

# **Methodik zur effizienten Applikation automatisierter Fahrfunktionen**

Zur Erlangung des akademischen Grades eines

**Doktors der Ingenieurwissenschaften (Dr.-Ing.)**

von der KIT-Fakultät für Maschinenbau  
des Karlsruher Instituts für Technologie (KIT)

genehmigte

**DISSERTATION**

von

**M. Sc. Nicolas Fraikin**

geb. in Arnsberg

Tag der mündlichen Prüfung:

04.12.2023

Hauptreferent:

Prof. Dr. rer. nat. Frank Gauterin

Korreferent:

Prof. Dr.-Ing. Hans-Christian Reuss



# Kurzfassung

Komplexer werdende Fahrerassistenzsysteme, steigende Kundenanforderungen und eine wachsende Derivatevielfalt stellen die fahrzeugbasierte Applikation vor neue Herausforderungen. Um dem gestiegenen Ressourcenaufwand gerecht zu werden, wird im Rahmen dieser Arbeit eine Methodik vorgestellt, welche das konventionelle Verfahren mit Simulationsmodellen und Ansätzen zur virtuellen Komplexitätsreduktion sowie Optimierung ergänzt.

Die beschriebene Applikationsmethodik basiert auf einem neuartigen Fahrzeugmodell, welches kurze Feedback-Schleifen zur Absicherung manuell gefundener Bedatungen und simulative Parameterstudien mit vertretbarem Rechenaufwand ermöglicht. Durch den Aufbau als Kombination aus analytischem Einspurmodell und neuronalem Netz kann die Komplexität und damit Rechenzeit gering gehalten und gleichzeitig eine hohe Prädiktionsgüte erreicht werden. Um den Anteil fahrzeugbasierter Applikationen zu senken, wird darüber hinaus eine Methodik zur Komplexitätsreduktion des Parameterraums und Erhöhung des Systemverständnisses basierend auf Sensitivitätsanalysen vorgestellt. Mit Hilfe des dadurch verkleinerten Suchraums erfolgt im Anschluss eine simulative Optimierung und Bereitstellung von Anfangsparametrierungen mit hohem Reifegrad für weitergehende Applikationen im Fahrversuch. Neben der Vorstellung und Erprobung für das Problem geeigneter Optimierungsmethoden erfolgt eine Bewertung der Übertragbarkeit auf den Fahrversuch mit Robustheitsanalysen.

Die Methodik trägt dazu bei, den hohen Aufwand fahrzeugbasierter Applikationen zu senken und gleichzeitig den Prozess zu systematisieren. Entgegen vergleichbarer Ansätze wird der Fahrversuch für die Feinapplikation weiterhin mit einbezogen, aber mit automatisierten virtuellen Analysen unterstützt. Das hybride Fahrzeugmodell und die Bewertung der Robustheit gegenüber auf der Zielhardware veränderten Bedingungen tragen der Diskrepanz zwischen Realität und Simulation Rechnung.



# Abstract

Increasing complexity of future driver assistance systems, higher customer demands and growing vehicle portfolios of OEMs (original equipment manufacturers) cause new challenges for the vehicle based calibration. To reduce costs and resources a new calibration method is presented that extends the established process with simulation models and approaches for virtual complexity reductions and optimizations.

The herein described calibration method is based on a novel vehicle model that enables short feedback loops for continuously assessing manually tuned parameter sets. Moreover, it serves as an accurate and efficient simulation model for virtual parameter studies such as optimization and sensitivity analyses. By combining the analytical single track model with a neural network comparably small complexities and thus computation times as well as high prediction accuracies are possible. To further decrease the ratio of costly vehicle based parameterizations a method is described to reduce complexity of the search space and increase system understanding (such as influence information or interdependencies) based on sensitivity analyses. Using the therewith reduced parameter space, virtual optimizations with an improved convergence behaviour can be performed to provide initial parameterizations with high maturity levels for further calibrations on the target hardware. Therefore, not only different optimization algorithms are compared with respect to their applicability for the problem but also the transferability of virtually optimal solutions to the vehicle test is evaluated with robustness analyses.

The methodology contributes to reducing high efforts of vehicle based calibrations and systematizes the process. On the contrary to existing approaches vehicle tests are still considered for fine tunings but supported by automated virtual analyses. The hybrid vehicle model and the evaluation of robustnesses towards changing environmental conditions on the target hardware take the discrepancy between reality and simulation into account.



# Danksagung

Die vorliegende Dissertation entstand während meiner Tätigkeit als Doktorand bei der BMW Group im Bereich Entwicklung Automatisiertes Fahren im Rahmen einer Kooperation mit dem Lehrstuhl für Fahrzeugsystemtechnik (FAST) des Karlsruher Institut für Technologie (KIT).

Ein herzlicher Dank gilt meinem Doktorvater Herrn Prof. Dr. rer. nat. Frank Gauterin, nicht nur für die Unterstützung des Promotionsvorhabens und das mir entgegengebrachte Vertrauen, sondern auch für sein stetes Interesse an dem bearbeiteten Thema und den konstruktiven, engen Austausch während der Promotionszeit.

Ebenso bedanke ich mich bei Herrn Prof. Dr.-Ing. Hans-Christian Reuss für die Übernahme des Korreferats und bei Herrn Prof. Dr.-Ing. Frederik Zanger für die Übernahme des Prüfungsvorsitzes und das damit verbundene Interesse an der Arbeit.

Weiterhin möchte ich mich bei meinem fachlichen Betreuer bei der BMW Group Herr Dr.-Ing. Kilian Funk für sein konstruktives Feedback, den regen wissenschaftlichen Austausch und die Handlungsfreiheit bei der Erstellung der Arbeit bedanken. Mein Dank gilt außerdem meinem Vorgesetzten Herr Dr.-Ing. Peter Waldmann, der das Forschungsvorhaben ermöglicht und aktiv mitgestaltet hat. Bei meinen Kollegen während der Zeit als Doktorand bedanke ich mich herzlich für das kollegiale und aufgeschlossene Arbeitsklima sowie die konstruktive Unterstützung.

Mein Dank gilt ebenfalls Herrn Dr.-Ing. Michael Frey für die fachliche Betreuung der Arbeit seitens des Lehrstuhls und die vielen, sachlichen Diskussionen, welche einen entscheidenden Beitrag zum Gelingen der Arbeit geleistet haben.

Schließlich bedanke ich mich ganz herzlich bei meiner Frau Kristina für die beständige Unterstützung und meinen Kindern Nela und Viola für ihr Verständnis für die entgangene Zeit.





# Inhaltsverzeichnis

<b>Kurzfassung</b> . . . . .	<b>i</b>
<b>Abstract</b> . . . . .	<b>iii</b>
<b>Danksagung</b> . . . . .	<b>v</b>
<b>Abkürzungen und Symbole</b> . . . . .	<b>xi</b>
<b>1 Einleitung und Motivation</b> . . . . .	<b>1</b>
<b>2 Stand der Technik und Wissenschaft</b> . . . . .	<b>7</b>
2.1 Applikation automatisierter Fahrfunktionen . . . . .	7
2.1.1 Einordnung in den Entwicklungsprozess . . . . .	8
2.1.2 Konventionelle Applikation im Fahrversuch . . . . .	11
2.2 Bestehende Ansätze zur Optimierung des Applikationsprozesses von Fahrzeugsystemen . . . . .	16
2.2.1 Applikationsmethoden . . . . .	16
2.2.2 Virtualisierung der Wirkkette . . . . .	21
2.3 Komplexitätsreduktion hochdimensionaler Parameterräume . . . . .	25
2.3.1 Numerische Sensitivitätsanalyse . . . . .	25
2.3.2 Ermittlung valider Parameterwertebereiche . . . . .	33
2.4 Automatisierte Optimierung von Applikationsparametern . . . . .	34
2.4.1 Optimierungsproblem . . . . .	34
2.4.2 Optimierungsverfahren . . . . .	37
2.4.3 Robustheitsanalysen . . . . .	44
2.5 Fahrzeugmodelle . . . . .	46
2.5.1 Analytische Fahrzeugmodelle . . . . .	48
2.5.2 Neuronale Netze zur Modellierung der Wirkkette . . . . .	53

2.5.3	Kombinierte Ansätze . . . . .	56
2.6	Modelle zur Objektivierung des Fahrverhaltens . . . . .	58
<b>3</b>	<b>Forschungsbedarf . . . . .</b>	<b>61</b>
3.1	Ableitung von Anforderungen an eine Applikationsmethodik . . . . .	61
3.2	Bewertung der Grundlagen und des Stands der Wissenschaft . . . . .	64
3.3	Forschungsansatz . . . . .	67
3.3.1	Forschungsfragen . . . . .	67
3.3.2	Konzeptionelles Vorgehen . . . . .	69
3.3.3	Validierung der Methodik . . . . .	71
<b>4</b>	<b>Dreistufige Applikationsmethodik . . . . .</b>	<b>75</b>
4.1	Hybrides Fahrzeugmodell zur effizienten virtuellen Applikation . . . . .	75
4.1.1	Design des hybriden Fahrzeugmodells . . . . .	76
4.1.2	Trainingsschema . . . . .	78
4.2	Komplexitätsreduktion des Parameterraums . . . . .	85
4.2.1	Sampling . . . . .	85
4.2.2	Ableitung gültiger Wertebereiche . . . . .	87
4.2.3	Dimensionalitätsreduktion und Steigerung des Systemverständnisses . . . . .	92
4.2.4	Evaluierung der Komplexitätsreduktion . . . . .	93
4.3	Simulative Optimierung von Applikationsparametern . . . . .	94
4.3.1	Auswahl eines Optimierungsverfahrens . . . . .	95
4.3.2	Nutzung der Komplexitätsreduktion für eine effizientere Optimierung . . . . .	102
<b>5</b>	<b>Anwendung der Methodik für die Applikation einer automatisierten Fahrfunktion . . . . .</b>	<b>107</b>
5.1	Applikationsproblem . . . . .	107
5.2	Ergebnisse . . . . .	110
5.2.1	Komplexitätsreduktion . . . . .	110
5.2.2	Optimierung . . . . .	116
5.2.3	Robustheitsanalyse . . . . .	119
5.2.4	Rechenzeitbetrachtungen . . . . .	121
5.3	Diskussion der Ergebnisse . . . . .	123

---

<b>6 Validierung</b>	<b>127</b>
6.1 Validierungskonzept	127
6.2 Ergebnisse	133
6.2.1 Validierung des hybriden Fahrzeugmodells	133
6.2.2 Validierung der sensitivitätsbasierten Optimierung	137
6.2.3 Robustheitsanalyse gegenüber Fahrzeugmodellparametern	140
6.3 Fazit	141
<b>7 Bewertung und Diskussion der Methodik</b>	<b>147</b>
<b>8 Zusammenfassung und Ausblick</b>	<b>155</b>
8.1 Zusammenfassung der Arbeit	155
8.2 Ausblick	158
<b>A Anhang</b>	<b>161</b>
A.1 Dreistufige Applikationsmethodik	161
A.1.1 Hybrides Fahrzeugmodell	161
A.1.2 Simulative Optimierung von Applikationsparametern	162
A.2 Anwendung der Methodik für die Applikation einer automatisierten Fahrfunktion	163
A.2.1 Komplexitätsreduktion	163
A.2.2 Optimierung	166
A.2.3 Robustheitsanalyse	166
A.3 Validierung	168
A.3.1 Validierungskonzept	168
A.3.2 Ergebnisse	171
<b>Abbildungsverzeichnis</b>	<b>175</b>
<b>Tabellenverzeichnis</b>	<b>177</b>
<b>Literaturverzeichnis</b>	<b>179</b>
<b>Eigene Veröffentlichungen</b>	<b>195</b>



# Abkürzungen und Symbole

## Abkürzungen

<b>DoE</b>	Design of Experiments
<b>EEM</b>	Elementareffektmethode
<b>ESM</b>	Einspurmodell
<b>FAS</b>	Fahrerassistenzsysteme
<b>GA</b>	Genetischer Algorithmus
<b>HM</b>	Hybrides Fahrzeugmodell
<b>HiL</b>	Hardware-in-the-Loop
<b>ISAR</b>	Integrierte Simulationsumgebung Fahrdynamik Regelsysteme (kommerziell genutztes Mehrkörper-Simulationsmodell)
<b>KNN</b>	Künstliches neuronales Netz
<b>LSTM</b>	Long Short Term Memory
<b>MKS</b>	Mehrkörpersystem
<b>MiL</b>	Model-in-the-Loop
<b>OAT</b>	One-at-a-time
<b>OEM</b>	Original Equipment Manufacturer
<b>PSO</b>	Partikelschwarmoptimierung

<b>RMSE</b>	Root Mean Square Error
<b>SA</b>	Sensitivitätsanalyse
<b>SiL</b>	Software-in-the-Loop
<b>TDD</b>	Test Driven Development
<b>VBSA</b>	Varianzbasierte Sensitivitätsanalyse
<b>ViL</b>	Vehicle-in-the-Loop
<b>XiL</b>	X-in-the-Loop

## Symbole und Variablen

Die in der Arbeit verwendeten Symbole und Variablen sind im Folgenden kapitel- und abschnittsweise zusammengetragen. Die Sortierung ist alphabetisch.

### Stand der Technik und Wissenschaft

#### Komplexitätsreduktion hochdimensionaler Parameterräume

$A$	Sampling-Matrix zur Annäherung der Varianzen für die VBSA
$A_B^i$	Kombinierte Sampling-Matrix aus $A$ und $B$ zur Annäherung der Varianzen für die VBSA
$B$	Sampling-Matrix zur Annäherung der Varianzen für die VBSA
$E$	Erwartungswert der VBSA
$E_{P_{-i}}$	Erwartungswert für alle Parameter außer $P_i$ der VBSA

$EE$	Elementareffekt
$F(P)$	Zielfunktion
$N$	Sampling-Zahl zur Annäherung der Varianzen für die VBSA
$P_i$	Parameter $i$
$R_{KI,i}$	Breite des Konfidenzintervalls für die Bootstrapping-Konvergenzanalyse
$S_i$	Haupteffekt der VBSA
$S_{Ti}$	Totaleffekt der VBSA
$S_i^{OG}$	Obere Intervallgrenze des Konfidenzintervalls der Bootstrapping-Konvergenzanalyse
$S_i^{UG}$	Untere Intervallgrenze des Konfidenzintervalls der Bootstrapping-Konvergenzanalyse
$V$	Varianz der VBSA
$V_{P_i}$	Varianz für Parameter $P_i$ der VBSA
$V_{P_{\sim i}}$	Varianz für alle Parameter außer $P_i$ der VBSA
$n_B$	Anzahl an Sampling-Plänen für die Bootstrapping-Konvergenzanalyse
$n_{EEM}$	Anzahl benötigter Systemevaluationen für die Elementareffektmethode
$n_p$	Anzahl Parameter
$n_{VBSA}$	Anzahl an Samples für die VBSA
$r$	Anzahl Elementareffekte
$\kappa_{Konf}$	Konfidenzniveau für die Bootstrapping-Konvergenzanalyse
$\Delta$	Änderung des Parameters zur Auswertung der Elementareffekte

$\mu$	Mittelwert der Elementareffekte
$\mu^*$	Alternativer Mittelwert der Elementareffekte
$\sigma$	Standardabweichung der Elementareffekte

### **Automatisierte Optimierung von Applikationsparametern**

$DoR$	Degree of Robustness (Robustheitsgrad)
$\vec{G}_t$	Position des Individuums mit der global besten Fitness zum Zeitpunkt $t$
$H$	Anzahl an Samples in einer $\delta$ -Hyperbox für Robustheitsanalysen
$N$	Populationsgröße
$\vec{V}_t^n$	Geschwindigkeit des Individuums $n$ zum Zeitpunkt $t$
$\vec{X}_{t,best}^n$	Historisch beste Position des Individuums $n$ zum Zeitpunkt $t$
$\vec{X}_{t,best}^n$	Historisch beste Position des Individuums $n$ zum Zeitpunkt $t$
$\vec{X}_t^n$	Position des Individuums $n$ zum Zeitpunkt $t$
$c_1$	Gewichtung der Ausrichtung eines Partikels an der individuell besten Lösung
$c_2$	Gewichtung der Ausrichtung eines Partikels an der global besten Lösung
$f(x)$	Einkriterielle Gütefunktion für die Optimierung
$\vec{f}(x)$	Gütefunktions-Vektor für die Optimierung
$\vec{g}(\vec{x})$	Ungleichheitsbedingungen für die Optimierung
$\vec{h}(\vec{x})$	Gleichheitsbedingungen für die Optimierung



$k_i$	Anzahl an Bits für Parameter $i$
$n_{P_i}^{min}$	Minimale Anzahl möglicher Werte zur Diskretisierung des Wertebereichs von Parameter $i$
$p_m$	Mutationswahrscheinlichkeit
$p_r$	Rekombinationswahrscheinlichkeit
$p_s$	Selektionswahrscheinlichkeit
$\vec{r}_t^n$	Zufallsvektoren zur Gewichtung der Ausrichtung eines Partikels an der individuell besten Lösung
$\vec{s}_t^n$	Zufallsvektoren zur Gewichtung der Ausrichtung eines Partikels an der global besten Lösung
$w_i$	Gewichtungsfaktor für die Methode der gewichteten Summe
$x_S$	Untersuchter Parametersatz für Robustheitsanalysen
$\vec{x}$	Vektor der Entscheidungsvariablen für die Optimierung
$\delta$	Hyperboxradius im <i>decision space</i> für Robustheitsanalysen
$\delta_1$	Minimale Hyperboxgröße
$\delta_{max}$	Maximale Hyperboxgröße zur Erfüllung der Robustheitsbedingung
$\eta$	Maximale Schwelle im <i>objective space</i> zur Definition von Robustheitsbedingungen

### Analytische Fahrzeugmodelle

$F_{l,f}$	Lat. Kraft VA
$F_{l,r}$	Lat. Kraft HA
$R$	Kurvenradius

$SP$	Massenschwerpunkt
$a_y$	Lat. Beschleunigung
$l_f$	Abstand VA-SP
$l_r$	Abstand HA-SP
$m$	Fahrzeugmasse
$v$	Geschw. im SP
$v_f$	Geschw. an der VA
$v_r$	Geschw. an der HA
$\Psi$	Gierwinkel
$\alpha_f$	Schräglaufwinkel VA
$\alpha_r$	Schräglaufwinkel HA
$\beta$	Schwimmwinkel
$\delta_f$	Lenkwinkel VA
$\theta_z$	Gierträgheitsmoment

### **Neuronale Netze zur Modellierung der Wirkkette**

$U_c$	Gewichtsmatrix des Outputs an der Memory Cell
$U_f$	Gewichtsmatrix des Outputs am Forget-Gate
$U_i$	Gewichtsmatrix des Outputs am Input-Gate
$U_o$	Gewichtsmatrix des Outputs am Output-Gate
$W_c$	Gewichtsmatrix des Inputs an der Memory Cell
$W_f$	Gewichtsmatrix des Inputs am Forget-Gate
$W_i$	Gewichtsmatrix des Inputs am Input-Gate
$W_o$	Gewichtsmatrix des Inputs am Output-Gate

$\vec{b}_c$	Bias-Vektor an der Memory Cell
$\vec{b}_f$	Bias-Vektor am Forget-Gate
$\vec{b}_i$	Bias-Vektor am Input-Gate
$\vec{b}_o$	Bias-Vektor am Output-Gate
$\vec{c}_t$	zustands-Vektor der Memory Cell zum Zeitpunkt $t$
$d$	Anzahl an Input-Variablen zur LSTM-Zelle
$\vec{f}_t$	Aktivitätslevel-Vektor des Forget-Gates zum Zeitpunkt $t$
$h$	Anzahl an Neuronen der verdeckten Schicht
$\vec{i}_t$	Aktivitätslevel-Vektor des Input-Gates zum Zeitpunkt $t$
$\vec{o}_t$	Aktivitätslevel-Vektor des Output-Gates zum Zeitpunkt $t$
$\vec{x}_t$	Input-Vektor der LSTM-Zelle zum Zeitpunkt $t$
$\vec{y}_t$	Output-Vektor der LSTM-Zelle zum Zeitpunkt $t$
$\sigma_c$	Aktivierungsfunktion des LSTM-Inputs
$\sigma_g$	Aktivierungsfunktion der Gates
$\sigma_h$	Aktivierungsfunktion der Memory Cell

### Modelle zur Objektivierung des Fahrverhaltens

$x_1, \dots, x_k$	fahrdynamische Messgrößen in der Regressionsfunktion
$y$	Fahrkomfort der Objektivierungsfunktion
$\beta_0, \dots, \beta_k$	Regressionskoeffizienten
$\epsilon$	Fehlerterm in der Regressionsfunktion

## Dreistufige Applikationsmethodik

### Hybrides Fahrzeugmodell zur effizienten virtuellen Applikation

$e_{ISAR}$	Prädiktionsfehler zwischen hybridem Modell und ISAR-Modell
$e_{Real}$	Prädiktionsfehler zwischen hybridem Modell und realem Prozess
$f$	Frequenz der Sollkrümmung zur Erzeugung von Trainingsdaten
$\vec{u}(t)$	Eingangsvektor zum hybriden Modell
$\vec{u}_{ESM}$	Eingangsvektor des Einspurmodells
$v(t)$	Längsgeschwindigkeit
$y_{ESM}$	Ausgang des Einspurmodells nach dem Postprocessing
$y_{ESM}^*$	Ausgang des Einspurmodells vor dem Postprocessing
$y_{ISAR}$	Prädiktion des ISAR-Modells
$y_{Real}$	Ausgang des realen Prozesses
$\tilde{y}(t)$	Modellausgang des hybriden Modells
$\tilde{y}_{LSTM}$	Prädiktion des LSTM-Netzes
$\delta_{soll}$	Sollvorgabe des Radlenkwinkels
$\hat{k}_{soll}$	Krümmungsamplitude der Sollkrümmung
$\hat{a}_y$	Maximale Querbeschleunigung zur Erzeugung von Trainingsdaten
$\kappa_{ESM}$	Modellausgang des ESM-Modells
$\kappa_{ISAR}$	Modellausgang des ISAR-Modells

$\kappa_{soll}$	Sollvorgabe der Krümmung
$\tilde{\kappa}$	Prädizierter Modellausgang des hybriden Modells
$\tilde{\kappa}_{ESM}$	Prädizierter Modellausgang des Einspurmodells vom LSTM-Netz
$\tilde{\kappa}_{ISAR}$	Prädizierter Modellausgang des ISAR-Modells vom LSTM-Netz
$\tilde{\kappa}_{LSTM}$	Prädizierter Modellausgang des LSTM-Netzes (Delta-Dynamik)
$\sigma_{ISAR}$	Übertragungsfunktion des ISAR-Modells zur Umwandlung von $\kappa_{soll}$ zu $\delta_{soll}$

### Komplexitätsreduktion des Parameterraums

$DR$	Dimensionalitätsreduktion
$OG_i$	Konservativ abgeschätzte initiale obere Grenze für Parameter $i$
$OG_i^*(k)$	Lokale obere Grenze des Intervalls $k$ für Parameter $i$
$OG_i^{neu}$	Sensitive obere Grenze für Parameter $i$
$PRR$	Parameterraumreduktion
$P_i$	Parameter $i$
$UG_i$	Konservativ abgeschätzte initiale untere Grenze für Parameter $i$
$UG_i^*(k)$	Lokale untere Grenze des Intervalls $k$ für Parameter $i$
$UG_i^{neu}$	Sensitive untere Grenze für Parameter $i$
$X_k^i$	Versuchsplan zur lokalen Analyse der Intervalle $k$ von Parameter $P_i$

$k_o$	Intervallindex für die Bestimmung der oberen Grenze
$k_u$	Intervallindex für die Bestimmung der unteren Grenze
$k_{ges}$	Vektor der die Indizes aller zur Auswertung berücksichtigten Intervalle enthält
$n_p$	Anzahl an Parametern
$n_{EEM}$	Anzahl an Samples der Elementareffektmethode zur Dimensionalitätsreduktion
$n_{EEM}^*$	Anzahl an Samples der Elementareffektmethode pro Intervall
$n_{Int}$	Anzahl an Intervallen zur Einteilung des Wertebereichs
$n_p^*$	Anzahl an Parametern, die als einflussreich bewertet wurden
$s_i$	Kombinierte Sensitivität des Parameters $i$
$s_{i,k_{ges}}^{rel}$	Relative Sensitivität des Parameters $i$ in den Intervallen $k_{ges}$
$s_i^{rel}$	Relative Sensitivität des Parameters $i$
$s_{min,WB}^{rel}$	Minimale Relevanzschwelle zur Bewertung von Intervallen als nicht sensitiv
$s_{rel}^{min}$	Minimale Relevanzschwelle zur Bewertung von Parametern als nicht sensitiv

### **Simulative Optimierung von Applikationsparametern**

$A$	Amplitude der Querablage als Parameter der Zielfunktion zum Vergleich von PSO und GA
-----	--

<i>Check_Dataset</i>	Boolescher Vektor der Auskunft darüber gibt ob eine Ersetzung von Szenariensimulationen mit einer Referenz möglich ist
<i>DB</i>	Datenbank
$F(f, A, v)$	Zielfunktion der Optimierung zum Vergleich von PSO und GA
<i>Non_influential_parameters</i>	Boolescher Vektor der Auskunft über die Relevanz eines Parameters gibt
<i>Non_influential_regions</i>	Boolescher Vektor der Auskunft darüber gibt ob ein Parameter im nicht-sensitiven Bereich liegt
$PB_{sens}$	Vektor der die sensitiven Ober- und Untergrenzen aller Parameter enthält
<i>PS</i>	Population die vom Optimierer erzeugt wird
$P_{sens}$	Boolescher Vektor der Auskunft über die Relevanz eines Parameters gibt
<i>Scenarios</i>	Szenarienkatalog
$\beta_0, \dots, \beta_3$	Regressionskoeffizienten der Zielfunktion zum Vergleich von PSO und GA
<i>db</i>	Parametersatz aus der Datenbank
<i>f</i>	Frequenz der Querablage als Parameter der Zielfunktion zum Vergleich von PSO und GA
<i>ps</i>	Parametersatz der Population
<i>s</i>	Szenario aus dem Szenarienkatalog <i>Scenarios</i>
<i>v</i>	Längsgeschwindigkeit als Parameter der Zielfunktion zum Vergleich von PSO und GA

## Anwendung der Methodik für die Applikation einer automatisierten Fahrfunktion

### Applikationsproblem

$UG_i$	description
$f(x)$	Zusammengesetzte Zielfunktion
$w_i$	Gewichtungsfaktor zur Gewichtung der einzelnen Zielfunktion mit Hilfe der Methode der gewichteten Summe

### Diskussion der Ergebnisse

$C$	Kosten zur Auswahl einer optimalen Trajektorie
$J$	Querruck
$T$	Manöverdauer
$d$	Laterale Spurbabweichung
$k_J$	Gewichtungsfaktor des Querrucks
$k_T$	Gewichtungsfaktor der Manöverdauer
$k_d$	Gewichtungsfaktor der lateralen Spurbabweichung

### Validierung

#### Validierungskonzept

$R_{krit}$	Anteil kritischer Parametersätze
$n_{Effekte}$	Anzahl relevanter Effekte (Haupteffekte und Wechselwirkungen) die für die Validierung berücksichtigt werden müssen



$n_{Init}$	Größe des initialen Versuchsplans an Basis-Samples pro Effekt
$n_{KR}$	Anzahl an Parametern dessen Wertebereich von der Komplexitätsreduktion verkleinert wurde oder die gänzlich ausgeschlossen wurden
$n_{Validierung}$	Anzahl an Samples pro Szenario für die Validierungsstudie
$n_{Var}$	Anzahl an Samples welche sich vom jeweiligen Basis-Sample nur innerhalb nicht sensitiver Bereiche unterscheiden
$n_{\Delta f(x) \leq 1}$	Anzahl unkritischer Parametersätze bei denen die Abweichung des Zielfunktionswerts immer kleiner als 1 ist
$n_{\Delta f(x) > 1}$	Anzahl kritischer Parametersätze bei denen die Abweichung des Zielfunktionswerts größer als 1 ist



# 1 Einleitung und Motivation

Seit Beginn der Markteinführung erster Fahrerassistenzsysteme (FAS) für Kraftfahrzeuge ist die Vielfalt und Funktionalität angebotener Fahrfunktionen stetig gestiegen. Während die zu Anfang entwickelten Systeme den Fahrer lediglich warnen konnten (z.B. Spurverlassenswarner, Totwinkelassistent), sind heutige FAS in der Lage die Längs- (z.B. Adaptive Cruise Control) und Querverführung (z.B. Spurhalteassistent, Stauassistent) vollständig zu übernehmen und werden von den meisten Automobilherstellern als Serien- oder Sonderausstattung angeboten. Auch wenn die Fahraufgabe theoretisch komplett von Assistenzsystemen übernommen werden könnte, ist der Fahrer für die Mehrheit heute angebotener FAS weiterhin zu jeder Zeit alleinig verantwortlich für die Fahrzeugführung und muss das Verkehrsgeschehen ständig überwachen. Entsprechend der SAE-Norm zur Klassifizierung und Definition autonomer Systeme im Straßenverkehr sind diese Fahrfunktionen den Stufen null bis zwei zuzuordnen [C<sup>+</sup>14].

Abbildung 1.1 stellt die Markteinführung wichtiger FAS-Vertreter der letzten Jahre zusammen und gibt einen Überblick zu Prognosen zukünftig angebotener Systeme. Der Grafik ist zu entnehmen, dass die nächste Entwicklungsstufe in dem Bereich die Ausstattung von Fahrzeugen mit Level-3-Systemen ist. Der damit verbundene Technologiesprung ist entscheidend, da dem Fahrer somit erstmals die Möglichkeit gegeben wird, sich vom Verkehrsgeschehen abzuwenden und einer Nebentätigkeit nachzugehen. Rund 60 % der Kunden sehen darin einen Komfortgewinn und würden sogar bis zu 20 % mehr für ein entsprechend ausgestattetes Fahrzeug ausgeben [Ins19]. Die vollständige Übernahme der Verantwortung durch Assistenzsysteme löst bei dem Großteil der Bevölkerung hingegen noch Sicherheitsbedenken aus [GVRP17]. Die Sicherstellung einer hohen Zuverlässigkeit auch in kritischen Situationen stellt auch für Automobilhersteller die größte Herausforderung beim Übergang zu Level drei dar, was sich in einer Verzehnfachung der weltweiten Ausgaben zwischen 2014 und 2018 im Gebiet Software und Entwicklung autonomer

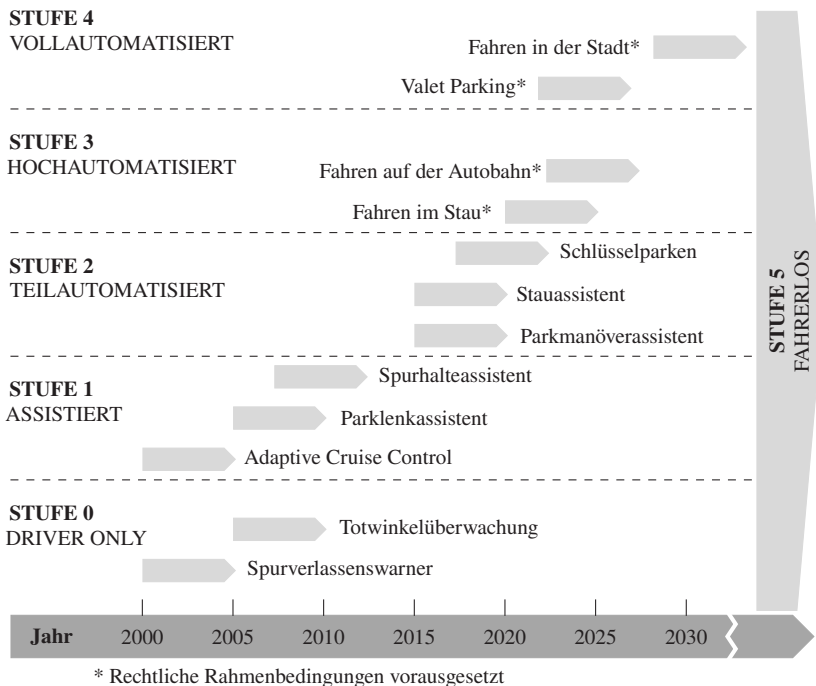


Abbildung 1.1: Vergangene und prognostizierte Markteinführung von Fahrerassistenzsystemen [VDA15].

Systeme widerspiegelt [Ahl19]. Die erhöhte Investitionssumme ist aber auch dadurch zu erklären, dass neben konventionellen Automobilherstellern neue Wettbewerber in den Markt drängen und den Konkurrenzdruck damit steigern. Um sich in dem Segment langfristig zu etablieren, sind neue Prozesse nötig die trotz gesteigener Anforderungen eine Entwicklung kundenwerter und verlässlicher Assistenzsysteme mit minimalem Ressourcen- und Zeitaufwand ermöglichen.

Auf dem Weg zur Serienreife automatisierter Fahrfunktionen spielt die Applikation entwickelter Systeme eine entscheidende Rolle. Sie stellt sicher, dass ein komfortables und sicheres Fahrverhalten in allen relevanten Szenarien und Fahrzeugen erreicht wird. Die Stellhebel dazu sind Applikationsparameter,

---

welche während der Funktionsentwicklung vorgesehen und typischerweise im Fahrversuch entsprechend der Anforderungen eingestellt werden. Die manuell durchgeführte Optimierung wird mit zunehmenden Komplexitätsgrad aufwändiger. Während Level-2-Systeme in den Anfängen nur für dedizierte Fahrsituationen ausgelegt werden mussten, umfassen Level-3-Systeme einen deutlich größeren Funktionsumfang. Die damit einhergehende Funktionsvielfalt und gestiegene Anzahl freier Parameter sorgen für einen hohen Ressourcenaufwand in der fahrzeuggestützten Applikation. Es müssen Prototypenfahrzeuge gefertigt, Arbeitszeiten von Ingenieuren gebunden und mehrwöchige Erprobungen finanziert werden. Auch die Bestrebungen vieler Fahrzeughersteller, eine breite Produktpalette aufzubauen, erschwert den konventionellen Applikationsprozess zusätzlich.

Um in einem kompetitiven Umfeld weiterhin hohe Applikationsgüten in vertretbarer Zeit zu erreichen, ist vor dem Hintergrund der aktuellen Herausforderungen im Entwicklungs- und Applikationsprozess automatisierter Fahrfunktionen eine Weiterentwicklung existierender Verfahren notwendig. Aufgrund des exponentiellen Wachstums der Rechenleistung von Computerchips [Moo98] entstehen immer performantere Simulationsmodelle, welche die Realität genau abbilden können. Das führt dazu, dass sich im Bereich der Fahrzeugtechnik vermehrt virtuelle Entwicklungsmethoden durchsetzen, welche das Potential haben, Kosten- und Zeitaufwände erheblich zu reduzieren. Aus dem Grund wird im Rahmen dieser Arbeit die Nutzung von Simulationsumgebungen zur Optimierung des konventionellen Applikationsprozesses von FAS analysiert. Das Ziel ist eine Weiterentwicklung des fahrzeuggestützten Parametrierungsprozesses, sodass eine Effizienzerhöhung ohne Einbußen des Applikationsergebnisses erreicht wird. Neben der Entwicklung eines geeigneten Simulationsmodells steht als weiterer Aspekt dieser Arbeit die virtuelle Komplexitätsreduktion des Parameterraums und Optimierung im Vordergrund. Somit soll die manuelle Parametereinstellung unterstützt werden, indem Informationen über Parametereinflüsse, -wechselwirkungen und virtuell optimierte Bedatungen zur Verfügung gestellt werden. Die Gültigkeit der theoretisch erarbeiteten Konzepte wird mit umfangreichen Anwendungsbeispielen untermauert. Dabei steht neben einer Effizienzsteigerung das Zusammenspiel zwischen Simulation und Fahrversuch sowie insbesondere die Übertragbarkeit virtuell erzielter Erkenntnisse auf reale Umgebungsbedingungen im Vordergrund.

Der schematische Aufbau der Arbeit kann Abbildung 1.2 entnommen werden. In Kapitel 2 erfolgt zunächst eine Einführung in den konventionellen Applikationsprozess (Kap. 2.1) sowie der damit verbundenen Herausforderungen. Außerdem werden bestehende Ansätze aus dem Bereich der Fahrwerks- und Antriebsstrangapplikation aufgezeigt, die das Potential besitzen die beschriebenen Probleme zu adressieren (Kap. 2.2). In den Kapiteln 2.3 - 2.6 wird der aktuelle Stand der Wissenschaft zur virtuellen Parameteroptimierung, Sensitivitätsanalysen und Fahrzeugmodellen präsentiert. Die Inhalte bilden gleichzeitig die technischen Grundlagen für die im Rahmen der Arbeit entwickelte Applikationsmethodik.

Basierend auf dem in Kapitel 2.1 und 2.2 beschriebenen Stand der Technik zu Applikationsverfahren wird in Kapitel 3.1 der zentrale Handlungsbedarf abgeleitet, indem Anforderungen an das bereitzustellende Konzept formuliert werden. Basierend auf dem in Kapitel 2 ebenfalls beschriebenen Stand der Wissenschaft erfolgt eine Bewertung bestehender Ansätze im Hinblick auf deren Eignung zur Erfüllung der Anforderungen (Kapitel 3.2). Aus den Analysen der Abschnitte 3.1 und 3.2 lässt sich in Kapitel 3.3 der Forschungsbedarf und -beitrag ableiten. Mit Hilfe von drei Forschungsfragen erfolgt eine erste Konzeptionierung der zu entwickelnden Applikationsmethodik und deren Validierung.

Die drei Forschungsschwerpunkte werden in Kapitel 4 adressiert. Das hybride Fahrzeugmodell zur effizienten, hochgenauen Simulation (Kap. 4.1) als auch Methoden zur Komplexitätsreduktion des Parameterraums (Kap. 4.2) und Optimierung (Kap. 4.3) werden vorgestellt. Die Eignung der weitgehend theoretisch hergeleiteten Ansätze wird in Kapitel 5 geprüft, indem eine beispielhafte Level-3-Fahrfunktion automatisiert appliziert wird. Der Fokus der Untersuchungen in dem Kapitel liegt darauf, die Effizienzsteigerung nachzuweisen. Im sich anschließenden Kapitel 6 erfolgt eine Validierung der Methodik mit dem Ziel die Gültigkeit der eingesetzten Verfahren zu prüfen und neben der Effizienz auch die Güte der Ergebnisse zu belegen. Dafür kommt eine umfangreiche Parameterstudie zum Einsatz. Des weiteren werden die Übertragbarkeit virtuell gefundener Lösungen auf den Fahrversuch und der Prädiktionsfehler des hybriden Fahrzeugmodells analysiert.

Basierend auf den Erkenntnissen der Kapitel 4 bis 6 erfolgt in Kapitel 7 eine Rückführung zum Forschungsbedarf aus Kapitel 3. Es wird im Speziellen diskutiert, ob die Methoden einen Beitrag zur Beantwortung der drei

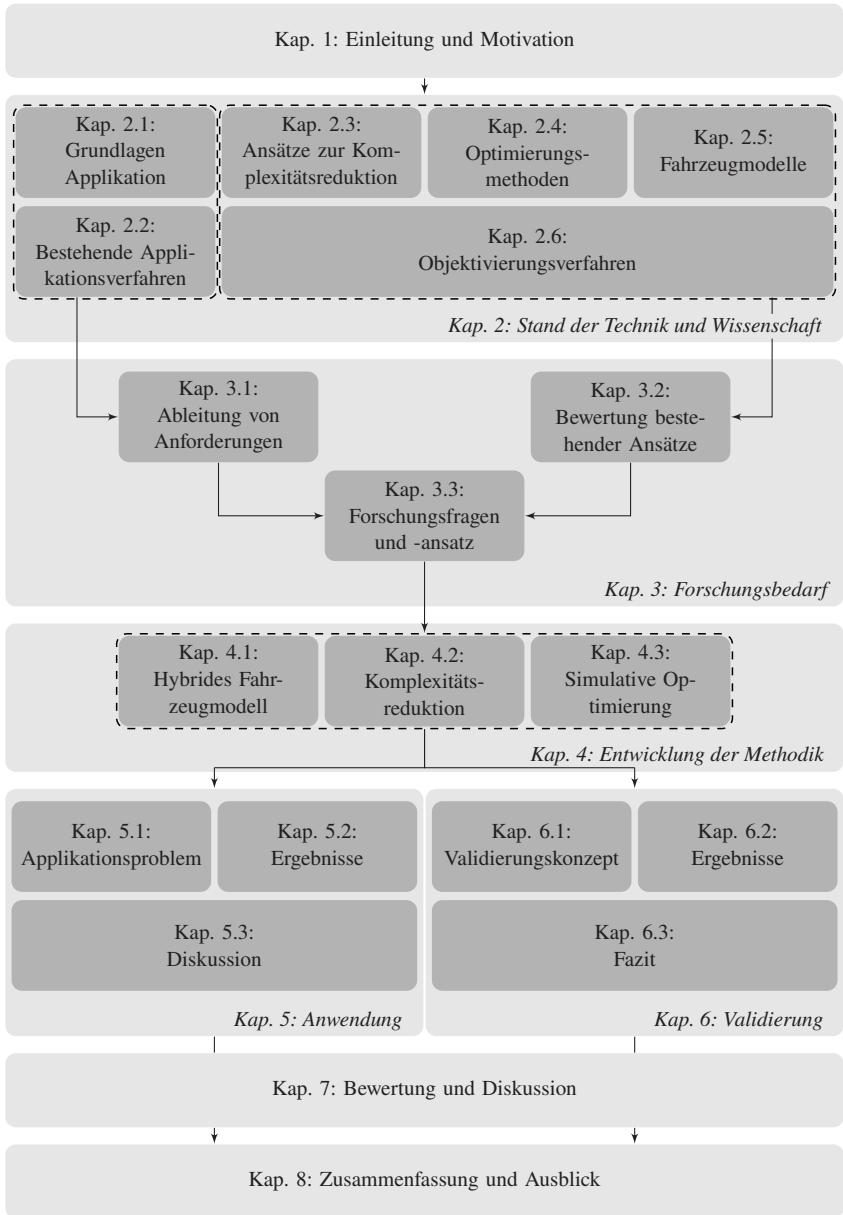


Abbildung 1.2: Aufbau der Arbeit.

