \frac{E_{\mathrm{Antrieb}}}{\eta} \quad und \quad E_{\mathrm{Reku}} = \eta \cdot E_{\mathrm{Brems}} \tag{5.17}$$

, wobei der Wirkungsgrad bei der Rekuperation entgegengesetzt wirkt. Von der am Rad verfügbaren Bremsenergie erreicht nur der Anteil η die Quelle. Die Fahrleistung selbst bleibt unverändert, da $M_{\max}$ und $P_{\max}$ bereits als Motorgrößen definiert sind. Der Wirkungsgrad wirkt ausschließlich auf die Energiebilanz. Damit lässt sich der am Rad ermittelte Bedarf ohne erneute Simulation auf einen angenommenen Antriebsstrang umrechnen und mit Herstellerangaben zum Verbrauch vergleichen.

Im Fahrzyklus-Modus lässt sich darüber hinaus eine nutzbare Energiekapazität C angeben, aus der die Reichweite bestimmt wird. Grundgedanke ist das wiederholte Abfahren des Zyklus, bis die Energie aufgebraucht ist. Da ein Normzyklus im Stillstand beginnt und endet, ist jede Wiederholung identisch; Energiebedarf E_{Z} und Weglänge s_{Z} einer Runde sind damit konstant. Die Reichweite ergibt sich aus der Anzahl vollständiger Wiederholungen und dem im letzten angebrochenen Zyklus noch zurückgelegten Weg. Die Anzahl vollständiger Durchläufe n folgt aus der nutzbaren Energiekapazität und dem Energiebedarf einer Runde:

$$n = \left\lfloor \frac{C}{E_{\mathrm{Z}}} \right\rfloor \tag{5.18}$$

Die Abrundung ist notwendig, da nur vollständig abgefahrene Zyklen die volle Weglänge s_{Z} beitragen. Die verbleibende Energie $(C - n \cdot E_{\mathrm{Z}})$ reicht für einen weiteren, unvollständigen Durchlauf, dessen Weganteil über die Umkehrfunktion des Energieverlaufs bestimmt wird:

$$s_R = n \cdot s_Z + s \cdot (C - n \cdot E_Z) \tag{5.19}$$

Dabei bezeichnet $s(E) = s \cdot (C - n \cdot E_{\mathrm{Z}})$ den Weg, an dem die kumulierte Energie innerhalb eines Zyklus den Wert E erreicht. Dieser Zusammenhang wird aus dem Energieverlauf eines einzigen Zyklus errechnet. Die Simulation muss den Zyklus dafür nicht mehrfach durchlaufen

– das Ergebnis ist identisch, der Rechenaufwand jedoch um den Faktor der Wiederholungszahl geringer, was bei Reichweiten von mehreren hundert Kilometern den Unterschied zwischen Sekunden und Minuten ausmacht.

Zwei Festlegungen bestimmen die Aussage des Ergebnisses. Ist ein Wirkungsgrad zugeschaltet, wird der Energiebedarf an der Quelle gemäß (5.17) zugrunde gelegt; andernfalls wird der Bedarf am Rad berücksichtigt. Die Bremsenergie wird nicht gutgeschrieben, da sie im Modell ausschließlich als Rekuperationspotenzial dargestellt wird. Die berechnete Reichweite stellt somit eine bewusst konservative Schätzung dar. Diese Option steht ausschließlich im Fahrzyklus-Modus zur Verfügung. Einmal abgerufene Routen lassen sich unter frei wählbaren Namen als JSON-Dateien speichern und später ohne erneute API-Abfrage laden. Somit können Simulationsläufe auf identischer Streckenbasis auch offline reproduziert werden. Falls der Stillstandswächter eine Simulation abbricht, zeigt die Benutzeroberfläche neben einer Meldung mit den Diagnosewerten – Kilometerposition, lokale Steigung, benötigte und verfügbare Kraft sowie die Kraftschlussgrenze – die betreffende Stelle als gelb markierten Punkt auf der Routenkarte an. Ergänzend nennt die Meldung die aus der aktuellen Parametrierung resultierende Anfahr-Grenzsteigung gemäß (5.15), sodass unmittelbar erkennbar ist, ob eine zu lange Gesamtübersetzung oder die Streckentopographie die Ursache ist.

Die Routenkarte zeigt den Streckenverlauf auf einer Karte, die aus vorgerenderten Ausschnitten eines Webdienstes besteht. Der Bezug dieser von den Servern des OpenStreetMap-Projekts erwies sich als nicht tragfähig: Diese Server werden ehrenamtlich betrieben und unterliegen einer strengen Nutzungsrichtlinie, die eine eindeutige Kennung (User-Agent), Zwischenspeicherung der Inhalte und Verzicht auf massenhafte Abrufe fordert. Die verwendete Bibliothek “contextily“ verletzt diese Bedingungen, daher wurde der Datenbezug in einer Funktion gekapselt, die drei Punkte umsetzt: Die Anwendung meldet sich mit einer eigenen Kennung an, Abrufe werden zwischengespeichert und die Karendaten stammen nun von einem Dienst, dessen Nutzungsbedingungen den Abruf durch eine Desktopanwendung erlauben. Dabei werden diese weiterhin von OpenStreetMap sowie zwei weiteren Anbietern (CARTO, Esri) bezogen, nur der Server, der die Bildkacheln rendert, ist ein anderer. Anzumerken bleibt, dass die Kapselung es erlaubt, den Kartendienst an einer Stelle gegen einen eigenen oder kommerziellen Dienst zu tauschen. Der Kartenhintergrund ist als Auswahlfeld in der Oberfläche vorhanden (siehe Abbildung 26), sodass der Nutzer zwischen mehreren Darstellungen wählen kann. Ein Wechsel aktualisiert nur die Routenkarte, ohne die Simulation zu unterbrechen.

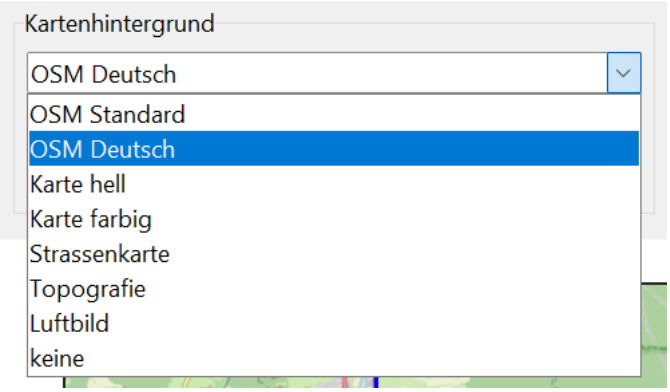


Abbildung 28: Auswahl Kartendarstellung in der Benutzeroberfläche

Für die Dokumentation von Simulationsläufen erzeugt die Schaltfläche „Bericht erstellen“ nach Abfrage des Speicherorts einen mehrseitigen PDF-Bericht: ein Deckblatt mit sämtlichen Eingangsparametern und Kennzahlen, die sechs Standarddiagramme, das Motorkennfeld und die Kennzahlentabelle sowie im Streckenmodus die Routenkarte. Die Berichtserzeugung nutzt die PDF-Ausgabe von *matplotlib* und kommt damit ohne zusätzliche Abhängigkeiten aus.