Forschungsfragen leisten können und inwiefern eine Übertragbarkeit der Konzepte auf vergleichbare Probleme möglich ist. Kapitel 8 schließt mit einer Zusammenfassung und Ausblick auf zukünftige Forschungsbemühungen.

Die Forschungsgemeinschaft profitiert von einer schnellen Veröffentlichung neuer Erkenntnisse. Aus dem Grund wurden Teile dieser Arbeit vom Autor bereits vorveröffentlicht. Das in Abschnitt 4.1 beschriebene Konzept des hybriden Fahrzeugmodells sowie dessen Validierung wurde erstmals in [FFFG19b] publiziert. Die in Abschnitt 4.2 eingeführte Methode zur Komplexitätsreduktion wurde in der Vorveröffentlichung [FFFG19a] erstmals thematisiert. Das in Abschnitt 4.3 beschriebene Verfahren zur effizienten, simulativen Optimierung von Applikationsparametern von Fahrerassistenzsystemen wurde zuvor in den Veröffentlichungen [FFFG20] und [Fra21] vorgestellt. Die in Kapitel 5 und 6 beschriebene Anwendung sowie Validierung des Verfahrens anhand eines exemplarischen Applikationsproblems sind zum Teil in [FFFG20] vorgestellt worden. Im Rahmen dieser Arbeit werden die zuvor veröffentlichten Ansätze zu einer ganzheitlichen Methodik zusammengeführt. Die Teile der Arbeit, welche auf bereits publizierte Sachverhalte referenzieren sind mit einem entsprechenden Verweis auf die relevante Quelle versehen.



## **2 Stand der Technik und Wissenschaft**

Im Folgenden werden die Grundlagen für die Entwicklung der in dieser Arbeit vorgestellten Methodik zur Applikation automatisierter Fahrfunktionen gelegt. Der erste Abschnitt enthält eine Prozessbeschreibung der konventionellen Applikation von FAS. Dabei werden etablierte Vorgehensweisen der fahrzeugbasierten Applikation vorgestellt und diese in den Entwicklungsprozess des Gesamtfahrzeugs eingeordnet. Der aktuelle Stand der Wissenschaft bestehender Applikationsmethoden und Konzepte zur Optimierung des Applikationsprozesses werden im sich anschließenden Abschnitt 2.2 beschrieben. In den Abschnitten 2.3 und 2.4 erfolgt eine Einführung in Ansätze der Komplexitätsreduktion hochdimensionaler Parameterräume und gängige Optimierungsverfahren. Die im Folgenden thematisierten Fahrzeugmodelle unterschiedlicher Komplexität und Modellgüte stellen die Grundlage dar, um eine simulationsbasierte Applikation erst zu ermöglichen (Abschn. 2.5). Da die in Kapitel 4 zu entwickelnde Methodik sowohl eine Optimierung des Parametrierungsprozesses als auch eine Verbesserung der Simulationsumgebung vorsieht, bilden diese Abschnitte (Abschn. 2.3 - 2.5) die theoretischen Grundlagen. Im letzten Teil dieses Kapitels werden Modelle zur Objektivierung des subjektiven Fahrverhaltens beschrieben, die als Gütekriterien für eine modellbasierte Optimierung verwendet werden können (Abschn. 2.6).

### **2.1 Applikation automatisierter Fahrfunktionen**

Für die Entwicklung einer ganzheitlichen Applikationsmethodik zur effizienten Parametrierung automatisierter Fahrfunktionen ist es notwendig, den etablierten Applikationsprozess zu untersuchen und deren Schwachstellen zu identifizieren. Die Kalibrierung von Fahrerassistenzfunktionen im Fahrzeug stellt den aktuellen Stand der Technik dar und soll deshalb ausführlich beschrieben

werden. Dabei wird insbesondere auf Ziele der Applikation, die Wirkkette automatisierter Fahrfunktionen und eingesetzte Ressourcen eingegangen. Für ein besseres Gesamtverständnis wird im Vorhinein der Prozessschritt der Systemapplikation in den Gesamtkontext der Entwicklung von Fahrzeugsystemen eingeordnet.

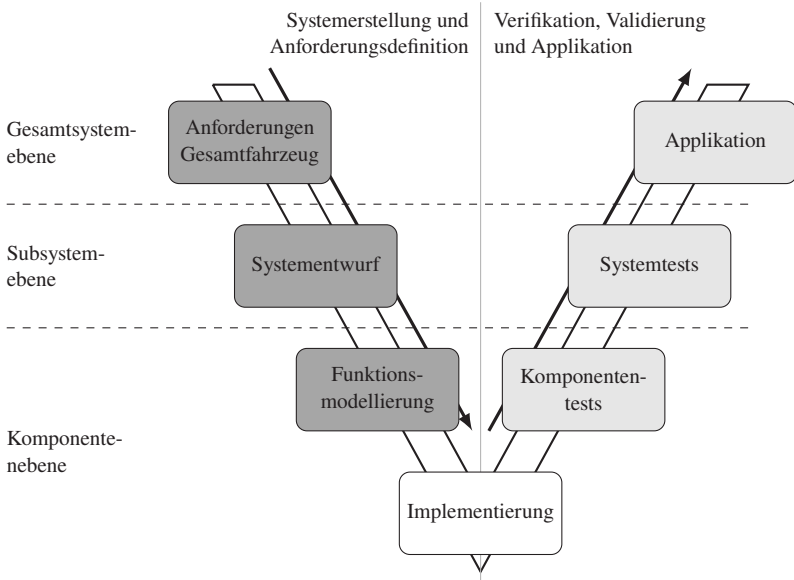
## 2.1.1 Einordnung in den Entwicklungsprozess

Die Entwicklung von Fahrzeugen hat in den letzten Jahren bedeutend an Komplexität gewonnen. Steigende Kundenanforderungen, die stetig wachsende Anzahl angebotener Derivate und kürzere Entwicklungszyklen erfordern eine strukturierte und transparente Ablaufplanung. Anhand dieser muss es möglich sein, Kundenanforderungen an das Gesamtprodukt in unterschiedliche Abstraktionsebenen herunterzubrechen. In der Fahrzeugentwicklung erfolgt typischerweise eine Unterteilung in Komponenten und Subsysteme, die von verschiedenen Entwicklerteams verantwortet werden. Auf diese Weise können die hohen Anforderungen auf Gesamtsystem-Ebene und die Komplexität des Produkts in Einklang gebracht werden [WPM18].

Ziele, die auf oberster Ebene definiert sind, richten sich nach der Positionierung des Produkts im Markt (z.B. Fahrdynamik, Fahrkomfort, Effizienz). Auf Subsystem-Ebene erfolgt eine Detaillierung dieser Ziele anhand fahrdynamischer Kennwerte, die in standardisierten Testszenarien eingehalten werden müssen. Die detailliertesten Eigenschaften werden auf Komponenten-Ebene definiert. Hierbei handelt es sich meist um bauteilspezifische Charakteristika wie beispielsweise Achssteifigkeiten oder Reifeneigenschaften.

Um eine Durchgängigkeit der Anforderungen bei der Partitionierung von Zieleigenschaften auf die unterschiedlichen Hierarchieebenen zu gewährleisten, existieren definierte Vorgehensweisen, die im weit verbreiteten V-Modell beschrieben werden (vgl. Abbildung 2.1). Aus dem Militärbereich kommend, ist es heute das etablierte Entwicklungsprinzip für sicherheits- und qualitätskritische Produkte im Automobilbereich und wird vielfach zur Softwareerstellung für Steuergeräte übernommen [BBB<sup>+</sup>06, SR08].

Das V-Modell beinhaltet einen Top-Down-Pfad auf der linken Seite, auf dem die Phasen der Systemerstellung von der Anforderungsdefinition auf den unterschiedlichen Hierarchieebenen bis hin zur Funktionsmodellierung und

Abbildung 2.1: V-Modell für die Entwicklung von Fahrzeugsystemen nach [WPM18, BBB<sup>+</sup>06]

-implementierung stattfinden. Auf dem Bottom-Up-Pfad erfolgt die Verifikation und Validierung mit Hilfe von Tests auf den jeweiligen Ebenen bis hin zur Applikation gegenüber den Kundenanforderungen und der finalen Systemabnahme.

Die Definition der Anforderungen an das Gesamtfahrzeug erfolgt als erster Schritt der Produktentwicklung. Im Bereich des automatisierten Fahrens sind dies typischerweise Sicherheitsanforderungen, Anforderungen an Verlässlichkeit oder Komfortkriterien [WPM18]. Daraus lassen sich wiederum Parameter und Grenzwerte für den Entwurf und die Modellierung der Einzelkomponenten ableiten. In den folgenden Phasen erfolgt die Definition einer Softwarearchitektur sowie der modellbasierte Entwurf aller Einzelkomponenten. Die Umsetzung (Implementierung) erfolgt meist in einer Rapid Prototyping Umgebung mit Hilfe von Tools und Frameworks, die ein vereinfachtes Debugging und Design zum Beispiel anhand von Signalfuss- und Blockdiagrammen ermöglichen [Hen18]. Dabei ist es unerlässlich, standardisierte Modellierungsrichtli-

nien und Vorgaben zur Softwarearchitektur einzuhalten um eine problemlose Portierung der Software aus der Prototyping-Umgebung auf die Zielhardware zu ermöglichen (z.B. der AUTOSAR-Standard [FMB<sup>+</sup>09]). Diese Portierung erfolgt im Implementierungs-Schritt und wird durch Codegeneratoren unterstützt, die für das Steuergerät lauffzeit- und prozessoroptimierten Softwarecode zur Verfügung stellen.

Die sich anschließende Verifikation und Validierung der Anforderungen erfolgt nach dem Bottom-Up-Prinzip angefangen mit der Softwarekomponente der Fahrfunktion als kleinster Einheit. Auf Basis der Komponententests werden im Anschluss System-Integrationstests durchgeführt, welche zum ersten Mal Steuergeräte-Software und Teile der Zielhardware beinhalten. Diese Tests werden zunächst nur unter Einbeziehung realer Steuergeräte (Hardware-in-the-Loop, vgl. Abschnitt 2.2.2) und im Anschluss auch mit Prototypenfahrzeugen durchgeführt [WHLS15]. Zur Erfüllung der Anforderungen auf höchster Ebene gewinnt die Applikation an Bedeutung, da somit die Ausprägung der Fahrfunktion und die Erstellung varianten- und länderspezifischer Parametrierungen möglich ist. Die Applikation, welche normalerweise im Fahrversuch erfolgt, gliedert sich somit an die System-Integrationstests an und stellt die letzte Instanz vor der finalen Systemabnahme dar.

Während sich das V-Modell in vielen Automobilkonzernen etabliert hat, gewinnen agile Arbeitsmethoden insbesondere bei der Entwicklung von automatisierten Fahrfunktionen zunehmend an Bedeutung (vgl. [Dor18, HSMS18]). Ursprünglich aus der Softwareentwicklung kommend, werden im agilen Arbeitsmodell häufige kleine Lieferungen wertschöpfender Produkte angestrebt. Somit wird das primäre Ziel verfolgt, Produkthanforderungen dynamisch anpassen zu können. Im Vergleich zum klassischen V-Modell kann somit flexibler auf sich ändernde Umgebungsbedingungen (z.B. wechselnde Kundenanforderungen, Technologiefortschritt, finanzielle Rahmenbedingungen) reagiert werden. Das Endprodukt wird inkrementell statt sequentiell aufgebaut [BBVB<sup>+</sup>01]. Die Softwareinkremente werden dabei von cross-funktional zusammengestellten Teams in kurzen Entwicklungszyklen (sogenannten Sprints) innerhalb von zwei bis vier Wochen erzeugt, in denen jeweils alle Phasen des V-Modells durchlaufen werden. Der entscheidende Unterschied besteht jedoch neben der Größe des zu entwickelnden Produkts darin, dass Tests nicht zur Verifikation der Anforderungen dienen sondern als Entwicklungswerkzeug eingesetzt werden. Das sogenannte Test Driven Development (TDD) sieht die Implemen-

tierung entsprechender Testfälle vor der Erstellung von Funktions-Code vor und ermöglicht somit ein kontinuierliches Testen von Anforderungen während der Softwareentwicklung [Bec03]. Da die Änderungskosten bei reiner Softwareentwicklung meist geringer sind als bei Hardwareabhängigkeiten, erfordert eine Umstellung auf agile Arbeitsmethoden in der Fahrzeugtechnik hochgenaue Simulationstools, um das Zielsystem bereits in den frühen Phasen der Entwicklung realitätsgetreu abbilden und die Integration neuer Features testen zu können [CRW10]. Um dem Anspruch der inkrementellen Erzeugung wertschöpfender Features für automatisierte Fahrfunktionen gerecht zu werden, muss auch die Applikation der jeweiligen Komponente während eines Entwicklungszyklus erfolgen. Die konventionelle Parametrierungsmethodik am Fahrzeug wäre innerhalb eines Sprints zu ressourcenaufwändig, sodass detaillierte Simulationsmodelle und automatisierte Applikationsverfahren für diesen Prozessschritt an Bedeutung gewinnen.

### **2.1.2 Konventionelle Applikation im Fahrversuch**

Die Applikation stellt im Entwicklungsprozess die letzte Möglichkeit vor der Auslieferung dar, um die automatisierten Fahrfunktionen entsprechend der Kundenanforderungen auszuprägen. Dabei können unterschiedliche Parametersätze in Abhängigkeit von Ländern und Fahrzeugderivaten entstehen. Die entsprechenden Kalibrierparameter liegen typischerweise in Form von Kennwerten, Kennlinien und Kennfeldern vor. Die Suche nach einer geeigneten Parameterkombination kann als iterativer Optimierungsprozess aufgefasst werden, welcher aufgrund der Abhängigkeit vom Fahrversuch sehr ressourcenaufwändig werden kann. Das etablierte Vorgehen ist in Abbildung 2.2 dargestellt.

Softwaremodule die gesamthaft die Funktionalität der automatisierten Fahrfunktion abbilden, werden vom Softwareentwickler mit einstellbaren Parametern versehen. Die zugehörige Wertebelegung wird im ersten Schritt auf Grundlage von Erfahrungswerten oder simulativen Tests definiert und stellen eine grundsätzliche Fahrbarkeit sicher. Sollte der Applikateur Parameteränderungen durchführen, werden die entsprechenden Werte in der Software aktualisiert, während unveränderte Werte weiterhin dem Initialsatz entsprechen. Als „Flashen“ wird das Überschreiben bestimmter Speicherstellen auf dem Steuergerät bezeichnet, welche in diesem Fall die Parameterwerte beinhalten. Eine Erlebbarkeit der Fahrfunktionen ist in den frühen Phasen der Entwicklung

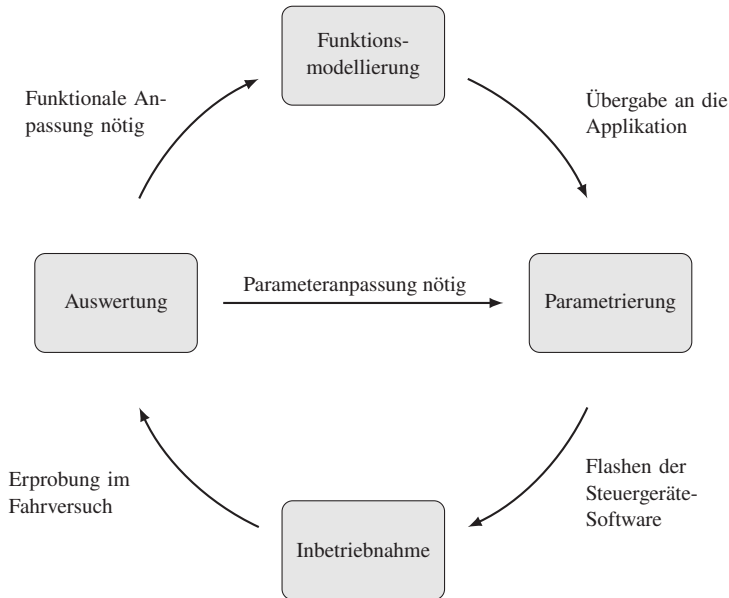


Abbildung 2.2: Applikationsprozess für automatisierte Fahrfunktionen nach [Hen15]

mit Hilfe von Rapid-Prototyping-Umgebungen möglich. Die Kalibrierung erfolgt jedoch auf Applikationssteuergeräten in einem Fahrzeug, welches bereits über seriennahe Aktuatorik, Sensorik und Regelsysteme verfügt. Mit zunehmendem Reifegrad entwickeln sich alle Komponenten und Subsysteme weiter, sodass sich das Prototypenfahrzeug immer mehr dem Serienfahrzeug annähert und eine aussagekräftigere Applikation möglich ist. Zwischen der Verfügbarkeit erster Prototypenfahrzeuge und dem Serienanlauf liegen im Schnitt circa zwei Jahre. In umfangreichen Fahrversuchen wird ein Systemverständnis der Wechselwirkungen zwischen eingestellten Parameterwerten und resultierendem Fahrverhalten erlangt. Die grundsätzlichen Zusammenhänge bleiben dabei zwar konstant, jedoch kann die Verfügbarkeit weiter entwickelter Aktuatorik und Sensorik das Übertragungsverhalten stark beeinflussen, sodass bereits optimierte Datensätze den neuen Gegebenheiten angepasst werden müssen. Falls eine entscheidende Optimierung durch die Ergänzung der Algorithmen und Regler möglich ist, wird die Software auch parallel zu der Applikation angepasst. Über mehrere Iterationen erfolgt somit idealerweise eine Konvergenz

zum anfangs definierten Fahrerwunschverhalten. Obwohl es Ansätze gibt, das Zielverhalten zu objektivieren (vgl. Abschnitt 2.6), wird das Fahrgefühl und der Komfort vom Applikateur meistens subjektiv im Fahrzeug bewertet. Somit ergeben sich teils unterschiedliche Wahrnehmungen und Parametersätze je nachdem, welcher Entwicklungsingenieur gerade die Kalibrierung vornimmt.

Um den Einfluss der Applikationsparameter auf das Systemverhalten besser verstehen zu können ist es hilfreich, die Wirkkette automatisierter Fahrfunktionen zu kennen. Durch die Schnittstellen des gerade applizierten Softwremoduls und dessen Verantwortung im Gesamtverbund lassen sich bereits Wechselwirkungen und Parametereinflüsse auf das Fahrverhalten abschätzen. Eine allgemeine Darstellung der Wirkkette unter Vernachlässigung etwaiger Rückkopplungen und Kaskaden zwischen den Modulen liefert Abbildung 2.3.

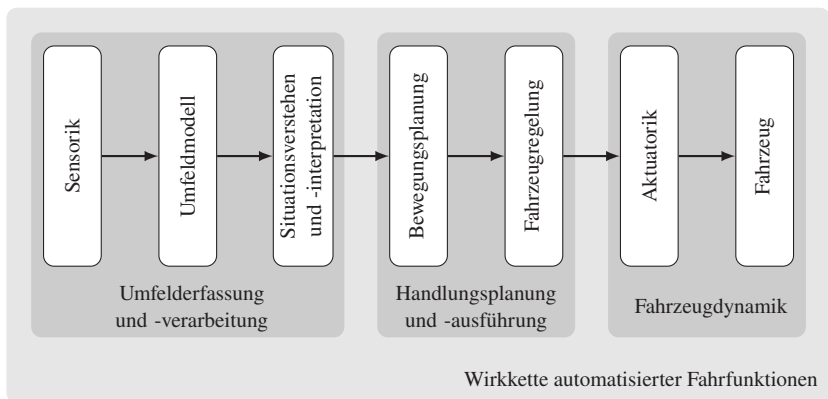


Abbildung 2.3: Allgemeine Wirkkette für automatisierte Fahrfunktionen nach [Die]

In dem linken Teil des Verbunds findet die Umfelderfassung und -verarbeitung statt. Für die Umfeldererkennung kommen typischerweise mehrere, teils redundante Sensorsysteme (z.B. Radar, Lidar, Laserscanner, Kamera) zum Einsatz. Je nach Einbauposition und Sensorkonzept können sie jedoch jeweils nur einen kleinen Teil des Verkehrsumfelds verlässlich erfassen. Erst durch die Kombination der Informationen ergibt sich eine gesamthafte Repräsentation der Umgebung sowie Dynamik und Position anderer Verkehrsteilnehmer. Dieser Prozess wird auch als Sensorfusion bezeichnet und findet im *Um-*

*feldmodell* statt [WHL15]. Die Erstellung einer für die Handlungsplanung verständlichen Umfeldarstellung wird von dem Modul *Situationsverstehen und -interpretation* durchgeführt. Dabei werden Verkehrsobjekte je nach Dimension und Dynamik klassifiziert, Fahrspuren erkannt und der befahrbare Raum definiert. Auch eine Verhaltensprädiktion anderer Verkehrsteilnehmer für den Planungshorizont ist möglich. Im Anschluss erfolgt die *Handlungsplanung und -ausführung*. Basierend auf dem Verkehrsumfeld, der eigenen Position und Fahrervorgaben erfolgt eine Trajektorienplanung in Längs- und Querrichtung. Die geplante Trajektorie wird anschließend von den aktuatornahen Regelsystemen umgesetzt. Obwohl eine Kalibrierung theoretisch für alle Softwaremodule entlang der Wirkkette möglich ist, stellt das Zusammenspiel aus Bewegungsplanung und -regelung den größten Stellhebel zur Erreichung des Fahrerwunschverhaltens dar. Auch weil das Umfeldmodell und die Situationsinterpretation teilweise vom Zulieferer umgesetzt wird, fokussiert sich die Applikation in Automobilkonzernen vorrangig auf die Module der Handlungsplanung und -ausführung. Für die Ausführung ist der Regler zuständig, welcher eine Stellgröße von den Aktuatorssystemen anfordert. Je nach Trägheit und Dynamik der Aktuatorik und des Fahrzeugs kann die Vorgabe der Bewegungsplanung unterschiedlich gut eingehalten werden. Die Aufgabe des Applikateurs besteht deshalb meist darin, einen Kompromiss zwischen der Regelabweichung und -dynamik des Gesamtsystems zu finden.

Obwohl sich die Parametrierung auf einen vergleichsweise kleinen Teil der Wirkkette beschränkt, sind Wechselwirkungen zu den restlichen Modulen zu erwarten, sodass eine aussagekräftige Applikation zwangsweise die Erprobung im Gesamtsystem erfordert. Nur auf diese Weise können alle Komponenten unter realen Betriebsbedingungen aufeinander abgestimmt werden. Aus diesem Grund erfolgt die Funktionskalibrierung heutzutage fast ausschließlich im Prototyp. Die fahrzeugbasierte Applikationsmethodik hat allerdings Nachteile, die abschließend aufgezeigt werden sollen [TCMA17, Hen18, WWJ18]:

- *Verfügbarkeit von Fahrzeugen*: Insbesondere in den frühen Phasen der Entwicklung ist nur eine sehr begrenzte Anzahl an Prototypenfahrzeugen verfügbar. Die ersten Applikationen werden typischerweise auf Versuchsträgern durchgeführt, dessen Subsysteme und Komponenten sich klar von dem Zielsystem differenzieren. Die auf dieser Basis identifizierten Bedatungen sind somit unter Umständen wenig aussagekräftig.



- *Ressourcenaufwand*: Insbesondere die gestiegene Anzahl angebotener Fahrzeugderivate und die Komplexität heutiger automatisierter Fahrfunktionen führt zu einem hohen Applikationsaufwand. Zunächst müssen eine große Anzahl an Prototypen bereitgestellt werden, dessen Herstellung nicht in Serienfertigung sondern oft in Einzelproduktion erfolgt und somit teuer ist. Außerdem erfordert die Erprobung der Fahrfunktionen die Bereitstellung einer Teststrecke, eine detaillierte Planung sowie die Bindung von Entwicklungsingenieuren für einen mehrwöchigen Zeitraum.
- *Komplexität des Parameterraums*: Mit den gestiegenen Funktionalitäten heutiger Fahrerassistenzsysteme geht unmittelbar eine Steigerung der Dimensionalität des Parameterraums einher. Da eine umfangreiche Analyse von Parameterkombinationen zu ressourcenaufwändig wäre, erfolgt die Kalibrierung meistens auf Basis von Erfahrungs- und Expertenwissen. Die Gefahr ist jedoch groß, dass weite Teile des Parameterraums nicht oder nur unzulänglich betrachtet werden. In diesen Bereichen könnten sich jedoch gute Lösungen befinden, die somit unberücksichtigt bleiben.
- *Subjektive Bewertung des Fahrverhaltens*: Wie bereits im oberen Teil beschrieben, stützt sich die Bewertung des Fahrverhaltens meist auf die subjektive Wahrnehmung des Applikationsingenieurs. Da eine allgemeingültige Objektivierung des Zielverhaltens nicht vorhanden ist, wird somit unter Umständen je nach Applikateur eine unterschiedliche Funktionsausprägung erzielt.
- *Fehlendes Systemverständnis*: Der Aufbau eines Systemverständnisses über Parametereinflüsse und -wechselwirkungen erfolgt zumeist experimentell oder mit Hilfe von Expertenwissen. Da im Fahrversuch jedoch nie alle Parameterkombinationen erprobt werden können und das Expertenwissen unter Umständen unvollständig ist, ist diese Vorgehensweise fehleranfällig. Die Gefahr ist somit gegeben, dass Parameteroptimierungen nicht zielführend durchgeführt werden, da das vorhandene Systemverständnis unzureichend ist. Eine Anwendung klassischer Verfahren der Sensitivitäts- und Einflussanalyse könnte diesem Problem begegnen, jedoch wäre dafür eine unrealistisch hohe Anzahl an Fahrversuchen nötig.

## **2.2 Bestehende Ansätze zur Optimierung des Applikationsprozesses von Fahrzeugsystemen**

Im folgenden Abschnitt werden im ersten Teil Applikationsmethoden aus dem Bereich der Fahrzeugtechnik vorgestellt, welche eine Systematisierung und Effizienzsteigerung des konventionellen Prozesses bereits ermöglichen. Inwiefern eine Anwendbarkeit dieser Methoden auf die Parametrierung automatisierter Fahrfunktionen möglich ist, wird in Abschnitt 3.2 diskutiert. Im zweiten Teil erfolgt eine Vorstellung existierender Verfahren zur Virtualisierung der Wirkkette (X-in-the-Loop Verfahren), die zur Effizienzsteigerung von Applikationsprozessen Anwendung finden.

### **2.2.1 Applikationsmethoden**

Das Problem einer zielgerichteten Applikation tritt nicht nur im Bereich der Entwicklung automatisierter Fahrfunktionen auf, sondern in jedem Bereich des Fahrzeugs, in dem die Systemausprägung durch unterschiedliche Parametrierungen beeinflusst werden kann. Insbesondere dann, wenn die Dimensionalität des Parameterraums und die Wechselwirkungen im Gesamtverbund keine praxisgerechte Darstellung des Zusammenhangs zwischen Bedatungen und Systemantworten mehr erlauben, entsteht der Bedarf nach Applikationsmethoden, die eine Beherrschbarkeit der Komplexität ermöglichen. Im Bereich der Antriebsstrangentwicklung sind diese Methoden bereits seit Anfang der Jahrtausendwende Gegenstand der Forschung und mittlerweile im Serienentwicklungsprozess etabliert [HSHK15].

Während Verbrennungsmotoren in den 70er Jahren vornehmlich mit dem Ziel der Leistungsmaximierung entwickelt wurden, sind seit circa 30 Jahren aufgrund gesetzlicher Grenzwerte die Anforderungen nach minimalen Emissionen und Kraftstoffverbrauch dazu gekommen. Die Motorenentwicklung muss somit einen Kompromiss zwischen den teils gegensätzlichen Zielen nach maximaler Fahrleistung, Fahrkomfort und Zuverlässigkeit sowie minimalem Schadstoffausstoß, Kraftstoffverbrauch und Entwicklungskosten finden [HSHK15, Nau02, Haa04]. Ein Weg diesem Zielkonflikt zu begegnen ist neben konstruktiver Maßnahmen die Einführung weiterer Freiheitsgrade bestehender

Steuerungs- und Regelungssysteme. Fortschritte im Bereich der Prozessorleistung von Steuergeräten, Sensorik und die Nutzung neuer Technologien wie Turboaufladung, Direkteinspritzung oder Abgasrückführung erlauben zunehmend präzisere Regelungskonzepte und Möglichkeiten die teils konträren Ziele zu erreichen [BZ17, Haf02, Mit00]. Mit der gesteigerten Komplexität heutiger Motoren geht unmittelbar eine Erhöhung der Anzahl an Applikationsparametern einher. Innerhalb von 20 Jahren ist eine ungefähr 17-fache Steigerung der Dimensionalität des Parameterraums für die Motorapplikation entstanden [SSR<sup>+</sup>18]. Die Tatsache, dass der Applikationsaufwand exponentiell mit der Anzahl an Parametern steigt [Hen18] motivierte somit die Entwicklung verbesserter Verfahren der Steuergeräteapplikation um die Komplexität zu beherrschen. Die Produktstrategie vieler Automobilhersteller, Motoren modellübergreifend einzusetzen und nur durch die Applikation zu differenzieren, trug ebenfalls zu dieser Entwicklung bei.

Schwerpunkt der Forschung in diesem Bereich ist das Ersetzen der konventionellen manuellen Applikation am Prüfstand oder im Realfahrzeug durch modellbasierte Verfahren, welche die Wirkkette teilweise oder komplett virtualisieren und auf Grundlage von Ersatzmodellen Optimierungen durchführen. Ein weiterer Schwerpunkt ist die Weiterentwicklung der Applikation dahingehend, dass Parametersätze durch intelligente Versuchspläne und Optimierer unter minimalem Versuchsaufwand am realen Motor herausgefahren werden. Diese beiden Ansätze werden nach [WMF<sup>+</sup>00] als Offline- und Online-Optimierung bezeichnet. Die Offline-Optimierung ist auch in anderen Bereichen bekannt als modellgestützte Optimierung und kommt immer dann zum Einsatz, wenn eine Approximation des realen Systems günstiger ist als die Auswertung jedes einzelnen Parametersatzes am Realsystem während der Optimierung [F<sup>+</sup>15].

Für die Motorapplikation hat sich eine Prozesskette wie in Abb. 2.4 dargestellt etabliert. Es wird zunächst ein Versuchsplan erstellt, dessen Betriebspunkte am realen Motor abgefahren werden und als Eingangsgröße zur Modellbildung dienen. Die Optimierung erfolgt dann auf Grundlage des Ersatzmodells, wird allerdings im Anschluss nochmals mit Messungen verifiziert. Ist die Modellgüte ausreichend, wird die optimierte Parametrierung übernommen, andernfalls wird der Prozess wiederholt. Dieses strukturierte Vorgehen ist unter dem Begriff der statistischen Versuchsplanung (Design of Experiments, DoE) bekannt und in mehreren Werken umfassend thematisiert [Puk06, SHB10, Mon17]. Beim Versuchsdesign wird ein Versuchsplan erstellt, welcher die minimale

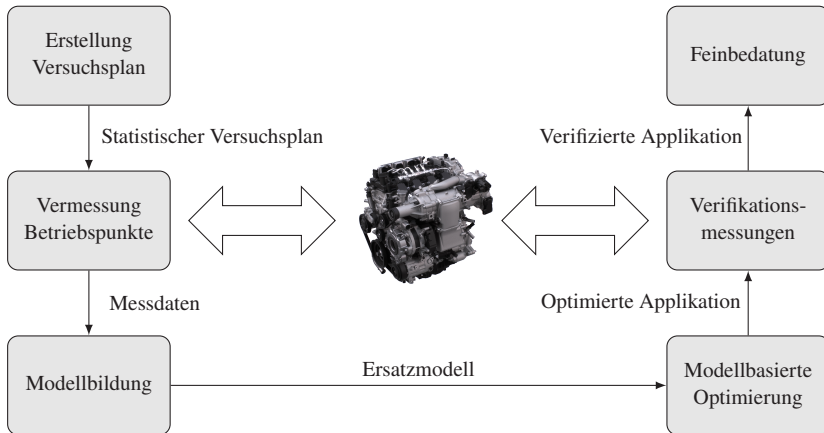


Abbildung 2.4: Prozesskette für die modellgestützte Applikation des Motors [WMF<sup>+</sup>00]

Anzahl aller relevanten Betriebspunkte zur Erstellung eines ausreichend genauen Ersatzmodells enthält. Für das Sampling ist die Anzahl der Parameter sowie deren Auflösung von entscheidender Bedeutung. Wenn diese durch Expertenwissen im Vorhinein eingeschränkt werden können, lässt sich der Versuchsaufwand exponentiell reduzieren. In der Praxis haben sich vor Allem teilfaktorielle Versuchspläne wie z.B. D-optimale Versuchspläne, quasi-zufällige Pläne (z.B. Latin Hypercube Sampling) oder mit Monte-Carlo-Verfahren erstellte Versuchspläne etabliert [SHB10].

Nachdem alle Betriebspunkte vermessen worden sind, erfolgt die Erstellung des Ersatzmodells. Dieses kann entweder unter Einbeziehung aller vorher definierten Applikationsparameter geschehen, die Dimensionalität kann aber auch in einem vorgelagerten Schritt durch Einflussanalysen reduziert werden um die Modellkomplexität klein zu halten. Hierfür eignen sich zum Beispiel Sensitivitätsanalysen. Diese können bei geschickter Wahl des Versuchsplans die gleichen Betriebspunkte wie die Modellbildung nutzen. Die Erstellung des Ersatzmodells kann auf unterschiedliche Arten erfolgen. Im einfachsten Fall kommen lineare Regressionsmodelle zum Einsatz. Zur Approximation einfacher nichtlinearer Zusammenhänge eignen sich Polynome oder Splines [Haa04]. Da allerdings meistens mit stark nichtlinearem Verhalten zu rechnen ist, kommen vermehrt neuronale Netze zum Einsatz [GDS14, Nau02, SHI00, Haf02].

Diese haben den Vorteil, dass keine komplexen analytischen Zusammenhänge hergeleitet werden müssen, sondern das Modell aus Messdaten lernt. Es bietet somit die Chance deutlich schneller rechnen zu können, sollten die Zusammenhänge anhand weniger Schichten und Neuronen abbildbar sein [SHI00]. Ein weiterer Approximator mit guten Ergebnissen ist das Kriging-Verfahren [BDV-CB09, SLKK11, CZL<sup>+</sup>15]. Kriging kommt ursprünglich aus der Geostatik und interpoliert die Systemantwort an Stellen, wo keine Messwerte vorliegen. Die Güte der Ersatzmodelle hängt entscheidend von der Vollständigkeit des Versuchsplans ab. Fehlen Betriebspunkte, in dessen Umgebung die Optimierung eine Auswertung von dem Ersatzmodell erwartet, ist das Optimierungsergebnis möglicherweise fehlerbehaftet.

Basierend auf dem Ersatzmodell kann eine Optimierung mit reduziertem Ressourcenaufwand erfolgen. Wie oben beschrieben, ist es die Aufgabe der Applikation teils konträre Ziele zu optimieren. Im Bereich der Motorenentwicklung sind dies typischerweise Fahrkomfort, Leistung, Emissionen und Kraftstoffverbrauch oder ein Subset von diesen [HSHK15, Mit00]. Es gibt somit nicht nur ein einziges Optimum, sondern eine Reihe von Kompromisslösungen. Wenn bei diesen Lösungen eine Verbesserung eines Zielkriteriums nicht ohne Verschlechterung anderer Kriterien möglich ist, sind diese Lösungen pareto-optimal [SHB10] (vgl. Abschnitt 2.4). Als Optimierungsalgorithmen für das multikriterielle Optimierungsproblem kommen vermehrt populationsbasierte Verfahren zum Einsatz. Insbesondere der genetische Algorithmus bildet einen guten Kompromiss zwischen Optimierungsergebnis und -aufwand und wird daher oft verwendet [Haa04, Jan08]. Auch neuronale Netze sind als Optimierer denkbar [GDS14]. Die automatisierte Kalibrierung liefert eine Vielzahl an pareto-optimalen Lösungen, aus denen der Applikateur einen für den jeweiligen Anwendungsfall geeigneten Kandidaten auswählen muss. Da die modellbasierte Optimierung allerdings unter geringem Zeitaufwand wiederholt werden kann, können Zielkriterien umformuliert werden um mehrere Bedatungen beispielsweise für unterschiedliche Fahrzeugderivate oder Länder zu generieren. Zuletzt werden die am Ersatzmodell erzielten Lösungen noch am Motorprüfstand verifiziert.

Im Gegensatz zur Offline-Optimierung steht die Online-Optimierung, die ausschließlich anhand der realen Zielhardware durchgeführt wird. Der entscheidende Vorteil dieses Verfahrens ist, dass keine Modellfehler in Kauf genommen werden müssen. Typischerweise wird der Motor in diesem Fall am Prüfstand

mit Hilfe eines vollfaktoriellen Versuchsplans vermessen und Applikationsparameter manuell vom Entwickler eingestellt. Mit der heutzutage vorliegenden Parameteranzahl stößt dieses Verfahren jedoch an seine Grenzen. Aus dem Grund gibt es Ansätze, welche die Offline-Optimierung zur Erzeugung einer Basisapplikation nutzen und die Feinapplikation am Prüfstand durchführen [Nau02, Jan08]. Um die Anzahl an Versuchsdurchläufen gering zu halten, wird Expertenwissen genutzt, das die Lösungssuche im Parameterraum zielgerichtet beeinflussen soll. Die Wissensbasis entsteht durch Anwendung von Fuzzy-Techniken, welche Motorprozesszusammenhänge in einfache Entscheidungsregeln für die Optimum-Suche umwandelt. Heuristische Suchverfahren verwenden diese Regeln um schneller zu einer optimalen Lösung konvergieren zu können. Der Vorteil gegenüber klassischen gradientenbasierten Optimierungsverfahren und populationsbasierten Ansätzen ist die deutlich reduzierte Anzahl an Prüfstandsdurchläufen. Aufbauend auf der vorgelagerten Offline-Optimierung am Ersatzmodell kann somit ein Parametersatz mit hohem Reifegrad erzielt werden.

Neben den etablierten Applikationsmethoden in der Motorapplikation existieren Ansätze zur automatisierten Parametrierung weiterer Fahrzeugsysteme. [Kah13] stellt eine Methode zur effizienten Getriebeapplikation vor, welche eine ersatzmodellgestützte Optimierung der Kriterien Verschleiß, Schaltzeit und Fahrkomfort mit Hilfe eines genetischen Algorithmus anstrebt. Das Getriebe-Ersatzmodell wird dabei mit dem Kriging-Verfahren erzeugt und in eine Simulationsumgebung mit einem detaillierten Fahrzeugmodell eingebettet, welches als Grundlage für die Optimierung dient. In der Arbeit von [Sit14] wird ein Verfahren zur automatisierten Applikation von Betriebsstrategien in Hybridfahrzeugen vorgestellt. Ziel der Kalibrierung ist in diesem Fall die Minimierung des Kraftstoffverbrauchs in Kombination mit weiteren Zielkriterien beim Fahren standardisierter Verbrauchszyklen. Die Optimierung erfolgt ebenfalls mit einem genetischen Algorithmus und stützt sich auf ein Gesamtfahrzeug-Simulationsmodell.

In den letzten Jahren entstehen vermehrt Forschungsbeiträge, die sich mit der Applikation von Fahrdynamik- und Fahrerassistenzsystemen beschäftigen [Lie17, Osc19]. Der entscheidende Unterschied im Vergleich zu der Applikation des Antriebsstrangs ist, dass das Fahrerwuschverhalten als objektives Zielkriterium bestimmt werden muss und nicht sofort anhand einzelner Kennwerte formuliert werden kann. Die vorliegenden Arbeiten führen deshalb im

Vorhinein zu der Applikation Probandenstudien durch, mit denen durch Regressionsanalysen analytische Zielfunktionen bestimmt werden können. Die Funktionsoptimierung erfolgt dann auf Grundlage definierter Referenzszenarien, die von der jeweiligen Assistenzfunktion in der Simulation gefahren werden. Im Gegensatz zu dem Applikationsproblem beim Antriebsstrang kann eine automatisierte Fahrfunktion nicht isoliert auf dem Prüfstand appliziert werden sondern muss immer im Kontext der gesamten Wirkkette betrachtet werden. Die Gefahr von Modellungenauigkeiten bei der simulativen Optimierung ist somit deutlich höher.

## 2.2.2 Virtualisierung der Wirkkette

Wie im letzten Abschnitt beschrieben, ist eine isolierte Betrachtung automatisierter Fahrfunktionen mit Abstraktion der Eingangs- und Ausgangsgrößen nicht zielführend. Die Performance des Gesamtsystems muss immer unter Berücksichtigung der gesamten Wirkkette bewertet werden, da Wechselwirkungen zwischen der Umfelderkennung und -verarbeitung, Bewegungsplanung, Regelung und der Fahrzeugdynamik zu erwarten sind (vgl. Wirkkette der Applikation, Abb. 2.3). Um diesem Problem zu begegnen und Tests nicht ausschließlich im Fahrzeug durchführen zu müssen, haben sich verschiedene Ansätze zur Virtualisierung einzelner Teile der Wirkkette etabliert. Simulationsbasiertes Testen hat insbesondere im Bereich der Fahrerassistenz den Vorteil der Reproduzierbarkeit von Verkehrsszenarien und rückt aufgrund verschärfter gesetzlicher Anforderungen an Testumfang und -komplexität immer mehr in den Fokus. Auch die Forderung agiler Arbeitsmethoden nach schnellen Feedbackschleifen und hoher Testabdeckung in den frühen Phasen der Entwicklung motiviert die Weiterentwicklung dieser Verfahren.

Je nach Virtualisierungsgrad des Gesamtsystems werden Model-in-the-Loop (MiL), Software-in-the-Loop (SiL), Hardware-in-the-Loop (HiL) und Vehicle-in-the-Loop (ViL) unterschieden. Gesamthaft werden diese Ansätze als X-in-the-Loop (XiL) bezeichnet. Die einzelnen Testverfahren unterscheiden sich in der Anzahl an Komponenten, die als reale Hardware zur Verfügung stehen. Mit der Definition von standardisierten Schnittstellen kann eine einfache Überführung der Methoden ineinander erfolgen. Abbildung 2.5 gibt einen Überblick über real und virtuell eingebundene Komponenten der einzelnen Verfahren. Im klassischen V-Modell der Produktentwicklung (vgl. Abschnitt 2.1) werden

Komponenten	Model in the Loop	Software in the loop	Hardware in the loop	Vehicle in the loop	Fahrversuch
Funktions-Code	virtuell	real			
Steuergerät	virtuell		real		
System	virtuell		real		
Fahrzeug	virtuell			real	
Fahrer	virtuell			virtuell/real	real
Fahrbahn	virtuell				real
Umwelt	virtuell				real

Abbildung 2.5: Überblick virtuell und real zur Verfügung gestellter Komponenten der X-in-the-Loop-Verfahren [WHLS15]

die Methoden mit zunehmendem Reifegrad der Funktion als Testwerkzeuge eingesetzt, sodass die Entwicklung von Kundenanforderungen, das Systemdesign und die Implementierung typischerweise mit MiL- und SiL-Verfahren erfolgt. Tests unterschiedlicher Detaillierung gegenüber den Anforderungen sowie die Applikation werden hingegen mit Hilfe von HiL, ViL und dem Fahrversuch durchgeführt [WHLS15]. Agile Arbeitsmodelle sehen hingegen die Verwendung sämtlicher Testumgebungen schon während der Produktentstehung vor um Softwarefeatures regelmäßig gegenüber sich ständig ändernden Kundenanforderungen testen zu können. Da die Funktionalität in der MiL-Umgebung durch modellbasierte Software abstrahiert wird und der eigentliche Funktionscode noch nicht zur Verfügung steht, kommt diese Umgebung nicht für die virtuelle Applikation in Frage. Es wird daher im Folgenden nur auf die drei übrigen XiL-Verfahren eingegangen.

### Software-In-the-Loop

Die SiL-Methode enthält bereits den realen Funktionscode. Dieser wird in eine Simulationsumgebung eingebettet, die das Zielsystem in Bezug auf Auflösungsgenauigkeit, Echtzeitverhalten und Rechenzeit bereits gut abbilden kann, jedoch hardwareunabhängig ist [MDF13]. Unter der Voraussetzung, dass alle



Komponenten des Gesamtsystems mit ausreichender Genauigkeit modelliert sind, kann das SiL-Verfahren bereits in den frühen Phasen der Entwicklung eingesetzt werden, um grundsätzliche Anforderungen an Teilmodule zu testen. Im Bereich der agilen Softwareentwicklung können somit Systemtests, Integrationstests und Akzeptanztests formuliert werden, die als Grundlage zur Entwicklung der Softwarekomponenten dienen [Som11]. Da der Software-in-the-Loop-Ansatz die günstigste und schnellste Alternative zum Testen des Gesamtsystems ist, hat es sich insbesondere in den letzten Jahren für die virtuelle Absicherung automatisierter Fahrfunktionen etabliert. Die hohe Anzahl an geforderten unfallfreien Kilometern für eine erfolgreiche Absicherung von Level-3-Funktionen wird zunehmend virtuell nachgewiesen, um den Testaufwand beherrschbar zu machen [WWJ18, PZE17, GKF<sup>+</sup>17]. Inwiefern SiL-Umgebungen sich auch für eine virtuelle Applikation eignen, hängt von der Güte des Simulationsmodells ab. Während das Ziel in der Absicherung die Einhaltung sicherheitskritischer Grenzwerte ist, strebt die Applikation eine Optimierung des Fahrkomforts und subjektiven Fahrgefühls an. Unabhängig dessen, dass somit ein objektiviertes Gütekriterium bestimmt werden muss, kann eine geringfügig geänderte Fahrdynamik einen großen Einfluss auf das Empfinden im Fahrzeug haben, die in Simulationsmodellen unter Umständen nicht abbildbar sind. Ob ein Einsatz für simulative Applikationen dennoch möglich ist, wird im späteren Teil der Arbeit diskutiert.

### **Hardware-In-the-Loop**

Beim Hardware-in-the-Loop-Verfahren werden typischerweise reale Steuergeräte verwendet, die äquivalent zur finalen Architektur miteinander kommunizieren und den realen Funktionscode enthalten. Es ist aber auch denkbar neben den Steuergeräten reale mechatronische Komponenten einzubetten. Während die einzelnen Funktionsmodule bei dem SiL-Verfahren meist in einer zielhardwareunabhängigen Rapid-Prototyping-Umgebung getestet werden, sind sie nun auf den Steuergeräten kompiliert. Aufgrund der gesteigerten Modellgüte durch die Einbindung realer Komponenten und der gleichzeitigen Möglichkeit der Testautomatisierung hält die HiL-Methodik immer mehr Einzug im Bereich der Regler- und Algorithmenentwicklung für automatisierte Fahrfunktionen [LZC<sup>+</sup>17, Sch10, Jos17]. Dem Vorteil der gesteigerten Modellgüte durch das Einbringen realer Komponenten steht allerdings der Nachteil einer eingeschränkten Skalierbarkeit der Versuche gegenüber. Während SiL-Simulationen

einfach parallelisiert werden können, müssten im Fall der HiL-Umgebung für die gleiche Versuchsanzahl viele Prüfstände aufgebaut werden, was mit teils hohen Kosten verbunden wäre. Der für eine Parameter-Optimierung in einer HiL-Umgebung erforderliche hohe Ressourcenaufwand sowie die fehlende Erlebbarkeit der Fahrfunktion im Fahrzeug rechtfertigen nicht den Einsatz für eine automatisierte Applikation. Der Haupt-Anwendungsbereich erstreckt sich vielmehr auf das entwicklungsbegleitende Testen der automatisierten Fahrfunktionen in Komponenten-, System- und Integrationstests [WHLS15].

### **Vehicle-In-the-Loop**

Dem Bedarf vieler Assistenzfunktionen, im realen Zielsystem getestet werden zu müssen, wird die Vehicle-in-the-Loop-Methodik gerecht. Sie sieht neben der Einbindung von Steuergeräten die Integration aller Komponenten in das Fahrzeug vor. Der einzige Unterschied zum Fahrversuch besteht darin, das Verkehrsumfeld nicht real zur Verfügung stellen zu müssen. Die Sensorik zur Umfelderkennung wird in diesem Fall oftmals mit synthetischen Eingangsdaten gespeist, die sich entweder aus realen Verkehrsszenarien ergeben oder vom Entwickler designt werden. Das Fahrzeug reagiert dann auf Merkmale und Ereignisse aus der virtuellen Umwelt [Boc08]. Obwohl der Testaufwand vergleichbar zum Fahrversuch ist, bietet diese Methodik den entscheidenden Vorteil der Reproduzierbarkeit von Testfällen. Darüber hinaus ergeben sich neue Chancen kritische Fahrmanöver, welche die Reaktion auf Hindernisse oder Verkehrsteilnehmer beinhalten, sicher zu testen. Die Anwendung als Testwerkzeug zur Funktionsentwicklung und Absicherung sicherheitskritischer Szenarien ist in mehreren Arbeiten thematisiert [Ber14, HLT<sup>+</sup>19, TSVT18] und der industrielle Einsatz mittlerweile etabliert. Im Gegensatz zu der HiL-Umgebung ist mit dem ViL-Versuch eine Erlebbarkeit der Fahrfunktionen im Fahrzeug möglich. Auch wenn der Test umfangreicher ist und eine automatisierte Applikation auf Grundlage von Optimierungsalgorithmen zu ressourcenintensiv wäre, bietet das Verfahren die Chance eine Feinapplikation im realen Zielsystem systematisch mit reproduzierbaren Verkehrssituationen durchzuführen. Insbesondere die ständig wachsende Anzahl an Fahrzeugderivaten, in denen automatisierte Fahrfunktionen angeboten werden und der damit verbundene Applikations- und Absicherungsaufwand motiviert den Einsatz von ViL-Methoden.

## 2.3 Komplexitätsreduktion hochdimensionaler Parameterräume

Wie in Abschnitt 2.1 beschrieben, stellt die Applikation automatisierter Fahrfunktionen ein komplexes, hochdimensionales Optimierungsproblem dar. Die Lösung dieses Problems erfolgt heutzutage weitgehend durch Erprobungen des Gesamtsystems im Fahrversuch auf Grundlage von Erfahrungs- und Expertenwissen der Applikateure und Funktionsentwickler. Die Komplexität des Parameterraums ist durch manuelles Tunen von Parametern jedoch nicht beherrschbar und die Gefahr besteht, dass einflussreiche Bereiche außer Acht gelassen werden. Für Optimierungsprobleme vergleichbarer Dimensionalität hat sich aus diesem Grund eine vorgelagerte Einflussanalyse der Parameter als vielversprechend erwiesen. Mit Hilfe dessen können nicht-sensitive Parameter oder Teile des Parameterraums von der Optimierung ausgeschlossen und die Konvergenzgeschwindigkeit gesteigert werden. Darüber hinaus liegt ein gesteigertes Systemverständnis vor, welches in die Kalibrierung mit einfließen kann [Mac15, Unt13]. Die Größe des Suchraums für die Optimierung ist festgelegt durch die Anzahl an Parametern (Dimensionalität) und die Wertebereiche beziehungsweise deren Diskretisierung. Im Folgenden soll auf die mögliche Reduktion beider Faktoren eingegangen werden. Sensitivitätsanalysen dienen in erster Linie der Identifikation von Parametereinflüssen und werden in Abschnitt 2.3.1 vorgestellt, während Abschnitt 2.3.2 auf Verfahren zur Einschränkung von Parameterwertebereichen eingeht.

### 2.3.1 Numerische Sensitivitätsanalyse

Unter Sensitivitätsanalysen (SA) wird die Untersuchung von Eingangsgrößen hinsichtlich ihrer Einflüsse in Bezug auf Änderungen von Ausgabegrößen eines Systems oder Modells verstanden. Im Bereich der Optimierung sind die Ausgabegrößen in Form von Gütekriterien oder Zielfunktionen gegeben [SRA<sup>+</sup>08]. Die Ergebnisse einer SA sind Informationen über den totalen Effekt und den Haupteffekt jedes Parameters. Der Haupteffekt beschreibt den Einfluss eines Faktors auf das Gütekriterium unter Vernachlässigung von Informationen über Wechselwirkungen und Nichtlinearitäten, die zusätzlich im totalen Effekt enthalten sind [SRA<sup>+</sup>08]. Können beide Effekte im Kontext des jeweiligen Optimierungsproblems als vernachlässigbar bewertet werden, lässt sich die

Dimensionalität des Parameterraums entscheidend reduzieren und eine höhere Konvergenzgeschwindigkeit zur Lösungssuche ist zu erwarten [STCR13]. Aufgrund dieses Potentials werden Sensitivitätsanalysen heute vielfach in Kombination mit Parameter-Optimierungen angewendet, die durch eine aufwändige Evaluation der Zielfunktion charakterisiert sind [IL15]. Im Bereich der Fahrzeugtechnik findet dieses Verfahren bereits für die Applikation des Antriebsstrangs (vgl. [Haa04, CS09, Kah13, Pil17]) oder Parameteridentifikation zur Modellbildung (vgl. [Unt13, Bau15]) breite Anwendung.

Methoden der Sensitivitätsanalyse lassen sich zunächst in lokale und globale sowie darüber hinaus in qualitative und quantitative Methoden unterteilen. Bei der lokalen SA wird nur jeweils ein Parameter variiert und der Einfluss dessen auf das Ergebnis untersucht. Dieses Vorgehen wird auch als One-at-a-time-Verfahren (OAT) bezeichnet. Der Vorteil dieser Methodik ist, dass sie bei vielen Parametern in vertretbarem Aufwand durchführbar ist. Der Nachteil ist jedoch, dass Wechselwirkungen zu den übrigen Faktoren vernachlässigt werden und die Ergebnisse immer nur im jeweiligen Arbeitspunkt gültig sind. Sind Wechselwirkungen und Nichtlinearitäten jedoch im Vorhinein auszuschließen, sind die Erkenntnisse der lokalen SA auf den gesamten Arbeitsbereich übertragbar. Im Gegensatz dazu werden bei der globalen SA auch Einflüsse von Parameterkombinationen auf das Ergebnis betrachtet, sodass Nichtlinearitäten und Wechselwirkungen in die Bestimmung der Einflussmetriken mit einfließen. Dem meist höheren Aufwand dieser Verfahren steht die Gültigkeit der Ergebnisse im gesamten Arbeitsbereich gegenüber. Da insbesondere bei komplexen Optimierungsproblemen nicht von einem rein linearen Modell ohne Interaktionen zwischen den Faktoren ausgegangen werden kann, wird die globale SA in den meisten Fällen bevorzugt [SRA<sup>+</sup>08, STCR13, CSC11].

Die Unterteilung nach qualitativen und quantitativen SA-Methoden richtet sich nach der Aussagekraft der Sensitivitätsmaße. Während qualitative Verfahren meist weniger Systemevaluationen benötigen, erlauben die Ergebnisse lediglich einen relativen Vergleich der Parametereinflüsse untereinander. Quantitative Methoden sind meist aufwändiger, ermöglichen aber eine Analyse des quantitativen Einflusses von jedem Parameter auf die Zielfunktion.

Für die Anwendung in der Applikation automatisierter Fahrfunktionen kommen aufgrund der Wechselwirkungen entlang der Wirkkette (vgl. Abb. 2.3) und anzunehmender Nichtlinearitäten nur globale SA-Verfahren in Frage. Es ist auch davon auszugehen, dass im Vorhinein kein Systemwissen vorliegt.

Eine qualitative Methode, welche mit geringem Versuchsaufwand alle Haupt- und Totaleffekte identifizieren kann, ist die Elementareffektmethode (EEM) oder Morris-Methode nach [Mor91]. Aufgrund der weiten Verbreitung und erfolgreichen Anwendung für vergleichbare Probleme soll diese im Folgenden vorgestellt werden. Im Anschluss wird die ebenfalls oft verwendete varianzbasierte Sensitivitätsanalyse (VBSA) als alternatives Verfahren vorgestellt, welches im Unterschied zur EEM quantitative Ergebnisse liefert, aber mit deutlich höherem Versuchsaufwand verbunden ist.

### Elementareffektmethode

Die Elementareffektmethode stellt ein qualitatives Verfahren der Sensitivitätsanalyse dar, welches den Parameterraum mit einem OAT-Ansatz entlang einer Trajektorie (als Bewegung im Parameterraum) durchsucht und die Größe der Effekte von Eingangsparametern abschätzt. Da das Verfahren Nichtlinearitäten und Wechselwirkungen berücksichtigt, sind die Ergebnisse im gesamten Arbeitsbereich gültig.

Die Grundlage für die Berechnung der Haupt- und Totaleffekte bildet das Sampling, welches im ersten Schritt erfolgt. Obwohl verschiedene Sampling-Strategien möglich sind, wird für die EEM meist ein radialer Ansatz verfolgt, da dieser einen guten Kompromiss zwischen der Anzahl an Punkten und einer gleichmäßigen Parameterraum-Abdeckung bildet [CSC11]. Ausgehend von einem quasi-zufällig erzeugten Versuchsplan (z.B. Latin Hypercube Sampling, Monte Carlo Sampling) mit  $r$  Parametersets werden alle Punkte im Suchraum jeweils entlang jeder der  $n_p$  Dimensionen variiert. Die restlichen Parameterwerte des Samples bleiben dabei unverändert (OAT-Ansatz). Abb. 2.6 zeigt das Sampling-Schema anhand eines dreidimensionalen Beispiels. Die Anzahl der benötigten Systemevaluation  $n_{EEM}$  berechnet sich wie folgt:

$$n_{EEM} = r \cdot (n_p + 1). \quad (2.1)$$

Für die OAT-Variationen der  $r$  Basis-Samples werden nun die relativen Änderungen der Zielfunktion  $F(P)$  in Abhängigkeit der Änderung  $\Delta$  des jeweiligen Parameters  $P_i$  berechnet. Die Variable  $P$  bildet alle Parameter ab:  $P \in [P_1, \dots, P_{n_p}]$ . Die somit approximativ über den Differenzenquotienten berechnete numerische Ableitung von  $F(P)$  nach  $P_i$  wird auch als Elementar-

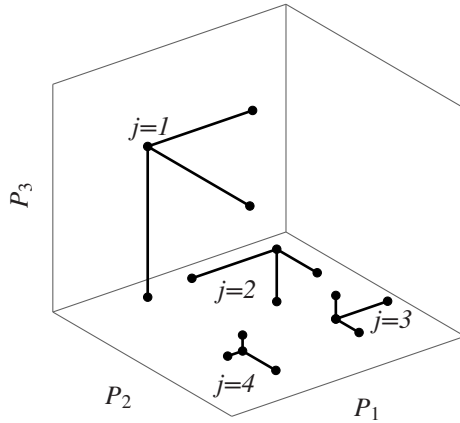


Abbildung 2.6: Radiales Sampling-Schema für die Elementareffektmethode mit  $r = 4$  und  $n_p = 3$  [FFFG19a]

effekt bezeichnet und berechnet sich entsprechend Gleichung 2.2. Der Index  $j$  beschreibt darin die Sampling-Gruppe (vgl. Abb. 2.6).

$$EE_i^j = \frac{F(P_1^j, P_2^j, \dots, P_{i-1}^j, P_i^j + \Delta, \dots, P_{n_p}^j) - F(P_1^j, P_2^j, \dots, P_{n_p}^j)}{\Delta}, \quad (2.2)$$

$i \in \{1, \dots, n_p\}, j \in \{1, \dots, r\}$

Für jeden Parameter  $P_i$  entstehen somit  $r$  Elementareffekte, dessen Mittelwert  $\mu_i$  zur Abschätzung des Haupteffekts dient. Zusätzlich wird die Standardabweichung  $\sigma_i$  berechnet, welche die Größe des Einflusses von Nichtlinearitäten und Wechselwirkungen approximiert:

$$\mu_i = \frac{1}{r} \cdot \sum_{j=1}^r EE_i^j, \quad (2.3)$$

$$\sigma_i = \sqrt{\frac{1}{r-1} \cdot \sum_{j=1}^r (EE_i^j - \mu_i)^2}. \quad (2.4)$$

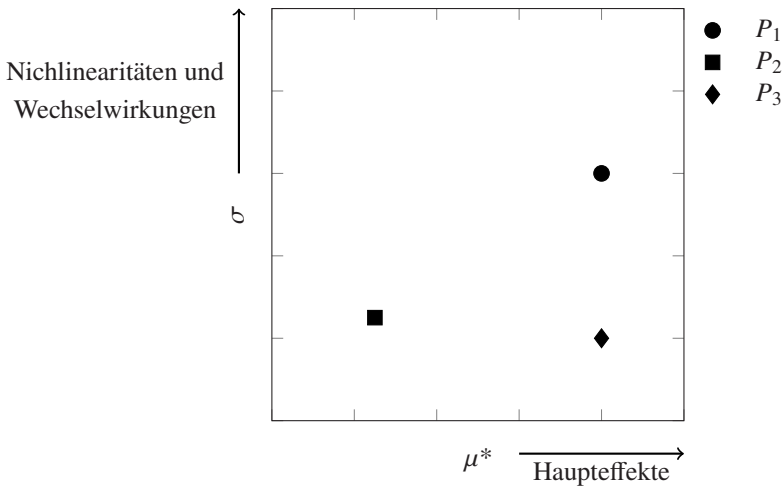
Bei der Berechnung des Erwartungswerts entsprechend Gleichung 2.3 besteht die Gefahr, dass sich Effekte unterschiedlichen Vorzeichens gegenseitig auslöschen, wenn das Modell nicht-monoton ist. Aus dem Grund wurde eine alternative Abschätzung der Haupteffekte eingeführt, welche diese Problematik auflöst [CCS07]:

$$\mu_i^* = \frac{1}{r} \cdot \sum_{j=1}^r |EE_i^j|, \quad (2.5)$$

Der Nachteil dieser Berechnungsvorschrift ist allerdings, dass Aussagen über die Richtung der Effekte verloren gehen. Es ist somit ratsam, sowohl  $\mu^*$  als auch  $\mu$  zu berechnen.

Aus der Mittelwertbildung aller Elementareffekte eines Parameters kann direkt auf dessen Haupteffekt geschlossen werden. Der Totaleffekt ergibt sich allerdings erst aus der kombinierten Betrachtung des Erwartungswerts und der Standardabweichung. Unabhängig davon sollte eine Eingangsgröße aber als einflussreich klassifiziert werden, wenn ein hohes  $\sigma$  auf starke Nichtlinearitäten oder Wechselwirkungen hindeutet. Für die Analyse der Sensitivitätsmaße wird häufig eine graphische Darstellung im  $\mu^* - \sigma$ -Diagramm bevorzugt. Parameter, welche darin auf der Hauptdiagonalen in der Nähe des Koordinatenursprungs liegen, werden als wenig einflussreich eingestuft. Je weiter ein Parameter rechts oben im Diagramm liegt, desto sensitiver ist er bezüglich der Gütefunktion  $F(P)$  und sollte daher in jedem Fall für die Optimierung herangezogen werden. Ein beispielhaftes  $\mu^* - \sigma$ -Diagramm ist in Abb. 2.7 dargestellt.

Das Morris-Verfahren ist ein qualitatives Verfahren, welches mit zunehmender Anzahl  $r$  an Elementareffekten genauere Ergebnisse liefert. Da die Werte für  $\mu^*$  und  $\sigma$  keine Aussage über den direkten Einfluss der Parameter auf  $F(P)$  liefern, sondern die Sensitivitäten relativ zueinander bewertet werden, wird meist auf eine Achsenbeschriftung im  $\mu^* - \sigma$ -Diagramm verzichtet. Aufgrund der vergleichbar geringen Anzahl an Systemevaluationen im Vergleich zu den im Folgenden beschriebenen varianzbasierten Verfahren und der Berücksichtigung sämtlicher relevanten Effekte wird die EEM heutzutage dennoch oftmals eingesetzt um die Dimensionalität von Optimierungsproblemen zu reduzieren und ein gesteigertes Systemverständnis zu erzielen (vgl. z.B. [CS09, RRGK04, PD-PS15]).

Abbildung 2.7: Sensitivitätsmaße der Elementareffektmethode im  $\mu^* - \sigma$ -Diagramm [FFFG19a]

### Varianzbasierte Sensitivitätsanalyse

Im Gegensatz zur EEM zählt die varianzbasierte Sensitivitätsanalyse (VBSA) zu den quantitativen Methoden, sodass Sensitivitätswerte direkt in Zusammenhang mit dem Gütefunktional gebracht werden können. Die Zerlegung der Varianz der Ausgangsgrößen oder Gütefunktionale in Abhängigkeit der Varianz der Eingangsgrößen oder Systemparameter bildet die Grundlage für das Verfahren. Die Sensitivitätsmaße können somit gesamthaft die Varianz der Zielfunktion  $F(P)$  erklären, indem sie diese auf die Streuungen der Eingangsparameter oder Kombinationen aus diesen beziehen [STCR04, SAA<sup>+</sup>10]. Entsprechend Gleichung 2.6 lässt sich der Haupteffekt  $S_i$  für Parameter  $P_i$  berechnen. Die Notation basiert auf [SRA<sup>+</sup>08], wobei  $V$  die Varianz und  $E$  den Erwartungswert kennzeichnet. Die Indizes  $P_i$  und  $P_{-i}$  implizieren, dass sich die zugehörige Operation auf den Parameter  $i$  beziehungsweise alle Parameter außer dem Parameter  $i$  bezieht.

$$S_i = \frac{V_{P_i}(E_{P_{-i}}(F(P)|P_i))}{V(F(P))} \quad (2.6)$$



Mit Hilfe des Haupteffekts lassen sich Effekte erster Ordnung bei alleiniger Variation des Parameters  $P_i$  auf die Varianz des Ausgangs gemittelt über mehrere Parameterkombinationen beschreiben. Für rein additive Modelle (wie z.B. lineare Regressionsmodelle) wäre somit eine Priorisierung der Eingangsparameter bereits möglich, da keine Wechselwirkungen zwischen den Parametern zu erwarten sind. Kann diese Annahme allerdings nicht getroffen werden, müssen Effekte höherer Ordnung berücksichtigt werden, da Abhängigkeiten zwischen den Variablen bestehen. Diese Effekte werden wie auch schon bei der EEM über den Totaleffekt  $S_{Ti}$  beschrieben. Für die VBSA erfolgt die Analyse der Varianz des Parameters  $P_i$  inklusive sämtlicher Faktorkombinationen in Abhängigkeit der Varianz der Zielfunktion:

$$S_{Ti} = \frac{E_{P_{-i}}(V_{P_i}(F(P)|P_{-i}))}{V(F(P))} = 1 - \frac{V_{P_{-i}}(E_{P_i}(F(P)|P_{-i}))}{V(F(P))}. \quad (2.7)$$

Mit Hilfe von  $S_{Ti}$  können also Wechselwirkungen erfasst werden. Im Gegensatz zu  $S_i$  ist die Summe der Sensitivitätsmaße aller Parameter jedoch nicht eins, da Parameterkombinationen mehrfach in die Berechnung einfließen, je nachdem zu welchem der Faktoren sie zugehörig sind.

Da die exakte Berechnung der Varianzen gemäß der Gleichungen 2.6 und 2.7 sehr viele Systemevaluationen benötigt, wurden mehrere approximative Verfahren vorgestellt, um die Anwendung der VBSA mit vertretbarem Aufwand durchführen zu können. Eine weit verbreitete Berechnungsvorschrift ist in [Jan99] und [SAA<sup>+</sup>10] beschrieben und sieht die Schätzung der Varianz und des Erwartungswerts gemäß der Gleichungen 2.8 und 2.9 vor.

$$V_{P_i}(E_{P_{-i}}(F(P)|P_i)) \approx V(F(P)) - \frac{1}{2N} \sum_{j=1}^N (F(B)_j - F(A_B^i)_j)^2. \quad (2.8)$$

$$E_{P_{-i}}(V_{P_i}(F(P)|P_{-i})) \approx \frac{1}{2N} \sum_{j=1}^N (F(A)_j - F(A_B^i)_j)^2. \quad (2.9)$$

Im ersten Schritt werden dafür zwei unabhängige Sampling-Matrizen  $A$  und  $B$  der Dimension  $N \times n_p$  erzeugt, wobei  $N$  die Anzahl an Samples kennzeichnet. Mit Hilfe dessen wird für jeden Parameter  $P_i$  jeweils eine dritte Matrix  $A_B^i$  erzeugt, welche der Matrix  $A$  entspricht aber die  $i$ -te Spalte aus  $B$  enthält. Für die Approximation der Varianz und des Erwartungswerts muss also für jeden

Parameter nur die neu zusammengesetzte Matrix  $A_B^i$  ausgewertet werden. Die Anzahl an Systemevaluation ergibt sich somit gemäß Gleichung 2.10 und liegt normalerweise über dem Wert der EEM (vgl. Gleichung 2.1).

$$n_{VBSA} = N \cdot (n_p + 2) \quad (2.10)$$

Sowohl bei der EEM als auch der VBSA bestimmt die Anzahl an Samples  $r$  beziehungsweise  $N$  die Güte des Ergebnisses der Sensitivitätsanalyse. Wird der Parameterraum nicht ausreichend abgetastet, bleiben modellspezifische Effekte in nicht untersuchten Bereichen möglicherweise unberücksichtigt. Die Aufgabe besteht darin, die Sampling-Parameter so zu wählen, dass gerade so viele Systemevaluationen wie nötig durchgeführt werden, um alle Effekte erforschen zu können. Für den Fall, dass das Modellwissen nicht ausreicht, um eine geeignete Abschätzung dieser Variablen durchzuführen, existieren Konvergenzanalysen mit Hilfe derer die Ergebnisse der Einflussanalyse im Nachhinein verifiziert werden können. Diese basieren auf der wiederholten Durchführung von Sensitivitätsanalysen (SAs) mit steigender Anzahl von Samples. Ändert sich das Ergebnis oberhalb einer Samplingzahl nicht mehr, ist Konvergenz erreicht [HKRW13]. Unter den verfügbaren Verfahren liefert der Bootstrapping-Ansatz von [SPW16] die besten Ergebnisse [VMCN15]. Dies wurde ebenfalls in einer vorausgegangenen Veröffentlichung des Autors bestätigt [FFFG19a]. Die Grundidee besteht darin, die SA mit konstanter Sampling-Größe, aber unterschiedlichen Parametersets mehrfach nacheinander durchzuführen und ein Konfidenzintervall der Sensitivitätsmaße zu bestimmen. Dabei werden  $n_B$  neue Sampling-Pläne mit  $n$  Parametersets aus der einmalig ausgewerteten Grundgesamtheit (mit  $n_{EEM}$  beziehungsweise  $n_{VBSA}$  Samples) durch ein Ziehen mit Zurücklegen generiert. Die kritische Sampling-Anzahl, bei der ein vorher definiertes Konvergenzkriterium erstmalig erreicht ist, stellt somit die mindestens benötigte Anzahl an Systemevaluationen dar. Zur Prüfung der Konvergenzreichung einer Sample-Größe  $n$  schlägt [SPW16] die Berechnung der Breite des Konfidenzintervalls  $R_{KI,i}$  bei  $n_B$ -maliger Durchführung der SA zu einem gegebenen Konfidenzniveau  $\kappa_{Konf}$  vor:

$$R_{KI,i} = S_i^{OG} - S_i^{UG}. \quad (2.11)$$

$S_i^{OG}$  und  $S_i^{UG}$  beschreiben die oberen und unteren Intervallgrenzen des jeweiligen Sensitivitätsmaßes.

### 2.3.2 Ermittlung valider Parameterwertebereiche

Neben der Möglichkeit, den Parameterraum in seiner Dimensionalität zu reduzieren, besteht ein großes Potential darin, die Wertebereiche jedes Parameters zu untersuchen und diese auf einflussreiche Grenzen zu beschränken. In der Praxis erfolgt die Definition der oberen und unteren Schranken meistens empirisch oder auf Grundlage von Erfahrungs- und Expertenwissen des Entwicklers. Auf eine analytische Herleitung der Grenzen wird oftmals verzichtet, da diese mit zu hohem Rechenaufwand verbunden wäre [MNPLW13]. Die Bestimmung verlässlicher Grenzen kann allerdings ausschlaggebend für das Optimierungsergebnis und vorgelagerte Sensitivitätsanalysen sein. Sind beispielsweise große Teile des Wertebereichs ohne Einfluss, würden bei der EEM viele sehr kleine Elementareffekte entstehen welche die Sensitivitätsmaße durch die Mittelung zu kleinen Werten tendieren lassen. Der entsprechende Parameter würde somit unter Umständen fälschlicherweise nicht für eine Optimierung priorisiert werden, obwohl es einen vergleichsweise kleinen Teil des Wertebereichs mit hohem Einfluss gibt. Korrekt definierte Grenzwerte stellen also eine notwendige Bedingung für die Durchführung von SAs dar [SRA<sup>+</sup>08].

Die Ermittlung valider Wertebereiche ist insbesondere im Bereich der Systemidentifikation Gegenstand der Forschung. Aufgrund von oftmals geringem Wissen über das Übertragungsverhalten des modellierten Systems können die freien Parameter nicht verlässlich durch Erfahrungs- oder Expertenwissen geschätzt werden. Ein weit verbreiteter Ansatz ist deshalb, das Problem an einem Ersatzmodell mit Hilfe algebraischer Methoden zu lösen. In [BABPA18] werden gültige Grenzen für Eingangsparameter eines genetischen Algorithmus bestimmt, indem experimentelle Brute-Force-Auswertungen an einem analytischen Metamodell durchgeführt werden. In den Arbeiten von [CR03, CR06] werden für die Parametrierung eines komplexen Regelsystems der nichtlineare Block mit Hilfe von Polynomen linearisiert und somit eine Überführung in ein einfacheres Optimierungsproblem ermöglicht. Die Lösung des Problems sieht dann eine Bestimmung gültiger Parametergrenzen vor, welche die Wertebereiche ohne Vernachlässigung relevanter Parametrierungen beschränken. Einen Überblick über weitere Anwendungsbeispiele im Kontext der Modellidentifikation liefert [MNPLW13]. Ist eine Ersatzmodellbildung oder analytische Bestimmung valider Grenzen nicht möglich, werden diese meist konservativ abgeschätzt, da eine experimentelle Auswertung mit hohem Rechenaufwand verbunden wäre.

## 2.4 Automatisierte Optimierung von Applikationsparametern

Wie bereits in Abschnitt 2.1 beschrieben, lässt sich die Applikation von Systemparametern als Optimierungsproblem auffassen. In diesem Abschnitt wird zunächst auf die Grundlagen der Optimierung eingegangen und das allgemeine Optimierungsproblem formuliert. Im Anschluss wird von der allgemeinen Definition auf die konkrete Anwendung für die Applikation automatisierter Fahrfunktionen geschlossen und das im Rahmen dieser Arbeit behandelte Applikationsproblem beschrieben. Der nächste Abschnitt stellt in Frage kommende Optimierungsverfahren vor, welche bereits breite Anwendung bei vergleichbaren Rahmenbedingungen finden. Zuletzt erfolgt noch eine kurze Einführung in Robustheitsanalysen, welche oft im Zusammenhang mit Optimierungen eingesetzt werden, um die Stabilität gefundener Lösungen zu bewerten.

### 2.4.1 Optimierungsproblem

Der Begriff Optimierung bezeichnet im Allgemeinen die Suche der besten Lösung eines bestimmten Problems. In der Mathematik wird darunter die Minimierung einer Gütefunktion  $f(x)$  mit Hilfe der Entscheidungs- oder Designvariablen  $x$  unter Berücksichtigung von Ungleichheits- und Gleichheitsbedingungen  $g(x)$  bzw.  $h(x)$  verstanden. Die Zielfunktion  $f(x)$  besteht entweder aus einem Gütefunktional (einkriterielle Optimierung) oder einem Gütefunktionsvektor (mehrkriterielle Optimierung). Da für die Entwicklung von Fahrzeugsystemen häufig mehrere Anforderungen vereint werden müssen (vgl. V-Modell 2.1), liegt im Kontext der Applikation automatisierter Fahrfunktionen selten nur ein einziges Gütekriterium vor. Eine allgemeine Definition als Gütefunktions-Vektor  $\vec{f}(x)$  ist in Gleichung 2.12 gegeben, wobei  $m$  die Anzahl der Zielfunktionen beschreibt.

$$\vec{f}(\vec{x}) = \begin{pmatrix} f_1(\vec{x}) \\ f_2(\vec{x}) \\ \vdots \\ f_m(\vec{x}) \end{pmatrix}. \quad (2.12)$$

Die Entscheidungsvariablen werden durch einen  $n$ -dimensionalen Vektor  $\vec{x}$  beschrieben:

$$\vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}. \quad (2.13)$$

Ein multikriterielles Optimierungsproblem lässt sich somit wie folgt definieren [Deb01]:

$$\begin{aligned} & \text{Minimiere } \vec{f}(\vec{x}) \\ & \text{mit } \vec{g}(\vec{x}) \geq 0 \\ & \text{und } \vec{h}(\vec{x}) = 0 \end{aligned} \quad (2.14)$$

wobei gilt  $\vec{x} \in \mathbb{R}^n$ ,  $\vec{f}(\vec{x}) \in \mathbb{R}^m$ ,  $\vec{g}(\vec{x}) \in \mathbb{R}^j$ ,  
 $\vec{h}(\vec{x}) \in \mathbb{R}^k$

Ziel ist, einen Vektor  $\vec{x}^*$  zu finden, welcher die Vektorfunktion  $\vec{f}(\vec{x})$  unter Einhaltung der  $j$  Ungleichheitsbedingungen  $\vec{g}(\vec{x})$  und  $k$  Gleichheitsbedingungen  $\vec{h}(\vec{x})$  minimiert. Mit Hilfe der Gleichungen 2.14 lässt sich problemlos ein einkriterielles Optimierungsproblem formulieren, wenn der Gütefunktionsvektor nur eine Zielfunktion enthält ( $m = 1$ ). Auch die Umwandlung in ein Maximierungsproblem ist möglich:

$$\min\{\vec{f}(\vec{x})\} = -\max\{-\vec{f}(\vec{x})\}. \quad (2.15)$$

Im Gegensatz zur einkriteriellen Optimierung können bei der multikriteriellen Optimierung teils konkurrierende Zielfunktionen vorliegen. Es ist somit für den Optimierungsalgorithmus nicht immer möglich eine Lösung zu finden. Vielmehr existiert eine Menge an Kompromisslösungen, von denen der Anwender eine für die gegebene Situation beste Lösung auswählen muss [CLVV<sup>+</sup>07]. Wenn eine Auswahl an Parametersets gefunden wurde, mit denen eine Verbesserung eines Gütefunktional nur mit gleichzeitiger Verschlechterung mindestens einer anderen Zielfunktion verbunden ist, liegt Pareto-Optimalität vor (vgl. [Par64]). Zur Lösung dieser Probleme existieren verschiedene Methoden. Alternativ lässt sich das multikriterielle in ein einkriterielles Problem überführen. Die Überführung kann entweder im Vorhinein (a-priori), im Nachhinein (a-posteriori) oder progressiv während der Optimierung passie-

ren [Deb01, CLVV<sup>+</sup>07]. Beispiele sind die *Methode der gewichteten Metrik*, *Goal Programming* oder die  $\epsilon$ -*constraint-Methode*. Eine detaillierte Beschreibung der Verfahren kann [Sch18] entnommen werden. Im Rahmen dieser Arbeit wird die *Methode der gewichteten Summe* verwendet, da sich die Systemanforderungen der Fahrfunktion direkt in Gewichtungsfaktoren  $w_i$  überführen lassen:

$$f(x) = \sum_{i=1}^m w_i f_i(x) \quad \text{mit} \quad w_i \in \{0, 1\} \quad (2.16)$$

Für die Anwendung der Methode sollten die Zielfunktionen normiert sein, sodass sie bezüglich ihrer Wertebereiche gleich gewichtet sind. Die Summe der Gewichtungsfaktoren wird meist zu eins gewählt ( $\sum_{i=1}^m w_i = 1$ ), sodass das Optimierungsergebnis gleich wie die Gütefunktionale definiert ist. Im Gegensatz zur Entscheidungsfindung aus der Pareto-Menge wird somit eine Präferenzierung im Vorhinein festgelegt, sodass nur eine optimale Lösung entsteht. Durch Variation der Wichtungen können somit unterschiedliche Priorisierungen und Ausprägungen des Ergebnisses erzeugt werden [Mie12].

Das Optimierungsziel für die Applikation automatisierter Fahrfunktionen besteht darin, sämtliche Kundenanforderungen zu erfüllen und das Fahrerwunschverhalten herbeizuführen. Dafür ist es notwendig, eine gleichermaßen gute Performance in allen für das System repräsentativen Situationen zu erreichen. Die Applikation im Fahrversuch erfolgt somit meist szenarienbasiert (vgl. Abschnitt 2.1) mit teils szenarienabhängigen Definitionen des Zielfunktionsvektors  $f(x)$ . Neben konkurrierenden Gütefunktionen können somit auch Konflikte bei der gleichzeitigen Zielerreichung in allen Szenarien entstehen, wenn die Software beispielsweise für Parkiermanöver und Autobahnsszenarien nur eine Kompromisslösung anbietet. Basierend auf der in Abbildung 2.3 vorgestellten Wirkkette liefert Abbildung 2.8 eine schemenhafte Darstellung des in dieser Arbeit bearbeiteten Optimierungsproblems. Die Entscheidungsvariablen sind in diesem Fall die Applikationsparameter, welche nur im Modul *Handlungsplanung und -ausführung* berücksichtigt werden und mit Hilfe von Wertebereichen in Form von Ungleichheitsbedingungen beschränkt sind. Die Auswertung einer Systemevaluation beinhaltet die Erprobung des jeweiligen Parametersatzes für alle Szenarien und kann entweder im Fahrversuch oder simulativ erfolgen. Auch wenn sich durch eine virtuelle Erprobung die Auswertung deutlich beschleunigt, ist die Rechenzeit immer noch vergleichsweise hoch.