6 Validierung

Im Folgenden wird das Modell anhand realer Messungen validiert. Hierfür wird als Vergleichsfahrzeug der Opel Corsa F Electric (Ultimate) aus dem Baujahr 2021 herangezogen. Die Fahrzeugparameter und deren Quelle werden in der untenstehenden Tabelle 13 dargestellt:

Parameter	Wert	Quelle
Masse	1650kg (Leergewicht 1530kg + Fahrer 80kg + Tankfüllung 40kg)	ADAC Database, Annahme (Fahrer + Tank)
Leistung	100kW	Angabe Fahrzeugschein
Drehmoment	260Nm	ADAC
Stirnfläche	2,13m^2	Automobil-guru.de
Cw-Wert	0,29	ADAC
Reifenradius	30,8cm	Berechnet aus Serienreifengröße 205/45R17 (ADAC)
Reibbeiwert	1,0	Annahme Asphalt (vgl. Abschnitt 5.5)
Drehmassenfaktor	1,1	Annahme (vgl. Abschnitt 2.2.4)
Übersetzung	10,8	Berechnet siehe nachfolgend
Anzahl Gänge	1	ADAC

Drehzahl	14000	Motordaten Schaeffler EMR3 [51], welcher im genannten Fahrzeug verbaut ist[52]
Temperatur	20°C	Annahme aufgrund Jahreszeit Frühling
Verkehrsstufe	1	Annahme da Morgenfahrten
Wirkungsgrad	85%	Annahme (vgl. Abschnitt 2.3.2)
Fahrertyp	Normal	Annahme

Tabelle 13: Fahrzeug- und Simulationsparameter Opel Corsa F Electric 100kW, Baujahr: 2021

Der Nennradius des Reifen wird aus der Serienreifengröße 205/45R17, mit Nennbreite B in mm, Querschnittsverhältnis Q in Prozent und Felgennenndurchmesser Z in Zoll, berechnet. Dafür werden folgende Formeln angewandt:

$$U = \pi \cdot D = \pi \cdot \left(Z \cdot 25{,}4mm + 2 \cdot \frac{B \cdot Q}{100}\right) \tag{6.1}$$

$$r = \frac{U}{2\pi} \tag{6.2}$$

Daraus ergibt sich $r = 0{,}308m$ [53, S. 76]. Die Gesamtübersetzung ist für dieses Fahrzeug nicht veröffentlicht und wird daher anhand der vorliegenden Kennzahlen bestimmt. Für eine grundlegende Näherung wird zur Vereinfachung mit dem Nennradius des Reifens gerechnet. Bei einem einstufigen Getriebe entfällt die Gangwahl, dadurch lässt sich die Übersetzung an eine Bedingung knüpfen: Beim Erreichen der Höchstgeschwindigkeit des Fahrzeugs muss die maximal festgelegte Drehzahl des Motors erreicht sein. Aus dem Verhältnis der Winkelgeschwindigkeiten von Motor und Antriebsrädern folgt damit unmittelbar:

$$i = \frac{\omega_{\text{Motor,max}}}{\omega_{\text{Rad,max}}} = \frac{\pi \cdot n_{\text{max}} \cdot r}{30 \cdot v_{\text{max}}} \tag{6.3}$$

Dabei ist die Abregelgeschwindigkeit des Fahrzeugs laut ADAC Database auf 150 km/h festgelegt. Die maximale Motordrehzahl an diesem Punkt geht mit 14000 1/min in die Berechnung ein. Damit ergibt sich $i \approx 10{,}8$ für die Gesamtübersetzung.

6.1 Reale Fahrerprobung

Als Vergleichsbasis dienen sieben Fahrten desselben Fahrzeugs auf der Strecke Im oberen Grund 24, 78166 Donaueschingen – DHBW Campus Horb, 72160 Horb am Neckar mit Anfahrt über Mühlen, aufgezeichnet zwischen Februar und April 2026 jeweils in den frühen Morgenstunden. Die Aufzeichnung erfolgte über die OBD-Schnittstelle und umfasst mehrere Messgrößen aus unterschiedlichen Messkanälen: Fahrgeschwindigkeit, Leistung und Strom der

Hochvoltbatterie, Ladezustand sowie den fahrzeugseitigen Energiezähler. Positionsdaten wurden nicht mitgeschrieben, weshalb die Route für die Simulation aus den Adressen erzeugt wird.

Zwei Eigenschaften der Aufzeichnungen bestimmen das methodische Vorgehen. Erstens wurde der Logger jeweils erst nach Fahrtbeginn gestartet: In allen sieben Aufzeichnungen fährt das Fahrzeug bereits beim ersten Datenpunkt mit 48 bis 102 km/h. Das Fahrtende hingegen ist vollständig erfasst. Als gemeinsame Bezugsachse von Messung und Simulation dient deshalb nicht die Zeit ab Fahrtbeginn, sondern der Restweg bis zum Ziel; verglichen wird ausschließlich der Streckenabschnitt, den beide Datensätze abdecken. Zweitens liegen die Messkanäle nicht auf einer gemeinsamen Zeitbasis. Die Aufzeichnungssoftware schreibt je Zeile nur denjenigen Wert, den das Steuergerät zu diesem Zeitpunkt übermittelt hat, während die übrigen Spalten der Zeile leer bleiben. Zudem werden die einzelnen Größen unterschiedlich häufig aktualisiert, die Fahrgeschwindigkeit beispielsweise im Mittel alle 1,1 Sekunden. Es existiert damit kein Zeitpunkt, zu dem alle Messgrößen gleichzeitig vorliegen. Für die Auswertung wird deshalb jeder Messkanal einzeln auf ein gemeinsames Zeitraster mit einem Wert je Sekunde umgerechnet. Liegt zu einer vollen Sekunde kein Messwert vor, wird er aus dem vorhergehenden und dem nachfolgenden Wert desselben Kanals bestimmt. Eine feinere Auflösung ist angesichts der Übertragungsraten der Steuergeräte nicht sinnvoll.

Für die Messfahrten liegt keine unabhängige Referenzmessung vor. Die Belastbarkeit der Daten wird deshalb über die Redundanz der aufgezeichneten Kanäle geprüft. Das Fahrzeug meldet zum einen einen fortlaufenden Zähler der verbrauchten Energie, zum anderen die momentane Leistung der Hochvoltbatterie. Summiert man die Leistung über die Fahrzeit auf, muss sich der Zählerstand ergeben; über alle sieben Fahrten beträgt die Abweichung zwischen beiden Größen weniger als 0,7 %. Darüber hinaus lässt sich aus der umgesetzten Energie und dem zugehörigen Rückgang des Ladezustands die nutzbare Kapazität der Batterie zurückrechnen. Alle sieben Fahrten führen dabei übereinstimmend auf 41,1 ± 1,0 kWh. Für das Neufahrzeug werden 46 kWh netto angegeben [54], woraus sich mit dem vom BMS gemeldeten State of Health von 92,4 % eine zu erwartende Kapazität von 42,5 kWh ergibt. Die verbleibende Differenz von 1,4 kWh liegt in der Größenordnung der Streuung zwischen den einzelnen Fahrten. Leistungs-, Energie- und Ladezustandskanal sind damit untereinander konsistent.