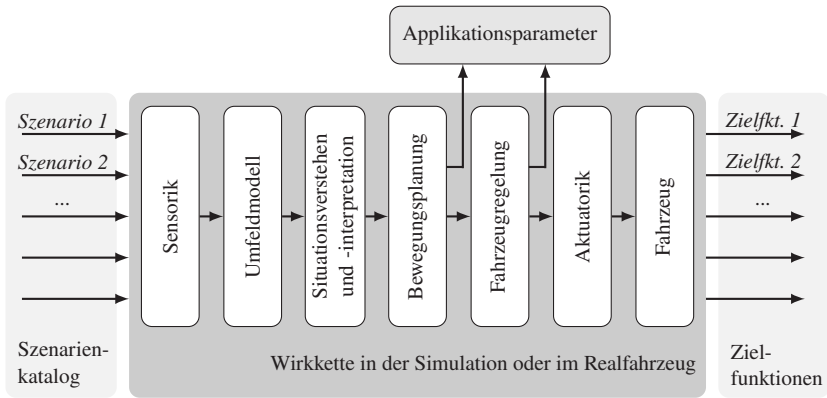


Abbildung 2.8: Schematische Darstellung des in dieser Arbeit behandelten Optimierungsproblems [FFFG20]

## 2.4.2 Optimierungsverfahren

In der Literatur existieren eine Vielzahl an Optimierungsverfahren, welche sich je nach Charakteristik des Optimierungsproblems unterschiedlich gut für dessen Lösung eignen. Im Allgemeinen lassen sich gängige Methoden in die Kategorien *gradientenbasierte* Verfahren und *gradientenfreie* Verfahren einteilen. Optimierer der ersten Klasse setzen die Verfügbarkeit von Ableitungen der Zielfunktion voraus und basieren auf deterministischen, mathematischen Operationen. Auch wenn die Zielfunktion nicht in analytischer Form gegeben ist, lässt sich dessen Gradient mit Hilfe von Differenzenquotienten approximieren. Neben der Tatsache, dass dadurch deutlich mehr Systemevaluationen nötig wären, arbeiten diese Algorithmen meistens sequentiell und bieten wenig Potential zur Parallelisierung. Desweiteren besteht gerade bei nichtlinearen Problemen die Gefahr, zu schnell in Richtung lokaler Optima zu konvergieren [Deb14]. Da bei dem Optimierungsproblem gemäß Abbildung 2.8 die Auswertung der Parametersets insbesondere bei sequentieller Berechnung sehr rechenaufwändig wäre und Nichtlinearitäten nicht auszuschließen sind, eignen sich gradientenbasierte Verfahren nicht für das in dieser Arbeit adressierte Problem. Gradientenfreie Optimierungsalgorithmen stehen dem gegenüber und basieren auf der direkten Suche optimaler Lösungen im Suchraum. Da das Fortschreiten nicht durch die Ableitung der Zielfunktion bestimmt ist, erlauben diese Verfahren

eine parallele Auswertung von Parametersets. Die teilweise Nutzung von Heuristiken bei der Lösungssuche führt allerdings dazu, dass die Optimierer nicht immer deterministisch arbeiten und ebenfalls Gefahr laufen, in lokalen Optima zu enden [RS13]. Aufgrund der Charakteristik des Applikationsproblems und einer im Vergleich kürzeren zu erwartenden Rechenzeit erscheinen die gradientenfreien Verfahren jedoch insgesamt vielversprechender und werden deshalb im Folgenden weiter thematisiert.

Eine weitere Unterteilung lässt sich in *lokale* und *globale* Suchmethoden vornehmen. Die erste Unterkategorie enthält Algorithmen, welche darauf basieren, neue Lösungen immer im direkten Umfeld zuvor ausgewerteter Parametersets zu suchen. Diese Verfahren erweisen sich dann als vielversprechend, wenn eine stetige, lineare Zielfunktion vorliegt. Sind allerdings mit Nichtlinearitäten zu rechnen, kann die optimale Lösung unter Umständen nicht erreicht werden, da keine ausgeprägte Breitensuche durchgeführt wird. Globale Verfahren begegnen diesem Nachteil und führen die Tiefen- und Breitensuche gleichermaßen durch. Im Zweifel kann sich die Konvergenzgeschwindigkeit damit stark reduzieren, jedoch ist beim Vorliegen von guten Lösungen an diskreten Peaks im Suchraum oder bei nichtlinearen, un stetigen Problemen mit einer verbesserten Performance zu rechnen [RS13, CSV09]. Die globalen Suchverfahren umfassen eine große Anzahl an Optimierungsmethoden (z.B. *Simulated Annealing*, *Response Surface Methods*, *Genetische Algorithmen*, *Partikelschwarmoptimierung*). Die Eignung dieser Verfahren zur Lösungsfindung hängt maßgeblich von dem untersuchten Problem ab. Wenn die Rahmenbedingungen aber einen erfolgreichen Einsatz eines speziellen Optimierers versprechen (z.B. basierend auf Vergleichsstudien), wird dieser bessere Ergebnisse als eine generalisierte Methode erbringen (*No-Free-Lunch-Theorem*, vgl. [Wei09]).

Aufgrund der erfolgreichen Anwendung genetischer Algorithmen (vgl. [Sit14, Kah13, Haa04, Jan08]) und der Partikelschwarmoptimierung (vgl. [Osc19, Pfe19, Pil17]) für Applikationsprobleme in der Fahrzeugtechnik werden diese beiden Verfahren für die Lösung des in dieser Arbeit beschriebenen Problems in Betracht gezogen. Eine experimentelle Vergleichsstudie beider im Folgenden detailliert beschriebenen Methoden erfolgt in Kapitel 4.



## Partikelschwarmoptimierung

Die Partikelschwarmoptimierung (PSO) zählt zu der Klasse populationsbasierter Verfahren, welche auch die evolutionären Algorithmen einschließt. Die PSO orientiert sich an dem biologischen Schwarmverhalten von Vögeln, dessen Bewegung durch die Suche nach einem konkreten Ziel wie beispielsweise Rastplätzen oder Futterstellen bestimmt wird. Erstmals wurden die Verhaltensregeln in [EK95a, EK95b] formalisiert und für die Simulation von Fisch- und Vögelschwärmen eingesetzt. Im Gegensatz zu anderen naturinspirierten Heuristiken (z.B. evolutionäre Optimierer) arbeiten die Individuen des Schwarms (oder der Population) zusammen an der Lösungssuche anstatt miteinander zu konkurrieren. Die Grundidee ist, dass die Partikelpopulation so lange durch den Suchraum bewegt wird, bis eine gute Lösung gefunden wird. Dabei erfolgt eine Neuberechnung der Position der Individuen in jeder Iteration.

Die Intelligenz, mit der sich die Individuen fortschreiten, basiert auf der Kombination von drei Operationen, welche auch in dem PSO-Algorithmus Anwendung finden: *Evaluation*, *Vergleich* und *Imitation* [Ken10]. Zu Anfang jeder neuen Partikel-Positionierung steht die *Evaluation*, welche den momentanen Zustand bewertet. Nur durch Kenntnis einer Bewertung aktueller Positionen kann zielgerichtet im Sinne einer Verbesserung gehandelt werden. Beim PSO wird dieser Schritt durch Auswertung der einkriteriellen Zielfunktion  $f(x)$  (auch Fitnessfunktion genannt) vorgenommen. Im Anschluss folgt der *Vergleich* beziehungsweise die Einordnung der Güte jedes Individuums. Während dies in der Natur beispielsweise durch Erfahrungswerte, Wissen oder Vergleich mit anderen Partikeln geschieht, wird die Fitness in der algorithmischen Umsetzung dem jeweils historisch bestem Wert sowie im Suchraum umliegenden Fitness-Bewertungen gegenübergestellt. Im dritten Schritt erfolgt die *Imitation*, bei dem basierend auf den zwei vorherigen Phasen Konsequenzen für „gute Verhaltensweisen“ abgeleitet werden. Da damit immer eine Interpretation verbunden ist, welche in der Natur subjektiv sein kann, ist dieser Schritt weder beim Algorithmus noch dem biologischen Vorbild deterministisch.

Die drei zuvor beschriebenen Phasen lassen sich mathematisch wie folgt beschreiben: In einem gegebenen Parameterraum wird zunächst eine zufällig gesampelte Anfangspopulation der Größe  $N$  erzeugt. Jedem Individuum (bzw. Parameterset)  $n$  der Population wird zum Zeitpunkt  $t$  eine Position  $\vec{X}_t^n \in \mathbb{R}^{n_p}$  und ein Geschwindigkeitsvektor  $\vec{V}_t^n \in \mathbb{R}^{n_p}$  zugeordnet. Zusätzlich wird der Pa-

rametersatz mit der historisch besten Fitness  $\vec{X}_{t,best}^n \in \mathbb{R}_p^n$  jedes Partikels sowie der global besten Fitness  $\vec{G}_t = \max_{n \in \{1, \dots, N\}} \vec{X}_{best}^n$  gespeichert. Die Aufgabe der *Imitation* ist nun, basierend auf der Orientierung an der historisch größten Fitness und dem für alle sichtbaren global besten Partikel die Positions- und Geschwindigkeitsvektoren anzupassen. Die Bewegung des Schwarms richtet sich nach folgenden Bewegungsgleichungen:

$$\vec{V}_t^n = \cdot \vec{V}_{t-1}^n + c_1 \cdot \vec{r}_t^n \cdot (\vec{X}_{t,best}^n - \vec{X}_{t-1}^n) + c_2 \cdot \vec{s}_t^n \cdot (\vec{G}_t - \vec{X}_{t-1}^n) \quad (2.17)$$

$$\vec{X}_t^n = \vec{X}_{t-1}^n + \vec{V}_t^n \quad (2.18)$$

Die Konstanten  $c_1$  und  $c_2$  regulieren die Orientierung des Partikels an seiner individuell besten Lösung und dem global besten Parametersatz. Die Faktoren  $\vec{r}_t^n \in \mathbb{R}^{n_p}$  und  $\vec{s}_t^n \in \mathbb{R}^{n_p}$  stellen Zufallsvektoren mit Einträgen im Bereich  $[0, 1]$  dar und werden in jedem Schritt neu berechnet. Sie geben dem Algorithmus die Möglichkeit aus lokalen Optima zu entkommen, führen aber dazu, dass der PSO nicht deterministisch arbeitet. In einigen Ausführungen wird ebenfalls die Definition einer Maximalgeschwindigkeit vorgeschlagen, um ein Überspringen von Optima im Suchraum bei hohen Gradienten zu verhindern. Der Ansatz läuft allerdings Gefahr, nur sehr langsam zu konvergieren [Bog13]. Bei Verwendung der Gleichungen 2.17 und 2.18 sollte berücksichtigt werden, dass von einem Maximierungsproblem ausgegangen wird. Eine Übertragbarkeit auf Minimierungsprobleme ist aber mit Hilfe von Gleichung 2.15 problemlos möglich. Die Bewegung der Partikel stellt Abbildung 2.9 beispielhaft dar. In umfangreichen Studien konnte nachgewiesen werden, dass der PSO nach einer gewissen Zeit immer gegen einen Punkt konvergiert [JLY07].

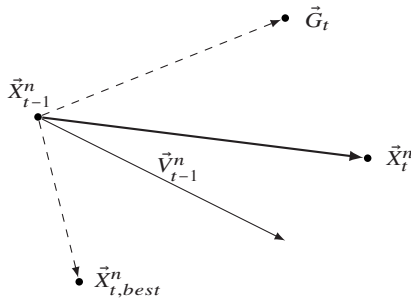


Abbildung 2.9: Bewegung eines Partikels im Suchraum beim PSO-Algorithmus [FFFG20]

## Genetischer Algorithmus

Genetische Algorithmen (GAs) gehören genau wie der PSO zur Klasse populationsbasierter Verfahren und orientieren sich an der biologischen Evolutionstheorie, welche Darwin's Prinzip des „Survival of the fittest“ nutzt, um das Überleben und die Weiterentwicklung bestimmter Spezies zu erklären [Rec94]. Demnach haben nur solche Organismen und Lebensformen Bestand, die sich kontinuierlich an veränderte Umweltbedingungen anpassen. Die dafür zum Einsatz kommenden Grundprinzipien *Selektion*, *Rekombination* und *Mutation* werden auch im GA verwendet um eine optimale Lösung im Suchraum zu finden. Im Gegensatz zur PSO arbeiten die Individuen der Population allerdings gegeneinander, indem sie den in der Natur stattfindenden Konkurrenzkampf unter den Spezies abbilden. In jeder Iteration entsteht basierend auf den Evolutionsprinzipien somit eine neue Generation mit mutierten, rekombinierten oder aus der Vorgeneration übernommenen Individuen. Die Übertragung des Konzepts auf den GA erfolgte in den 70er Jahren und ist ausführlich in [H<sup>+</sup>92, Hol92] beschrieben.

Analog zum biologischen Vorbild ist jedes Individuum durch seinen Genotyp (oder Chromosom) charakterisiert. Genotypen sind wiederum in Gene unterteilt, welche im Algorithmus die als Binärstring codierten Parameter des Suchraums sind. Der Wertebereich muss also diskretisiert sein, wobei jeder mögliche Parameterwert durch eine Binärkombination dargestellt ist. Eine Nutzung reell codierter Skalen ist ebenfalls möglich, aber mit deutlich höherem Rechenaufwand verbunden [Hol92]. In der Praxis hat sich für kontinuierliche

Skalen die Festlegung einer minimalen Anzahl möglicher Werte pro Parameter  $n_{P_i}^{min}$  als sinnvoll erwiesen, die mit konstantem Abstand entlang des Wertebereichs definiert werden. Die binäre Repräsentation muss dann über  $k_i$  Bits verfügen, sodass folgende Bedingung erfüllt ist:  $2^{k_i} > n_{P_i}^{min}$  [FFFG20]. Durch die Aneinanderreihung aller Parameter-Strings entsteht der individuelle Genotyp, welcher die Grundlage für die Anwendung der drei GA-Operatoren bildet. Abbildung 2.10 zeigt drei Bit-Codierungen von Individuen sowie eine beispielhafte Anwendung des Rekombinations- und Mutationsoperators.

Ausgehend von einer Startpopulation werden in der *Selektions*-Phase zunächst überdurchschnittlich gute Individuen ausgewählt und in die Folgegeneration kopiert (sogennanter Elitismus), während schlechte Lösungen eliminiert werden. Je höher der Zielfunktionswert eines Parametersets, desto höher ist die Chance übernommen zu werden. Aus der verbleibenden Menge an Individuen wird außerdem zufällig eine Untermenge gebildet, auf die im Anschluss Rekombinations- und Mutationsoperatoren angewendet werden. Für die Auswahl der Parametersätze existieren verschiedene Verfahren wie beispielsweise die Roulette-, Ranking- oder Turnier-Selektion. Weil die Turnierselektion ein gleichwertiges Konvergenzverhalten wie seine Alternativen bei gleichzeitig geringem Rechenaufwand bietet, kommt sie oft zum Einsatz [GD91]. Das Grundprinzip besteht darin, dass wiederholt eine beliebige Menge an Individuen zufällig ausgewählt werden und der entsprechend des Zielfunktionswerts beste Parametersatz selektiert wird. Die Konvergenz des Algorithmus wird maßgeblich durch den Selektionsdruck bestimmt, welcher über die Parametrierung der Selektionswahrscheinlichkeit  $p_s$  beeinflussbar ist. Je größer  $p_s$ , desto mehr Individuen werden für die Turnierselektion ausgewählt. Damit kann zwar schneller eine Lösung gefunden werden, diese ist allerdings möglicherweise nur lokal optimal.

Die Individuen der selektierten Untermenge werden im Anschluss durch Anwendung von *Crossover*-Operationen rekombiniert. Dabei erfolgt eine zufällige Auswahl von zwei Parametersets, dessen Genotypen stückweise ausgetauscht werden. Der Tausch kann anhand einer (1-Punkt-Crossover) oder mehrerer (N-Punkt-Crossover) Kreuzungsstellen erfolgen. Die Anzahl an Rekombinationen pro Generation ist zufällig, kann aber über die Rekombinationswahrscheinlichkeit  $p_r$  beeinflusst werden. Im Extremfall von  $p_r = 1$  erfolgt eine Kreuzung aller selektierten Elternindividuen. Das 1-Punkt-Crossover ist in Abbildung 2.10 dargestellt und hat den Vorteil schneller zu konvergieren als

das N-Punkt-Crossover ( $N > 1$ ). Es führt allerdings zu einer weniger ausgeprägten Breitensuche [GKK13].

Der Mutationsoperator wird im dritten Schritt angewendet und fördert die Erkundung nicht erforschter Bereiche im Parameterraum. Durch zufällige Variation genetischer Informationen werden rekombinierte Parametersets erneut leicht verändert und somit neu positioniert. Das Ziel der *Mutation* ist es, eine zu schnelle Konvergenz zu verhindern. Darüber hinaus wird sichergestellt, dass die Wahrscheinlichkeit, jeden im Definitionsbereich liegenden Punkt zu erreichen, immer größer als Null ist. Die Anpassung der Genotypen erfolgt normalerweise durch das ein- oder mehrfache Invertieren eines Bits (Bit-Flip, vgl. Abbildung 2.10) oder den Austausch zweier Stellen entlang des Binärstrangs. Genau wie die Rekombination kann auch die Mutation unterschiedlich im Hinblick auf die Breitensuche parametrisiert werden. Die Anzahl an Mutationsoperationen pro Iteration wird über die Mutationswahrscheinlichkeit  $p_m$  beeinflusst. Eine zu häufig durchgeführte Veränderung der Individuen sollte allerdings vermieden werden, da der GA ansonsten eher einer Zufallssuche gleicht und möglicherweise niemals konvergiert [HLV98].

Zum Schluss erfolgt ein Zusammenführen der durch Elitismus in die neue Generation kopierten Lösungsmenge und der verarbeiteten Submenge so, dass die Populationsgröße konstant bleibt. In den meisten Fällen werden dafür die besten Individuen ausgewählt, es kann aber auch eine „Alterung“ implementiert werden um einer vorzeitigen Konvergenz vorzubeugen. Das bedeutet, dass Individuen, die eine bestimmte Anzahl an Iterationen unverändert geblieben sind automatisch entfernt werden [GKK13].

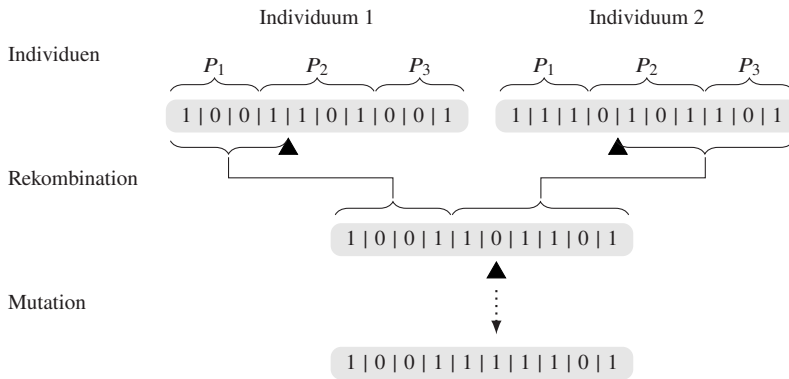


Abbildung 2.10: Rekombination und Mutation beim binär codierten genetischen Algorithmus [FFFG20]

### 2.4.3 Robustheitsanalysen

Robustheitsanalysen werden im Kontext von Parameteroptimierungen mit dem Ziel eingesetzt, die Stabilität von Parametersätzen gegenüber sich ändernden Rahmenbedingungen zu prüfen. Die Analyse kann dabei sowohl gegenüber parametrischen Änderungen im Suchraum als auch externen Einflüssen erfolgen. In der Literatur existieren bereits einige Verfahren für einkriterielle [Sta98, JS03, BS05] und multikriterielle [Hug01, DG05, BA06] Optimierungsprobleme. Da sich viele multikriterielle in einkriterielle Probleme mit Hilfe gängiger Verfahren umwandeln lassen (vgl. Abschnitt 2.4.1) wird im Folgenden insbesondere auf Robustheitsanalysen für eine einzige Zielfunktion  $f(x)$  eingegangen.

Die Bestimmung von Robustheitsmetriken erfordert normalerweise eine Analyse der direkten Umgebung der gefundenen Lösung. Für eine Untersuchung gegenüber parametrischen Unsicherheiten werden Funktionswerte innerhalb einer  $n_p$ -dimensionalen Hyperbox mit Radius  $\delta$  um den Ausgangszustand ausgewertet.  $n_p$  bezeichnet in diesem Fall die Anzahl an Parametern. Im Anschluss wird geprüft, inwiefern Abweichungen von der Ausgangslösung unterhalb einer vorher definierten Schwelle  $\eta$  bleiben. Eine Überschreitung würde bedeuten, dass das Robustheitskriterium nicht erfüllt ist. Abbildung 2.11 verdeutlicht das

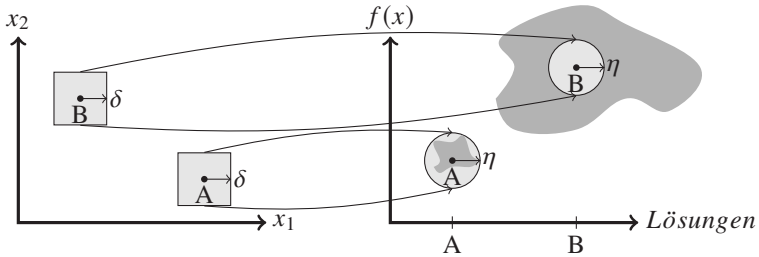


Abbildung 2.11: Links: Definition einer maximalen Hyperbox im *decision space* für die Lösungen A und B. Rechts: Auswertung der Zielfunktionswerte in einer  $\delta$ -Umgebung im *objective space* (dunkelgraues Polygon) und Prüfung gegenüber des Robustheitskriteriums definiert durch  $\eta$  (hellgraues Polygon). Lösung A erfüllt die Robustheitsbedingung, Lösung B nicht [FFFG20]

Prinzip für einen Parameterraum mit zwei Parametern  $(x_1, x_2)$ . Die innerhalb der Hyperbox mit Radius  $\delta$  gesampelten Parametersätze erzeugen Zielfunktionswerte im dunkelgrauen Polygon im *objective space*. Weil das der Lösung B zugeordnete Polygon die Hyperbox mit Radius  $\eta$  deutlich überschreitet, ist die Robustheitsbedingung in diesem Fall nicht erfüllt. Der zu A zugehörige Parametersatz kann allerdings als stabil gegenüber parametrischen Abweichungen bezeichnet werden.

Die Anwendung einer Robustheitsanalyse für einen Parametersatz  $x_S$  mit Zielfunktionswert  $f(x_S)$  erfordert zunächst die Erzeugung von  $H$  neuen Parametersätzen, die innerhalb einer  $\delta$ -Hyperbox gesampelt werden. Die zugehörigen Funktionswerte sind durch  $f(x_i)$  gegeben, wobei gilt:  $i = 1, \dots, H$ . Als Kriterium zur Entscheidungsfindung ob Robustheit erreicht ist, kann die mittlere Abweichung zwischen den neuen Funktionswerten  $f(x_i)$  und  $f(x_S)$  berechnet und  $\eta$  gegenübergestellt werden:  $\frac{1}{H} \sum_{i=1}^H \frac{|f(x_i) - f(x_S)|}{|f(x_S)|} < \eta$ . Ein alternatives Kriterium verhindert die Vernachlässigung von Ausreißern und wertet die maximale absolute Differenz aus:  $\max_{i=1, \dots, H} \frac{|f(x_i) - f(x_S)|}{|f(x_S)|} < \eta$ . Auch eine faktorabhängige Evaluierung ist möglich:  $\frac{|f(x_i) - f(x_S)|}{|f(x_S)|} < \eta_i, i = 1, \dots, H$  [DG05]. In der Praxis wird üblicherweise bei gegebenem  $\eta$  eine maximale Hyperboxgröße  $\delta_{max}$  bestimmt, welche eine der genannten Robustheitskriterien erfüllt. Eine Gegenüberstellung der verschiedenen Werte für  $\delta_{max}$  jeder Lösung ermöglicht dann eine Bewertung und einen Vergleich von Lösungen im Hinblick auf Robustheit. Dabei gilt: Je größer die maximal noch zulässige Hyperbox, desto

stabiler ist eine Lösung gegenüber parametrischen Abweichungen vom Ausgangszustand. Ein Wert für  $\delta_{max}$  kann entweder mit einem Optimierer oder iterativ gefunden werden. Ein iteratives Verfahren ist der *degree-of-robustness*-Ansatz von [BA06], welcher eine schrittweise Erhöhung der Hyperboxgröße ausgehend von einem Initialwert  $\delta_1$  vorsieht:  $\delta = \delta_1, \dots, k\delta_1$ . Eine Vergrößerung des Raums erfolgt so lange, bis die Robustheitsbedingung das erste Mal nicht mehr erfüllt ist. Das maximale Vielfache von  $\delta_1$  welches das Stabilitätskriterium noch erfüllt, bezeichnet den Robustheitsgrad *DoR* (*degree of robustness*) der jeweiligen Lösung. Je höher *DoR*, desto robuster ist ein Parametersatz zu bewerten.

Analog zu der Analyse des direkten parametrischen Umfelds im Parameterraum kann auch die Stabilität von Lösungen gegenüber externen Einflüssen untersucht werden. In Anlehnung an das in dieser Arbeit behandelte Optimierungsproblem (vgl. Abbildung 2.8) wäre beispielsweise eine Robustheitsanalyse gegenüber unterschiedlichen Szenarienausprägungen oder injizierten Fehlern entlang der Wirkkette (in der Sensorik, Umfeldmodell oder in der Fahrzeugdynamik) denkbar. In diesem Fall würden die Werte der Applikationsparameter konstant gehalten und stattdessen die Fehlerinjektionen systematisch variiert.

## 2.5 Fahrzeugmodelle

Wie Abbildung 2.8 zeigt, ist die Performance der automatisierten Fahrfunktion im Gesamtverbund neben der Umfeldwahrnehmung und -verarbeitung maßgeblich abhängig von den fahrdynamischen Eigenschaften des Fahrzeugs sowie der Aktuator- und Regelsysteme. Während Sensormodelle sowie die Softwariemodule der Umfeldverarbeitung und Handlungsplanung bereits in frühen Phasen der Entwicklung zur Verfügung stehen, werden Prototypen-Fahrzeuge mit entsprechender Aktuatorik erst später aufgebaut. Um dennoch eine virtuelle Applikation für unterschiedliche Derivate und eine Übertragbarkeit simulativ ermittelter Ergebnisse auf den Fahrversuch zu ermöglichen, kommen deshalb Fahrzeugmodelle zum Einsatz. Diese beschreiben das Übertragungsverhalten zwischen Fahrer- oder Fahrerassistenzsystem-Anforderungen und der resultierenden Fahrdynamik [Raj11].

Im Laufe der Jahre sind viele analytische Fahrzeugmodelle unterschiedlicher Komplexität entstanden. Die bekanntesten Vertreter sind kinematische Mo-



delle, Einspurmodelle, Zweispurmodelle sowie kommerziell genutzte Finite-Elemente-Modelle und Mehrkörpersysteme. Inwiefern sich ein Einsatz für den konkreten Anwendungsfall eignet, hängt von der Genauigkeit und Komplexität ab, die normalerweise im Zielkonflikt zueinander stehen. Dieser Zielkonflikt wurde vom Autor bereits in [FFFG19b] herausgestellt und lässt sich entsprechend Abbildung 2.12 darstellen. Während das Einspurmodell beispielsweise lediglich über zwei algebraisch einfach zu lösende Differentialgleichungen verfügt, erfordert die Entwicklung eines Mehrkörpersystem-Modells die aufwändige Modellierung sämtlicher Einzelkomponenten. Demgegenüber steht allerdings der Vorteil komplexerer Modelle, mehr Freiheitsgrade des Realfahrzeugs abbilden zu können und damit eine höhere Genauigkeit zu besitzen.

Mit dem Ziel diesem Trade-Off zu begegnen, sind in den letzten Jahren zunehmend Machine-Learning-Ansätze zum Einsatz gekommen, welche in vielen Situationen effizienter zu rechnen sind als analytische Modelle und die Möglichkeit bieten, fahrdynamische Zusammenhänge aus Messdaten zu lernen (vgl. [Cao08, HNR19]). Die am weitesten verbreiteten künstlichen neuronalen Netze (KNNs) weisen insbesondere dann Vorteile gegenüber klassischen Ansätzen auf, wenn das Übertragungsverhalten Nichtlinearitäten oder Unstetigkeiten aufweist und nur durch aufwändige mathematische Modellierungen abbildbar ist. Gleichzeitig muss die Generierung ausreichender Trainingsdaten mit vertretbarem Aufwand durchführbar sein [CB92]. Trotz der erfolgreichen Anwendung in vielen Bereichen steht den genannten Vorzügen der Nachteil gegenüber, dass künstliche neuronale Netze (KNNs) Black-Box-Charakter haben und die Ursache von Fehlprädiktionen somit nur sehr schwer nachzuvollziehen ist. Auch die große Menge an Trainingsdaten, die benötigt wird, um eine ausreichende Modellgüte zu erreichen, kann eine Herausforderung darstellen [H<sup>+</sup>09].

Im folgenden Abschnitt werden im ersten Teil zwei Vertreter analytischer Modelle beschrieben: Das weit verbreitete Einspurmodell sowie ein komplexes Mehrkörpermodell, welches bei der BMW AG kommerziell zur Funktionsentwicklung und -absicherung von Regelsystemen zum Einsatz kommt. Im Anschluss folgt eine Vorstellung KNN-basierter Verfahren zur Abbildung der Fahrdynamik und die detaillierte Beschreibung eines Netzwerktyps, welcher für den Einsatz als Fahrzeugmodell prädestiniert ist. Final werden noch Möglichkeiten aufgezeigt, die Vorteile analytischer und KNN-basierter Modelltypen zu kombinieren.

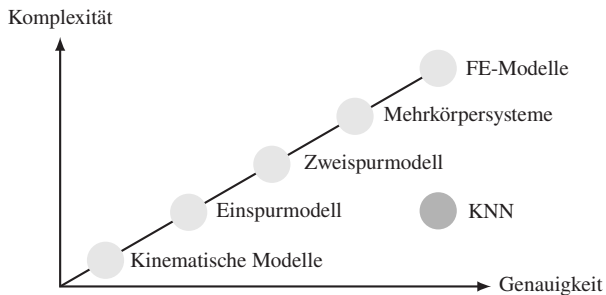


Abbildung 2.12: Komplexitäts-Genauigkeits-Trade-Off von Fahrzeugmodellen [FFFG19b]

## 2.5.1 Analytische Fahrzeugmodelle

Das im Folgenden beschriebene Einspurmodell (ESM) und kommerziell genutzte, sogenannte ISAR-Modell dienen dazu, dem Leser Unterschiede in der Komplexität analytischer Modelltypen und dem sinnvollen Anwendungsbereich dieser aufzuzeigen.

### Einspurmodell

Das lineare Einspurmodell ist eines der einfachsten analytischen Fahrzeugmodelle zur näherungsweise aber gleichzeitig plausiblen Beschreibung der Fahrzeug-Querdynamik. Es wurde erstmals in [RS40] eingeführt und kommt heutzutage aufgrund seiner geringen Komplexität und hohen Genauigkeit im dedizierten Anwendungsbereich vielfach für die Entwicklung fahrdynamischer Regelungssysteme zum Einsatz. Der Gültigkeitsbereich beschränkt sich auf den linearen Bereich der Querdynamik, welcher mit Überschreiten der Haftgrenze zwischen Reifen und Fahrbahn endet [MW14]. Da mit vielen Fahrerassistenzsystemen hochdynamische Betriebspunkte nie erreicht werden, reicht das ESM oftmals für die Erklärung des fahrdynamischen Übertragungsverhaltens in Simulationsumgebungen aus [WHLS15].

Die Modellierung beruht auf einigen vereinfachenden Annahmen, die im Folgenden aufgelistet sind [RS40, MW14]:

- Die Vorder- und Hinterachse sind jeweils zu einem mittig liegenden Rad zusammengefasst. Die Reifenaufstandspunkte liegen in der Fahrzeuglängsachse.
- Die Geschwindigkeit des Fahrzeugschwerpunkts ist entlang seiner Bahnkurve konstant
- Alle Hub-, Wank- und Nickbewegungen werden vernachlässigt
- Die Fahrzeugmasse ist im Fahrzeugschwerpunkt zusammengefasst
- Zwischen Vorder- und Hinterachse existiert eine konstante Radlastverteilung
- Rückstellmomente und Reifennachläufe infolge von Schräglaufwinkeln werden vernachlässigt
- Eine Lenkbarkeit der Hinterachse wird vernachlässigt

Es existieren zwar Weiterentwicklungen für den nichtlinearen Bereich oder mit zusätzlichen Wankfreiheitsgraden, auf diese soll im Folgenden aber nicht weiter eingegangen werden.

Die Geometrie des in dieser Arbeit beschriebenen ESMs inklusive im Folgenden genutzter Variablenbeschreibungen ist in Abbildung 2.13 dargestellt. Das Fahrzeug bewegt sich in jedem Betriebspunkt entlang des  $v$ -Vektors, der mit der Fahrzeuglängsachse den Schwimmwinkel  $\beta$  einschließt. Um das Fahrzeug in eine Kurvenbewegung zu lenken, müssen die lateralen Kräfte  $F_{l,f}$  und  $F_{l,r}$  an der Vorder- und Hinterachse aufgebracht werden. Diese ergeben sich aus den Schräglaufwinkeln  $\alpha_f$  und  $\alpha_r$ , welche zwischen den Bewegungsrichtungen und Längsachsen der Reifen liegen. Die Größe der lateral induzierten Kräfte lässt sich dann zusammen mit den reifenspezifischen Schräglaufsteifigkeiten  $c_{\alpha,f}$  und  $c_{\alpha,r}$  wie folgt berechnen:

$$F_{l,i} = c_{\alpha,i} \cdot \alpha_i \quad \text{mit} \quad i \in \{f, r\}. \quad (2.19)$$

Durch Aufstellen des Kräftegleichgewichts in  $y$ -Richtung und des Momentengleichgewichts um den Schwerpunkt  $SP$  ergeben sich die Bewegungsgleichun-

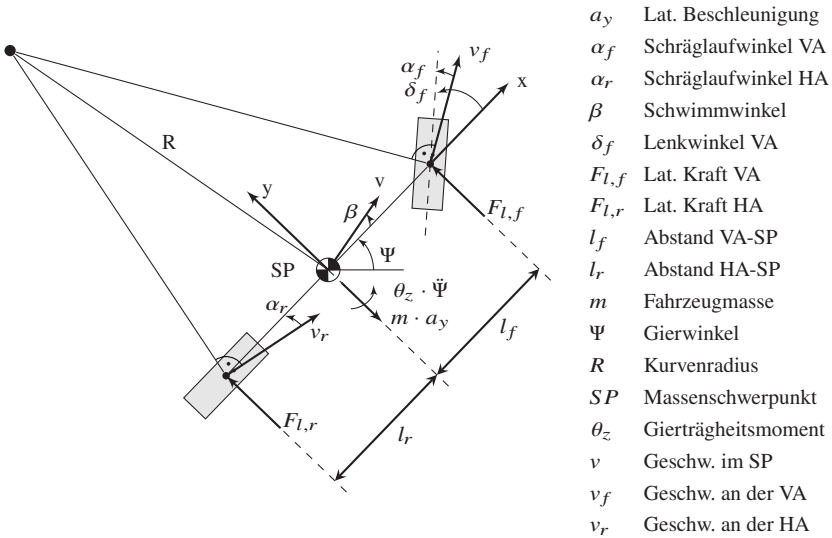


Abbildung 2.13: Geometrie des Einspurmodells mit vernachlässigter Hinterachselenkung. HA: Hinterachse, VA: Vorderachse [FFFG19b]

gen des ESMs (Gleichungen 2.20 und 2.21). Eine detaillierte Herleitung der mathematischen Beziehungen kann [Raj11] entnommen werden.

$$\dot{\beta}(t) = -\frac{c_{\alpha,f} + c_{\alpha,r}}{m \cdot v(t)} \cdot \beta(t) + \frac{mv(t)^2 - (c_{\alpha,r}l_r - c_{\alpha,f}l_f)}{mv(t)^2} \cdot \dot{\Psi}(t) \quad (2.20)$$

$$\Psi(t) = -\frac{c_{\alpha,r}l_r - c_{\alpha,f}l_f}{\theta_z} \cdot \beta(t) - \frac{c_{\alpha,r}l_r^2 + c_{\alpha,f}l_f^2}{\theta_z v(t)} \cdot \dot{\Psi}(t) + \frac{c_{\alpha,f}l_f}{\theta_z} \cdot \delta_f(t) \quad (2.21)$$

Das ESM modelliert den Schwimmwinkel  $\beta(t)$  und Gierwinkel  $\Psi(t)$  des Fahrzeugs in Abhängigkeit der Lenkwinkel-Anforderung  $\delta_f(t)$  und der aktuellen Geschwindigkeit in Längsrichtung  $v(t)$ . Weitere fahrzeugspezifische Größen wie die aktuelle Krümmung  $\kappa(t)$  oder Position des Fahrzeugs im Raum können aus den Modelloutputs berechnet werden. Eine Einbindung in einer Gesamtfahrzeugsimulation ist somit möglich.

Die Parametrierung des ESMs erfordert lediglich die Lösung eines sechs-dimensionalen, einkriteriellen Optimierungsproblems indem die freien Parameter  $m, c_{\alpha,f}, c_{\alpha,r}, l_f, l_r, \theta_z$  so eingestellt werden, dass ein minimaler Modellfehler entsteht. Die Dimensionen und Masse des Fahrzeugs liegen meistens bereits in Form von Produktspezifikationen vor oder können leicht gemessen werden. Die reifenspezifischen Eigenschaften  $c_{\alpha,f}$  und  $c_{\alpha,r}$  sowie das Gierträghheitsmoment  $\theta_z$  lassen sich experimentell bestimmen.

### Komplexe Mehrkörpersysteme

Mehrkörpersysteme (MKS) basieren typischerweise auf dem Zweispurmodell und erweitern dieses um Modelle realer physikalischer Komponenten. Der Fokus liegt darauf, die Kinematik und Elastokinematik der Radaufhängung, der Aktuatorik, des Fahrwerks und der Reifen realitätsgetreu nachzubilden. Das Zweispurmodell besteht als Weiterentwicklung des ESMs aus mindestens fünf Massen (Fahrzeugaufbau und vier Radträger inklusive der Räder) und verfügt über sechs Starrkörperfreiheitsgrade. Mit Hilfe eines detaillierten Reifenmodells und einer vereinfachten Radaufhängungskinematik können Hubbewegungen der Räder und damit fahrdynamisch bedingte Achs- und Radlastverteilungen abgebildet werden. Folglich ist eine Modellierung der Fahrzeugaufbaubewegung und Abbildung von Hub-, Nick- und Wankbewegungen möglich [Unt13].

Das in dieser Arbeit beschriebene ISAR-Modell verfügt über 14 Freiheitsgrade und wird für die Entwicklung, Parametrierung und Erprobung fahrdynamischer Regelsysteme eingesetzt [KPD<sup>+</sup>06, KD13]. Das Modell ist modular aufgebaut und basiert wie die meisten MKS ebenfalls auf einem Zweispurmodell. Die grundlegende Struktur ist in Abbildung 2.14 gezeigt, wobei die abgebildeten Verbindungen mechanischer Komponenten üblicherweise mit Hilfe von Feder-Dämpfer-Elementen oder Gelenken realisiert sind. Das Chassis bildet die Basis und ist als Starrkörper mit sechs Freiheitsgraden modelliert. Die Radaufhängung verfügt über vier Freiheitsgrade und ist mit Hilfe dreidimensionaler Kennfelder parametrierbar, sodass fahrzeugspezifische Kinematiken und Elastokinematiken abbildbar sind. Ähnlich wie die Aufhängung lassen sich

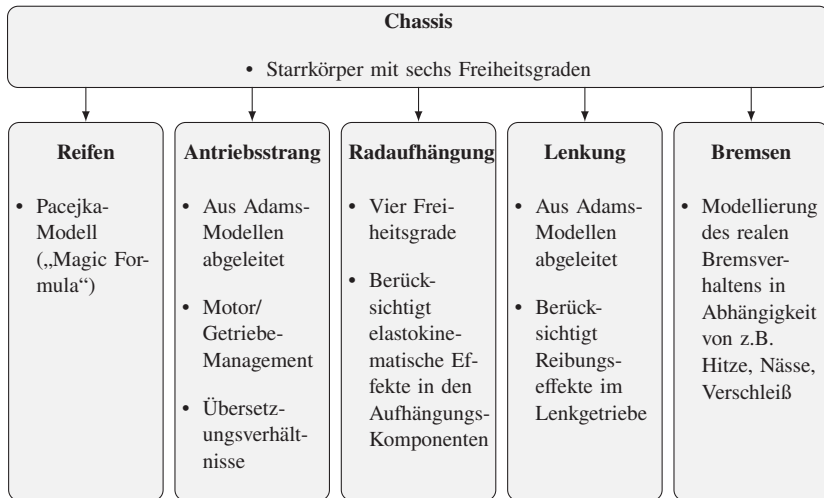


Abbildung 2.14: Grundlegender Aufbau des ISAR-Modells. Die Verbindungen mechanischer Komponenten sind mit Feder-Dämpfer-Elementen realisiert [FFFG19b]

auch die Aktuator- und Antriebsstrangmodelle parametrieren, deren Adams<sup>1</sup>-Simulationsmodelle als kompilierte Black-Box-Modelle in das ISAR-Modell integriert werden. Sie ermöglichen beispielsweise die Modellierung nichtlinearer Reibungseffekte im Lenkaktuator, Motor und Getriebe sowie die Einbeziehung der jeweiligen Übersetzungen. Da die Querdynamik insbesondere außerhalb des linearen Betriebsbereichs nur durch hochgenaue Reifenmodelle abbildbar ist, wird im ISAR-Modell das in Mehrkörpersimulationen oft verwendete Pacejka-Modell (auch als „Magic Formula“ bekannt, vgl. [Pac05]) verwendet. Durch die verwendeten Differentialgleichungen wird eine Näherung des verzögerten Auf- und Abbaus der Reifenkräfte bei schnellen Änderungen von Eingangsgrößen ermöglicht, was insbesondere bei hoher Querdynamik für ein nichtlineares Übertragungsverhalten sorgt.

<sup>1</sup> Mehrkörper-Simulationsprogramm für kinematische/kinetische Analysen mit dem das Bewegungsverhalten dreidimensionaler mechanischer Komponenten realitätsgetreu modelliert und animiert werden kann.

Die Parametrierung der Einzelkomponenten erfolgt zu großen Teilen experimentell mit Hilfe standardisierter Manöver basierend auf Messungen am realen Fahrzeug oder Vergleichsmessungen aus den Adams-Modellen der Einzelkomponenten. Neben dem in Abbildung 2.14 beschriebenen Fahrzeugmodell sind im ISAR-Modell Fahrzeugregelsysteme vom Chassis, des Fahrwerks und der Aktuatorik integriert. Der rechte Teil der Wirkkette ab der Bewegungsplanung ist somit realitätsgetreu abbildbar (vgl. Abbildung 2.3). Mit Hilfe von Verkehrssimulationsmodellen oder synthetischer Sensordaten und den Softwaremodulen der automatisierten Fahrfunktion kann somit auch der linke Teil der Wirkkette zur Verfügung gestellt und eine Closed-Loop-Simulation ermöglicht werden. Das ISAR-Modell entspricht dem in der Veröffentlichung [FFF19b] verwendeten Mehrkörpersimulationsmodell.

## 2.5.2 Neuronale Netze zur Modellierung der Wirkkette

Wie oben beschrieben, eignen sich KNNs insbesondere dann, wenn das zu modellierende Übertragungsverhalten mathematisch nur mit sehr großem Aufwand vollumfänglich zu beschreiben ist (wie z.B. bei MKS) und ausreichend Messdaten zur Verfügung stehen, um die Zusammenhänge zu lernen. Es können dann Netzwerke entstehen, die nur aus wenigen Schichten und Neuronen bestehen und damit effizienter zu rechnen sind als analytische Modelle. Im Folgenden soll kurz auf die für diese Arbeit benötigten theoretischen Grundlagen von KNNs eingegangen werden, bevor Anwendungsbeispiele aus der Modellbildung in der Fahrzeugtechnik und ein für dynamische, nichtlineare Systeme verbreiteter Netztyp detailliert beschrieben werden.

Mit Hilfe künstlicher neuronaler Netze wird versucht, die Vorgänge im zentralen Nervensystem des Menschen nachzubilden. Die Struktur eines KNNs richtet sich dabei im Wesentlichen nach dem natürlichen Vorbild, wobei Zellen oder Neuronen über Kantenverbindungen miteinander kommunizieren. Die Neuronen sind in Schichten angeordnet, wobei jedes Neuron einer Schicht immer mit allen Neuronen der jeweils nachfolgenden Schicht verbunden ist (Feedforward-Netze). Eine Erweiterung stellen rekurrente Netze dar, die darüber hinaus auch rückgerichtete Kanten enthalten. Beginnend mit der Eingabeschicht fließen Informationen über die Zwischenschichten zur Ausgabeschicht, wobei der Output des einen Neurons den Input des nächsten darstellt. Jede Kantenverbindung ist mit Gewichten belegt, die die Intensität des Informati-

onsflusses definieren. Das zugehörige Neuron gibt also jeweils die gewichtete Information nach Addition einer Neuronen-spezifischen Verzerrung (Bias) an die Nachfolgeschicht weiter. Vor der Weitergabe wird die Information oft durch eine Aktivierungsfunktion geleitet, die den Aktivitätslevel des Neurons wiedergibt. Im einfachsten Fall wird dafür eine Identitätsfunktion verwendet, welche die gewichtete, verzerrte Information als Aktivitätslevel weitergibt. Die Benutzung nichtlinearer Funktionen (z.B. Sigmoidfunktion, Sprungfunktion) ermöglichen dem Netz allerdings das Erlernen hochkomplexer, nichtlinearer Zusammenhänge und machen es besonders mächtig. Durch Schaltung mehrerer Neuronen hintereinander können somit hochdimensionale mathematische Funktionen abgebildet werden. Die Stellschrauben für eine hohe Modellierungsgüte sind neben der Architektur des Netzes die Kantengewichte, Verzerrungen und Aktivierungsfunktionen. Die Parametrierung stellt wiederum ein Optimierungsproblem dar, dessen Ziel die Erreichung einer hohen Generalisierungsfähigkeit ist. Der Trainingsdatensatz dient dabei als Grundlage zum Erlernen der Zusammenhänge, wobei die Performance auf einem während der Optimierung nicht genutzten Testdatensatz geprüft wird. Die Parametrierung der Netzparameter und Optimierungsparameter (wie z.B. die Lernrate, Fehlerfunktion, Anzahl Trainingsepochen) wird auch als „Hyperparametertuning“ bezeichnet [H<sup>+</sup>09, Roj13, Sch15].

Obwohl die Nutzung künstlicher neuronaler Netze zur Approximation mathematischer Funktionen bereits in den fünfziger Jahren in [MP43] beschrieben wurde, erfahren KNNs erst seit ca. 10 Jahren wieder zunehmende Beliebtheit. Der Grund dafür ist, dass Rechenkapazitäten erst heutzutage ausreichen, um aufwändige Hyperparametertunings durchzuführen und somit das gesamte Potential zu nutzen. Auch in der Fahrzeugtechnik sind sie mittlerweile weit verbreitet. Die Arbeit von [OEB00] beschreibt erste Ansätze zur Modellierung der Fahrzeugbeschleunigung in Crash-Situationen für die Entwicklung und Erprobung sicherheitskritischer Systeme zum Insassenschutz. Auch für die Querdynamik gibt es bereits Ansätze das Fahrzeugübertragungsverhalten im nichtlinearen Bereich zu lernen und als Prädiktionsmodell für Trajektorienplaner oder adaptive Regelsysteme zu nutzen (vgl. [DK01, KD03, PD11]). Eine Zusammenstellung unterschiedlicher Anwendungsbereiche von KNNs in der Fahrzeugtechnik kann [EGP93] und [Dou95] entnommen werden. Da sich für die Modellierung dynamischer Zusammenhänge insbesondere rekurrente Netzwerkarchitekturen als performanter erwiesen haben, soll im Folgenden einer der bekanntesten Vertreter dynamischer rekurrenter Netze vorgestellt



werden: Das Long-Short-Term-Memory-Netzwerk (LSTM). Es ist im Gegensatz zu herkömmlichen rekurrenten Netzen in der Lage, durch Nutzung einer Art Langzeitgedächtnis, deutlich komplexere Zusammenhänge mit vertretbarem Trainingsaufwand zu erlernen [HS97].

Dynamische rekurrente Netze besitzen die Fähigkeit den jeweils nächsten Betriebspunkt einer Zeitreihe auf Grundlage vorausgegangener Zeitschritte präzisieren zu können. Durch Implementierung eines Gedächtnisses können LSTMs Informationen, die länger zurückliegen speichern und später für die Prädiktion nutzen. Da die Fahrzeugreaktion bei Trägheiten und Totzeiten entlang der Wirkkette durchaus mit großer Verzögerung auf die Soll-Vorgabe folgen könnte, haben LSTMs für den Einsatz als Fahrzeugmodell besonderes Potential. Neben dem Gedächtnis (sog. *memory cell*) verfügt die LSTM-Zelle über mehrere Gates, welche die interne Verarbeitung hereinkommender Informationen steuern (vgl. Abbildung 2.15). Die *Input-Gates* und *Output-Gates* kontrollieren den internen Informationsfluss und bestimmen, ob ein Wert in der *memory cell* gespeichert wird und in welchem Umfang er für die Berechnung des Aktivitätslevels am Zellenausgang verwendet wird. Das *forget gate* entscheidet, wie lange Werte in der *memory cell* gespeichert werden. Der Input-Vektor zur LSTM-Zelle  $\vec{x}_t$  sowie die Aktivitätslevel-Vektoren der Gates  $\vec{f}_t$ ,  $\vec{i}_t$  und  $\vec{o}_t$  enthalten typischerweise vergangene Werte der jeweiligen Variablen und dienen als Grundlage zur Vorhersage des nächsten Zeitschritts. Der Output-Vektor  $\vec{y}_t$  entspricht dem um einen Zeitschritt verschobenen Input-Vektor und enthält somit den prädierten Wert für den Zeitpunkt  $t$ . Der Index  $t - 1$  beschreibt den vorherigen Durchlauf. Die in der *memory cell* hinterlegten Zustände sind im Vektor  $\vec{c}_t$  gespeichert. Als Aktivierungsfunktionen der jeweiligen Gates kommen hyperbolische Tangens-Funktionen ( $\sigma_c$  und  $\sigma_h$ ) und Sigmoid-Funktionen ( $\sigma_g$ ) zum Einsatz, dessen mathematische Herleitung [HS97] entnommen werden kann. Mit den Gewichtsmatrizen  $W_f, W_i, W_o, W_c \in \mathbb{R}^{h \times d}$  und  $U_f, U_i, U_o, U_c \in \mathbb{R}^{h \times h}$  sowie den Bias-Vektoren  $\vec{b}_f, \vec{b}_i, \vec{b}_o, \vec{b}_c \in \mathbb{R}^h$  ergibt sich die formelhafte Beschrei-

bung der Zusammenhänge eines LSTM-Netzwerks mit  $d$  Input-Variablen und  $h$  Neuronen in der verdeckten Schicht [FFFG19b]:

$$\begin{aligned}f_t &= \sigma_g(W_f \vec{x}_t + U_f \vec{y}_{t-1} + \vec{b}_f) \\i_t &= \sigma_g(W_i \vec{x}_t + U_i \vec{y}_{t-1} + \vec{b}_i) \\o_t &= \sigma_g(W_o \vec{x}_t + U_o \vec{y}_{t-1} + \vec{b}_o) \\c_t &= \vec{f}_t \circ \vec{c}_{t-1} + \vec{i}_t \circ \sigma_c(W_c \vec{x}_t + U_c \vec{y}_{t-1} + \vec{b}_c) \\y_t &= \sigma_t \circ \sigma_h(\vec{c}_t)\end{aligned}\tag{2.22}$$

Der Operator  $\circ$  steht für das Hadamard-Produkt (elementweise Multiplikation zweier Matrizen). Die Bestimmung der Gewichts- und Biasmatrizen ist Aufgabe des Trainingsprozesses. LSTM-Netzwerke werden heutzutage aufgrund ihrer guten Performance oftmals für die Sprach- und Schrifterkennung oder zur Vorhersage von Aktienkursen verwendet (vgl. [SSB14, GLF<sup>+</sup>08, CZD15]). Auch im Bereich automatisierter Fahrfunktionen gibt es beliebte Anwendungsbeispiele wie z.B. die Bewegungsprädiktion anderer Verkehrsteilnehmer (vgl. [AGR<sup>+</sup>16, ALF17]).

### 2.5.3 Kombinierte Ansätze

Aufgrund der guten Performance neuronaler Netze für die Lösung komplexer Probleme und der gleichzeitig oft kritisierten fehlenden Nachvollziehbarkeit des Modellverhaltens existieren Ansätze analytische Modelle und KNNs miteinander zu kombinieren. Das unter dem Begriff der „hybriden Modellbildung“ bekannte Verfahren kann insbesondere dann vorteilhaft sein, wenn Expertenwissen eingebracht und durch analytische Zusammenhänge beschrieben werden kann. Erfolgreiche Anwendungsbeispiele existieren sowohl für Zeitreihenprädiktionen (vgl. [AEK09, KAV15]) als auch für den Ausgleich von Parameteränderungen analytischer Fahrzeugsimulationsmodelle (vgl. [HNHI99]). In den meisten Fällen benötigt das KNN weniger Trainingsdaten und ist flexibler gegenüber Modelladaptionen [PU92].

Für die Verschaltung des KNNs und analytischen Modells existieren drei grundlegende Schemata, welche im folgenden näher beschrieben werden sollen [Aga97]:

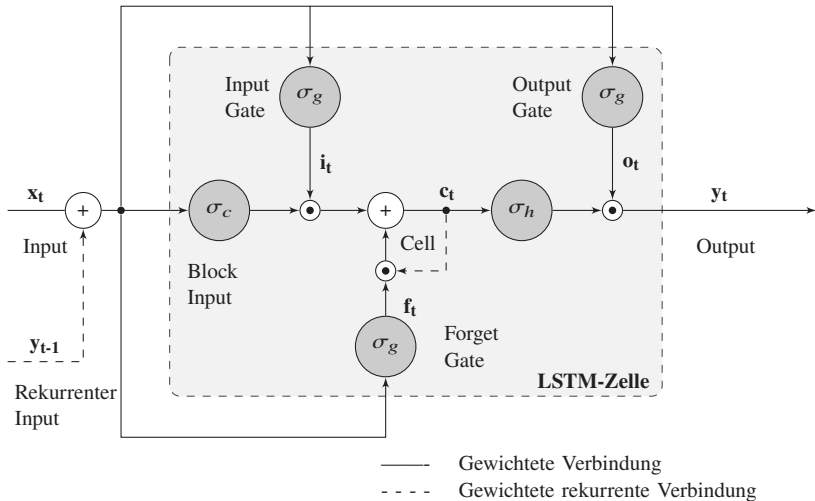


Abbildung 2.15: Aufbau einer LSTM-Zelle. Alle Kanten sind gewichtete Verbindungen [FFFG19b]

- Parallele Struktur:** In der parallelen Verschaltung ergibt sich der Modellausgang als Addition des KNN-Outputs und analytischen Modells. Die Grundidee dieser Anordnung ist, das neuronale Netz zur Kompensation von Modellfehlern zu nutzen. Das analytische Modell wird also typischerweise zuerst parametrisiert bevor das KNN im Hinblick auf eine Minimierung des hybriden Modellfehlers trainiert wird. Der parallele Ansatz eignet sich insbesondere dann, wenn das analytische Modell strukturelle oder parametrische Fehler aufweist. Wenn die Qualität allerdings bereits sehr hoch ist, muss das KNN ein stochastisches Messrauschen ausgleichen, was eine Herausforderung darstellen kann.
- Serielle Hammerstein-Struktur:** Die Hammerstein-Kombination sieht eine Anordnung des analytischen Modells nach dem KNN vor. Das neuronale Netz soll systematische Modellfehler des analytischen Modells präzisieren und dieses bei der Prädiktion des Modelloutputs unterstützen. Im Gegensatz zu der parallelen Verschaltung muss ein zusätzlicher Eingang am analytischen Modell für den KNN-Output vorgesehen werden.

- *Serielle Wiener-Struktur*: In der sogenannten Wiener Anordnung befindet sich das neuronale Netz im Prozessablauf nach dem analytischen Modell und kann als Filter für dessen Ausgang interpretiert werden, indem es Modellfehler darin erkennt und ausgleicht. Der Nachteil gegenüber den anderen beiden Ansätzen ist, dass der Modellausgang direkt vom KNN geliefert wird und damit wiederum Black-Box-Charakter hätte.

Die Eingänge der Einzel-Modelle sind meistens die Gleichen, können aber je nach Charakteristik des Problems auch unterschiedlich sein.

## 2.6 Modelle zur Objektivierung des Fahrverhaltens

Das Ziel der Optimierung von Applikationsparametern im Kontext automatisierter Fahrfunktionen ist eine bestmögliche Erreichung des Fahrerwunschverhaltens. Dieses beinhaltet neben der Einhaltung grundlegender Sicherheitsanforderungen das Bestreben nach maximalem Fahrkomfort. Während Grenzwerte für Sicherheitsanforderungen aus gesetzlichen Richtlinien oder anderweitig getroffenen Vereinbarungen abgeleitet werden können, ist die subjektive Empfindung des Fahrkomforts stark fahrerabhängig.

Das konventionelle Verfahren zur Bewertung der Applikation automatisierter Fahrfunktionen sieht eine Einschätzung von ausgewählten Experten vor, welche während des Entwicklungsprozesses immer wieder befragt werden und ihr persönliches Empfinden auf einer Komfort-Skala einordnen. Obwohl dieses Vorgehen mit geringem Aufwand durchführbar ist, mangelt es an Objektivität und Verlässlichkeit. Die Einschätzungen variieren nicht nur zwischen den Experten, sondern auch in Abhängigkeit der mentalen/physischen Verfassung und der Dauer des Experiments [Dec09]. Aus diesem Grund weichen Expertenbefragungen heutzutage immer mehr umfangreichen Probandenstudien, bei denen die subjektive Komfortwahrnehmung der Teilnehmer einer aussagekräftigen Stichprobe in synthetisch erzeugten Fahrsituationen erfragt wird. Die subjektive Wahrnehmung wird dann mit fahrdynamischen Messgrößen mittels Korrelations- oder Regressionsanalysen in Verbindung gebracht. Aus dem Bereich der Fahrdynamik-Applikation kommend, werden in der

Fahrerassistenzsystem-Entwicklung zumeist lineare multiple Regressionsmodelle verwendet, welche z.B. die maximale laterale Abweichung der Spurmitte, Geschwindigkeit, Beschleunigung oder Gierrate verwenden [ESJ15]. Das allgemeine Regressionsmodell lässt sich wie folgt formulieren:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k + \epsilon_i \quad (2.23)$$

Die Variable  $y$  bezeichnet in diesem Fall den Fahrkomfort während die Variablen  $x_1, \dots, x_k$  fahrdynamische Messgrößen darstellen, welche als vielversprechend zur Erklärung des zu beschreibenden Zusammenhangs gelten. Diese Auswahl kann durch Sensitivitätsanalysen unterstützt werden. Im Rahmen der Regressionsanalyse erfolgt eine Bestimmung der Regressionskoeffizienten  $\beta_0, \dots, \beta_k$  so, dass der Fehlerterm jedes Probanden  $\epsilon_i$  minimiert wird.

Im Zuge der Einführung des hochautomatisierten Fahrens gewinnen Fahrerassistenzsysteme zunehmend an Komplexität, was eine große Anzahl an Studien erfordert um alle relevanten Fahrsituationen und deren Ausprägungen erfassen zu können. Um den Versuchsaufwand gering zu halten, kommen deshalb zum Teil Machine-Learning-Ansätze zum Einsatz, welche in der Lage sind, komplexe mathematische Zusammenhänge aus Messdaten zu extrahieren. Obwohl KNNs sich prinzipiell für diese Probleme eignen, kommt in dieser Arbeit aufgrund des Black-Box-Charakters ein alternativer Machine-Learning-Ansatz zum Einsatz. Das in [MHS19] beschriebene multivariate *Dynamic-Time-Warping*-Verfahren nutzt ein geschwindigkeitsunabhängiges Abstandsmaß um einen ähnlichen Datensatz zu der aktuellen Zeitreihe in einer gelabelten Referenz-Datenbasis zu finden und somit eine Komfort-Klassifizierung vorzunehmen. Der Vorteil gegenüber linearen Regressionsmodellen ist, dass Zeitreihen nicht auf Features wie beispielsweise den Mittelwert, Maximum oder Minimum reduziert werden müssen, wodurch Informationen verloren gingen. In Studien konnte gezeigt werden, dass sich das Verfahren ohne aufwändige Modifikationen für Spurhalte-, Spurwechsel- und Beschleunigungs-/Verzögerungsmanöver anwenden lässt [Mos20]. Dafür werden lediglich eine gelabelte Datenbasis und geeignete fahrdynamische Messgrößen benötigt. Der Komfort-Output des beschriebenen Objektivierungs-Modells ist zwischen 0 und 7 definiert, wobei 7 den maximalen Komfort darstellt. Die Bedeutung der Skalenwerte kann Tabelle 2.1 entnommen werden.

Tabelle 2.1: Komfortskala für das *Dynamic-Time-Warping*-Modell [MHS19]

1	2	3	4	5	6	7
Almost crash	Strong irritation	Irritation	Strong dissatisfaction	Slight dissatisfaction	Satisfaction	Great satisfaction

## **3 Forschungsbedarf**

Die Darstellung des Stands der Technik und Wissenschaft hat gezeigt, dass die Applikation automatisierter Fahrfunktionen heutzutage vom Entwicklungsingenieur weitgehend manuell im Fahrversuch erfolgt. Während Potentiale zur Automatisierung und Virtualisierung des Prozesses noch wenig Beachtung finden, zeigen Ansätze aus dem Bereich der Antriebsstrang-Applikation die Vorteile simulativer Parameteroptimierungen auf. Auch der Einzug verbesserter X-in-the-Loop-Modelle in die Fahrerassistenzsystem-Entwicklung begünstigt eine Effizienzsteigerung des konventionellen Vorgehens. Zusammen mit den theoretischen Grundlagen zur Komplexitätsreduktion, Optimierung und Simulationsmodellen sollen im folgenden Kapitel der Forschungsbedarf und -ansatz dieser Arbeit hergeleitet werden.

Im ersten Schritt erfolgt eine Zusammenfassung zukünftiger Herausforderungen bei der Kalibrierung von Fahrerassistenzsystemen und Ableitung von Anforderungen an eine Applikationsmethodik. Dem wird der aktuelle Stand der Technik gegenübergestellt und es wird auf Möglichkeiten der Nutzung vorgestellter theoretischer Grundlagen eingegangen. Basierend auf diesem Vergleich lässt sich der Forschungsbedarf ableiten und ein entsprechender Ansatz zur Adressierung aktueller Herausforderungen in der Applikation erarbeiten. Dazu werden Forschungsfragen formuliert, auf deren Basis eine anschließende Skizzierung des konzeptionellen Vorgehens und Validierungsansatzes erfolgt.

### **3.1 Ableitung von Anforderungen an eine Applikationsmethodik**

In Abschnitt 2.1 wurden bereits die fahrzeugbasierte Applikation und die damit einhergehenden Nachteile ausführlich beschrieben. Mit Einführung des hochautomatisierten Fahrens (SAE Level 3) und zunehmender Derivatevielfalt

werden sich die damit verbundenen Herausforderungen noch verstärken, sodass eine termingerechte Fertigstellung der Software zu Lasten der Applikationsgüte erfolgen könnte. Aufgrund gestiegener Kundenanforderungen und steigendem Wettbewerbsdruck kann sich das ein Automobilhersteller nicht leisten, was die Einführung alternativer Applikationsprozesse zur Adressierung der Probleme unerlässlich macht.

Die in Abschnitt 2.1.2 angeführte mangelnde *Verfügbarkeit von Fahrzeugen* und der hohe *Ressourcenaufwand* manueller Applikationen sind maßgeblich durch die starke Abhängigkeit des aktuellen Prozesses von Prototypen-Fahrzeugen verursacht. Wenn Parametrierungen nur im Fahrversuch durchgeführt werden, müssen nicht nur ausreichend Versuchsträger in aufwändigen Herstellungsprozessen zur Verfügung gestellt werden, sondern auch Teststrecken gebucht, Erprobungen finanziert und Arbeitszeiten mehrerer Ingenieure gebunden werden. Darüber hinaus lassen sich die Systeme zwar im Realsystem, aber immer nur in Echtzeit manuell applizieren. Damit können normalerweise nicht alle relevanten Szenarien vollumfänglich getestet und Wechselwirkungen zwischen Fahrfunktionen unter Umständen nicht aufgedeckt werden. Im Sinne einer testgetriebenen Entwicklung (TDD, vgl. Abschnitt 2.1.1) wäre es jedoch ratsam, Änderungen an der Software kontinuierlich mit umfangreichen Tests abzusichern, um Abhängigkeiten frühzeitig zu erkennen. Eine weitere Herausforderung stellt die Diskrepanz der in frühen Entwicklungsphasen im Prototyp verwendeten Subsysteme und Komponenten zum Zielsystem dar, welche eine durchgängige Applikation über mehrere Reifegrade zusätzlich erschwert. Eine Anforderung an die Applikationsmethodik ist folglich die sinnvolle Einbeziehung von Simulationsumgebungen zum Testen und Parametrieren der Fahrerassistenzsysteme so, dass eine Unterstützung der fahrzeugbasierten Applikation möglich ist. Um kurze Feedback-Schleifen zu ermöglichen, wird im Idealfall ein Simulationsmodell verwendet, das realitätsgetreue Ergebnisse mit vertretbarem Rechenaufwand liefert.

Das in Abschnitt 2.1 angesprochene Problem des *fehlenden Systemverständnisses* von Applikateuren kann durch Bereitstellung verlässlicher Simulationswerkzeuge ebenfalls adressiert werden. Unter der Voraussetzung einer performanten Simulation und ausreichend Rechenleistung besteht die Möglichkeit, mangelndes Verständnis von Parametereinflüssen und -wechselwirkungen experimentell mit vertretbarem Aufwand auszugleichen. Das fehleranfällige Kalibrieren auf Grundlage von Expertenwissen des Applikateurs oder Funktions-



entwicklers lässt sich somit durch fundierte Parameterstudien untermauern. Diese erlauben es, relevante Wechselwirkungen zwischen Szenarien oder Parametern frühzeitig zu erkennen und in die Optimierungsstrategie mit einzu beziehen. Neben der Abschätzung des zu erwartenden Effekts von Parametern auf das Fahrverhalten in bestimmten Situationen stellt die Festlegung sinnvoller Wertebereiche eine weitere Herausforderung dar. Diese werden meist vom Entwickler aufgrund seines Systemverständnisses oder historischer Referenzen konservativ festgelegt. Die effiziente Identifikation relevanter Einflussbereiche im Parameterraum stellt somit eine zweite Anforderung dar, welche für ein validiertes und weniger fehlerbehaftetes Systemverständnis sorgen soll.

Die steigende Anzahl angebotener Fahrerassistenzfunktionen und deren Funktionalitäten tragen zu einer in den letzten Jahren gestiegenen *Dimensionalität* und *Komplexität des Parameterraums* bei, welche mit der konventionellen Herangehensweise in vertretbarer Zeit nicht mehr beherrschbar sind. Auch die zunehmende Derivatevielfalt erhöht den Versuchsaufwand, da die Applikation ein optimales Fahrverhalten für alle relevanten Fahrsituationen in jeglichen angebotenen Fahrzeugen sämtlicher belieferten Länder sicherstellen muss. Eine Möglichkeit, den Applikationsprozess zu beschleunigen ist der Einsatz von Optimierungsalgorithmen, welche basierend auf X-in-the-Loop-Modellen den Suchraum screenen und Parametrierungen automatisiert herausfahren (in Anlehnung an den Prozess der Motorapplikation aus Abbildung 2.4). Da der Applikateur hierfür idealerweise nicht mehr benötigt wird, müssen geeignete Modelle zur Objektivierung des Fahrverhaltens (vgl. Abschnitt 2.6) genutzt werden, die das Optimierungsziel beschreiben. Das in Abschnitt 2.1.2 zusätzlich herausgestellte Problem der *subjektiven Bewertung des Fahrverhaltens* muss somit durch den Einsatz automatisierter Applikationswerkzeuge ebenfalls adressiert werden, da diese objektiv messbare Gütefunktionale zwingend benötigen. Als letzte Anforderung an die Applikationsmethodik ergibt sich somit der Bedarf nach Optimierungsmethoden, welche eine schnelle Parametrierung automatisierter Fahrfunktionen ermöglichen und Ergebnisse liefern, die den Ressourcenaufwand fahrzeuggestützter Applikationen entlasten können.

Zusammenfassend ergeben sich die folgenden drei Anforderungen:

- Einbeziehung von Simulationsumgebungen zur Unterstützung fahrzeuggestützter Applikationen
- Effiziente Identifikation relevanter Einflussbereiche im Parameterraum

- Bereitstellung von Optimierungsmethoden zur schnellen Parametrierung

## **3.2 Bewertung der Grundlagen und des Stands der Wissenschaft**

Basierend auf den im vorherigen Abschnitt formulierten Anforderungen wird nun das Potential bestehender Verfahren zur Adressierung der identifizierten Probleme evaluiert. Des Weiteren erfolgt eine kritische Prüfung der in Kapitel 2 beschriebenen theoretischen Grundlagen im Hinblick auf deren Anwendbarkeit zur Optimierung konventioneller Applikationsverfahren.

Die Forderung nach zunehmender Virtualisierung im Applikationsprozess wird im Bereich der Motorapplikation bereits bedient (vgl. Abschnitt 2.2). Applikationen auf dem Prüfstand oder in einer SiL-Umgebung mittels DoE-Methoden stellen dort bereits den Stand der Technik dar. Auch in der Fahrerassistenzentwicklung kommen vermehrt X-in-the-Loop-Verfahren zum Einsatz, um eine Reproduzierbarkeit und Skalierung von Szenarien zu ermöglichen (vgl. Abschnitt 2.2.2). Durch eine teilweise Modellierung der Wirkkette kann das Closed-Loop-System ohne Abhängigkeiten zu noch nicht verfügbaren Softwaremodulen oder Hardware getestet werden. Während eine Abstraktion der Schnittstellen an Motorprüfständen jedoch in Open-Loop-Simulationen ohne Inkaufnahme großer Modellierungsfehler möglich ist, müssen für das Testen automatisierter Fahrfunktionen deutlich mehr Wechselwirkungen entlang der Wirkkette berücksichtigt werden (vgl. Abbildung 2.3). Erst das Zusammenwirken der Umfelddarstellung, Bewegungsplanung und -regelung sowie der Fahrzeugdynamik erzeugt das vom Kunden empfundene Fahrverhalten, dessen Bewertung das Gütekriterium der Optimierung darstellt. Während eine Umfeldsimulation vergleichsweise einfach durch Einspeisen synthetischer Sensordaten ersetzt werden kann, wird die Fahrdynamik insbesondere durch das Zusammenspiel aus Fahrfunktion, Regelung und Fahrzeugdynamik bestimmt. Wie in Abschnitt 2.4.1 beschrieben, stellen die Applikationsparameter für den Fahrzeughersteller den wichtigsten Stellhebel zur Erreichung des Fahrerwunschverhaltens dar, während der linke Teil der Wirkkette (vgl. Abbildung 2.3) zu großen Anteilen durch Anforderungen spezifiziert und von einem Zulieferer entwickelt wird. Unter der Voraussetzung, dass die Softwaremodule und Regelung zum Zeitpunkt der Applikation bereits vorliegen, muss also ein

Fahrzeugmodell zur Verfügung stehen, welches genau ist, nicht viel Rechenleistung benötigt und leicht an verschiedene Derivate anpassbar ist. Unter den in Abschnitt 2.5 beschriebenen Modellen erfüllt das ESM zwar den Wunsch nach geringer Komplexität und vergleichsweise einfacher Parametrierbarkeit. Jedoch stößt es in dynamischen Manövern an seine Grenzen, welche bei einigen Fahrerassistenzfunktionen (z.B. Ausweichassistenten, Nothaltesystemen) durchaus auftreten können. Das ISAR-Modell bietet auf der anderen Seite eine hohe Genauigkeit, ist aber rechenaufwändig und lässt sich nur mit hohem Aufwand parametrieren. Dem in Abschnitt 2.5 ausführlich beschriebenen Trade-Off zwischen Genauigkeit und Komplexität kann prinzipiell mit neuronalen Netzen begegnet werden. Diese weisen aber den Nachteil auf, bei Fehlprädiktionen nur schwer nachvollziehbar zu sein. Die in Abschnitt 2.5.3 vorgestellten kombinierten Ansätze bieten zwar die Chance, dieser konzeptionellen Schwäche von KNNs zu begegnen. Eine Anwendung als Fahrzeugmodell ist allerdings noch nicht erforscht.

Zur Steigerung des Systemverständnisses und Identifikation einflussreicher Parameter kommen insbesondere für Optimierungsprobleme mit hochdimensionalen Suchräumen oftmals Sensitivitätsanalysen zum Einsatz. Diese benötigen zwar zu Anfang relativ viele Systemevaluationen, können aber eine schnellere Konvergenz des Optimierers ermöglichen und für die Applikation relevante Parameterwirkungen herausstellen (vgl. Abschnitt 2.3). Die in Abschnitt 2.3.1 eingeführte Elementareffektmethode wird oft angewendet, wenn wenig Informationen über Sensitivitäten des Parameterraums verfügbar sind und Systemevaluationen mit hohem Rechenaufwand verbunden sind. Trotz der vergleichsweise recheneffizienten Anwendbarkeit des Verfahrens besteht im Gegensatz zu der ebenfalls vorgestellten VBSA der Nachteil, dass die Ergebnisse lediglich qualitativen Charakter haben und somit keine direkten Rückschlüsse von Parameterwirkungen auf das Gütekriterium erlauben. Da Modellgenauigkeiten bei einer simulativen Auswertung jedoch nicht ausgeschlossen werden können, wäre eine Übertragbarkeit der Sensitivitätswerte auf das Realsystem aus vielerlei Hinsicht problematisch, sodass Informationen über relative Sensitivitäten für eine erste Abschätzung ausreichend sind. Der Parameterraum ist neben seiner Dimensionalität durch die Wertebereiche jedes Faktors bestimmt. Wie im letzten Abschnitt herausgestellt, erfolgt eine Abschätzung dieser meist basierend auf Expertenwissen. Eine Gültigkeit der Parametergrenzen ist jedoch nicht nur hilfreich für eine zielgerichtete Kalibrierung, sondern auch notwendige Voraussetzung für die Anwendung von Sensitivitätsanalysen (vgl. Abschnitt

2.3). In Abschnitt 2.3.2 sind einige Ansätze zur Identifikation valider Bereiche beschrieben, welche ihren Ursprung größtenteils in der Systemidentifikation haben. Um eine Auswertung mit vertretbarem Aufwand durchzuführen werden oft analytische Ersatzmodelle erstellt, die das Übertragungsverhalten des Zielsystems ausreichend genau abbilden. Die Herleitung gültiger Wertebereiche erfolgt dann mit Hilfe algebraischer Methoden. Weil eine Überführung der zu simulierenden Wirkkette automatisierter Fahrfunktionen entweder zu Lasten der Genauigkeit ginge oder zu einem komplexen analytischen Modell (wie z.B. dem ISAR-Modell) führen würde und eine effiziente Anwendung algebraischer Methoden damit nicht mehr gegeben ist, ist eine Ersatzmodellierung nicht zielführend. Verfahren, welche eine verlässliche Bestimmung einflussreicher Wertebereiche mit dem ISAR-Modell ermöglichen, würden approximierte Gradientendarstellungen (z.B. durch Differenzenquotienten) benötigen um gültige Ober- und Untergrenzen mathematisch herzuleiten. Die dafür notwendige Anzahl an Simulationsdurchläufen stellt einen unrealistisch hohen Versuchsaufwand dar. Auch die Tatsache, dass der Stand der Wissenschaft erst wenige Arbeiten aufweist, welche eine verlässliche Bestimmung gültiger Parametergrenzen untersuchen, bestätigt den Forschungsbedarf in diesem Bereich.

Als letzte Anforderung an die Applikationsmethodik wurde der Bedarf nach effizienten Optimierungsmethoden formuliert, welche den Ressourcenaufwand manueller Parametrierungen im Fahrversuch reduzieren sollen. Da automatisierte Applikationen mit Hilfe populationsbasierter Algorithmen bereits in vergleichbaren Bereichen (wie z.B. der Motor-, Getriebe-, und Fahrwerkapplikation, vgl. Abschnitt 2.1) gute Ergebnisse erzielen konnten, erscheint eine Anwendung für automatisierte Fahrfunktionen vielversprechend. Wie Abschnitt 2.6 herausstellt, lässt sich die Optimierung von Fahrerassistenzsystemen nicht anhand weniger klar definierter Zielkriterien wie beispielsweise in der Motorapplikation durch Emissionsgrenzwerte, Leistung und Kraftstoffverbrauch durchführen. Stattdessen müssen aufwändig durch Probandenstudien bestimmte Objektivierungsmodelle zum Einsatz kommen, welche das Fahrerwunschverhalten in der Simulation zwar statistisch fundiert abbilden, den Fahrversuch aber nicht gänzlich ersetzen können. Da die für eine virtuelle Applikation zu berücksichtigende Szenarienvielfalt aufgrund gesteigerter Funktionalitäten stetig wächst, ist es außerdem zur Erreichung kurzer Feedback-Schleifen erstrebenswert, die Ergebnisse der Einflussanalyse in die automatisierte Kalibrierung mit einzubeziehen. Sensitivitätsbasierte Optimierungen werden in vielen Veröffentlichungen beschrieben. Das in dieser Arbeit behandelte einkriterielle

Optimierungsproblem (vgl. Abschnitt 2.4.1) weist allerdings die Besonderheit auf, dass eine Systemevaluation die Auswertung eines Szenarienkatalogs bedeutet und folglich szenarienabhängige Einflüsse von Parametern zu erwarten sind. Eine aus Sensitivitätsinformationen abgeleitete globale Reduktion des Parameterraums für alle Manöver würde folglich wichtige Informationen vernachlässigen. Durch eine intelligente Einbeziehung szenarienabhängiger Sensitivitäten könnte die Effizienz einer virtuellen automatisierten Parametrierung jedoch deutlich gesteigert werden.

## **3.3 Forschungsansatz**

Basierend auf den in Abschnitt 3.1 definierten Anforderungen und der Prüfung von Potentialen existierender Verfahren, den damit verbundenen Herausforderungen zu begegnen, wird im Folgenden zunächst der in diese Arbeit adressierte Forschungsbedarf abgeleitet. Im Anschluss erfolgt eine kurze Beschreibung des konzeptionellen Vorgehens zur Entwicklung und Validierung der Methodik, welche die Struktur der Kapitel 4, 5 und 6 näher erläutert.

### **3.3.1 Forschungsfragen**

Als Grundvoraussetzung für zukünftige Applikationsprozesse wurde die Nutzung von Simulationsumgebungen gefordert, welche mit geringem Rechenaufwand genaue Ergebnisse liefern und somit die Basis für virtuelle Applikationen und kurze Feedback-Schleifen im Entwicklungsprozess legen. Da bei einer Übertragbarkeit simulativ gewonnener Ergebnisse auf das Realsystem Ungenauigkeiten zu erwarten sind, sollte eine geeignete Integration der Modelle in den fahrzeuggesteuerten Applikationsprozess erarbeitet werden. Die Bewertung des Stands der Technik und Wissenschaft hat außerdem gezeigt, dass das von Fahrerassistenzfunktionen ausgelöste Fahrverhalten maßgeblich durch die Fahrzeugdynamik bestimmt wird und somit eine sinnvolle Virtualisierung der Wirkkette die Bereitstellung eines hochgenauen, schnellen und adaptiven Fahrzeugmodells beinhalten sollte. Auch wenn eine Vielzahl an Modellen unterschiedlicher Komplexität existieren, weisen sie Schwächen auf, welche einen sinnvollen Einsatz als Applikationswerkzeug erschweren. Daraus ergibt sich die erste Forschungsfrage:

**1. Forschungsfrage:** Wie können Simulationsumgebungen sinnvoll in den Applikationsprozess eingebunden werden und mit welchem Konzept lässt sich eine hohe Genauigkeit, Effizienz und Adaptivität von Fahrzeugmodellen erreichen?

Aus der Herausforderung komplexer werdender Parameterräume und umfassenderer Funktionalitäten zukünftiger Fahrfunktionen ist die Anforderung entstanden, durch Anwendung von Einflussanalysen das Systemverständnis zu steigern. In der Literatur kommen dafür üblicherweise Sensitivitätsanalysen zum Einsatz, welche Effekte einzelner Parameter auf die Zielfunktion abschätzen können. Eine Evaluierung der Benutzung bestehender Verfahren hat ergeben, dass die vergleichsweise effiziente Elementareffektmethode für den Anwendungsfall ausreichende Genauigkeiten liefert. Neben dem globalen Einfluss von Parametern auf die Zielfunktion wurde zusätzlich die Relevanz der Bestimmung sensibler Wertebereiche herausgestellt. Diese sind nicht nur notwendige Bedingung für die Anwendung von SAs sondern bieten auch die Chance, den für die Applikation definierten Suchraum deutlich zu reduzieren. Bestehende Verfahren sehen eine Bestimmung gültiger Grenzen mit algebraischen Methoden basierend auf analytischen Ersatzmodellen vor und wurden aufgrund von mangelnder Genauigkeit und Übertragbarkeit der Ergebnisse als nicht hilfreich bewertet. In Folge dessen soll im Rahmen der Arbeit eine Methodik entwickelt werden, welche eine verlässliche Einflusssschätzung der Dimensionen des Parameterraums und zugehöriger Wertebereiche mit vertretbarem Rechenaufwand ermöglicht.

**2. Forschungsfrage:** Gibt es eine Methode, mit der einflussreiche Parameter und Wertebereiche im Suchraum verlässlich mit vertretbarem Rechenaufwand bestimmt werden können?

Über die Identifikation sensibler Bereiche und damit verbundene Erhöhung des Systemverständnisses hinaus soll eine simulative Applikation ermöglicht werden. Bekannte populationsbasierte Optimierungsalgorithmen haben bereits unter vergleichbaren Rahmenbedingungen gute Ergebnisse geliefert und wurden als vielversprechend für die Lösung des in dieser Arbeit beschriebenen Problems bewertet. Im Gegensatz zu Applikationen in anderen Bereichen ist die Szenarienvielfalt für automatisierte Fahrfunktionen jedoch deutlich größer, sodass eine Einbeziehung von Sensitivitätsinformationen der Parameter und Szenarien in den Optimierungsprozess zur Reduktion der Rechenzeit ange-

strebt wird. Während Algorithmen zur automatisierten sensitivitätsbasierten Kalibrierung zwar existieren, sehen diese meistens keine Berücksichtigung szenarienspezifischer Einflüsse vor. Weil dadurch aber erst das gesamte Potential der Sensitivitätsinformationen genutzt werden kann, wurde ein Vorsehen dieser Funktionalität im Rahmen der automatisierten Optimierung als zielführend bewertet. Die dritte Forschungsfrage ergibt sich damit wie folgt:

**3. Forschungsfrage:** Welcher Optimierungsalgorithmus eignet sich am besten für eine virtuelle automatisierte Applikation und wie können szenarienspezifische Einflussinformationen zur Reduktion der Rechenzeit genutzt werden?

### 3.3.2 Konzeptionelles Vorgehen

Wie zuvor beschrieben, stellt die Bereitstellung eines hochgenauen, effizienten Fahrzeugmodells die Basis zur virtuellen Ermittlung sensibler Parameterbereiche und Optimierungen dar. Da analytische Modelle dem Komplexitäts-Genauigkeits-Trade-Off unterliegen und KNNs als Grundlage für simulative Applikationen aufgrund ihres Black-Box-Charakters auf geringe Akzeptanz stoßen würden, wird in dieser Arbeit ein kombinierter Ansatz (vgl. Abschnitt 2.5.3) verfolgt. Die Basis bildet das analytische Einspurmodell, welches leicht zu implementieren und recheneffizient ist. Parametrische und strukturelle Modellfehler des ESMs werden durch ein neuronales Netz ausgeglichen, das Prädiktionsungenauigkeiten lernen kann. Aufgrund der für Simulationen geforderten langfristigen Prädiktionsgüte kommen dafür LSTM-Netze zum Einsatz. Der Vorteil kombinierter Strukturen besteht darin, dass ESM-Parameter aus „Katalogdaten“ hergeleitet werden können und sich das KNN mit geringem Aufwand aus Messdaten trainieren lässt. Der weiterhin hohe Anteil fahrzeugbasierter Applikationen begünstigt die Nutzung von Messdaten zur kontinuierlichen Verbesserung des LSTM-Netzes (vgl. Abbildung 3.1). Das Konzept wurde vom Autor erstmals in [FFFG19b] veröffentlicht.