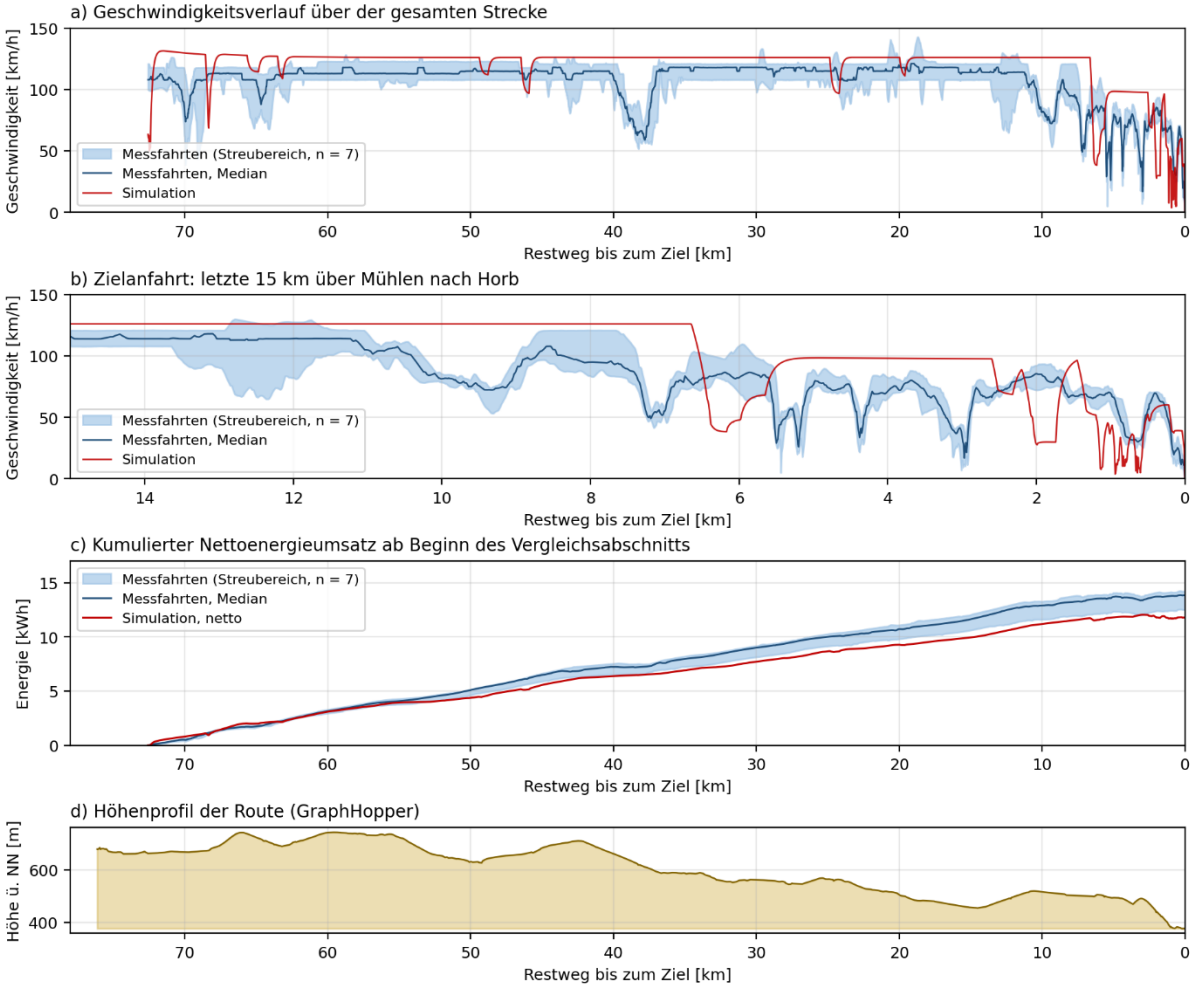


Abbildung 29: Vergleich von Simulation und der sieben Messfahrten

Der kumulierte Energieumsatz in Abbildung 29 (Diagramm c) zeigt, dass die Simulation den Verlauf über die gesamte Strecke qualitativ richtig wiedergibt, den Bedarf jedoch durchgehend leicht unterschätzt. Die Abweichung wächst gleichmäßig mit dem zurückgelegten Weg und nicht sprunghaft an einzelnen Streckenpunkten, was auf eine systematische und nicht auf eine lokale Ursache hindeutet. Die quantitative Gegenüberstellung fasst die nachfolgende Tabelle 14 zusammen.

Größe	Messung (n = 7)	Simulation	Abweichung
Nettoverbrauch [kWh/100 km]	18,90 ± 0,78	15,86	−16,1 %
Bruttoverbrauch [kWh/100 km]	20,61 ± 0,82	18,30	−11,2 %
Energiebedarf netto [kWh]	14,16 ± 0,60	12,07	−14,8 %
Rekuperierte Energie [kWh]	1,28 ± 0,24	1,86	+45,3 %
Anteil am Bruttobedarf [%]	8,3	13,4	+5,1 Pp.

Tabelle 14: Energiebilanz der Simulation verglichen mit der realen Fahrtmessung

Der Nettoverbrauch, also der an der Batterie gemessene Bedarf abzüglich der zurückgespeisten Energie, liegt in der Simulation um 16,1 % unter dem Mittel der Messfahrten. Zur Einordnung: Die sieben Messfahrten streuen untereinander lediglich zwischen 17,40 und 19,72 kWh/100 km. Die Modellabweichung übersteigt die Streubreite der Realfahrten deutlich und ist damit systematischer Natur.

Die Abweichung setzt sich aus mehreren Anteilen der Modellvereinfachung zusammen. Der erste betrifft die Nebenverbraucher: Heizung, Gebläse, Licht und Steuergeräte gehen vollständig in den gemessenen Batteriebedarf ein, sind im reinen Fahrwiderstandsmodell aber nicht abgebildet. Auf der Bruttoebene entspricht die verbleibende Differenz von 1,76 kWh bei einer Fahrzeit von rund 48 Minuten einer mittleren Zusatzleistung von etwa 2,3 kW. Diese Größenordnung deckt sich mit dem erwartbaren Bedarf der Nebenverbraucher in der Jahreszeit der Messfahrten von Februar bis April. Gestützt wird die Deutung dadurch, dass die Abweichung nach Abbildung c) über die gesamte Strecke gleichmäßig und nicht sprunghaft anwächst, wie es für einen weitgehend konstanten Grundverbrauch zu erwarten ist.

Der zweite Anteil liegt beim Rekuperationspotenzial. Auf der Bruttoebene, also ohne dessen Gutschrift, verringert sich die Abweichung von 16,1 % auf 11,2 %. Das Modell weist mit 1,86 kWh einen um 45,3 % höheren Wert aus als die Messung mit 1,28 kWh; sein Anteil am Bruttobedarf beträgt 13,4 % gegenüber 8,3 % in der Messung. Ursache ist die Modellierung des Bremsvorgangs: Der Simulationskern erfasst die gesamte Bremsarbeit als Rekuperationspotenzial und rechnet sie mit dem konstanten Wirkungsgrad um, während das reale Fahrzeug nur einen kleineren Teil zurückspeisen kann. Der ausgewiesene Wert ist damit erwartungsgemäß eine Obergrenze und als solche zu interpretieren.

Der dritte Anteil liegt beim konstanten Wirkungsgrad, der bereits in 2.3.2 erwähnt wurde. Der Antriebsstrang wird über einen konstanten Wirkungsgrad von 85 % abgebildet. Da der reale Wirkungsgrad vom Betriebspunkt abhängt und gerade im Teillastbereich deutlich niedriger liegt, verzerrt diese Annahme den Energiebedarf, insbesondere dort, wo die Maschine den Großteil der Zeit arbeitet.