Auf der Grundlage des im Folgenden als „hybrides Modell“ bezeichneten Fahrzeugmodells erfolgt eine virtuelle Identifikation relevanter Bereiche des Suchraums, um eine zielgerichtete Applikation in der Simulation und im Fahrversuch zu ermöglichen. Die Methodik zur Komplexitätsreduktion des Parameterraums basiert auf einer lokalen Anwendung der Elementareffektmethode

und nutzt Synergien bei der Bestimmung von Parametereinflüssen und Identifikation sensitiver Wertebereiche durch ein intelligentes Sampling-Schema. Das Konzept wurde vom Autor in [FFFG19a] erstmals veröffentlicht.

Basierend auf den szenarienabhängigen Sensitivitätsinformationen wird eine automatisierte Kalibrierung in der Simulation angestrebt. Um die prinzipielle Eignung eines Optimierungsalgorithmus festzustellen, werden die beiden in Abschnitt 2.4.2 vorgestellten Methoden (GA und PSO) als vielversprechendste Vertreter populationsbasierter Verfahren experimentell gegenübergestellt. Dafür wird ein realistisches Applikationsproblem definiert und Optimierungen mit beiden Ansätzen durchgeführt. Das im Hinblick auf Optimierungsergebnis und Rechenzeit geeignetere Verfahren wird im Anschluss weiterverwendet. Um die Effizienz weiter steigern zu können und das Ziel schneller Feedback-Schleifen zu erreichen, wird darüber hinaus eine Methodik vorgestellt, um die Ergebnisse der Komplexitätsreduktion bestmöglich zur Reduktion von Szenariensimulationen zu nutzen. Diese ist erstmals in den Vorveröffentlichungen [FFFG20] und [Fra21] beschrieben. Die besten Lösungen der virtuellen Optimierung werden final einer Robustheitsanalyse unterzogen, welche die Stabilität gegenüber Parameteränderungen prüft und die Eignung zur Weiterverwendung im Fahrversuch bewertet. Somit wird bei der Auswahl vielversprechender Lösungen neben dem reinen Optimierungsergebnis, die Toleranz im Hinblick auf parametrische Unsicherheiten entlang der Wirkkette in Betracht gezogen. Eine Prüfung der zu Grunde liegenden Annahme, dass im Hinblick auf Änderungen von Applikationsparametern robuste Lösungen eine verbesserte Übertragbarkeit in den Fahrversuch versprechen, erfolgt im Rahmen der Validierung.

Obwohl somit theoretisch eine vollständig in der Simulation durchgeführte Parametrierung möglich wäre, haben die Auswertungen in den Abschnitten 3.1 und 3.2 gezeigt, dass die fahrzeugbasierte Applikation nicht gänzlich zu ersetzen ist. Neben Modellungenauigkeiten entlang der gesamten Wirkkette ist die Verwendung validierter Objektivierungsmodelle immer noch kritisch zu bewerten, da in den zugehörigen Probandenstudien immer nur ein kleiner Teil tatsächlich notwendiger Situationen evaluiert wird und der vom Kunden empfundene Fahrkomfort möglicherweise durch Messgrößen nicht vollständig abbildbar ist. Darüber hinaus wäre eine Markteinführung rein simulativ applizierter Fahrfunktionen aus politischer und rechtlicher Sicht von einem Automobilhersteller nicht zu vertreten. Aus den genannten Gründen



zeigt Abbildung 3.1 ein Konzept auf, wie die fahrzeugbasierte Applikation durch simulative Auswertungen unterstützt werden kann. Auch wenn sich die Wirkkette in der Simulation möglicherweise anders verhält als im Fahrzeug, wird mit Nutzung des hybriden Modells angenommen, dass die Unterschiede klein genug sind, um qualitative Sensitivitätsinformationen als Anhaltspunkt für ein zielgerichtetes Einstellen der Parameter nutzen zu können. Auch bei Verwendung virtuell gefundener optimaler Parametersätze im Fahrzeug ist keine fehlerfreie Übertragbarkeit der Ergebnisse zu erwarten. Die Lösungen der simulativen Optimierung werden aber dennoch einen höheren Reifegrad als zufällig gewählte Anfangsparametrierungen aufweisen und damit eine „gute Absprungbasis“ für ein anschließendes Fine-Tuning bieten. Eine Prüfung dieser Annahmen geschieht im Rahmen der Validierung, welche im nächsten Abschnitt konzeptionell vorgestellt wird. Um im Fahrversuch ein zielorientiertes Applizieren zu garantieren und Wechselwirkungen aufzudecken, wird das hybride Fahrzeugmodell zur kontinuierlichen Absicherung neuer Parametersätze verwendet. Aufgrund der Effizienz des Simulationsmodells lassen sich damit im Fahrversuch erprobte Situationen durch Simulation eines umfassenden Manöverkatalogs skalieren. Ein relativer Vergleich der Objektivgrößen zu den vorherigen Ergebnissen erlaubt dann eine qualitative Abschätzung der Auswirkungen neu gefundener Bedatungen und somit eine inkrementelle globale Verbesserung des Applikationsstands.

### 3.3.3 Validierung der Methodik

Im Rahmen der Validierung soll zum Einen die Genauigkeit sowie Recheneffizienz des hybriden Fahrzeugmodells nachgewiesen und zum Anderen die Validität der simulativen Komplexitätsreduktion und Optimierung bestätigt werden. Darüber hinaus erfolgt eine Prüfung der Übertragbarkeit in der Simulation gefundener Lösungen auf den Fahrversuch.

Entsprechend des etablierten Vorgehens zur Validierung neuronaler Netze erfolgt eine Prüfung der Generalisierungsfähigkeit des in dieser Arbeit entwickelten Fahrzeugmodells mit Hilfe eines Testdatensatzes, der dem Netz während des Trainings nicht präsentiert wurde. Die Szenarien setzen sich aus realen Applikationsfahrten zusammen und decken einen für Fahrerassistenzsysteme repräsentativen Bereich der Fahrdynamik ab. Bei der Auswertung unterschiedlicher Geschwindigkeiten und Krümmungsverläufe werden auch die Prädikti-

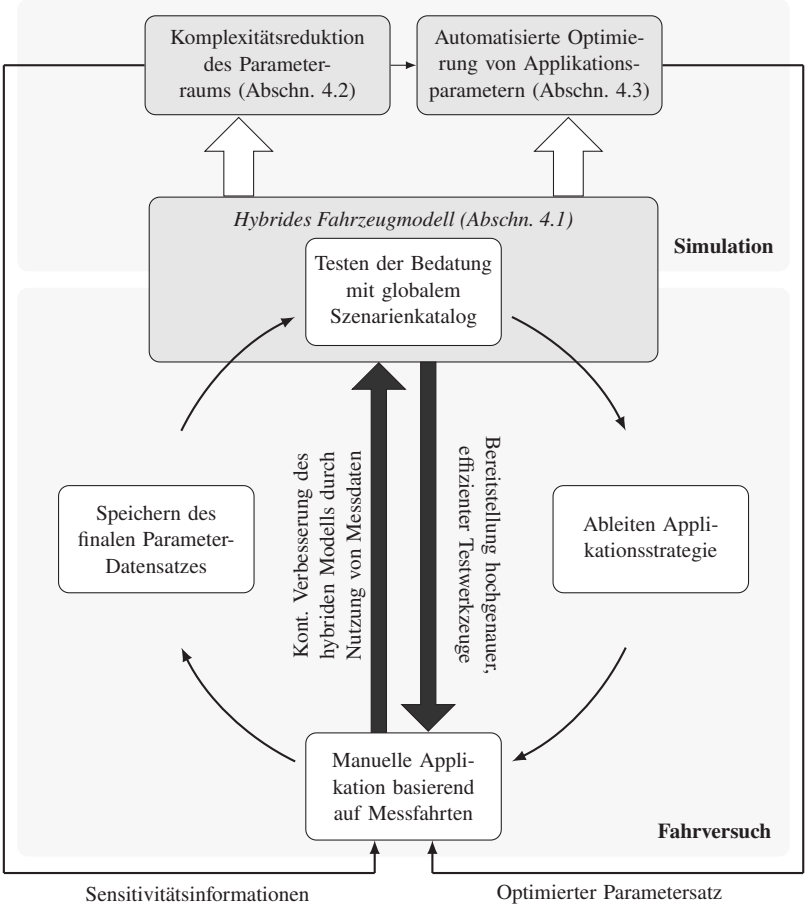


Abbildung 3.1: Konzept der Applikationsmethodik für automatisierte Fahrfunktionen

ongüte für hochdynamische Manöver im nichtlinearen Dynamikbereich und Szenarien mit besonders langer Dauer evaluiert. Dieses Validierungskonzept entspricht dem in der Vorveröffentlichung [FFFG19b] beschriebenen Ansatz.

Für die Validierung der Komplexitätsreduktion und darauf basierenden Optimierung wird durch ein umfangreiches Resampling im Parameterraum nachgewiesen, dass Parameteränderungen in als nicht sensitiv bewerteten Bereichen keinen relevanten Einfluss auf die Zielfunktion haben. Als Konsequenz kann bestätigt werden, dass eine sensitivitätsbasierte Reduktion von Szenariensimulationen den Optimierungsverlauf nicht negativ beeinflusst und die damit verbundene Effizienzsteigerung ohne Einschränkung des Ergebnisses angewendet werden kann. Dieses Verfahren zur Validierung der Komplexitätsreduktion wird vom Autor erstmals in [FFFG20] beschrieben.

Als dritter Aspekt der Validierung wird die Aussagekraft virtuell gefundener optimaler Lösungen hinsichtlich ihrer Performance im Zielsystem getestet. Da die Optimierung auf Grundlage des hybriden Modells durchgeführt wird, gibt die Validierung der Genauigkeit dessen bereits Aufschluss über die Übertragbarkeit der Ergebnisse in den Fahrversuch. Unter realen Umgebungsbedingungen sind Situationen jedoch nicht genau reproduzierbar, sodass sich die Rahmenbedingungen bei jeder Messfahrt ändern können. Unterschiede treten insbesondere in der Charakteristik des Szenarios und des Übertragungsverhaltens der Wirkkette auf, welches maßgeblich durch die Fahrzeugdynamik bestimmt wird. Um den Nachweis der Validität zu erbringen, kommen aus dem Grund Robustheitsanalysen zum Einsatz, welche die Stabilität simulativ gefundener Bedatungen prüfen. Eine Robustheit im Hinblick auf veränderte Manövercharakteristiken ist implizit durch die Formulierung als einkriterielles Problem zur gleichgewichteten Optimierung der Performance in allen Szenarien des Szenarienkatalogs gegeben. Je mehr Fahr Situationen und Variationen dieser in die automatisierte Applikation mit einfließen, desto robuster sind die gefundenen Lösungen in Bezug auf sich ändernde Manöverausprägungen. Um jedoch eine Stabilität hinsichtlich der Fahrzeugdynamik nachzuweisen, wird eine Robustheitsanalyse virtuell gefundener Lösungen in Abhängigkeit von Fahrzeugmodellparametern durchgeführt. Kann eine hohe Stabilität nachgewiesen werden, ist davon auszugehen, dass eine ähnlich gute Performance für verwandte Derivate oder aufgrund veränderter Umgebungsbedingungen angepasstem Übertragungsverhalten zu erwarten ist. Die Bestätigung einer validen Weiterverwendung simulativ optimierter Parametersätze als Startpunkt für die

fahrzeugbasierte Applikation kann somit erbracht werden. Dieser Ansatz zur Validierung der Übertragbarkeit auf den Fahrversuch wird in der Vorveröffentlichung [FFFG20] erstmals beschrieben.

## 4 Dreistufige Applikationsmethodik

Wie im vorherigen Abschnitt konzeptionell beschrieben und in Abbildung 3.1 zusammengefasst, soll im Rahmen dieser Arbeit ein hybrides Fahrzeugmodell entwickelt werden, auf dessen Basis eine virtuelle Identifikation einflussreicher Bereiche im Suchraum und eine Optimierung von Applikationsparametern erfolgen können. Da die Optimierungsmethodik auf der Komplexitätsreduktion aufbaut und beide Verfahren das hybride Fahrzeugmodell nutzen, erfolgt eine Vorstellung der Ansätze entsprechend der in Abbildung 3.1 gezeigten Wirkzusammenhänge. Kapitel 4 beschreibt lediglich die jeweiligen Konzepte, welche im Anschluss implementiert und in einem realistischen Anwendungsszenario erprobt werden (Kapitel 5). Die Validierung erfolgt in Kapitel 6 entsprechend des in Abschnitt 3.3.3 vorgestellten Konzepts.

### 4.1 Hybrides Fahrzeugmodell zur effizienten virtuellen Applikation

Die Diskussion und Bewertung existierender Ansätze zur Modellierung der Fahrzeugdynamik haben ergeben, dass ein geeignetes Modell zur simulativen Applikation und Erprobung automatisierter Fahrfunktionen den Komplexitäts-Genauigkeits-Trade-Off entsprechend Abbildung 2.12 abmildern muss. Da reine neuronale Netze durch ihren Black-Box-Charakter bei Fehlprädiktionen systembedingt schwer nachvollziehbar sind und teilweise eine große Anzahl an Messdaten benötigen, kommt in dieser Arbeit ein kombinierter Ansatz zum Einsatz. Bestehend aus einem Einspurmodell und LSTM-Netzwerk bietet das Konzept des hybriden Modells (HM) aufgrund seiner geringen Komplexität das Potenzial, die geforderte Dynamik hochgenau und recheneffizient wiederzugeben. Auch eine verbesserte Nachvollziehbarkeit des Modelloutputs ist möglich. Im Folgenden wird zunächst das Design des Modells hergeleitet und dabei auf die grundlegende Architektur und Modellgrößen eingegangen. Im

zweiten Schritt erfolgt eine Vorstellung des Trainingskonzepts und der verwendeten Messdaten. Das in den Abschnitten 4.1.1 und 4.1.2 beschriebene Modelldesign und -training wurden in der Vorveröffentlichung [FFFG19b] vom Autor erstmals vorgestellt.

### 4.1.1 Design des hybriden Fahrzeugmodells

Für die Herleitung einer geeigneten Architektur muss im ersten Schritt basierend auf bekannten Modellierungskonzepten (vgl. Abschnitt 2.5.3) eine Verschaltung des ESMs und LSTMs gewählt werden. Die serielle Anordnung beider Modelle entsprechend der *Hammerstein-Struktur* erfordert einen zusätzlichen Modelleingang beim ESM zur Verarbeitung des KNN-Inputs. Sie eignet sich daher nicht für den vorliegenden Anwendungsfall. Die *serielle Wiener-Struktur* stellt eine Alternative serieller Strukturen dar. Da das LSTM in diesem Ansatz die Prädiktion des hybriden Modells übernimmt, besteht die Gefahr einer geringen Nachvollziehbarkeit ähnlich wie bei der alleinigen Nutzung von KNNs. Ein entscheidender Vorteil kombinierter Strukturen ist damit nicht mehr gegeben. Die Wiener-Anordnung eignet sich folglich auch nicht für die vorliegende Problemstellung. Bei einer *parallelen Anordnung* ergibt sich die Prädiktion hingegen durch eine Addition der Ausgänge beider Modelle. Somit ist eine verbesserte Nachvollziehbarkeit und vergleichsweise einfache Parametrierung möglich, solange der vom KNN auszugleichende Modellfehler systematisch ist und nicht durch stochastisches Messrauschen bestimmt wird. Da das Einspurmodell beim Erreichen der Haftgrenze konzeptbedingte Grenzen aufweist oder durch parametrische Ungenauigkeiten fehlerhaft sein könnte, sind systematische Abweichungen zu erwarten. Diese sind durch neuronale Netze abbildbar, sodass eine Anordnung gemäß der parallelen Struktur am vielversprechendsten ist. Sie wird deshalb im Folgenden verwendet und weiter spezifiziert.

Der konzeptionelle Aufbau des hybriden Modells ist in Abbildung 4.1 dargestellt. Entsprechend der Idee paralleler Anordnungen analytischer Modelle und neuronaler Netze ergibt sich der Modellausgang  $\tilde{y}(t)$  als Summe der ESM-Prädiktion  $y_{ESM}(t)$  und dem LSTM-Ausgang  $\tilde{y}_{LSTM}(t)$ . Der Modelleingang  $\vec{u}(t)$  stellt die Soll-Anforderung des Fahrers oder der Fahrerassistenzfunktion dar und wird für das in dieser Arbeit behandelte Problem vom Ausgang der Fahrzeugregelung (vgl. Wirkkette in Abbildung 2.3) vorgegeben. Weil sowohl

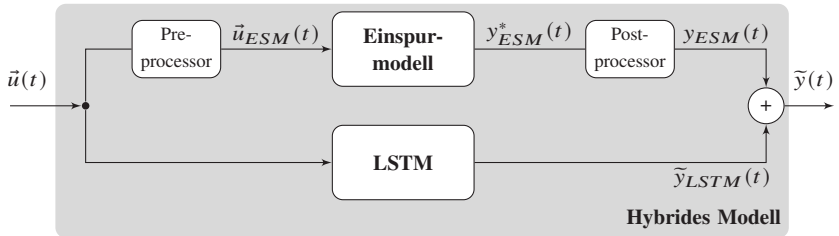


Abbildung 4.1: Aufbau des hybriden Fahrzeugmodells [FFFG19b]

die Trajektorienplanung als auch -regelung mit Krümmungsverläufen  $\kappa_{soll}(t)$  arbeiten und das Übertragungsverhalten neben der Vorgabe einer lateralen Stellgröße maßgeblich durch die Längsgeschwindigkeit  $v(t)$  bestimmt ist (vgl. Abschnitt 2.5.1), wird der Modelleingang wie folgt definiert:

$$\vec{u}(t) = \begin{pmatrix} \kappa_{soll}(t) \\ v(t) \end{pmatrix} \quad (4.1)$$

Da das LSTM-Netz als Input Zeitreihen benötigt, stellen die Variablen  $\kappa_{soll}(t)$  und  $v(t)$  die jeweiligen Zeitreihen einer definierten Länge zum Zeitpunkt  $t$  dar. Das Einspurmodell benötigt statt der Soll-Krümmung den angeforderten Radlenkwinkel  $\delta_{soll}(t)$  als Eingangsgröße (vgl. Abschnitt 2.5.1). Der Zusammenhang zwischen der angeforderten Krümmung der Trajektorienregelung und dem am Rad dazu äquivalenten Lenkwinkel wird maßgeblich durch Übersetzungsverhältnisse und aktuatornahe Regelsysteme entlang der Wirkkette (Lenkung, Differential) beeinflusst. Eine triviale mathematische Umwandlung ist damit nicht möglich. Aus dem Grund wird die Konvertierung von  $\kappa_{soll}(t)$  zu  $\delta_{soll}(t)$  mit Hilfe des ISAR-Modells durchgeführt und das dafür nötige Übertragungsverhalten im Folgenden als Funktion  $\sigma_{ISAR}(\kappa(t))$  abgebildet. Die im *Preprocessor* stattfindende Umwandlung ermöglicht somit die Definition von  $\vec{u}_{ESM}(t)$ :

$$\vec{u}_{ESM}(t) = \begin{pmatrix} \sigma_{ISAR}(\kappa_{soll}(t)) \\ v(t) \end{pmatrix} = \begin{pmatrix} \delta_{soll}(t) \\ v(t) \end{pmatrix}. \quad (4.2)$$

Der Ausgang des hybriden Modells  $\tilde{y}(t)$  stellt die prädizierte Ist-Krümmung  $\tilde{\kappa}(t)$  dar und ergibt sich als Addition der ESM- und LSTM-Prädiktion mit  $y_{ESM}(t)$  und  $\tilde{y}_{LSTM}(t)$  gemäß

$$\tilde{y}(t) = y_{ESM}(t) + \tilde{y}_{LSTM}(t) \Leftrightarrow \tilde{\kappa}(t) = \kappa_{ESM}(t) + \tilde{\kappa}_{LSTM}(t) \quad (4.3)$$

Weil das analytische Modell die Krümmung nicht direkt modelliert, muss diese durch ein *Postprocessing* auf Grundlage des ESM-Outputs  $y_{ESM}^*(t)$  berechnet werden, bevor eine Verrechnung mit dem LSTM-Ausgang möglich ist. Mit Hilfe der aus den Differentialgleichungen des Einspurmodells ableitbaren Gierrate  $\dot{\Psi}(t)$  und der Änderungsrate des Schwimmwinkels  $\dot{\beta}(t)$  lässt sich die Krümmung ausgehend von den geometrischen Beziehungen einer stationären Kreisfahrt

$$v(t) = \dot{\Psi}(t) \cdot R \quad (4.4)$$

unter Berücksichtigung eines Schwimmwinkels ( $\beta \neq 0$ ) wie folgt ableiten:

$$v(t) = (\dot{\Psi}(t) + \dot{\beta}(t)) \cdot R \Leftrightarrow \kappa_{ESM}(t) = \frac{\dot{\Psi}(t) + \dot{\beta}(t)}{v(t)} \quad \text{mit} \quad \kappa_{ESM} = \frac{1}{R} \quad (4.5)$$

Durch Verwendung des Pre- und Postprocessors kann mit Hilfe des hybriden Modells nun eine Prädiktion der Krümmung basierend auf der Soll-Krümmung als Stellgröße des Trajektorienreglers und der aktuellen longitudinalen Geschwindigkeit vorgenommen werden. Eine Einbettung in die Wirkkette als Simulationsmodell der Fahrzeugdynamik ist somit an die vorliegenden Schnittstellen ohne weitere Umwandlung von Messgrößen möglich. Der detaillierte Aufbau des neuronalen Netzes und das Trainingskonzept werden im folgenden Abschnitt beschrieben.

### 4.1.2 Trainingsschema

Basierend auf der bereits festgelegten Architektur des hybriden Modells (Abbildung 4.1) soll nun das Konzept zur Parametrierung der Submodelle vorgestellt werden. Weil das Einspurmodell die grundlegende Dynamik abbildet und das LSTM-Netz lediglich zur Korrektur eingesetzt werden soll, erfolgt die ESM-Parametrierung im ersten Schritt. Hierfür kommen typischerweise Optimierungsalgorithmen zum Einsatz welche die sechs freien Modell-Parameter basierend auf experimentellen Studien im Zielsystem oder eines Referenz-



modells im Hinblick auf eine Minimierung des Prädiktionsfehlers einstellen. Weil die meisten Größen bereits früh im Entwicklungsprozess zur Verfügung stehen (z.B. Masse, Schräglaufsteifigkeiten, Abmessungen) und Referenzbedatungen vergleichbarer Fahrzeuge zumeist ausreichend genau sind, kann eine aufwändige Optimierung in vielen Fällen vermieden werden. Darüber hinaus sind neuronale Netze in der Lage, parametrische Fehler zu lernen, sodass der Aufwand einer ESM-Parametrierung insbesondere im Kontext des hybriden Modells gering gehalten werden sollte.

Ausgehend von einem grob parametrierten Einspurmodell ist es die Aufgabe des neuronalen Netzes Prädiktionsfehler zu erkennen und diese durch Addition einer „Delta-Dynamik“ auszugleichen. Der Trainingsdatensatz muss folglich neben den realen Größen die vom Einspurmodell modellierte Dynamik enthalten. Zur Generierung von Trainingsdaten werden normalerweise Messdaten des realen Prozesses aufgenommen, was in diesem Fall einer Erprobung im Fahrzeug entsprechen würde. Weil aber das in Abschnitt 2.5.1 beschriebene ISAR-Modell die Realität im gesamten Arbeitsbereich automatisierter Fahrfunktionen mit einer hohen Güte abbilden kann, werden Simulationsdaten zum LSTM-Training benutzt. Der Vorteil gegenüber einer Nutzung von Messfahrten besteht neben dem deutlich reduzierten Ressourcenaufwand in der Möglichkeit Manöver synthetisch vorgeben und reproduzierbar simulieren zu können. Außerdem lässt sich die Zusammensetzung des Trainingsdatensatzes frei festlegen, sodass eine höhere Ausgewogenheit von Fahrsituationen erreicht wird, während im Fahrversuch einige Szenarien (wie z.B. hohe Querdynamiken und Geschwindigkeiten) möglicherweise aufgrund der aufwändigen Umsetzung nicht in ausreichendem Umfang zur Verfügung gestellt werden können. Der Nachteil dieses Vorgehens besteht darin, dass das HM im besten Fall nur die Güte des Referenzmodells erreichen kann. Wenn dabei allerdings die Rechenzeit des Modells deutlich reduziert wird, ist das HM dem ISAR-Modell bereits überlegen. Inwiefern ein solches Vorgehen gerechtfertigt ist, wird durch eine Validierung gegenüber dem realen Prozess in Kapitel 6 geprüft, wobei der Nachweis einer Generalisierungsfähigkeit für im Training nicht verwendete Messfahrten erbracht wird. Das beschriebene Trainings- und Validierungskonzept ist in Abbildung 4.2 zusammengefasst. Analog zu der Notation im Schaubild 4.1 stellen  $y_{ISAR}$  und  $y_{Real}$  die (prädizierten) Ist-Krümmungen der jeweiligen Prozesse dar.

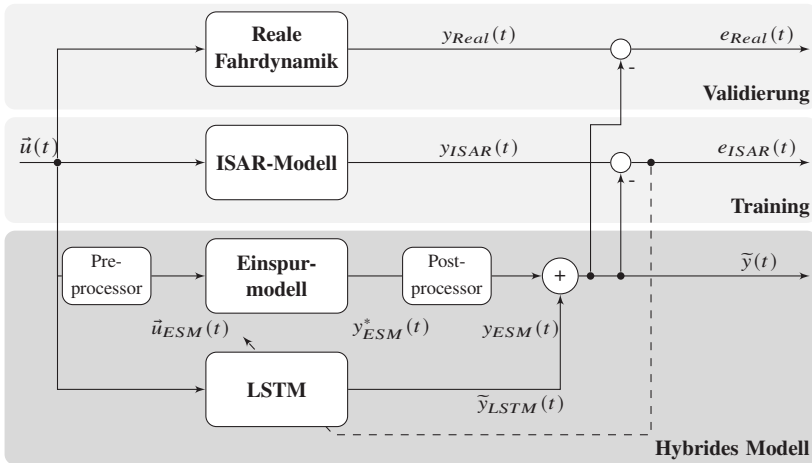


Abbildung 4.2: Trainings- und Validierungskonzept des hybriden Modells [FFFG19b]

Das Training wird anhand eines mit dem ISAR-Modell erzeugten Datensets basierend auf dem Prädiktionsfehler  $e_{ISAR}$  durchgeführt während die Validierung gegenüber des realen Prozesses mit  $e_{Real}$  erfolgt. Die Nutzung des ISAR-Modells zur Messdatenerzeugung bietet die Freiheit, ein breites Spektrum an Szenarien definieren zu können, welche das neuronale Netz bestmöglich zum Erlernen der Fahrzeugdynamik befähigen. Die Softwaremodule der Bewegungsplanung sind darauf optimiert, stetige Trajektorien mit kleinen Gradientenänderungen zu planen, um eine Regelbarkeit zu ermöglichen und einen hohen Fahrkomfort zu erzielen. Trajektorienverläufe, welche diese Charakteristika besitzen, sind beispielsweise durch Polynomfunktionen oder Kreisfunktionen gegeben. In Anlehnung an aus der Regelungstechnik bekannte Ansätze zur Systemidentifikation werden in dieser Arbeit Sinusverläufe unterschiedlicher Frequenz und Amplitude erzeugt, die mit dem ISAR-Modell sowie dem ESM simuliert werden. Die Trajektorien der Soll-Krümmung des Trainingsdatensatzes lassen sich folglich durch Sinusfunktionen beschreiben:

$$\kappa_{soll} = \hat{\kappa}_{soll} \cdot \sin(2\pi f) \quad (4.6)$$

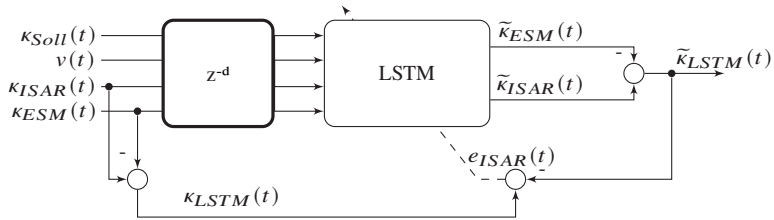
Da der minimal befahrbare Kurvenradius entscheidend von der Geschwindigkeit  $v(t)$  abhängt, wird die Krümmungsamplitude  $\hat{k}_{soll}$  über die maximale Querbeschleunigung  $\hat{a}_y$  festgelegt:

$$\hat{k}_{soll} = \frac{\hat{a}_y}{v^2} \quad (4.7)$$

Über eine vollfaktorielle Variation für die Applikation von Fahrerassistenzfunktionen repräsentativer Querbeschleunigungen  $\hat{a}_y$ , Geschwindigkeiten  $v$  und Frequenzen  $f$  kann mit geringem Aufwand ein breiter Bereich der Fahrdynamik abgebildet werden (Werte finden sich in Tabelle A.1). Da jede Trajektorie eine unterschiedliche Konfiguration der drei Szenarienparameter hat, ist der Datensatz gleichverteilt. Das Risiko, dass das LSTM-Netz aufgrund unzureichend abgebildeter Fahrdynamiken im Trainingsset in der Simulation für diese Situationen eine nicht zufriedenstellende Prädiktionsgüte erreicht, kann somit reduziert werden.

Mit Hilfe der simulativ gewonnenen Sinus-Verläufe kann ein LSTM-Training basierend auf dem Prädiktionsfehler des HMs durchgeführt werden (vgl. Abbildung 4.2). Wie in Abschnitt 2.5.2 beschrieben, nutzen dynamische neuronale Netze zurückliegende Zeitreihen, um den jeweils nächsten Schritt der Ausgangsgröße zu präzisieren. In für LSTMs üblichen Anwendungsfällen (z.B. Spracherkennung, Aktienkursprädiktion) werden dafür oftmals nur kurzfristige Vorhersagen über wenige Zeitschritte getroffen, bevor neue Daten des realen Prozesses wieder zur Verfügung stehen. Für den Einsatz als Fahrzeugmodell ist allerdings nicht davon auszugehen, dass reale Prozessdaten während der Simulation zur Verfügung stehen. Stattdessen ist es die Aufgabe des LSTMs, die reale Fahrdynamik in Abhängigkeit der Sollvorgabe, aktuellen Geschwindigkeit und basierend auf der eigenen Prädiktion vorausgegangener Schritte abzubilden. Da sich das neuronale Netz folglich auf den eigenen Modellausgang verlässt, besteht mit zunehmender Simulationszeit die Gefahr eines kumulierten Prädiktionsfehlers. Um dem Risiko entgegenzuwirken, muss das LSTM-Netz im Hinblick auf die Nutzung rekurrenter Ausgangsgrößen optimiert werden. Dafür kann während des Trainings auf Daten des realen Prozesses (abgebildet durch das ISAR-Modell) zurückgegriffen werden. Das im Folgenden in Anlehnung an [YO04] vorgestellte zweistufige Trainingskonzept sieht ein initiales Open-Loop-Training basierend auf realen Prozessdaten und ein darauf aufbauendes Closed-Loop-Training mit zurückgeführtem Modellausgang vor (vgl.

**Open Loop Training**



**Closed Loop Training**

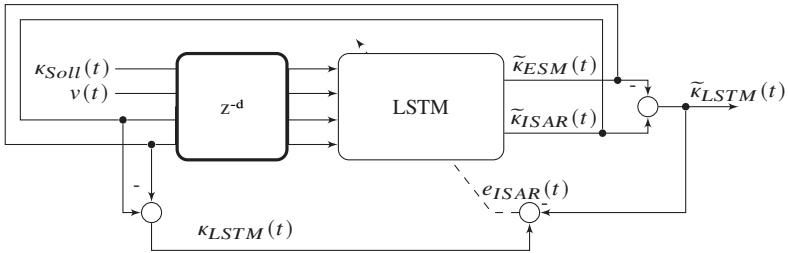


Abbildung 4.3: Zweistufiges Trainingsschema zur Erzielung einer langfristigen Prädiktionsgüte [FFFG19b]

Abbildung 4.3). Die Grundidee besteht darin, die Kantengewichte im Open-Loop-Training zunächst so einzustellen, dass die reale Fahrdynamik kurzfristig basierend auf zurückliegenden Prozessdaten modelliert werden kann. Im zweiten Schritt wird der Kreis geschlossen und das Netz dazu befähigt, den eigenen Modellfehler im Closed-Loop-Setup zu lernen und somit ausgleichen zu können. Die Herausforderung besteht darin, das Training so zu steuern, dass das im ersten Schritt implizit gewonnene Modellwissen über fahrdynamische Zusammenhänge im zweiten Teil des Trainings nicht verloren geht, sondern lediglich für die Nutzung in Closed-Loop-Systemen optimiert wird. Mit dem Konzept, im Training bereits die Ziel-Konfiguration zu verwenden, konnte für vergleichbare Probleme eine Steigerung der Prädiktionsgüte erreicht werden (vgl. [YO04]).

Bei näherer Betrachtung der LSTM-Konfiguration in Abbildung 4.3 fällt auf, dass am Modellausgang zwei Größen berechnet werden: Die zu erwartende Krümmung des ESMs  $\tilde{\kappa}_{ESM}$  und die des Referenzprozesses  $\tilde{\kappa}_{ISAR}$ . Die vom LSTM-Netz bereitgestellte Delta-Dynamik  $\tilde{\kappa}_{LSTM}$  ergibt sich dann aus der Differenz beider Größen. Von einer direkten Modellierung des zu erwartenden Unterschieds zwischen der realen Fahrdynamik und der vom ESM prädizierten wird abgesehen, da experimentelle Untersuchungen gezeigt haben, dass das LSTM diesen nur schwer prädizieren kann. Stattdessen erfolgt eine explizite Modellierung der zu erwartenden Krümmung in beiden Prozessen und eine explizite Berechnung der Abweichung zwischen den Outputs, welche sonst implizit innerhalb des Modells stattfinden müsste. Die Unterschiede in der Prädiktionsgüte lassen sich dadurch begründen, dass das Einspurmodell möglicherweise bereits eine hohe Genauigkeit aufweist und lediglich in dedizierten Situationen fehlerhaft ist. Die zu erlernende Delta-Dynamik hat somit einen hohen Anteil, der durch stochastisches Messrauschen bestimmt ist. Wie bereits in Abschnitt 2.5.3 herausgestellt, stellt das Erlernen von Signalen dieser Charakteristik für neuronale Netze eine besondere Herausforderung dar. Darüber hinaus besteht eine Ähnlichkeit in den Verläufen des Eingangssignals und der Ausgangssignale bei getrennter Modellierung der Prozesse, was das Erlernen der Zusammenhänge vereinfacht.

Das Training erfolgt mit dem Ziel einer Minimierung des Fehlers zwischen dem optimalen LSTM-Output  $\kappa_{LSTM}$  und der aktuellen Prädiktion  $\tilde{\kappa}_{LSTM}$ . Es erfolgt damit eine Minimierung des Fehlers  $e_{ISAR}$ , also des Unterschieds zwischen der prädizierten Ist-Krümmung des ISAR-Modells und derjenigen des hybriden Modells. Der Nachweis, dass die Definitionen von  $e_{ISAR}$  in den Abbildungen 4.3 und 4.2 äquivalent sind, kann erbracht werden, indem die HM-Prädiktion  $\tilde{\kappa}$  (entspricht  $\tilde{y}$  in Abbildung 4.2) als Addition der Einzelmodelle dargestellt wird:

$$\begin{aligned} e_{ISAR} &= \kappa_{ISAR} - \tilde{\kappa} \\ \Leftrightarrow e_{ISAR} &= \kappa_{ISAR} - (\kappa_{ESM} + \tilde{\kappa}_{LSTM}) \end{aligned} \quad (4.8)$$

Wird der ISAR-Ausgang  $\kappa_{ISAR}$  ebenfalls als Summe des ESM-Ausgangs  $\kappa_{ESM}$  und optimalem LSTM-Output  $\kappa_{LSTM}$  dargestellt, ergibt sich die Definition des Trainingsfehlers entsprechend Abbildung 4.3:

$$\begin{aligned} e_{ISAR} &= (\kappa_{LSTM} + \kappa_{ESM}) - (\kappa_{ESM} + \tilde{\kappa}_{LSTM}) \\ \Leftrightarrow e_{ISAR} &= \kappa_{LSTM} - \tilde{\kappa}_{LSTM} \end{aligned} \quad (4.9)$$

Neben den während des Trainings optimierten Kantengewichten müssen im Vorhinein LSTM-Parameter festgelegt werden, welche beispielsweise die Architektur, den Optimierer, die Verlustfunktion oder die Trainingsdauer definieren. Diese im Kontext neuronaler Netze als „Hyperparameter“ bezeichneten Einflussfaktoren werden basierend auf einem hierarchischen Kalibrierungsschema eingestellt, welches in [GSK<sup>+</sup>16] näher beschrieben ist. In der Veröffentlichung wurde eine umfangreiche Sensitivitätsanalyse von Lernrate, Netzwerkgröße und Aktivierungsfunktion anhand bekannter Anwendungsbeispiele durchgeführt. Die Studien konnten zeigen, dass die meisten Hyperparameter unabhängig voneinander eingestellt werden können und die Lernrate sowie Anzahl an Neuronen und Schichten den größten Einfluss haben. Ausgehend von den Erkenntnissen werden zunächst basierend auf 200 Trainingsepochen verschiedene Lernraten gegenübergestellt und das Verfahren im Anschluss unter Evaluierung mehrerer Netzwerkarchitekturen wiederholt. Die im Hinblick auf die Performance in der Closed-Loop-Architektur beste Konfiguration wird weiterverwendet. Da ein hoher Einfluss in der Verteilung der Trainingsepochen zwischen dem Open- und Closed-Loop-Teil zu erwarten ist, werden auch hierfür unterschiedliche Einstellmöglichkeiten erprobt.

Die experimentell bestimmte Parametereinstellung mit der besten Generalisierungsfähigkeit ist zusammenfassend in Tabelle A.3 dargestellt. Darüber hinaus gibt Tabelle A.2 Aufschluss über die verwendete ESM-Parametrierung, welche basierend auf Erfahrungswerten für einen BMW der 7er Baureihe abgeschätzt wurde. Die hohen Werte für die Masse und Gierträchtigkeit ergeben sich aufgrund des hohen Gewichts verbauter Messinstrumente und Sensorik. Die Optimierung des hybriden Modells erfolgt im Hinblick auf dieses Forschungsfahrzeug.

## 4.2 Komplexitätsreduktion des Parameterraums

Neben der Virtualisierung wurde mit dem Ziel der Effizienzsteigerung konventioneller Applikationsprozesse der Komplexitätsreduktion des Parameterraums und Steigerung des Systemverständnisses ein großes Potential zugewiesen (vgl. Abschnitt 3.2). Das etablierte Verfahren sieht ein Parametertuning basierend auf Erfahrungswissen oder experimenteller Studien vor, was ein vollständiges Erfassen aller Wechselwirkungen und situationsspezifischer Einflüsse unmöglich macht. Auch die Identifikation sensitiver Wertebereiche birgt gerade im Vergleich zu konservativen Abschätzungen die Chance eine erhöhte Konvergenzgeschwindigkeit für automatisierte als auch manuelle Parametrierungen zu erreichen. Während der Einsatz von Sensitivitätsanalysen im Zusammenhang mit Optimierungsalgorithmen bereits weit verbreitet ist, kann eine Ableitung einflussreicher Parametergrenzen basierend auf bekannten Ansätzen aus der Systemidentifikation nur mit hohem Aufwand durchgeführt werden. Um eine effiziente Einschränkung des Suchraums auf relevante Bereiche zu ermöglichen, wird deshalb im Folgenden ein Verfahren vorgestellt, welches basierend auf der Elementareffektmethode (vgl. Abschnitt 2.3.1) szenarienabhängig sensitive Parameter und Wertebereiche herausstellen kann. Das Verfahren wurde in [FFFG19a] veröffentlicht und wird in diesem Kapitel 4.2 nochmals näher ausgeführt. Die Integration der Ergebnisse in die automatisierte Applikation wird im nachfolgenden Abschnitt 4.3 beschrieben.

### 4.2.1 Sampling

Da bei der Durchführung einer EEM üblicherweise das Erstellen eines Versuchsplans (oder das Samplen des Suchraums) an erster Stelle steht, wird die Aufteilung des Parameterraums zur lokalen Anwendung der SA und das dazugehörige Sampling-Schema auch in dieser Arbeit als Erstes thematisiert. Das Sampling-Konzept hat dabei die Aufgabe, einen Kompromiss zwischen der ausreichenden Abdeckung relevanter Bereiche und Minimierung der dafür benötigten Punkte zu finden.

Um gültige Parametergrenzen zu identifizieren, wird der gesamte Wertebereich im Hinblick auf einflussreiche Teilabschnitte durchsucht. Da das Verfahren auf der Annahme basiert, dass konservativ abgeschätzte Grenzen  $UG_i$  und  $OG_i$

für den Parameter  $P_i$  als Ausgangsbasis vorliegen, wird davon ausgegangen, dass der Wertebereich zu groß gewählt wurde und folglich reduzierte Einflüsse in den äußeren Regionen zu erwarten sind. Inwiefern diese Annahme gerechtfertigt ist, wird im späteren Teil der Arbeit in Abschnitt 5.3 diskutiert. Im ersten Schritt erfolgt eine Unterteilung des Definitionsbereichs der Parameter in der Form, dass  $n_{Int}$  gleich große Intervalle gebildet werden. Mit  $k \in \{1, \dots, n_{Int}\}$  als Intervallindex können somit die lokalen Grenzen  $UG_i^*$  und  $OG_i^*$  der Subregionen definiert werden:

$$\begin{aligned}UG_i^*(k) &= UG_i + (k - 1) \cdot \frac{OG_i - UG_i}{n_{Int}}, \\OG_i^*(k) &= UG_i + k \cdot \frac{OG_i - UG_i}{n_{Int}}.\end{aligned}\tag{4.10}$$

Wird nun eine EEM innerhalb einer dieser Subregionen durchgeführt, können die lokalen Einflusswerte des jeweiligen Parameters in diesem Teilbereich berechnet werden. Die resultierenden Sensitivitätsindizes geben Aufschluss darüber, wie groß der Effekt in dem untersuchten Abschnitt bei ansonsten unveränderten Wertebereichen anderer Parameter ist. Für das Sampling können prinzipiell alle für Morris-SAs verfügbaren Schemata zum Einsatz kommen, jedoch wird im Folgenden zur Verdeutlichung ein radialer Ansatz mit Latin Hypercube Sampling verwendet. Der zugehörige Versuchsplan  $X_k^i$  für Parameter  $P_i$  und Intervall  $k$  definiert sich entsprechend Gleichung 4.11, wobei die Dimensionen der Matrix durch die Anzahl an EEM-Samples pro Intervall  $n_{EEM}^* = r \cdot (n_p + 1)$  (vgl. Gleichung 2.1) und die Anzahl an Parametern  $n_p$  bestimmt ist:

$$X_k^i \in \mathbb{R}^{n_{EEM}^* \times n_p}\tag{4.11}$$

Bei der Definition von  $X_k^i$  sollte berücksichtigt werden, dass ein Sampling durchaus auch in mehr als einem Intervall stattfinden kann. Eine beispielhafte Notation wäre  $X_{[1, n_{Int}]}^1$ , welche die Erstellung eines Versuchsplan zur Analyse des Einflusses von Parameter  $P_1$  in den Intervallen 1 und  $n_{Int}$  beschreiben würde. Eine Umsetzung dieses Samplings im zweidimensionalen Raum ist in Abbildung 4.4 dargestellt. Analog zu der Analyse nur eines Intervalls geben die auf Grundlage dessen berechneten Sensitivitätswerte Aufschluss über den Einfluss des Parameters bei Variation der Werte innerhalb Intervall 1 und  $n_{Int}$ . Werden hingegen alle Intervalle untersucht, handelt es sich um eine Anwendung der Elementareffektmethode entsprechend des gängigen Vorgehens,



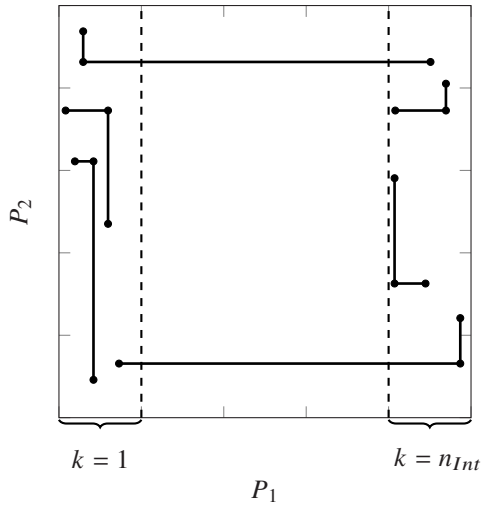


Abbildung 4.4: Beispielhafter Versuchsplan  $X_{[1, n_{Int}]}^1$  zur Sensitivitätsanalyse in den Intervallen 1 und  $n_{Int}$  [FFFG19a]

wobei die globalen Sensitivitäten der Parameter im gesamten Suchraum berechnet werden. Die entsprechende Notation sei im Folgenden durch  $X_{[1, \dots, n_{Int}]}^i$  gegeben.

### 4.2.2 Ableitung gültiger Wertebereiche

Wie zuvor herausgestellt, ist das Vorliegen gültiger Parametergrenzen eine Voraussetzung für die globale Anwendbarkeit der SA. Sind die Ober- und Untergrenzen zu groß abgeschätzt, ist die einflussreiche Region eines Parameters auf einen kleinen Teil des Wertebereichs beschränkt. In Folge dessen werden die in diesem Abschnitt großen Elementareffekte von den kleinen Elementareffekten des übrigen nicht-sensitiven Bereichs überkompensiert. Der Grund dafür ist die Berechnung der Sensitivitätsindizes, welche eine Mittelung über alle Elementareffekte entlang des Wertebereichs vorsieht (vgl. Gleichungen 2.4 und 2.5). Dem zugehörigen Parameter würde folglich fälschlicherweise eine zu geringe Sensitivität zugeordnet.

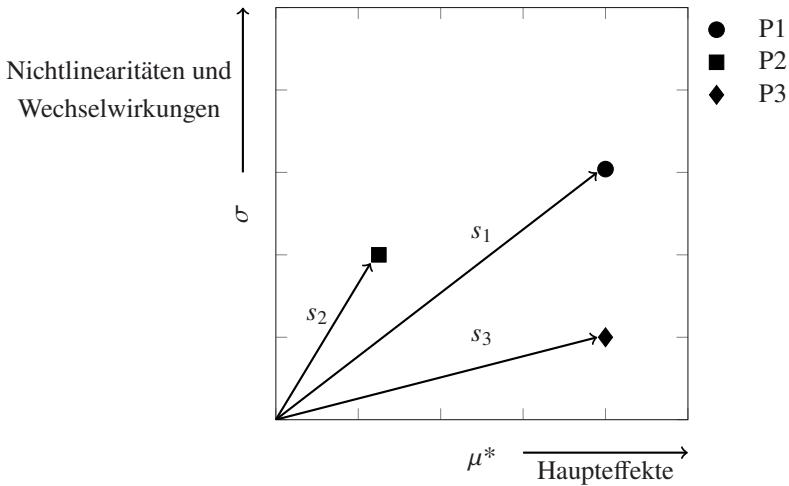


Abbildung 4.5: Bildung der euklidischen Distanz im  $\mu^*$ - $\sigma$ -Diagramm zur Zusammenfassung und Normierung von Sensitivitätswerten [FFFG19a]

Auf Grundlage des im letzten Abschnitt eingeführten Sampling-Konzepts kann die EEM in einem oder mehreren Intervallen durchgeführt und somit lokale Sensitivitätsinformationen berechnet werden. Die von der EEM zur Verfügung gestellten Metriken  $\mu^*$  und  $\sigma$  geben Aufschluss über den Haupteffekt und Wechselwirkungen beziehungsweise Nichtlinearitäten im  $\mu^*$ - $\sigma$ -Diagramm (vgl. Abbildung 2.7). Um einen Vergleich zwischen Szenarien und Parametern anhand eines Sensitivitätswerts zu ermöglichen, werden die Ausgänge der EEM zusammengefasst und auf einer Skala zwischen 0 und 1 normiert. Entsprechend der graphischen Äquivalenz, nach der Parameter einflussreicher klassifiziert werden, je weiter oben rechts sie sich im Diagramm befinden, wird die euklidische Distanz zwischen dem Koordinatenursprung und dem jeweiligen Punkt gebildet (vgl. Abbildung 4.5):

$$s_i = \sqrt{\mu_i^{*2} + \sigma_i^2} \quad (4.12)$$

Aufgrund des qualitativen Charakters der EEM sind die tatsächlichen Werte von  $\mu^*$  und  $\sigma$  wenig aussagekräftig und damit auch die kombinierte Sensitivität  $s_i$ . Eine Normierung der Einflüsse in Bezug auf den Parameter mit der

größten Sensitivität bietet allerdings die Chance, Sensitivitätswerte mehrerer Anwendungen der SA mit gleichen Parametern vergleichen zu können. Vor dem Hintergrund der Notwendigkeit, eine EEM für alle Szenarien des Szenarienkatalogs durchführen zu müssen, werden die kombinierten Sensitivitäten normiert. Die relative Sensitivität  $s_i^{rel} \in [0, 1]$  kann wie folgt berechnet werden:

$$s_i^{rel} = \frac{s_i - \min_{j=1, \dots, n_p} s_j}{\max_{j=1, \dots, n_p} s_j - \min_{j=1, \dots, n_p} s_j} \quad (4.13)$$

Basierend auf dem vorgestellten Sampling-Schema und der eingeführten Berechnungsvorschrift für relative Sensitivitäten wird zur Ableitung einflussreicher Parametergrenzen der Wertebereich jedes Parameters sukzessive verkleinert, indem Intervalle auf Grundlage lokaler Sensitivitäten als nicht relevant klassifiziert werden. Das methodische Vorgehen ist in Abbildung 4.6 dargestellt.

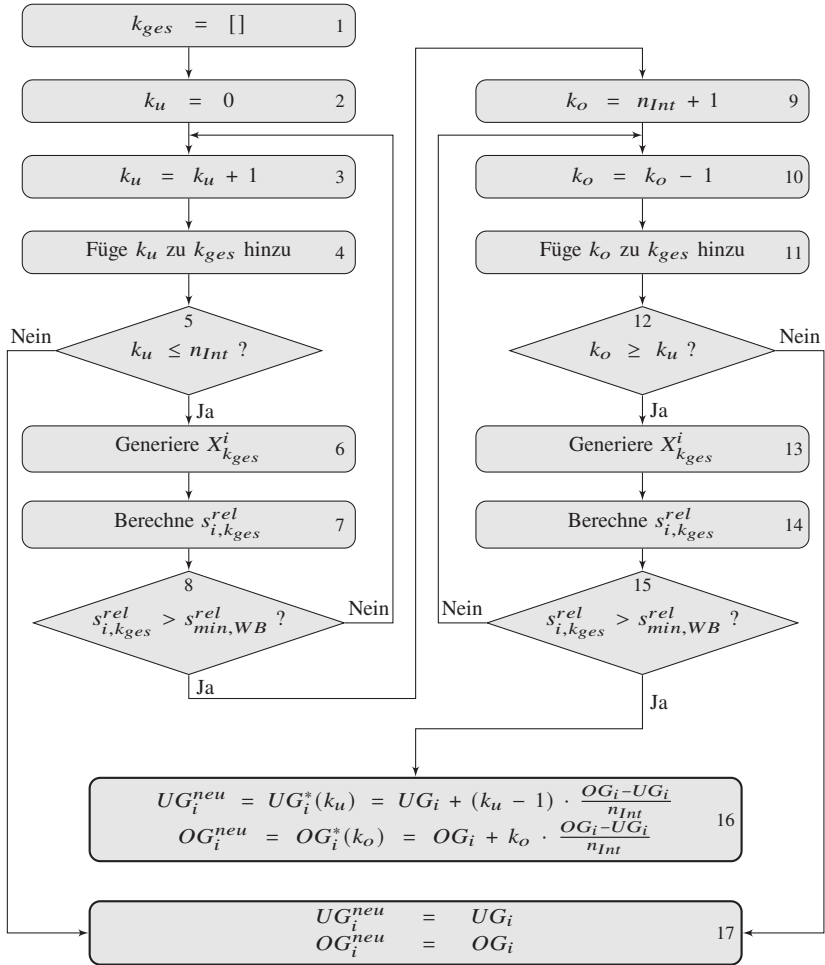


Abbildung 4.6: Methodik zur Identifikation einflussreicher Parametergrenzen für Parameter  $P_i$  [FFFG19a]

Im ersten Schritt (1) wird ein Vektor  $k_{ges}$  definiert, welcher die Indizes der Intervalle speichert, die bei der nächsten lokalen Anwendung der EEM berücksichtigt werden sollen. Zur Analyse der Einflüsse in den Subregionen wird der

Intervallindex für die Bestimmung der unteren Grenze  $k_u$  als Erstes initialisiert (2), in Folge immer um 1 erhöht (3) und dem Vektor  $k_{ges}$  hinzugefügt (4). Solange der Index noch unter der Anzahl definierter Intervalle  $n_{Int}$  liegt (5), wird im Anschluss ein Versuchsplan für den Parameter  $P_i$  in den in  $k_{ges}$  definierten Abschnitten erzeugt (6). Auf Grundlage dessen lassen sich im nächsten Schritt die Sensitivitätsmetriken der EEM berechnen, welche mit Hilfe der Gleichungen 4.12 und 4.13 in eine relative Sensitivität  $s_{i,k_{ges}}^{rel}$  des jeweiligen Parameters  $P_i$  in den Intervallen laut  $k_{ges}$  umgewandelt werden können (7). Um auf Grundlage dessen eine Entscheidung hinsichtlich der Wichtigkeit geprüfter Intervalle zu fällen, muss eine Relevanzschwelle  $s_{min,WB}^{rel}$  definiert werden. Liegt der Parametereinfluss  $s_{i,k_{ges}}^{rel}$  unterhalb dieser Schwelle (8), wird das entsprechende Intervall als nicht sensitiv eingestuft und der Intervallindex erhöht (3). Es werden so lange neue Intervalle der Auswertung hinzugefügt (4) und ausgewertet (5-7) bis das Einschließen eines Intervalls das erste Mal ein Überschreiten der Relevanzschwelle auslöst (8) oder  $k_{ges}$  alle Abschnitte des Wertebereichs beinhaltet und der Einfluss trotzdem noch vernachlässigbar klein ist (5). In diesem Fall entspricht die lokale SA einer globalen Einflussanalyse, sodass der Parameter theoretisch für weitere Betrachtungen nicht mehr berücksichtigt werden müsste. Zu diesem Zeitpunkt werden die Grenzen jedoch trotzdem noch unverändert gelassen (17). Eine Diskussion der Nutzung dieser Informationen zum Ausschluss von Parametern erfolgt in Abschnitt 4.2.3. Ist auf der anderen Seite die Suche nach dem ersten einflussreichen Intervall beginnend von der unteren Grenze abgeschlossen, wird der Wertebereich ausgehend von dem obersten Intervall weiter reduziert (9-10). Ein Abbruch des Verfahrens erfolgt genau wie in Block (5) gemäß der Vorschrift in (12) dann, wenn durch Verkleinerung von  $k_o$  ein bereits durch  $k_u$  ausgewertetes Intervall erreicht wird. In dem Fall verändert sich genau wie zuvor die lokale Analyse zu einer globalen Anwendung der SA. Der Vektor mit bereits gefundenen Indizes vernachlässigbarer Intervalle  $k_{ges}$  wird im zweiten Teil der Methode lediglich erweitert (11), sodass das Sampling (13) und die Anwendung der EEM (14) immer innerhalb aller bisher als nicht sensitiv klassifizierten Bereiche einschließlich neu hinzugefügter Intervalle durchgeführt wird. Analog zu dem Entscheidungsblock (8) wird auch in (15) überprüft, ob eine Relevanzschwelle überschritten wird. Solange dies nicht der Fall ist, werden neue Intervalle mit absteigendem Index hinzugefügt und die EEM erneut vollzogen. Führt eine Erhöhung von  $k_o$  zur Erfüllung der Bedingung in (15), ist die Bestimmung

einflussreicher Bereiche abgeschlossen und neue Grenzen können gemäß der Berechnungsvorschrift in Block (16) definiert werden.

Zusammenfassend basiert die beschriebene Methodik darauf, den Wertebereich jedes Parameters schrittweise einzuschränken und die recheneffiziente Elementareffektmethode wiederholt lokal durchzuführen. Das Verfahren muss für jeden Parameter nacheinander durchgeführt werden, wobei bereits reduzierte Grenzen durchgeführter Auswertungen für nachfolgende Anwendungen des Verfahrens Berücksichtigung finden sollten. Global gesehen besteht die Chance, den Suchraum mit jedem neuen Durchlauf der Methode für neue Parameter zu verkleinern. Entsprechend des Vorgehens in Abbildung 4.6 wird mit dem kleinsten Intervallindex gestartet und die untere Grenze solange erhöht, bis ein nicht mehr zu vernachlässigender Einfluss entsteht. Im Anschluss erfolgt in ähnlicher Art und Weise eine Reduktion der oberen Grenze unter Berücksichtigung bereits als nicht einflussreich klassifizierter Intervalle im unteren Abschnitt des Wertebereichs. Genauso kann das Verfahren aber auch umgekehrt werden und eine schrittweise Reduktion ausgehend von der oberen Grenze erfolgen. Kann mit der Methode eine Verkleinerung der Grenzen erreicht werden bedeutet das, dass Parametervariationen innerhalb nicht-sensitiv klassifizierter Bereiche einen vernachlässigbaren Einfluss auf das Gütekriterium haben. Im Umkehrschluss kann gefolgert werden, dass eine Erhöhung oder Verkleinerung eines Parameterwerts über die Grenzen des relevanten Bereichs hinaus keine großen Unterschiede mehr im Ergebnis hervorrufen wird. Eine Einschränkung von Definitionsbereichen entsprechend des vorgestellten Verfahrens ist also gerechtfertigt. Die Methode hat insbesondere dann großes Potential, wenn ein Abfall des Parametereinflusses in den äußeren Intervallen des Suchraums zu erwarten ist. Ob diese Charakteristik für das beschriebene Optimierungsproblem gegeben ist, wird im Laufe der Arbeit bei der Anwendung und Diskussion des Vorgehens evaluiert.

### **4.2.3 Dimensionalitätsreduktion und Steigerung des Systemverständnisses**

Wie bereits einleitend erwähnt, kann eine sinnvolle Anwendung der globalen Sensitivitätsanalyse nur unter Voraussetzung korrekt bestimmter einflussreicher Wertebereiche durchgeführt werden. Da eine lokale Anwendung von Sensitivitätsanalysen entsprechend des in Abschnitt 4.2.2 beschriebenen Ver-

fahrens unter Berücksichtigung aller Intervalle bereits einer globalen SA entspricht, können die dadurch als nicht einflussreich klassifizierten Parameter bereits für die weitere Betrachtung ausgeschlossen werden. Für die übrigen Parameter wird ein Sampling innerhalb sensitiver Grenzen  $UG_i^{neu}$  und  $OG_i^{neu}$  durchgeführt, welches als Grundlage für die Anwendung der Elementareffekt-methode dient. Um auch in diesem Fall die Ergebnisse zwischen den Szenarien vergleichen zu können, erfolgt eine Zusammenfassung und Normierung der Sensitivitätsmetriken mit Hilfe der Gleichungen 4.12 und 4.13. Ähnlich zu dem Vorgehen zur Reduktion von Wertebereichen wird für den Ausschluss nichtsensitiver Parameter eine Relevanzschwelle  $s_{min}^{rel}$  eingeführt, welche sich von der weiter oben beschriebenen Schwelle unterscheiden kann. Im Hinblick auf den qualitativen Charakter der verwendeten SA und der Tatsache, dass Ungenauigkeiten entlang der gesamten Wirkkette auftreten können, welche eine Übertragbarkeit der Ergebnisse in den Fahrversuch zusätzlich erschweren, sollte eine konservative Abschätzung der Relevanzgrenze vorgenommen werden. Um eine Reproduzierbarkeit und Gültigkeit der Ergebnisse innerhalb des verwendeten Setups sicherzustellen, kann die Anzahl verwendeter Samples erhöht werden. Zur Findung eines optimalen Kompromisses zwischen dem benötigten Rechenaufwand und Genauigkeit eignen sich Konvergenzanalysen des Sampling-Parameters  $n_{EEM}$  (vgl. z.B. Bootstrapping-Ansatz, Abschnitt 2.3.1)

#### 4.2.4 Evaluierung der Komplexitätsreduktion

Mit dem zuvor beschriebenen Vorgehen lassen sich in einem zweistufigen Prozess zunächst Wertebereiche von Parametern und im Anschluss Dimensionen des Parameterraums einschränken. Um die erzielte Komplexitätsreduktion anhand von Kennwerten messbar zu machen, werden im Folgenden zwei Metriken beschrieben, welche vom Autor erstmals in [FFFG19a] veröffentlicht wurden. Weil ein Vergleich der Volumen  $n_p$ -dimensionaler Hyperräume nur dann valide Aussagen liefert, wenn die Anzahl an Dimensionen gleich ist, erfolgt eine Definition von zwei unabhängigen Kennwerten. Die Dimensionalitätsreduktion  $DR$  ist definiert als die relative Verkleinerung der Parameteranzahl im Vergleich zur initialen Dimensionalität des Suchraums. Eine Berechnung ist

mit Gleichung 4.14 möglich, wobei  $n_p^*$  die reduzierte Anzahl einflussreicher Dimensionen darstellt.

$$DR = 1 - \frac{n_p^*}{n_p} \quad (4.14)$$

Die Bewertung des Potentials reduzierter Wertebereiche wird mit dem als Parameterraum-Reduktion  $PRR$  definierten Kennwert durchgeführt. Dieser ist definiert als relative Verkleinerung des Suchraum-Volumens basierend auf den als einflussreich klassifizierten Dimensionen  $n_p^*$ :

$$PRR = 1 - \prod_{i=1}^{n_p^*} \sqrt{\left( \frac{OG_i^{neu} - UG_i^{neu}}{OG_i - UG_i} \right)^2} \quad (4.15)$$

Je höher die Kennwerte  $DR$  und  $PRR$  der Komplexitätsreduktion, desto kleiner ist der für die Applikation als einflussreich definierte Suchraum im Vergleich zu dem am Anfang vorliegenden Raum. Die Metriken sollten lediglich als Vergleichsgrundlage dienen, um beispielsweise die Reduktion in mehreren Szenarien gegenüberzustellen. Ein direkter Zusammenhang zu der eingesparten Rechenzeit für die Optimierung kann nicht geschlussfolgert werden.

### 4.3 Simulative Optimierung von Applikationsparametern

Basierend auf einer Einschränkung des Suchraums gemäß des im letzten Abschnitt vorgestellten Verfahrens sieht die Applikationsmethodik entsprechend Abbildung 3.1 eine darauf aufbauende virtuelle Optimierung vor. Diese nutzt die Zeit- und Kostenvorteile von Simulationswerkzeugen, um einen Parametersatz mit hohem Reifegrad zu bestimmen und in Folge dessen den Aufwand fahrzeugbasierter Applikationen zu verringern. Zur Erarbeitung eines Vorgehens zur Optimierung werden im ersten Teil basierend auf den theoretischen Grundlagen zwei Optimierungsmethoden experimentell gegenübergestellt, welche eine vielversprechende Performance für das in dieser Arbeit adressierte Problem vermuten lassen (vgl. Abschnitt 2.4.2). Im Anschluss wird eine Methodik eingeführt, die eine Skalierbarkeit der automatisierten Applikation auf umfangreichere Szenarienkataloge und Parameterräume ermöglicht, indem die Ergebnisse der Komplexitätsreduktion in den Optimierungsprozess



einbezogen werden. Damit besteht die Chance, die Anzahl an Systemevaluationen gegenüber dem konventionellen Verfahren zu reduzieren.

### 4.3.1 Auswahl eines Optimierungsverfahrens

Wie bereits in Abschnitt 2.4.2 herausgestellt, ist die Auswahl in Frage kommender Optimierungsverfahren für das untersuchte Optimierungsproblem (vgl. Abbildung 2.8) aufgrund seiner Charakteristik beschränkt. Ein Einsatz gradientenbasierter Verfahren wäre mit einem nicht gerechtfertigten Rechenaufwand zur numerischen Approximation von Funktionsableitungen verbunden. In der Kategorie gradientenfreier Methoden können aufgrund der Nichtlinearität des Problems lediglich globale Suchalgorithmen sinnvoll zum Einsatz kommen. Da die Eignung eines Verfahrens stark von den Eigenschaften des Suchraums abhängt (*No-Free-Lunch-Theorem*, vgl. Abschnitt 2.4.2) und sich die Partikelschwarmoptimierung (PSO) sowie der genetische Algorithmus (GA) für Probleme vergleichbarer Charakteristik als vielversprechend erwiesen haben, werden diese im Folgenden experimentell gegenübergestellt. Um mehrere Konfigurationen der Verfahren mit vertretbarem Aufwand vergleichen zu können, wird ein, gegenüber dem in Abbildung 2.8 gezeigten, vereinfachtes Applikationsproblem formuliert. Das Grundkonzept der Vergleichsstudie wurde vom Autor in [FFFG20] bereits veröffentlicht und dient damit als Referenz. Im Rahmen dieser Arbeit wird jedoch ein anderes Optimierungsproblem verwendet.

#### Konzeption und Optimierungsproblem

Sowohl der GA als auch die PSO gehören zur Klasse populationsbasierter Algorithmen, weshalb ihnen viele Eigenschaften gemein sind. Beide Verfahren haben ein biologisches Vorbild. Individuen der PSO werden durch Partikel eines Schwarms dargestellt, welche gesamthaft eine Orientierung zu der global besten Lösung anstreben. Die Individuen des GA arbeiten gegeneinander und versuchen unabhängig voneinander zur besten Lösung zu gelangen (Prinzip des *Survival of the fittest*). Vergleiche beider Methoden zeigen bei einer Vielzahl von einkriteriellen Problemen ähnlich gute Ergebnisse, wobei die PSO in einigen Untersuchungen einen Effizienzvorteil aufweisen konnte [HCD-WV05, EHG05]. Welche Suchstrategie sich allerdings für den konkreten Anwendungsfall am besten eignet, wird experimentell untersucht. Eine Gegen-

überstellung der Ergebnisse erfolgt basierend auf dem Optimierungsergebnis sowie der dafür benötigten Rechenzeit.

Wie Abschnitt 2.4.2 herausstellt, verfügen beide Ansätze über Zufallsanteile, die das Fortschreiten der Population im Suchraum beeinflussen, sodass Ergebnisse unter Umständen nicht reproduzierbar sind. Eine Beeinflussung der Suchstrategie kann allerdings über die freien Parameter der Algorithmen vorgenommen werden. Diese dienen in erster Linie dazu, die Ausprägungen der Breiten- und Tiefensuche zu steuern. Während die Tiefensuche (*exploitation*) beim GA maßgeblich durch die Selektion getrieben wird, tragen der Rekombinations- und Mutationsoperator eher zur Breitensuche (*exploration*) bei. Die *exploitation* wird bei der PSO durch die Ausrichtung an der global besten Lösung realisiert und die *exploration* durch Orientierung an der lokal besten bisherigen Lösung des Partikels (vgl. Abbildung 2.9). Um eine Vergleichbarkeit beider Optimierungsverfahren zu ermöglichen, werden diese für die im folgenden beschriebenen Studien gleichermaßen im Hinblick auf Tiefen- und Breitensuche parametrisiert.

Die PSO verfügt über die zwei Einstellparameter  $c_1$  und  $c_2$ , welche Gewichtungsfaktoren für die *exploration* und *exploitation* darstellen (vgl. Gleichung 2.17). Für eine gleichgewichtete Berücksichtigung beider Suchstrategien ergibt sich somit die Forderung nach  $c_1 = c_2 = 0,5$  [Ped10, Tre03]. Der genetische Algorithmus verfügt hingegen über mehr Einstellmöglichkeiten, welche die Eigenschaften und Parametrierung der evolutionären Phasen betreffen. Wie bereits in Abschnitt 2.4.2 herausgestellt, hat sich die Turniers Selektion in den meisten Anwendungsfällen als geeignetes Verfahren erwiesen und wird im Folgenden verwendet. Um eine Konvergenz der Optimierung sicherzustellen, sollte die Selektionswahrscheinlichkeit mindestens 0,5 betragen. Ein zu hoher Selektionsdruck birgt allerdings die Gefahr, nur lokal optimale Lösungen zu finden. Die Wahrscheinlichkeit  $p_s$  wird deshalb auf 0,8 gesetzt [PCM14]. Die auf selektierte Individuen angewandte Rekombination ist durch die Wahrscheinlichkeit  $p_r$  parametrisierbar und sollte ebenfalls nicht zu groß gewählt werden um eine zu schnelle Konvergenz zu vermeiden. Nach [Ho192] sollte  $p_r$  mindestens 0,6 betragen und wird deshalb für diese Arbeit zu 0,7 gewählt, da dieser Wert für vergleichbare Probleme gute Ergebnisse geliefert hat [Sit14]. Als Kreuzungsverfahren kommt das 2-Punkt-Crossover aufgrund des guten Kompromisses zwischen Tiefen- und Breitensuche zum Einsatz [PCM14]. Die letzte Phase des GA stellt die Mutation dar. Weil Individuen in diesem Schritt

zufällig modifiziert werden, muss eine konservative Schätzung der Mutationswahrscheinlichkeit  $p_m$  erfolgen um eine zu starke Tendenz hin zur Zufallssuche zu vermeiden. Als Verfahren wird der weit verbreitete Bit-Flip-Ansatz verwendet, welcher das Invertieren von Bits entlang des Binärstrangs vorsieht (vgl. Abschnitt 2.4.2). Die Wahrscheinlichkeit  $p_m$  wird auf 0,2 gesetzt und somit eine zu ausgeprägte Breitensuche unterbunden [HM90]. Die Parametrierung beider Optimierungsalgorithmen ist zusammenfassend in Tabelle A.4 dargestellt.

Neben der Parametrierung hat die Populationsgröße einen großen Einfluss auf das Konvergenzverhalten. Mit steigender Anzahl an Individuen pro Generation steigt der Anteil nicht-dominierter Lösungen. Damit kann ein zu geringer Selektionsdruck und damit langsamere Konvergenz einhergehen, da populationsbasierte Algorithmen diese Individuen gleich in ihrer Fitness bewerten. Wird sie allerdings zu gering gewählt, bleiben Teile des Suchraums eventuell unerforscht. Die Populationsgröße stellt für beide Verfahren einen Stellhebel dar und wird deshalb systematisch variiert (350, 500, 1000 und 2000 Individuen). Um den Zufallscharakter der Optimierung zu entkräften, wird jeder Durchlauf drei mal wiederholt und der Median aller Durchgänge (in Bezug auf Optimierungsergebnis und Konvergenz) zum Vergleich herangezogen. Darüber hinaus wird zur Initialisierung der Algorithmen kein zufälliges Sampling-Verfahren genutzt, sondern ein deterministischer Ansatz (Halton-Sequenz, vgl. [KN12]), welcher für eine gleichmäßige Verteilung der Samples im Raum sorgt, aber bei gleicher Anzahl an Individuen immer den gleichen Versuchsplan liefert.

Wie zuvor erwähnt, erfolgt der experimentelle Vergleich beider Studien an einem vereinfachten System. Das in Abbildung 2.8 dargestellte Setup wird dafür in seiner Komplexität reduziert, um bei vergleichbarer Charakteristik eine umfangreiche Gegenüberstellung der beiden Optimierer in vertretbarer Rechenzeit zu ermöglichen. Da die Systemdynamik in erster Linie durch die Wirkkette beeinflusst wird, bleibt diese unverändert. Zum Einsatz kommt eine Level-2-Fahrerfunktion, welche den Fahrer bei der Querführung in der Fahrspur unterstützt (Lane-Keeping-System). Der Szenarienkatalog besteht in diesem Fall lediglich aus einem Szenario, welches ein Spurhalte-Manöver bei konstanter Geschwindigkeit auf einer Referenzstrecke beschreibt. Mit Hilfe von acht Applikationsparametern, die in erster Linie in der Trajektorienplanung verortet sind, wird eine Minimierung der Pendelbewegung (laterale Abweichung von der Spurmitte) angestrebt. Zur Messung der Güte verschiedener

Parametrierungen kommt eine Regressionsfunktion  $F(f, A, v)$  zum Einsatz, welche der gemessenen Pendelbewegung einen Komfortwert auf einer Skala von 0 bis 26 ( $[0; 26]$ ) zuordnet und von der BMW Group in Probandenstudien herausgefunden wurde. Die Zielfunktion der Optimierung nimmt die aktuelle Geschwindigkeit  $v$  sowie Amplitude  $A$  und Frequenz  $f$  der Querablage als Eingangsparameter. Die Variablen  $\beta_0, \beta_1, \beta_2$  und  $\beta_3$  stellen Regressionskonstanten dar:

$$F(f, A, v) = \beta_0 + \beta_1 A + \beta_2 f + \beta_3 v. \quad (4.16)$$

### **Partikelschwarmoptimierung**

Die Optimierungsergebnisse der Partikelschwarmoptimierung sind in Abbildung 4.7 dargestellt. Für alle betrachteten Populationsgrößen wurden maximal 100 Iterationen berechnet und der jeweils beste Durchlauf im Plot aufgetragen. Zunächst wird deutlich, dass offensichtlich viel weniger Durchläufe benötigt werden und der Algorithmus bereits nach circa 30 Iterationen konvergiert. Das beste Optimierungsergebnis ( $F(f, A, v) = 21, 1$ ) wird bei 2000 Individuen pro Generation nach 13 Optimierungsschritten erreicht. Die dafür benötigte Anzahl an Systemevaluationen ( $13 \cdot 2000 = 26000$ ) ist größer als für alle anderen Populationsgrößen. Da ein gutes Optimierungsergebnis jedoch gegenüber der Anzahl an Systemevaluationen priorisiert wird und sich die tatsächlich benötigte Rechenzeit nach dem Potential der Parallelisierbarkeit von Simulationen richtet, ist eine Durchführung mit 2000 Individuen basierend auf diesem Ergebnis zu bevorzugen.

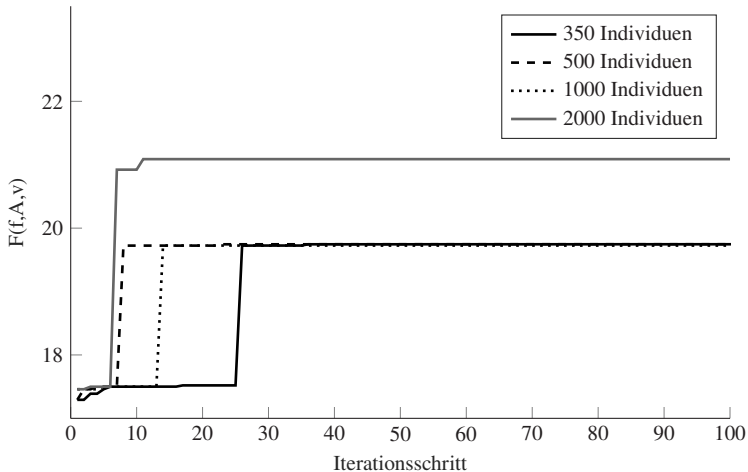


Abbildung 4.7: Optimierungsergebnisse der Partikelschwarmoptimierung

### Genetischer Algorithmus

Analog zur Analyse der PSO erfolgt die Durchführung der Studien mit dem genetischen Algorithmus. Die Ergebnisse in Abbildung 4.8 zeigen zunächst die gleiche Tendenz, dass Durchläufe mit mehr Samples pro Generation einen besseren Zielfunktionswert mit weniger Iterationsschritten ermöglichen. Sowohl der Durchlauf mit 1000 als auch mit 2000 Individuen sind nach 30 Iterationen konvergiert und weisen ein im Gegensatz zur PSO verbessertes Optimierungsergebnis auf ( $F(f, A, v) = 24, 2$ ). Aufgrund der deutlich kleineren Anzahl benötigter Simulationen wird allerdings in diesem Fall eine Optimierung mit 1000 Individuen bevorzugt. Die späte Konvergenz der Durchläufe und das vergleichsweise schlechte Optimierungsergebnis mit 350 und 500 Samples pro Generation deuten auf eine zu kleine Populationsgröße hin.

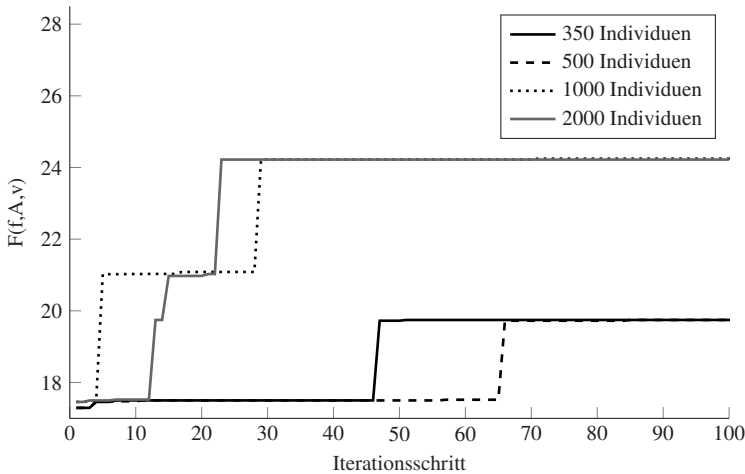


Abbildung 4.8: Optimierungsergebnisse für den genetischen Algorithmus

## Diskussion der Ergebnisse

Die qualitative Ähnlichkeit der Kurvenverläufe beider Algorithmen bestätigen die in der Literatur bereits festgestellten geringen Unterschiede beider Verfahren. Es wird deutlich, dass das beste Optimierungsergebnis in beiden Fällen mit der höchsten Populationsgröße erreicht wird. Mit kleineren Individuenzahlen sind damit meistens nur lokal optimale Lösungen möglich. Das Konvergenzverhalten scheint jedoch nicht mit der Populationsgröße zu korrelieren. Bei der PSO wird eine Konvergenz mit 500 Individuen schneller erreicht als mit 1000 Samples, obwohl die Abdeckung des Suchraums im zweiten Fall deutlich höher ist. Analog dazu konvergiert der Kurvenverlauf mit 350 Individuen schneller als der mit 500 Samples beim genetischen Algorithmus. Es liegt somit die Vermutung nahe, dass ein gutes Ergebnis bei kleineren Populationsgrößen zwar möglich ist, der Optimierungsverlauf aber stochastischen Streuungen unterliegt. Diese werden durch die Suchheuristiken der Algorithmen hervorgerufen. Mit einer größeren Anzahl an Samples pro Iteration kann der Suchraum dichter abgetastet werden und die Wahrscheinlichkeit gute Lösungen reproduzierbar zu erreichen steigt. Die Ergebnisse bestätigen die in [FFFG20] gewonnenen Erkenntnisse. Abbildung 4.9 zeigt jeweils drei Optimierungsdurchgänge für

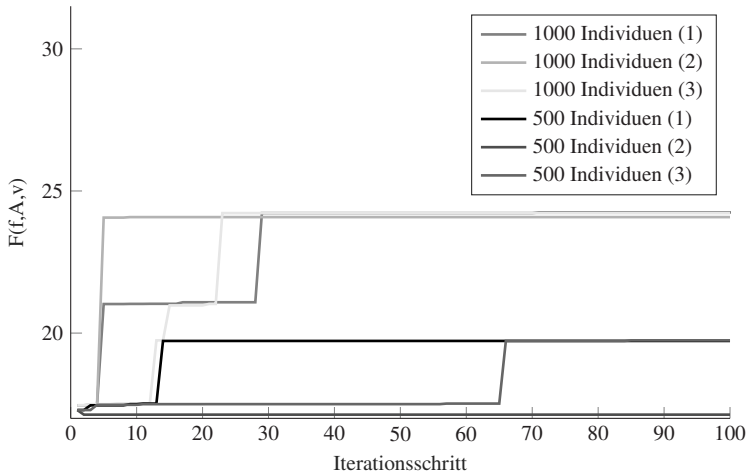


Abbildung 4.9: Vergleich der Reproduzierbarkeit von Optimierungsverläufen des GA bei gleicher Populationsgröße

500 und 1000 Individuen des genetischen Algorithmus. Die Verläufe bestätigen, dass eine Reproduzierbarkeit des Optimierungsergebnisses nur mit einer ausreichenden Anzahl an Samples pro Generation möglich ist. Weil im Rahmen dieser Arbeit davon ausgegangen wird, dass Simulationen auf einem Rechencluster gerechnet werden können, welches eine Parallelisierbarkeit von mindestens 1000 Simulationen ermöglicht, kann der Unterschied benötigter Systemevaluationen vernachlässigt werden. Die Auswertung einer Generation beansprucht damit für 1000 und 500 Individuen die gleiche Rechenzeit.

Die Tatsache, dass mehr Samples pro Iteration bessere Ergebnisse liefern und die Kurven teilweise sehr steile Gradienten aufweisen, lässt darauf schließen, dass gute Lösungen offensichtlich an diskreten Punkten im Suchraum liegen. Der stufenförmige Charakter des Optimierungsfortschritts deutet darauf hin, dass der Algorithmus für längere Zeit die direkte Umgebung guter Lösungen erforscht (Tiefensuche), ohne ein besseres Ergebnis zu finden. Erst nach einiger Zeit wird ein Parameterset erreicht, das den Zielfunktionswert deutlich erhöht. Diese Charakteristik lässt erahnen, dass sich die neue Lösung an einem steilen Peak befindet, in dessen direkter Umgebung Samples mit schlechteren Lösungen liegen. Das Finden solcher Punkte im Suchraum wird stark durch

die *exploration* (Breitensuche) begünstigt. Da die Gradienten der Zielfunktion für sämtliche Verläufe entweder null oder sehr groß sind, ist davon auszugehen, dass gute und schlechte Lösungen im gesamten Parameterraum sehr nah beieinander liegen und die Breitensuche folglich einen hohen Stellenwert einnimmt. Die Kurvenverläufe mit 1000 Individuen in Abbildung 4.9 untermauern die Beobachtungen. Während sich das Ergebnis für die Verläufe 1 und 3 schrittweise von anfangs 17.5 auf 21 und zuletzt zur finalen Lösung verbessern, zeigt Verlauf 2 eine Verbesserung von 17.5 auf 24.2 innerhalb einer Iteration. Eine stärkere Gewichtung der Breitensuche beider Algorithmen könnte auf Grundlage dieses Ergebnisses die Lösungssuche sogar noch beschleunigen. In jedem Fall wird diese aber durch eine groß gewählte Populationsgröße begünstigt. Eine weitergehende Analyse der Ursachen für diese Eigenschaften des Suchraums erfolgt in Abschnitt 5.3.

Für die weitere Verwendung eines Optimierungsverfahrens kommen sowohl der GA als auch die PSO in Frage. Auch wenn der genetische Algorithmus Lösungen mit höherem Funktionswert liefert, ist zu erwarten, dass auch die Partikelschwarmoptimierung bei entsprechender Parametrierung der Breitensuche dieses Ergebnis erzielen könnte. Beim Vergleich der Rechenzeit ist allerdings zu berücksichtigen, dass für die PSO normalerweise immer alle Individuen jeder Generation ausgewertet werden müssen. Der GA hingegen führt die genetischen Operationen lediglich mit einer selektierten Submenge durch, sodass der dafür nicht ausgewählte Teil unverändert in die nächste Iteration übernommen wird. Der Zielfunktionswert ändert sich folglich unter Voraussetzung deterministischer Simulationsumgebungen ebenfalls nicht. Aufgrund dieses Rechenzeitvorteils wird der genetische Algorithmus für nachfolgende Applikationen verwendet.