Zudem sind stochastische Einflussfaktoren nicht berechenbar. Fahrbahnzustand durch Nässe, Schnee oder Verschmutzung, Wind sowie lokale Temperaturunterschiede wirken über den Roll- und Luftwiderstand auf das Ergebnis, werden im Modell aber als konstant angenommen. Auch das Verkehrsgeschehen wird nur statistisch über die Verkehrsstufe und eine Wahrscheinlichkeit für rote Ampeln nachgebildet. Schließlich kommt die Qualität der Kartendaten

hinzu: Tempolimits sind in OpenStreetMap nicht ausnahmslos akkurat, sodass für fehlende Abschnitte auf Ersatzwerte nach Straßenklasse zurückgegriffen wird.

6.2 WLTP-Vergleich

Neben der realen Fahrerprobung wird das Modell gegen die WLTP-Werte, einem Verbrauch von 15,8 kWh (mit Nebenverbrauchern) und einer Reichweite von 353 km, geprüft. Dazu wird der WLTC der Klasse 3b im zyklusbasierten Modus mit identischem Fahrzeugparametersatz simuliert, dieser Klasse ist das Fahrzeug aufgrund seines Leistungsgewichts von 65,4 W/kg zugeordnet. Das Leistungsgewicht Pmr berechnet sich aus

$$Pmr = \frac{P_{\mathrm{Nenn}}}{m_{\mathrm{Nenn}} + m_{\mathrm{Fahrer}}} \tag{6.4}$$

mit $P_{Nenn} = 100\ kW$, $m_{Nenn} = 1530\ kg$, $m_{Fahrer} = 75\ kg$ [1, Abs. Anhang XXI, Unteranhang 1, Nr. 2]. Anders als bei der Streckensimulation ist das Geschwindigkeitsprofil hier fest vorgegeben, sodass sich zusätzlich die Güte des Fahrerreglers beurteilen lässt. Nachfolgend sind in Tabelle 15 die Simulationsergebnisse, verglichen mit denen der WLTP-Prüfung dargestellt.

Größe	Simulation	Referenz
Zykluslänge [km]	23,278	23,266
Folgefehler (RMSE) [km/h]	0,68	–
Bruttoverbrauch [kWh/100 km]	17,03	15,8
Nettoverbrauch [kWh/100 km]	14,5	–
Rekuperationspotenzial [kWh]	0,91	–
WLTP-Reichweite [km]	318,5	353

Tabelle 15: Ergebnisse der Zyklussimulation - WLTC Klasse 3b

Der Fahrerregler folgt dem Sollprofil mit einem Folgefehler von 0,68 km/h, die zurückgelegte Strecke weicht um 0,05 % von der Zykluslänge ab. Die Umsetzung des vorgegebenen Profils ist damit als korrekt anzusehen; Abweichungen im Energiebedarf sind folglich dem Fahrwiderstandsmodell und nicht der Regelung zuzuordnen.

Beim Verbrauch ist die Bezugsebene zu beachten. Die Herstellerangabe nach WLTP wird an der Ladesäule ermittelt und enthält damit die Ladeverluste, während das Modell den Bedarf am Eingang des Antriebsstrangs ausweist. Der WLTP-Verbrauchswert von 15,8 kWh/100 km liegt zwischen den beiden Modellgrenzen: 14,5 kWh/100 km am Rad und 17,03 kWh/100 km am Motor. Das Modell schließt den WLTP-Wert somit ein, ohne ihn punktgenau zu treffen. Bei der Reichweite weicht die Simulation mit 40 km zu wenig ab, dies ist durch den höheren Durchschnittverbrauch am Motor und die genannten Abweichungen begründet.

7 Ergebnisse

Nachdem das Modell im vorangegangenen Kapitel gegen reale Messfahrten und den WLTC-Zyklus geprüft wurde, wird nun gezeigt, welchen Nutzen die erzeugte Simulation für die Auslegung einer elektrischen Maschine hat. Hierzu wird weiterhin das gewählte Fahrzeug verwendet, da es die ermittelten Ergebnisse gut mit der Realität vergleichbar und bewertbar macht. Abschließend werden die Modellannahmen kritisch betrachtet.

7.1 Ableitung von Anforderungen zur Grobauslegung elektrischer Maschinen

Die in Abschnitt 2.3.2 beschriebenen Kenngrößen einer elektrischen Maschine – Maximalmoment, Maximaldrehzahl sowie die Unterscheidung zwischen kurzzeitiger Überlast und Dauerleistung – lassen sich unmittelbar aus dem simulierten Lastzyklus quantifizieren. Hinzu treten Größen, die erst aus dem Betriebsprofil entstehen: die über die Fahrt geforderte Dauerleistung, die Verteilung von Moment und Drehzahl sowie das Rekuperationspotenzial. Für die Dauerleistung wird der quadratische Mittelwert der Antriebsleistung herangezogen, da die Erwärmung der Wicklung näherungsweise mit dem Quadrat des Stroms und damit des Moments steigt. Tabelle 16 stellt wichtige Auslegungsgrößen dar.

Kenngröße	Wert	Bedeutung für die Auslegung
Spitzenleistung am Rad	85 kW	Für diese Strecke wird die maximale Radleistung abgerufen
Spitzenleistung am Eingang	100 kW	Die maximale Eingangsleistung wird ebenfalls benötigt
Dauerleistung (quadrat. Mittel)	23,9 kW	Unter anderem nützlich für thermische Auslegung der Maschine
mittlere Leistung	17,4 kW	Energiebedarf über die Fahrt
maximales Moment	260 Nm	Maximales Drehmoment wird abgerufen
95%-Quantil des Moments	56 Nm	Bereich der Wirkungsgradoptimierung
Höchstdrehzahl	12.207 1/min	Maximale Drehzahl von 14.000 1/min wird nicht genutzt
Energiebedarf je Fahrt	13,86 kWh	Auslegung der Batteriekapazität

Tabelle 16: Ermittelte Lastgrößen aus dem Simulationslauf (Opel Corsa F Electric)

Aussagekräftiger als die Einzelwerte ist ihre Verteilung über die Fahrt. Die volle Leistung wird nur während 1,0 % der Fahrzeit abgerufen, während 51,8 % der Zeit auf Leistungen unterhalb von 20 kW entfallen. Das Verhältnis von Spitzen- zu Dauerleistung beträgt rund 3,5 zu 1; die Maschine muss also kurzzeitig deutlich überlastbar sein, thermisch aber nur für weniger als ein Drittel der Spitzenleistung ausgelegt werden. Eine Auslegung allein auf den Spitzenwert würde zu einer erheblich überdimensionierten Maschine führen.

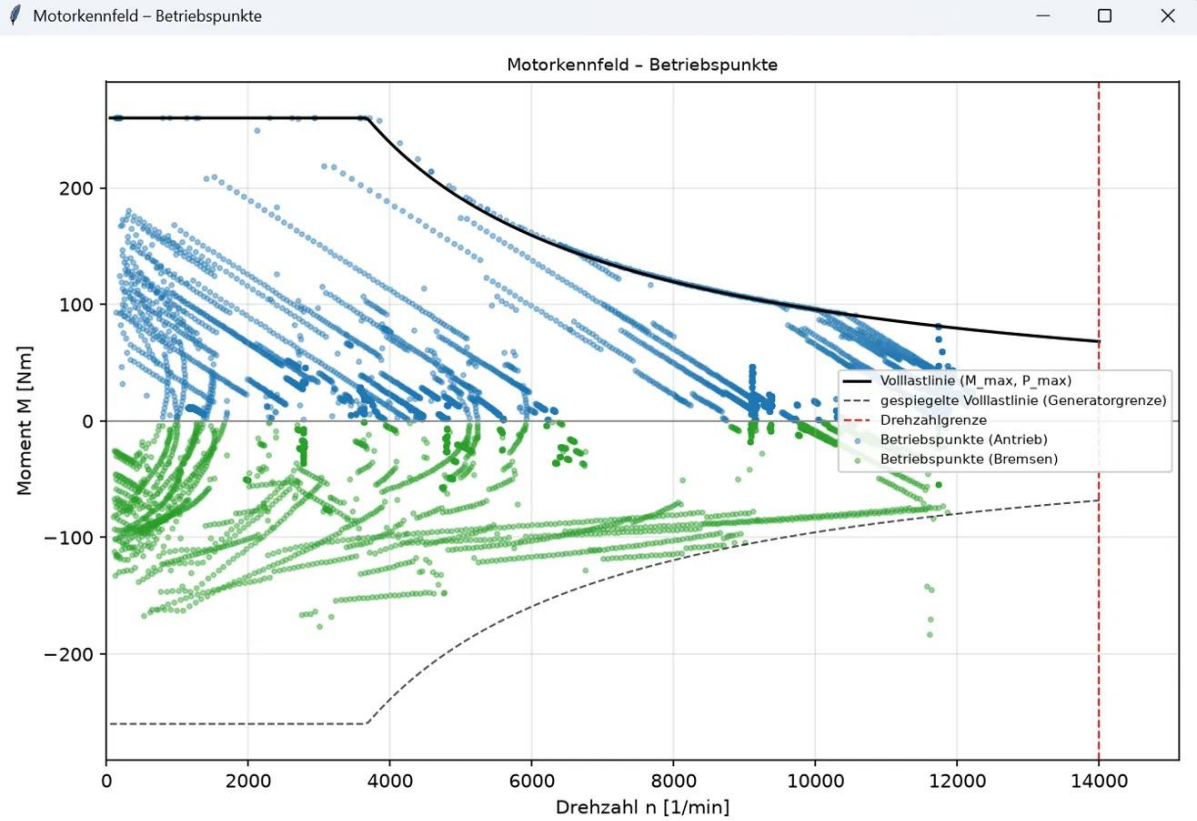


Abbildung 30: simuliertes Motorkennfeld und dessen Betriebspunkte (Opel Corsa F Electric)

Ebenso deutlich fällt die Verteilung über dem simulierten Kennfeld aus, welches in Abbildung 30 grafisch dargestellt ist. In 95,4 % der Fahrzeit liegt das Moment unterhalb von 60 Nm, also unter einem Viertel des verfügbaren Maximalmoments, und in 73,3 % der Zeit arbeitet die Maschine bei Drehzahlen zwischen 7.000 und 13.000 1/min. Dies liegt daran, dass der größte Streckenanteil bei nahezu konstantem Tempo und leichtem Gefälle gefahren wird. Der Betrieb konzentriert sich damit auf den oberen Drehzahlbereich bei kleinem Moment – ein Gebiet, das im Kennfeld einer elektrischen Maschine typischerweise in der Feldschwächung liegt. Für den betrachteten Anwendungsfall wäre folglich eine Optimierung des Wirkungsgrads in genau diesem Bereich wirksamer als eine Anhebung des Spitzenmoments.

7.2 Bewertung der Ergebnisse

Die entwickelte Simulationsumgebung liefert die gesuchten Vorauslegungsgrößen allein aus Start- und Zieladresse, dem Fahrzeugparametersatz und den Eckdaten zur Umgebung. Alle wichtigen Simulationsergebnisse, darunter auch die für die Vorauslegung nützlichen Diagramme, werden in der Benutzeroberfläche dargestellt.

Für die Belastbarkeit dieser Größen ist entscheidend, wie präzise die Parameter gewählt sind und wie umfangreich Umgebungsbedingungen abgebildet sind. Die in Kapitel 6 festgestellte Unterschätzung und Abweichung des Energiebedarfs betreffen die Energiebilanz, jedoch kaum die Spitzen- und Dauerleistung oder den genutzten Drehzahlbereich. Die Ursachen der Abweichung sind Modellvereinfachungen und werden in Abschnitt 6.1 behandelt. Die abgeleiteten Auslegungsgrößen sind damit belastbarer als die absolute Energiebilanz.

Dass das reale Serienfahrzeug deutlich über den aus diesem Nutzungsprofil abgeleiteten Anforderungen liegt, ist erklärlich. Denn in diesem Fall wurde zur Veranschaulichung eine Strecke befahren, die die Maschine nicht dauerhaft erheblich belastet. Die daraus ermittelten Auslegungsgrößen sind gut für ein Fahrzeug mit genau diesem Beanspruchungsprofil, jedoch unzureichend, wenn das Fahrzeug nun beispielsweise im Anhängerbetrieb, mit Volllast im Gebirge oder bei extremen Temperaturen fahren soll. Daher sollten für eine serienfahrzeugnahe Auslegung weitere Strecken mit unterschiedlichsten Anforderungen simuliert werden.

8 Zusammenfassung und Ausblick

Ziel dieser Arbeit war eine streckenbasierte Fahrsimulationsumgebung, die aus einer frei wählbaren realen Strecke einen Lastzyklus erzeugt und daraus die Vorauslegungsgrößen elektrischer Fahrzeugantriebe ableitet. Umgesetzt wurde ein Werkzeug, das Routen-, Höhen- und Kartendaten öffentlich zugänglicher Dienste zu einer Streckenrepräsentation aufbereitet und diese mit einem vorausschauenden Fahrermodell und einem längsdynamischen Fahrzeugmodell zu einer Fahrsimulation verarbeitet. Ergänzt wird es durch eine Fahrzeug- und eine Fahrzyklusdatenbank sowie eine grafische Benutzeroberfläche, über die sämtliche Parameter eingestellt und die Ergebnisse ausgewertet werden können. Der Quellcode ist durchgängig kommentiert und modular aufgebaut, sodass sich einzelne Modellbausteine ohne Eingriff in die übrige Struktur ersetzen lassen.

Die Validierung erfolgte zweistufig gegen sieben reale Messfahrten und gegen den WLTC der Klasse 3b. Der Energiebedarf wird gegenüber den Messfahrten um rund 15 Prozent unterschätzt, wobei die Ursachen benannt und quantifiziert werden konnten. Der Geschwindigkeitsverlauf und die daraus folgenden Leistungs- und Drehzahlanforderungen werden dagegen zutreffend wiedergegeben.

Genau diese Größen bilden das Anforderungsprofil, das in Kapitel 7 aus dem Lastzyklus abgeleitet wurde. Die Arbeit zeigt damit, dass sich belastbare Vorauslegungsgrößen allein aus öffentlich verfügbaren Streckendaten und wenigen Fahrzeugeckdaten gewinnen lassen, ohne Messkampagne und ohne Prüfstandsversuch.

Für die Weiterentwicklung ergeben sich mehrere Ansatzpunkte. Den größten Gewinn verspricht die Ablösung des konstanten Wirkungsgrads durch ein Wirkungsgradkennfeld, da die Maschine den überwiegenden Teil der Zeit im Teillastbereich arbeitet, in dem die Vereinfachung am stärksten wirkt. In dieselbe Richtung zielen ein Rekuperationsmodell mit Leistungsgrenze sowie die Berücksichtigung der Nebenverbraucher; beide würden die verbleibende Abweichung in der Energiebilanz deutlich verringern. Da ein Kennfeld in der frühen Auslegungsphase noch nicht vorliegt, wäre ein iteratives Vorgehen denkbar: Aus dem Anforderungsprofil werden die Eckdaten des Motors grob dimensioniert, dessen Kennfeld anschließend in eine zweite Simulation zurückfließt. Methodisch bietet es sich an, nicht mehr einzelne Strecken, sondern ganze Streckenkollektive zu betrachten. Da Verkehr und Ampelzustände stochastisch nachgebildet werden, liefert jeder Lauf nur eine mögliche Realisierung. Eine Auswertung über viele Läufe und mehrere Streckentypen würde statistisch abgesicherte Anforderungen ergeben und den in Abschnitt 7.2 beschriebenen Nachteil eines einzelnen Nutzungsprofils aufheben. Auf Datenseite wäre die Einbindung von Wetter- und Verkehrsdiensten der nächste Schritt, um die bislang als konstant angenommenen Umgebungsbedingungen abzubilden.

Über den betrachteten Anwendungsbereich hinaus ließe sich der Ansatz auf weitere Fahrzeugarten übertragen. Schienenfahrzeuge bieten sich dabei besonders an, da ihr Fahrweg fest vorgegeben ist und sich das Streckenmodell entsprechend vereinfacht. Luft- und Wasserfahrzeuge erfordern dagegen eine grundlegend andere Beschreibung des Fahrwegs und der Widerstände, sodass hier eher die Auswertungssystematik als das Streckenmodell übertragbar wäre.

Literaturverzeichnis

[1] „Emissionen des Verkehrs“, Umweltbundesamt: [Online]. Verfügbar unter: https://www.uba.de/n11168de
[2] *VERORDNUNG (EU) 2023/851 DES EUROPÄISCHEN PARLAMENTS UND DES RATES*. 2023.
[3] *VERORDNUNG (EU) 2024/1610 DES EUROPÄISCHEN PARLAMENTS UND DES RATES*. 2024.
[4] Kraftfahrt-Bundesamt (KBA), „Fahrzeugzulassungen im Juli 2026“, Flensburg, Pressemitteilung 32/2026, 08 2026. [Online]. Verfügbar unter: https://www.kba.de/DE/Presse/Pressemitteilungen/Fahrzeugzulassungen/2026/pm32_2026_n_07_26_pm_komplett.html
[5] A. Süßmann, „Kundenspezifische Bewertung von Maßnahmen zur Reduktion des Kraftstoffverbrauchs bei schweren Nutzfahrzeugen“, Technische Universität München, 2020. [Online]. Verfügbar unter: https://mediatum.ub.tum.de/doc/1546563/document.pdf
[6] U. Seiffert und S. Pischinger, *Vieweg Handbuch Kraftfahrzeugtechnik*, 9., Bd. ATZ/MTZ-Fachbuch. Wiesbaden: SpringerVieweg, 2021.
[7] T. J. Barlow, *A reference book of driving cycles for use in the measurement of road vehicle emissions.* TRL Limited, 2009. [Online]. Verfügbar unter: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/4247/ppr-354.pdf
[8] *Verordnung (EU) 2017/1151 - Typgenehmigungsverfahren*. 2017.
[9] United Nations Economic Commission for Europe (UNECE), *Global Technical Regulation No. 15: Worldwide Harmonized Light Vehicles Test Procedure (WLTP)*. 2014.
[10] *Verordnung (EU) 2016/427 des DES EUROPÄISCHEN PARLAMENTS UND DES RATES zur Änderung der Verordnung (EG) Nr. 692/2008 hinsichtlich der Emissionen von leichten Personenkraftwagen und Nutzfahrzeugen (Euro 6)*. 2016.
[11] K.-H. Waldmann und W. E. Helm, *Simulation stochastischer Systeme: Eine anwendungsorientierte Einführung*. Heidelberg: Springer Gabler Berlin, 2016. doi: https://doi.org/10.1007/978-3-662-49758-6.
[12] K. Reif (Hrsg.), *Grundlagen Fahrzeug- und Motorentechnik*. in Bosch Fachinformation Automobil. DHBW Ravensburg, Campus Friedrichshafen: SpringerVieweg, 2017.
[13] T. Schütz, *Hucho - Aerodynamik des Automobils*, 7. Aufl. in ATZ/MTZ-Fachbuch. Springer Vieweg Wiesbaden, 2023.
[14] M. Mitschke und H. Wallentowitz, *Dynamik der Kraftfahrzeuge*, 5. Aufl. Wiesbaden: Springer Vieweg, 2014. doi: https://doi.org/10.1007/978-3-658-05068-9.
[15] „Innenläufer Synchronmaschine mit vergrabenen Magneten“, Hybrid-Autos.info. [Online]. Verfügbar unter: https://www.hybrid-autos.info/index.php?option=com_content&view=article&id=62:pmsm-mit-vergrabenen-magneten&catid=321&Itemid=311
[16] M. Doppelbauer, *Grundlagen der Elektromobilität*, 2. Aufl. Springer Vieweg Wiesbaden, 2025. doi: https://doi.org/10.1007/978-3-658-44828-8.

[17] A. Binder, „Elektrische Maschinen und Antriebe“. Institut für Elektrische Energiewandlung, Technische Universität Darmstadt, 2023. [Online]. Verfügbar unter: https://www.etit.tu-darmstadt.de/media/ew/rd/ew_vorlesungen/lv_ema/EMA_Skript_2022_23_komp.pdf
[18] C. J. Date, *An Introduction to Database Systems*, 8. Aufl. Boston: Addison-Wesley, 2003.
[19] R. Elmasri und S. B. Navathe, *Fundamentals of Database Systems*, 7. Aufl. Boston: Pearson, 2016.
[20] A. Kemper und A. Eickler, *Datenbanksysteme: Eine Einführung*, 10. Aufl. Berlin, Boston: De Gruyter Oldenbourg, 2015.
[21] H.-P. Schoener, „Erprobung und Absicherung im dynamischen Fahrsimulator“, Daimler AG, Sindelfingen, Conference Paper 17, Nov. 2014.
[22] T. Markel et al, „ADVISOR: A Systems Analysis Tool for Advanced Vehicle Modeling“. Elsevier B.V., National Renewable Energy Laboratory, Golden, CO 80401, USA, 3. Juni 2002. [Online]. Verfügbar unter: https://research-hub.nlr.gov/en/publications/advisor-a-systems-analysis-tool-for-advanced-vehicle-modeling-2/
[23] Europäische Kommission, „Vehicle Energy Consumption calculation Tool - VECTO“. [Online]. Verfügbar unter: https://climate.ec.europa.eu/areas-action/transport-decarbonisation/road-transport/vehicle-energy-consumption-calculation-tool-vecto_en
[24] C. Gaete-Morales, H. Kramer, W. Schill, und et al, „An open tool for creating battery-electric vehicle time series from empirical data, emobpy“. Sci Data 8, 152, DIW Berlin (German Institute of Economic Research), 2021. doi: https://doi.org/10.1038/s41597-021-00932-9.
[25] A. Wolf, „Map-Based Driving Cycle Generation“. Karlsruhe, 12. Oktober 2015. [Online]. Verfügbar unter: https://publikationen.bibliothek.kit.edu/1000049932
[26] „GraphHopper“. [Online]. Verfügbar unter: https://www.graphhopper.com/de/impressum/
[27] „Open Source Routing Machine“. [Online]. Verfügbar unter: https://project-osrm.org/
[28] HeiGIT gGmbH, „openrouteservice“. [Online]. Verfügbar unter: https://openrouteservice.org/
[29] „Routes API - Google Maps Platform“. [Online]. Verfügbar unter: https://developers.google.com/maps/documentation/routes?hl=de
[30] HERE Technologies, „HERE Routing“. [Online]. Verfügbar unter: https://www.here.com/platform/routing
[31] Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR), „25 Jahre SRTM – die Shuttle Radar Topography Mission“. [Online]. Verfügbar unter: https://www.dlr.de/de/forschung-und-transfer/themen/das-system-erde-im-blick/25-jahre-srtm
[32] R. Pereira und et al, „Ranking programming languages by energy efficiency - Science of Computer Programmin, Vol. 205“. 2021. doi: https://doi.org/10.1016/j.scico.2021.102609.
[33] EPFL IC SCALA, „Scala Center“. [Online]. Verfügbar unter: https://scala.epfl.ch/
[34] „Python Software Foundation: Python 3 Documentation.“ [Online]. Verfügbar unter: https://docs.python.org/3/license.html

[35] S. Cass, „The Top Programming Languages 2025“, Sep. 2025. [Online]. Verfügbar unter: https://spectrum.ieee.org/top-programming-languages-2025
[36] C. R. Harris, K. J. Millman, und et al, „Array programming with NumPy“, in *Nature*, 2020, S. 357–362. doi: 10.1038/s41586-020-2649-2.
[37] „ADAC Autokatalog“. [Online]. Verfügbar unter: https://www.adac.de/rund-ums-fahrzeug/autokatalog/marken-modelle/?sort=SORTING_DESC
[38] European Environment Agency, „Monitoring of CO2 emissions from passenger cars Regulation (EU) 2019/631“. [Online]. Verfügbar unter: https://www.eea.europa.eu/en/datahub/datahubitem-view/fa8b1229-3db6-495d-b18e-9c9b3267c02b
[39] United States Environmental Protection Agency, „Data on Cars used for Testing Fuel Economy“. [Online]. Verfügbar unter: https://www.epa.gov/compliance-and-fuel-economy-data/data-cars-used-testing-fuel-economy
[40] United States Environmental Protection Agency, „Dynamometer Drive Schedules“.
[41] *ISO/IEC 25010:2023 -Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE)*, 15. November 2023. [Online]. Verfügbar unter: https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-2:v1:en
[42] Bundesministerium für Digitales und Verkehr, *Straßenverkehrs-Ordnung (StVO)*. 2013.
[43] „GraphHopper Entwicklerdokumentation“. 22. Juli 2026. [Online]. Verfügbar unter: https://docs.graphhopper.com/openapi/section/authentication
[44] „Was ist ein API (Application Programming Interface)?“, SAP. [Online]. Verfügbar unter: https://www.sap.com/germany/resources/what-is-api
[45] J. Erickson, „What Is JSON?“, ORACLE. [Online]. Verfügbar unter: https://www.oracle.com/de/database/what-is-json/
[46] H. Nahrstedt, *Regelungstechnik in: Excel + VBA für Ingenieure*. Wiesbaden: SpringerVieweg, 2025.
[47] F. Gauterin und H.-J. Unrau, „Skript - Grundlagen der Fahrzeugtechnik I“. KIT, Karlsruhe, Wintersemester /2020 2019.
[48] M. Dehli, E. Doering, und H. Schedwill, *Grundlagen der Technischen Thermodynamik*, 11. Aufl. SpringerVieweg, 2025.
[49] „Untersuchungen zur Kraftübertragung zwischen Pkw-Reifen und Fahrbahn unter besonderer Berücksichtigung der Kraftschlusserkennung im rotierenden Rad.“, TH Darmstadt, Düsseldorf, 1993.
[50] A. Meister, *Iterative Verfahren. In: Numerik linearer Gleichungssysteme*. Wiesbaden: Springer Spektrum, 2025.
[51] G. Mühlberg, W. Hackmann, und K. Buzziol, „Highly Integrated Electric Powertrain“, in *ATZelektronik worldwide*, Bd. 12, 2017, S. 42–45. doi: https://doi.org/10.1007/s38314-017-0048-8.
[52] „Electronics Motor Reducer“, Wikipedia. [Online]. Verfügbar unter: https://de.wikipedia.org/w/index.php?title=Electronics_Motor_Reducer&oldid=266330734
[53] Dr. H. Paknia, „Skript Fahrdynamik - Kapitel RF: Reifen“. 9. April 2025.
[54] „ADAC Database: Opel Corsa-e Ultimate“. [Online]. Verfügbar unter: https://www.adac.de/rund-ums-fahrzeug/autokatalog/marken-modelle/opel/corsa/f/316973/#technische-daten

Abbildungsverzeichnis

Abbildung 1: v(t)-Diagramm NEFZ, Quelle: Tüv Süd 6

Abbildung 2: v(t)-Diagramm des WLTC der Klasse 3 mit den Phasen Low, Medium, High und Extra High. (eigene Darstellung) 7

Abbildung 3: Fahrwiderstände am Fahrzeug, Quelle: eigene Darstellung 9

Abbildung 4: Grundlegender Aufbau eines elektrischen Antriebsstrangs, mit Energiefluss (eigene Darstellung) 13

Abbildung 5: Prinzipieller Aufbau einer Permanentmagneterregten Synchronmaschine im Querschnitt (Quelle: [14], Bezeichnungen auf Deutsch geändert) 13

Abbildung 6: Vier-Quadranten-Betrieb einer elektrischen Maschine (eigene Darstellung) 14

Abbildung 7: Drehmoment- und Leistungsverlauf einer elektrischen Maschine über der Drehzahl (eigene Darstellung, Prinzipdarstellung) 15

Abbildung 8: Simuliertes Kennfeld einer PMSM zum Antrieb eines Elektroautos (Quelle:[15, S. 29]) 16

Abbildung 9: Beispielhaftes relationales Datenmodell mit zwei Relationen (eigene Darstellung) 17

Abbildung 10: Notationselemente und Kardinalitäten des Entity-Relationship-Modells (eigene Darstellung) 18

Abbildung 11: Gesamtarchitektur der Fahrsimulationsumgebung (eigene Darstellung) 27

Abbildung 12: Minimum-Prinzip der vorausschauenden Sollwertbildung (schematisch, eigene Darstellung) 29

Abbildung 13: Logisches Datenmodell des Gesamtsystems als ER-Diagramm (eigene Darstellung) 32

Abbildung 14: Methodik der Lastzyklusanalyse (schematisch, eigene Darstellung) 33

Abbildung 15: Programmablauf des implementierten Gesamtsystems (eigene Darstellung) 36

Abbildung 16: Streckenrepräsentation als Knoten-Segment-Modell 43

Abbildung 17: v-s-Diagramm einer Bremsung an einer roten Ampel (vorausschauende Sollwertbildung) 44

Abbildung 18: Struktur des Geschwindigkeitsregelkreises 45

Abbildung 19: Auswirkung des Verkehrsmodells auf das Geschwindigkeitsprofil anhand einer simulierten Beispielstrecke (eigene Darstellung) 46

Abbildung 20: Zyklusbasierte Fahrsimulation (WLTC Class 1)..... 48

Abbildung 21: Rollwiderstandsbeiwert (grafisch, Quelle:[44] S.2-7) 48

Abbildung 22: Bandbreite der auf ebenen, homogenen Fahrbahnen im öffentlichen Straßenverkehr gemessenen Reibwert-Schlupf-Kurven (Quelle: [46]) 49

Abbildung 23: Zugkraftdiagramm des implementierten Antriebsstrangmodells (Beispielfahrzeug: 6 Gang, 250Nm, 96kW)..... 51

Abbildung 24: Eckdaten-Kennfeld einer elektrischen Maschine (420Nm, 258kW, 16000 1/min) mit den Betriebspunkten einer Simulation auf einer Stadt-Landstraßen-Autobahn-Strecke 52

Abbildung 25: Exemplarisches Simulationsergebnis mit Ampelstopp..... 54

Abbildung 26: Grafische Benutzeroberfläche Fahrsimulationstool 56

Abbildung 27: Grafische Benutzeroberfläche Fahrsimulationstool 56

Abbildung 28: Auswahl Kartendarstellung in der Benutzeroberfläche 60

Abbildung 29: Vergleich von Simulation und der sieben Messfahrten 63

Abbildung 30: simuliertes Motorkennfeld und dessen Betriebspunkte (Opel Corsa F Electric)..... 67

Tabellenverzeichnis

Tabelle 1: Bauarten elektrischer Traktionsmaschinen (nach [15, Kap. 7.4, 7.5]) .. 14

Tabelle 2: Frei zugängliche Fahrsimulations- und Fahrzykluswerkzeuge im Vergleich 19

Tabelle 3: Vergleich verbreiteter Routing-Dienste 21

Tabelle 4: Funktionale Anforderungen an die Fahrsimulationsumgebung 25

Tabelle 5: Nichtfunktionale Anforderungen an die Fahrsimulationsumgebung 26

Tabelle 6: Parametergruppen der Fahrzeugdatenbank und ihre Modellwirkung .. 31

Tabelle 8: Schema der Fahrzeugdatenbank (vehicle_database.xlsx) 39

Tabelle 9: Beispielfahrzeuge der Fahrzeugdatenbank (angenommene Werte).... 39

Tabelle 10: Parametrierung der Fahrertypen 45

Tabelle 11: Parametrierung der Verkehrsstufen 46

Tabelle 12: Wichtige Werte des Ergebnis-DataFrames df_hist 53

Tabelle 13: Fahrzeug- und Simulationsparameter Opel Corsa F Electric 100kW, Baujahr: 2021 61

Tabelle 14: Energiebilanz der Simulation verglichen mit der realen Fahrtmessung63

Tabelle 15: Ergebnisse der Zyklussimulation - WLTC Klasse 3b 65

Tabelle 16: Ermittelte Lastgrößen aus dem Simulationslauf (Opel Corsa F Electric)66

Anhang

Anlage 1: Verwendete Python-Bibliotheken und ihr Einsatzzweck

Bibliothek	Einsatzzweck
numpy, pandas	Numerik und tabellarische Datenhaltung (Strecken- und Ergebnisdaten)
requests	HTTP-Anfragen an die GraphHopper-API
python-dotenv	Einlesen des API-Schlüssels aus der .env-Datei
osmnx	Abfrage von Ampelpositionen aus OpenStreetMap
vincenty, shapely	Geodätische Distanzberechnung und Geometrieoperationen
gpxpy	Einlesen von GPX-Routendateien (alternativer Routendienst)
matplotlib	Diagramme in GUI und PDF-Bericht, Kartenrendering
contextily	OpenStreetMap-Kartenkacheln als Hintergrund der Routenkarte
openpyxl	Lesen und Schreiben der Fahrzeugdatenbank (xlsx)
tqdm	Fortschrittsanzeige bei längeren Abfragen
Tkinter (Standardbibliothek)	Grafische Benutzeroberfläche

Anlage 2: Installationsanleitung der Fahrsimulationsumgebung

```
1. VS Code installieren

2. Python Add-on installieren

3. Workspace Ordner in eigenes Dateienverzeichnis kopieren

4. unten rechts die Python Version anklicken, dann wird oben in der Leiste "Create Virtual Environment" angezeigt.
   --> Create VE --> Venv --> Delete and Recreate (falls angezeigt) --> aktuelle Python Version auswählen
5. Befehl: "pip install -r requirements.txt" ausführen

6. Eigenen GraphHopper-API-Schlüssel anlegen (jeder Nutzer braucht seinen eigenen, Schlüssel werden nicht weitergegeben):
   --> auf https://www.graphhopper.com/dashboard/signup kostenlos registrieren (keine Kreditkarte nötig)
   --> im Dashboard unter "API Keys" einen neuen Schlüssel erstellen und kopieren
   --> der kostenlose Tarif umfasst 500 Credits pro Tag und ist auf nicht-kommerzielle Nutzung beschränkt;
       eine Routenabfrage dieses Programms kostet 1 Credit, das reicht für den normalen Gebrauch aus

7. Im Workspace-Ordner (dort wo driving_cycle_desktop.py liegt) eine Textdatei mit dem Namen ".env" anlegen
   und den Schlüssel in folgender Form eintragen:
       GRAPHHOPPER_API_KEY=hier_den_eigenen_Schluessel_einfuegen
   --> keine Anführungszeichen, keine Leerzeichen um das "=" herum
   --> die Datei wird beim Programmstart automatisch eingelesen, im Quelltext selbst steht kein Schlüssel
   --> die .env-Datei nicht mitgeben, wenn der Workspace-Ordner an andere weitergegeben wird

8. driving_cycle_desktop.py über den Play-Button oben rechts ausführen.


Wenn eine Fehlermeldung beim Ausführen erscheint, dann nochmal probieren. Manchmal ist der Routing-service ausgelastet.

Meldet das Programm einen fehlenden oder ungültigen Schlüssel, dann Schritt 6 und 7 prüfen:
Dateiname wirklich ".env" (nicht ".env.txt"), Datei im richtigen Ordner, Schlüssel vollständig kopiert.
Bei sehr vielen Abfragen an einem Tag kann das kostenlose Tageskontingent aufgebraucht sein; dann am
nächsten Tag erneut versuchen oder bereits gespeicherte Routen offline weiterverwenden.
Bei Login auf www.graphhopper.com ist der verbleibende Tagesbedarf prozentual dargestellt.
```

Institut für
DLR Fahrzeugkonzepte

DHBW
Duale Hochschule
Baden-Württemberg

Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe und diese Arbeit bei keiner anderen Prüfung mit gleichem oder vergleichbarem Inhalt vorgelegt habe und diese bislang nicht veröffentlich wurde. Des Weiteren versichere ich, dass die eingereichte elektronische Fassung mit der gedruckten Ausfertigung übereinstimmt. Ich habe bei der Erstellung der Arbeit KI-Werkzeuge zur Überprüfung der Rechtschreibung und Grammatik eingesetzt.

Stuttgart, den 31.08.2026



(Ort, Datum) (Unterschrift)