### **4.3.2 Nutzung der Komplexitätsreduktion für eine effizientere Optimierung**

Mit dem Ziel eines minimalen benötigten Rechenaufwands zur simulativen Applikation geht die Forderung einer Reduktion von Szenariensimulationen einher. Wie bereits in Abschnitt 3.3.2 erwähnt, bietet die Einbeziehung von Ergebnissen der Komplexitätsreduktion (vgl. Abschnitt 4.2) dafür ein großes Potential. Das in diesem Abschnitt beschriebene Verfahren wurde vom Autor bereits in den Veröffentlichungen [FFFG20] und [Fra21] vorgestellt und wird



im Rahmen dieser Arbeit nochmals aufgegriffen. Wie zuvor beschrieben, wird der GA-Algorithmus weiter verwendet. Eine Übertragbarkeit des Verfahrens auf alternative Optimierer ist aber problemlos möglich.

Eine Anwendung des in Abschnitt 4.2 beschriebenen Vorgehens liefert Informationen über einflussreiche Wertebereiche und Parameter (Dimensionen) des Suchraums in Abhängigkeit eines Szenarios. Die Schnittmenge einflussreicher Bereiche aller Szenarien kann bereits einen reduzierten Suchraum für die Optimierung liefern. Es ist jedoch nicht davon auszugehen, dass es Überschneidungen zwischen allen szenarienspezifischen einflussreichen Regionen gibt, sondern sich die Sensitivitäten in Abhängigkeit der Szenariencharakteristik ändern. Darüber hinaus würden bei einer Bildung der kleinsten gemeinsamen Schnittmenge Potentiale szenarienabhängiger Einflussinformationen ungenutzt bleiben. Aus dem Grund wird eine Prüfung der Notwendigkeit einer Simulation jedes Parametersatzes in den Optimierungsablauf integriert. Wie in Abbildung 2.8 verdeutlicht, erfordert die Auswertung eines vom Optimierer bereitgestellten Parametersatzes die Simulation des gesamten Szenarienkatalogs. Wenn jedoch ein Parameterset generiert wird, welches sich im Vergleich zu einem Referenzset nur innerhalb für das jeweilige Szenario nicht relevanter Bereiche verändert, kann eine Simulation entfallen. In diesem Fall ist ein vernachlässigbarer Fehler zu erwarten, da der Einfluss von Parameteränderungen sowie die berechneten Sensitivitäten innerhalb dieser Regionen unterhalb einer Relevanzschwelle liegen (vgl. Abbildung 4.6). Parameter, welche sich im einflussreichen Bereich befinden, müssen allerdings in beiden Parametersätzen den gleichen Wert annehmen, da kleine Änderungen bereits einen großen Einfluss haben könnten. Der Funktionswert zu dem neuen Individuum kann folglich von der Referenz kopiert werden, ohne dass dabei ein relevanter Fehler entsteht.

Abbildung 4.10 zeigt Beispiele für Parametersätze, welche sich für eine Ersetzung von Szenariensimulationen eignen oder dafür nicht in Frage kommen. Der graue Bereich stellt den einflussreichen Parameterraum dar. Im linken Diagramm ist der vom Optimierer neu generierte Parametersatz mit der durchgezogenen Linie gekennzeichnet. Die mit der gestrichelten und gepunkteten Linie dargestellten Referenzen unterscheiden sich davon nur innerhalb des weißen als nicht einflussreich klassifizierten Bereichs während sie im grauen Bereich dem Ausgangsdatensatz entsprechen. Die Funktionswerte beider Referenzen können somit das Simulationsergebnis ersetzen und wertvolle Rechen-

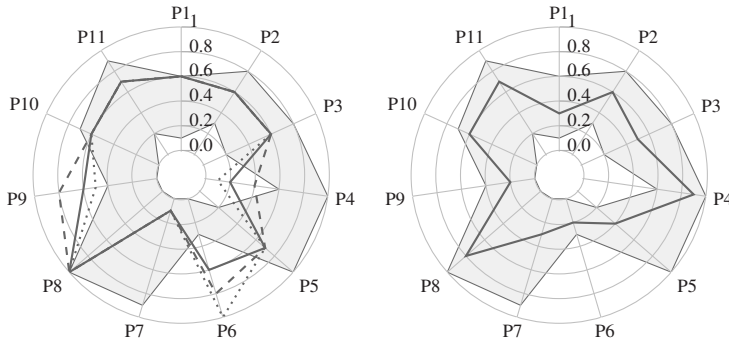


Abbildung 4.10: Graue Fläche: Einflussreicher Bereich des Parameterraums. Links: Beispiel, in dem eine Einsparung der Simulation des Parametersets (durchgezogene Linie) durch Ersetzung des Funktionswerts mit zuvor ausgewerteten Parametersätzen (gestrichelte und gepunktete Linie) möglich ist. Rechts: Beispiel für einen Parametersatz, welcher simuliert werden muss da er sich vollständig im einflussreichen Bereich befindet [FFFG20].

zeit einsparen. Der Parametersatz im rechten Plot liegt hingegen vollständig im einflussreichen Wertebereich und muss damit simuliert werden. Um eine effiziente virtuelle Optimierung zu ermöglichen sollte eine Datenbank gepflegt werden, welche die Ergebnisse aller simulierten Individuen und Parameterwerte enthält.

In Abbildung 4.11 ist das Optimierungsverfahren in einem Ablaufdiagramm zusammengetragen. Es kann als Vorgehensweise zur effizienten Auswertung einer vom Optimierer erzeugten Population unter Nutzung von Sensitivitätsinformationen verstanden werden. Die Variablen  $P_{sens}$  und  $PB_{sens}$  bezeichnen in der Abbildung Vektoren, welche Auskunft über die Relevanz eines Parameters beziehungsweise dessen sensitive Grenzen geben. Die Informationen liegen als Ergebnis der Komplexitätsreduktion vor und sind in Abhängigkeit des Szenarios (erste Dimension) sowie des Parameters (zweite Dimension) gespeichert. Die dritte Dimension des Vektors  $PB_{sens}$  beschreibt sensitive Unter- und Obergrenzen während  $P_{sens}$  entweder 1 oder 0 ist und damit die Notwendigkeit zur Berücksichtigung der jeweiligen Dimension angibt. Ausgehend von der vom Optimierer erzeugten zu simulierenden Population  $PS$  werden für jedes Szenario  $s$  (1) und jeden Parametersatz  $ps$  (2) im ersten Schritt die nicht-sensitiven Parameter identifiziert (3). Die Variable *Non\_influential\_parameters* stellt

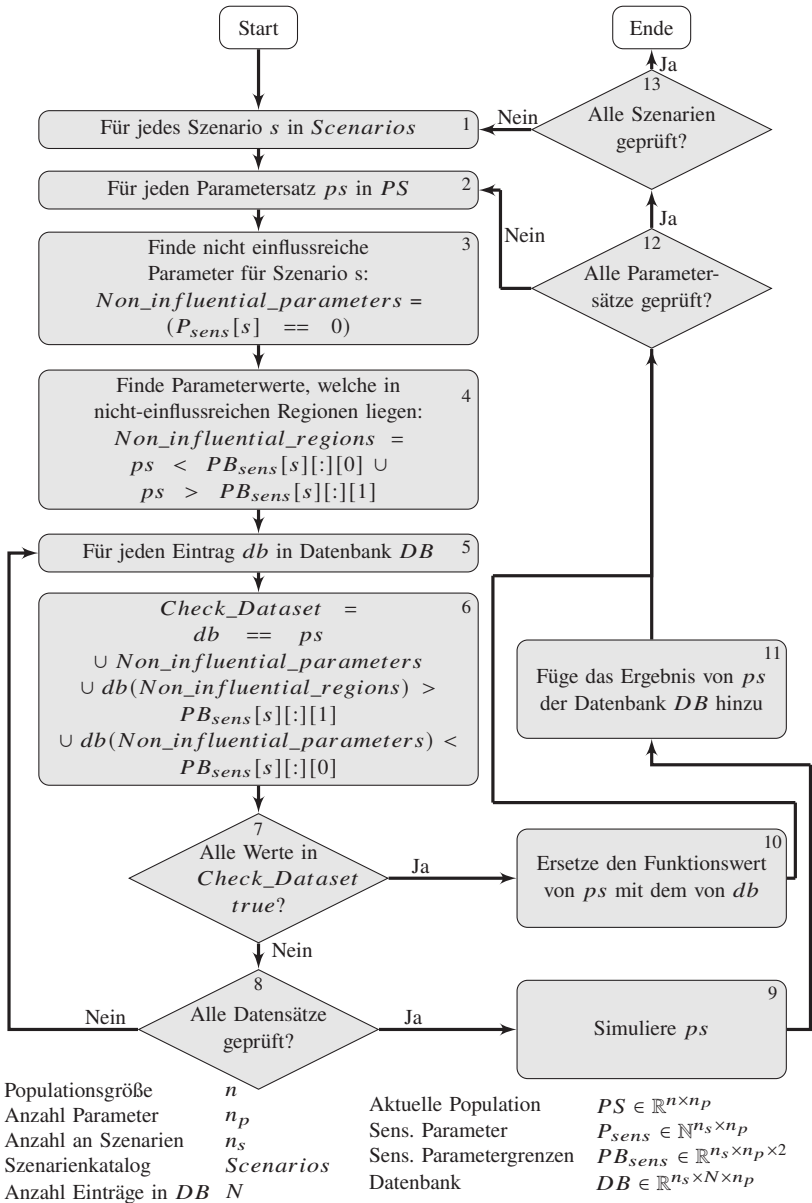


Abbildung 4.11: Ablaufdiagramm für die effiziente Evaluierung einer Population von Parametersets unter Berücksichtigung szenarienspezifischer Einflussinformationen [FFFG20, Fra21].

einen  $n_p$ -dimensionalen booleschen Vektor dar, welcher für Parameter ohne relevanten Einfluss den Wert *true* hat.  $n_p$  gibt in diesem Fall die Anzahl von Parametern an. Im Folgenden wird geprüft, welche Parameterwerte des aktuellen Sets  $ps$  in nicht-sensitiven Regionen liegen (4). Analog zu dem Vektor aus (3) enthält auch *Non\_influential\_regions* Einträge mit dem Wert *true* für Parameter, dessen Werte in nicht-einflussreichen Regionen liegen. Basierend darauf wird im Folgenden die Datenbank nach gültigen Referenzsätzen durchsucht (5). Eine valide Referenz  $db$  ist dann gefunden, wenn die Parameter entweder den gleichen Wert wie der aktuelle Parametersatz  $ps$  haben ( $db == ps$ ) oder als nicht-einflussreich klassifiziert sind (*Non\_influential\_parameters*) oder sich innerhalb der gleichen nicht-sensitiven Regionen wie der aktuelle Parametersatz befinden ( $db(\text{Non\_influential\_regions}) > PB_{sens}[s][:][1] \cup db(\text{Non\_influential\_parameters}) < PB_{sens}[s][:][0]$ ). Führt die Vereinigungsmenge aller Bedingungen in (6) zu einem *Check\_Dataset*-Vektor, dessen Einträge alle den Wert *true* haben, ist eine gültige Referenz gefunden (7) und der Funktionswert zu  $ps$  kann der Datenbank entnommen werden (10). Wird allerdings kein passender Vergleichs-Datensatz gefunden (8) muss der Funktionswert zu  $ps$  simulativ bestimmt werden (9). Das Ergebnis wird im Anschluss der Datenbank hinzugefügt und erhöht damit die Chance, Simulationen zukünftig auftretender Parametersätze ersetzen zu können (11). Die Datenbank erhält mit jeder Iteration neue Parametersets, sodass die Chancen zur Einsparung von Simulationen steigen je weiter der Optimierungsalgorithmus fortgeschritten ist.

Wie eingangs erwähnt, ist die Anwendung des Verfahrens mit jedem beliebigen Optimierungsalgorithmus möglich. Das größte Potential zur Einsparung von Rechenzeit besteht allerdings für Optimierer, welche eine diskrete Skala verwenden. Durch eine gleichmäßige Teilung des Wertebereichs wie es beispielsweise für die bitweise Darstellung des GAs gefordert ist, sind nur eine endliche (wenn auch sehr große) Anzahl an Punkten im Parameterraum erreichbar. Die Wahrscheinlichkeit, Referenzen zu finden, die von neu auszuwertenden Sätzen lediglich entlang weniger Dimensionen abweichen, ist somit größer als für Optimierer mit kontinuierlichen Skalen, die theoretisch jeden Punkt im Suchraum erreichen können. Aufgrund der geringen Wahrscheinlichkeit in diesem Fall eine exakte Übereinstimmung der Parameterwerte von zwei Individuen im einflussreichen Bereich zu finden, kann eine Abschätzung der maximal hinnehmbaren Abweichung durchgeführt werden, welche noch zu einem vernachlässigbarem Fehler führt.

# 5 Anwendung der Methodik für die Applikation einer automatisierten Fahrfunktion

Die im letzten Kapitel vorgestellte Methodik zur Komplexitätsreduktion und effizienten simulativen Optimierung von Applikationsparametern soll im Folgenden unter Nutzung des in Abschnitt 4.1 eingeführten hybriden Fahrzeugmodells für ein repräsentatives Applikationsproblem Anwendung finden. Das zu Grunde liegende Optimierungsproblem sowie die Simulationsumgebung werden im ersten Abschnitt beschrieben, bevor in Abschnitt 5.2 eine Vorstellung der Ergebnisse erfolgt. Die gewonnenen Erkenntnisse werden abschließend diskutiert und Schlussfolgerungen abgeleitet.

## 5.1 Applikationsproblem

Das in diesem Abschnitt beschriebene Applikationsproblem entspricht in weiten Teilen dem aus Abbildung 2.8, wird aber im Folgenden näher spezifiziert. Das in diesem Abschnitt beschriebene Applikationsproblem wurde erstmals in der Vorveröffentlichung [FFFG20] vorgestellt. Wie bereits erwähnt, ist das Ziel der Applikation die Optimierung des Fahrverhaltens, das mit Hilfe von Gütefunktionen in allen notwendigen Szenarien bewertbar ist. Im Gegensatz zu dem vereinfachten Problem, welches in Abschnitt 4.3 verwendet wurde, soll nun die Parametrierung einer Level-3-Fahrfunktion (nach SAE-Standard, vgl. [C<sup>+</sup>14]) erfolgen. Als Stellhebel werden elf freie Parameter der Trajektorienplanung verwendet. Diese definieren in erster Linie Gewichtungsfaktoren sowie Terme einer Kostenfunktion als auch maximal erlaubte laterale und longitudinale Dynamikgrenzwerte. Die verwendeten Stellhebel erlauben damit eine Beeinflussung der Trajektoriencharakteristik, die sich auf das Fahrverhalten im Closed-Loop-System auswirkt. Da das Level-3-System die Übernahme

von längs- und querführenden Fahraufgaben in Highway-Szenarien ermöglicht, wird ein Szenarienkatalog definiert, welcher einen minimalen Satz aller relevanten Use-Cases abbildet. Die Szenarien definieren sich wie folgt:

- S1: Spurwechsel auf gerader Straße (Überholen eines langsam fahrenden Fahrzeugs auf der rechten Fahrspur)
- S2: Spurwechsel auf gekrümmter Straße (Überholen eines langsam fahrenden Fahrzeugs auf der rechten Fahrspur)
- S3: Spurhalten auf gerader Straße
- S4: Spurhalten auf gekrümmter Straße
- S5: Geschwindigkeitsanpassung an ein Verkehrsschild (Beschleunigung)
- S6: Geschwindigkeitsanpassung an eine Kurve (Verzögerung)

Die definierten Szenarien stellen damit die drei wichtigsten Grundfunktionalitäten für das untersuchte System dar: Lane-Change, Lane-Keeping und Adapt-Speed. Für die Bewertung des Fahrverhaltens bedarf es einer Differenzierung nach den drei Manöverklassen, da das vom Kunden wahrgenommene Fahrgefühl von der Charakteristik des angeforderten Szenarios abhängt. In Abschnitt 2.6 wurden bekannte Ansätze zur Messung des Fahrkomforts anhand von Objektivierungsmodellen zusammengetragen. Entgegen des konventionellen Vorgehens der Bestimmung von Regressionsfunktionen basierend auf Probandenstudien hat das ebenfalls vorgestellte *Dynamic-Time-Warping*-Verfahren den Vorteil, dass es zeitreihenbasiert arbeitet und für Spurhalte-, Spurwechsel- und längsführende Manöver bessere Ergebnisse liefert. Darüber hinaus ist es ohne aufwändige Modifikationen anwendbar (vgl. Abschnitt 2.6). Die in den simulierten Szenarien gewonnenen Zeitreihen werden dem Objektivierungs-Algorithmus zugeführt, welcher über ein Abstandsmaß ein bereits von Probanden bewertetes Szenario mit möglichst hoher Ähnlichkeit identifiziert. Über die Höhe des Unterschieds zur Referenz wird mit Hilfe des *Dynamic-Time-Warping*-Verfahrens ein Komfortwert entsprechend der in Tabelle 2.1 definierten Skala abgeleitet. Aufgrund der statistisch nachgewiesenen Aussagekraft des Verfahrens und der unproblematischen Anwendbarkeit für die definierten Szenarien kommt der *Dynamic-Time-Warping*-Ansatz im Rahmen dieser Arbeit als Gütefunktion zum Einsatz.

Die Evaluierung eines Parametersatzes erfordert die Simulation aller sechs Szenarien und individuelle Auswertung der Komfortwerte. Um eine a-posteriori-Auswahl der besten Lösung durch den Applikateur entlang einer Pareto-Front zu vermeiden, wird das multikriterielle Optimierungsproblem als einkriterielles Problem definiert (vgl. Abschnitt 2.4.1). Somit wird vermieden, dass schlechte Lösungen einzelner Szenarien zu Gunsten eines guten Gesamtergebnisses hingenommen werden. Stattdessen erfolgt eine a-priori-Gewichtung mit Hilfe der *Methode der gewichteten Summe*. Die Funktionswerte aller Szenarien werden dafür gleich gewichtet um ein möglichst ausgeglichenes Fahrverhalten in allen relevanten Situationen zu erzielen. Bei komplexer werdendem Szenarienkatalog oder wenn verschiedene Ausprägungen realisiert werden sollen, kann die automatisierte Applikation mit Hilfe unterschiedlicher Gewichtungen gesteuert werden. Dies ist insbesondere dann von Vorteil, wenn beispielsweise länderspezifische oder Derivate-abhängige Applikationen realisiert werden sollen oder ein bestimmtes Fahrverhalten (z.B. sportliche/ökonomische Ausprägung) gewünscht ist. Im Rahmen dieser Arbeit ergibt sich ausgehend von Gleichung 2.16 folgende zusammengesetzte Zielfunktion  $f(x)$ :

$$f(x) = \sum_{i=1}^m w_i f_i(x) \quad \text{mit} \quad w_i = \frac{1}{6}, i = 1, \dots, 6 \quad (5.1)$$

Die szenarienspezifischen Gütefunktionen sind mit  $f_i(x)$  bezeichnet. Als Simulationsumgebung steht eine von der BMW AG bereitgestellte Software zur Verfügung, welche im Wesentlichen die in Abbildung 2.3 dargestellte Wirkkette abbildet. Zur Umfelderfassung kommen vom Zulieferer entwickelte Black-Box-Verhaltensmodelle der Sensorik (Frontradar, Seitenradar, Kamera, Laserscanner) zum Einsatz. Die sich entlang der Wirkkette anschließenden Softwaremodule zur Umfeldverarbeitung und Handlungsplanung entsprechen den realen Komponenten, welche auch im Zielsystem verwendet werden. Die Fahrzeugdynamik wird mit Hilfe des in Abschnitt 4.1 vorgestellten hybriden Modells abgebildet. Entgegen klassischer Ansätze wie zum Beispiel dem Einspurmodell wird somit sowohl die Aktuatorik als auch die Eigendynamik des Fahrzeugs abgebildet, indem das ISAR-Modell zur Erzeugung von Trainingsdaten verwendet wird und die von der Trajektorienregelung angeforderte Stellgröße als HM-Modelleingang dient (vgl. Abschnitt 4.1). Die Szenariengenerierung ist in der eingesetzten Simulationsumgebung modular aufgebaut. Es kann auf vorgegebene Fahrtstrecken wie beispielsweise Autobahnabschnitt-

te, Versuchsgelände oder fiktive Streckenverläufe zurückgegriffen werden. In dieser Umgebung können die Bewegungen anderer Verkehrsteilnehmer (z.B. Fahrzeuge, Fußgänger) sowie das vom Fahrer angeforderte Soll-Verhalten des Ego-Fahrzeugs (z.B. durch Aktivieren einer Fahrfunktion, Einstellen einer Wunschgeschwindigkeit) vorgegeben werden. Die Ausführung der Simulationen erfolgt auf vier parallel rechnenden Simulationscomputern (CPU 2,6 GHz, RAM 8 GB), mit Hilfe derer eine Rechenzeit von ca. 10 Sekunden pro Szenario erreicht werden kann.

## 5.2 Ergebnisse

Die Vorstellung der Ergebnisse gliedert sich in vier Teilabschnitte. Die in 5.2.1 beschriebene Reduktion des Parameterraums wird für die sensitivitätsbasierte Optimierung verwendet, deren Umsetzung in Abschnitt 5.2.2 vorgestellt wird. Wie in Kapitel 3 beschrieben, werden die gefundenen Lösungen im Anschluss noch einer Robustheitsanalyse unterzogen, welche eine weitere Klassifizierung im Hinblick auf die Weiterverwendung zur Feinapplikation im Fahrversuch ermöglicht (Abschnitt 5.2.3). Im letzten Teil erfolgt eine Analyse der eingesparten Simulationszeit, die durch Nutzung von Einflussinformationen während der Optimierung erreicht wird (Abschnitt 5.2.4). Die im Folgenden beschriebenen Ergebnisse wurden vom Autor bereits in [FFFG20] veröffentlicht. Die Erkenntnisse werden aber im Rahmen dieser Arbeit detaillierter analysiert und diskutiert.

### 5.2.1 Komplexitätsreduktion

Bevor die im vorherigen Kapitel vorgestellte Methodik zur Komplexitätsreduktion Anwendung finden kann, müssen initiale Parametergrenzen für den 11-dimensionalen Suchraum vorgegeben werden. Weil davon ausgegangen wird, dass diese entweder nicht vorliegen oder vom Softwareentwickler geschätzt werden, erfolgt eine konservative Definition von Grenzwerten. Aus Gründen der Darstellbarkeit werden die Ober- und Untergrenzen im Folgenden relativ



angegeben. Die initialen Wertebereiche für alle Parameter sind somit durch 0 und 1 begrenzt. Entsprechend der Notation aus Abschnitt 4.2 gilt somit:

$$\begin{aligned}UG_i &= 0 \quad \forall \quad i \in [1, \dots, 11] \\OG_i &= 1 \quad \forall \quad i \in [1, \dots, 11]\end{aligned}\tag{5.2}$$

Die tatsächlichen Werte können Tabelle A.5 entnommen werden. Wie in Abschnitt 4.2 beschrieben, erfolgt im ersten Schritt die Bestimmung gültiger Wertebereiche, innerhalb dessen Parameteränderungen einen relevanten Einfluss auf die Zielfunktion haben. Basierend darauf können im Anschluss bereits einige Parameter für weitere Analysen ausgeschlossen werden. Dimensionen, für die lediglich eine Reduktion des einflussreichen Bereichs erfolgen konnte, werden jedoch im Anschluss mit einem neuen Sampling hinsichtlich ihrer globalen Sensitivität untersucht.

Die Ableitung gültiger Wertebereiche erfordert zunächst eine Unterteilung des Wertebereichs von jedem Parameter in gleich große Intervalle. Je nach gewünschtem Detaillierungsgrad kann die Anzahl an Intervallen  $n_{Int}$  dafür beliebig groß gewählt werden. Für den untersuchten Anwendungsfall ist eine Diskretisierung in zehn Teilabschnitte ausreichend, sodass gilt:  $n_{Int} = 10$ . Darüber hinaus muss festgelegt werden, wie viele Samples im Versuchsplan für jedes Intervall enthalten sein sollen. Dafür ist die Anzahl an Elementareffekten  $r$  ausschlaggebend (vgl. Gleichung 2.1). Weil in vorausgegangenen Studien für vergleichbare Probleme eine Konvergenz der Ergebnisse mit dem Bootstrapping-Ansatz bereits mit einem relativ kleinen Versuchsplan nachgewiesen werden konnte ( $r = 15$ , vgl. [FFFG19a]), wird in dieser Arbeit die Variable  $r$  zu 50 festgelegt. Das entspricht  $n_{EM}^* = 600$  Samples, die pro Intervall, Parameter und Szenario ausgewertet werden müssen. Für die Klassifizierung evaluierter Intervalle im Hinblick auf deren Relevanz erfolgt ein Vergleich der relativen Sensitivität  $s_{i,k_{ges}}^{rel}$  mit der minimalen Relevanzschwelle  $s_{min,WB}^{rel}$  (vgl. Abbildung 4.6). Diese wird aufgrund des qualitativen Charakters der Sensitivitätsanalyse konservativ zu  $s_{min,WB}^{rel} = 0.01 = 1\%$  abgeschätzt. Im Detail bedeutet das, dass der Einfluss eines Parameters in der betrachteten Region des Suchraums als nicht relevant bewertet wird, wenn der zugehörige Sensitivitätswert kleiner als 1% des Werts des einflussreichsten Parameters ist. Die basierend auf der Anwendung des Verfahrens bestimmten

einflussreichen Ober- und Untergrenzen können Abbildung 5.1 entnommen werden. Die Werte für *PRR* (vgl. Abschnitt 4.2.4) erlauben einen quantitativen Vergleich der Parameterraum-Reduktionen zwischen den Szenarien. Für Parameter-Szenarien-Kombinationen, welche mit *o.S.* gekennzeichnet sind, kann kein relevanter Einfluss entlang des gesamten Wertebereichs nachgewiesen werden.

	S1	S2	S3	S4	S5	S6
<i>P</i> <sub>1</sub>	0-1	0-1	0,1-1	0-0,7	0-0,8	0-0,8
<i>P</i> <sub>2</sub>	0-1	0,2-0,9	0-1	0,1-1	0,1-0,7	0-1
<i>P</i> <sub>3</sub>	0-1	0,3-1	0-1	0,1-1	o.S.	0,1 -0,6
<i>P</i> <sub>4</sub>	0-1	0-1	0,2-1	0-1	o.S.	0,5-1
<i>P</i> <sub>5</sub>	0-1	0-1	0,1-0,9	0-1	o.S.	0,1-0,4
<i>P</i> <sub>6</sub>	0,1-1	0-1	0-1	0-1	o.S.	o.S.
<i>P</i> <sub>7</sub>	0-0,9	0-1	0-1	0-1	0,3-0,9	o.S.
<i>P</i> <sub>8</sub>	0-1	0-1	0-1	0,1-1	0-0,9	o.S.
<i>P</i> <sub>9</sub>	0-1	0-1	0-1	0-1	0,4 -0,7	o.S.
<i>P</i> <sub>10</sub>	0-1	0,1-1	0-1	0,1-1	o.S.	o.S.
<i>P</i> <sub>11</sub>	0-0,9	0-1	0-1	0,2-0,8	o.S.	o.S.
<i>PRR</i>	0, 271	0, 559	0, 424	0, 724	0, 922	0, 94

Abbildung 5.1: Sensitive Wertebereiche jedes Parameters in Abhängigkeit der Szenarien. o.S. : ohne Sensitivität [FFFG20]

Die Ergebnisse zeigen, dass sich der sensitive Suchraum in Abhängigkeit des Fahrmanövers unterscheidet. Während für die Szenarien S1 - S4 nur geringe Einschränkungen möglich sind, offenbaren die neu definierten Grenzwerte für S5 - S6 eine deutliche Verkleinerung des relevanten Suchraums. Die Tatsache, dass weite Teile des Parameterraums als nicht sensitiv klassifiziert sind, könnte mit der Szenariencharakteristik zusammenhängen. Da die Szenarien S5 und S6 im Gegensatz zu den übrigen Manövern der Basis-Funktionalität *Adapt Speed* zugeordnet sind, bewertet das Objektivierungsmodell in diesen Fällen

vornehmlich den Fahrkomfort der Längsdynamik. Es liegt also die Vermutung nahe, dass ein Großteil der untersuchten Parameter in erster Linie der lateralen Trajektorienplanung dienen und Wechselwirkungen zu der longitudinalen Planung als gering beziehungsweise nicht existent (z.B. für  $P_6, P_{10}, P_{11}$ ) bewertet werden können.

Im als einflussreich klassifizierten Teil des Parameterraums wird im nächsten Schritt eine globale Sensitivitätsanalyse und potentielle Dimensionalitätsreduktion durchgeführt. Auch für diesen Teil der Auswertung ist eine Definition der Versuchsplangröße  $n_{EEM}$  und minimalen Relevanzschwelle  $s_{min}^{rel}$  notwendig (vgl. Abschnitt 4.2.3). Wie weiter oben beschrieben, kann für die Abschätzung der Anzahl an Samples auf vorausgegangene Studien zurückgegriffen werden. Dort wurde die Konvergenz ab einer Elementareffektgröße von  $r = 110$  nachgewiesen. Für eine Abschätzung zur sicheren Seite wird deshalb  $r$  zu 150 festgelegt. In Folge dessen ergeben sich insgesamt  $n_{EEM} = 1800$  zusätzliche Systemevaluations für die globale Sensitivitätsanalyse. Die minimale Relevanzschwelle wird genau wie für den ersten Teil der Auswertung zu 1% definiert:  $s_{min}^{rel} = 0.01$ . Die relativen Sensitivitätswerte sowie Kennwerte für die Dimensionalitätsreduktion  $DR$  sind in Abbildung 5.2 dargestellt. Auch wenn die bereits als nicht sensitiv klassifizierten Parameter für die Szenarien S5 und S6 nicht weiter betrachtet werden müssten, sind dessen Einflusswerte der Vollständigkeit halber mit ausgewertet. Die zugehörigen  $\mu^*$ - $\sigma$ -Diagramme befinden sich im Anhang in den Abbildungen A.1 und A.2. Zunächst kann wie auch für den ersten Teil der Auswertung festgestellt werden, dass sich die Sensitivitätsinformationen in Abhängigkeit der Szenariencharakteristik unterscheiden. Darüber hinaus wird herausgestellt, dass der Parameter  $P_1$  offensichtlich in allen Szenarien einen gegenüber den anderen Parametern dominierenden Einfluss hat. Die Sensitivitäten der übrigen Dimensionen scheinen sich jedoch nach dem Szenario zu richten. Auf Grundlage dieser Erkenntnisse kann nun für jedes Fahrmanöver eine Applikationsstrategie abgeleitet werden, welche sowohl die individuellen Verteilungen der Sensitivitäten pro Manöver (spaltenweise Betrachtung) als auch Wechselwirkungen im Szenarienkatalog (zeilenweise Betrachtung) berücksichtigen sollte. Wenn nachgewiesen werden kann, dass ein Parameter nur für wenige Szenarien einen großen Einfluss hat (wie z.B.  $P_2$  für S2 und S4), kann dieser exklusiv zur Optimierung der jeweiligen Situationen eingesetzt werden ohne eine gleichzeitige Verschlechterung übriger Use-Cases befürchten zu müssen. Auf der anderen Seite können auf Grundlage der veränderten Parameter im Vergleich zu einem Referenz-Datensatz Szenario

	S1	S2	S3	S4	S5	S6
$P_1$	1	0,649	1	1	1	1
$P_2$	0,08	1	0,141	0,609	0,056	0,141
$P_3$	0,08	0,111	0,167	0,015	0,004	0,167
$P_4$	0,37	0,134	0,14	0,088	0,003	0,14
$P_5$	0,069	0,169	0,108	0,023	0,008	0,108
$P_6$	0,287	0,113	0,081	0,056	0,002	0,008
$P_7$	0,293	0,424	0,141	0,034	0,445	0,001
$P_8$	0,144	0,089	0,362	0,315	0,016	0,004
$P_9$	0,049	0,13	0,159	0,037	0,134	0,002
$P_{10}$	0,358	0,34	0,241	0,022	0,002	0,002
$P_{11}$	0,119	0,219	0,035	0,017	0,002	0,004
DR	0	0	0	0	0,545	0,545

Abbildung 5.2: Relative Sensitivitäten jedes Parameters in Abhängigkeit der Szenarien [FFFG20]

rien identifiziert werden, die simulativ oder im Fahrzeug mit dem gefundenen Datensatz ausgewertet werden sollten um einer Verschlechterung des globalen Applikationsstands vorzubeugen. Der Rechenaufwand der in Abbildung 3.1 gezeigten Übertragung des Fahrversuchs in die Simulation lässt sich damit begrenzen.

Die Polardiagramme in den Abbildungen 5.3 und 5.4 fassen die in diesem Abschnitt gewonnenen Erkenntnisse nochmal zusammen und bestätigen die Annahme, welche zur Entwicklung der sensitivitätsbasierten Optimierung (Abschnitt 4.3.2) geführt haben: Der einflussreiche Bereich des Suchraums wird maßgeblich von der Szenariencharakteristik beeinflusst, sodass eine individuelle Betrachtung der Haupteffekte und Wechselwirkungen in Abhängigkeit der Fahrsituationen in nachfolgenden Applikationen empfohlen wird.

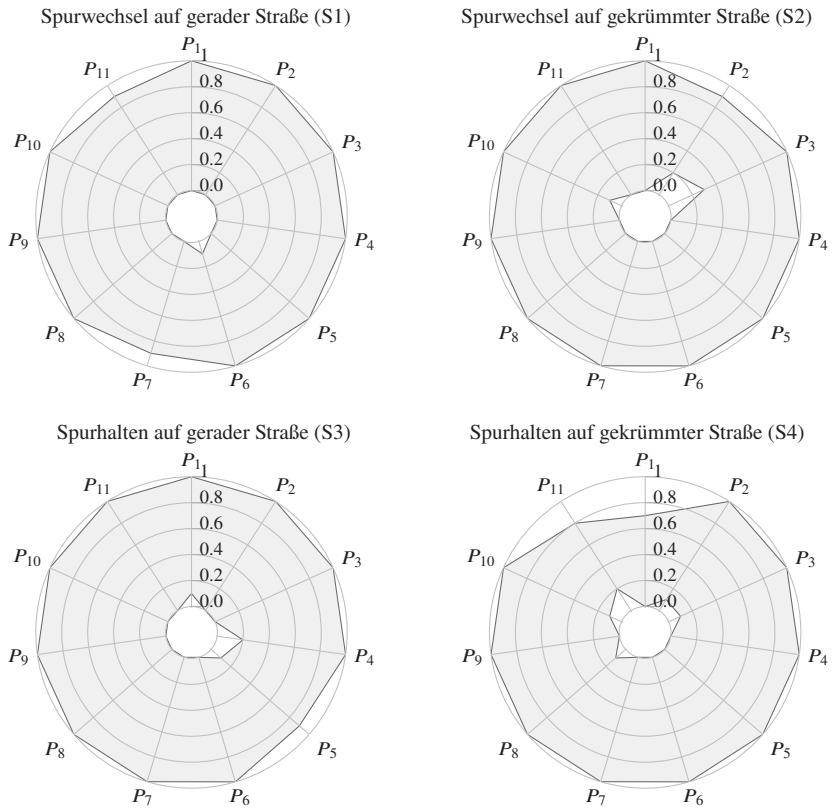


Abbildung 5.3: Einflussreiche Parameterräume (grau) der Szenarien S1 bis S4 [FFFG20]

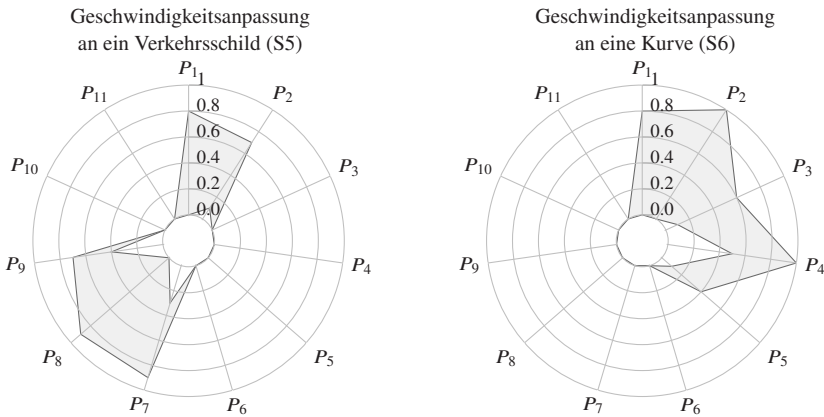


Abbildung 5.4: Einflussreiche Parameterräume (grau) der Szenarien S5 und S6 [FFFG20]

## 5.2.2 Optimierung

Basierend auf den Ergebnissen der Komplexitätsreduktion wird eine Optimierung der Applikationsparameter für das in Abschnitt 5.1 beschriebene und in Gleichung 2.16 definierte Optimierungsproblem durchgeführt. Wie in Abschnitt 4.3 beschrieben, wird dafür ein genetischer Algorithmus (GA) eingesetzt, der entsprechend der Optimierungsmethodik aus Abbildung 4.11 die Notwendigkeit von Szenarien-Simulationen in jeder Iteration prüft und damit eine effiziente Lösungsfindung ermöglicht.

Zur Anwendbarkeit des GA muss eine Darstellung von Parametersätzen als Binärstrang erfolgen, was eine Diskretisierung der Wertebereiche erfordert. In der Praxis wird dafür eine minimale Anzahl an geforderten Werten  $n_{P_i}^{min}$  für jeden Parameter definiert. Basierend darauf kann die Anzahl an Bits  $k_i$  mit folgender Bedingung abgeleitet werden:  $2^{k_i} > n_{P_i}^{min}$  (vgl. Abschnitt 2.4.2). Für das vorliegende Optimierungsproblem liegen Werte zur minimalen Diskretisierung jedes Wertebereichs vor, mit Hilfe dessen die genannten Variablen bestimmt werden können. Eine Übersicht findet sich im Anhang in Tabelle A.6. Mit Hilfe der genetischen Operatoren des Algorithmus kann eine Beeinflussung hinsichtlich der Breiten- und Tiefensuche erfolgen. Auf Grundlage

der guten Ergebnisse, welche mit der Parametrierung aus Abschnitt 4.3.1 an einem vergleichbaren Problem erzielt werden konnten, erfolgt die Einstellung von Selektion, Rekombination und Mutation mit Hilfe der gleichen Werte (vgl. Tabelle A.4). Des Weiteren wird eine Populationsgröße von 1000 gewählt, da diese einen guten Kompromiss zwischen Optimierungsergebnis und Rechenaufwand erwarten lässt (vgl. Abschnitt 4.3.1). Um stochastische Schwankungen ausschließen zu können, werden aber trotzdem Durchläufe mit 500 und 2000 Individuen durchgeführt. Deren Ergebnisse werden allerdings nicht weiter thematisiert, da auch in diesem Beispiel der Durchlauf mit 1000 Samples pro Generation die beste Performance liefert. Abbildung 5.5 gibt einen Überblick über den Verlauf der Zielfunktionswerte aller Individuen in Abhängigkeit der Iteration. Während die gepunktete Kurve das Ergebnis des jeweils besten Parametersets zeigt, ist das am schlechtesten bewertete Individuum durch die gestrichelte Linie dargestellt. Den Mittelwert über die Funktionswerte aller 1000 Parametersätze pro Iteration zeigt die durchgezogene Linie.

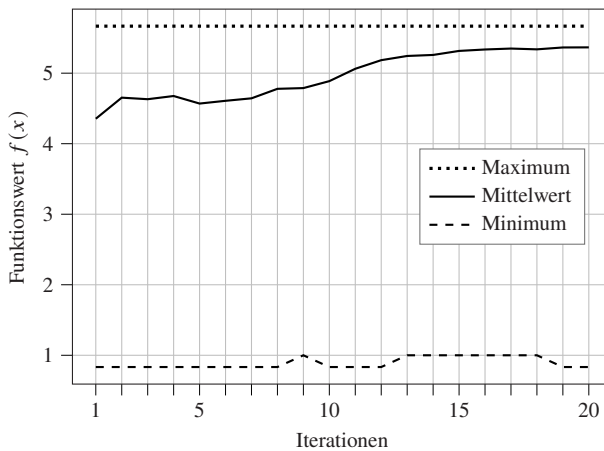


Abbildung 5.5: Minimaler, maximaler und gemittelter Funktionswert der Individuen-Population über die Iterationen [FFFG20]

Die Ergebnisse zeigen, dass bereits in der ersten Iteration ein Parametersatz gefunden wird, dessen Funktionswert im Laufe der Optimierung nicht mehr übertroffen wird. Die Entwicklung des Mittelwerts beweist allerdings, dass sich die gesamte Population über die Iterationen im Hinblick auf die Zielfunktion

zu verbessern scheint. Es gibt also über die Zeit mehr Parametersätze, welche dem bereits zu Anfang gefundenen Ergebnis näher kommen oder sogar diesem entsprechen.

Die Beobachtungen spiegeln damit nicht den typischen Verlauf einer Optimierung wieder. In den meisten Fällen wird die beste Lösung erst im Verlauf der Optimierung gefunden und erst wenn das Maximum der Population sich über mehrere Iterationen nicht mehr ändert, kann Konvergenz angenommen und der Algorithmus abgebrochen werden. Weil sich die Population aber insgesamt verbessert, liegt die Vermutung nahe, dass zwar keine bessere Lösung existiert, stattdessen aber alternative Parametrierungen gefunden werden, welche den gleichen Funktionswert liefern. In Abbildung 5.6 ist die Anzahl alternativer Parametersätze über die Iterationen aufgetragen, deren Funktionswerte dem der besten Lösung entsprechen ( $f(x) = 5,67$ ).

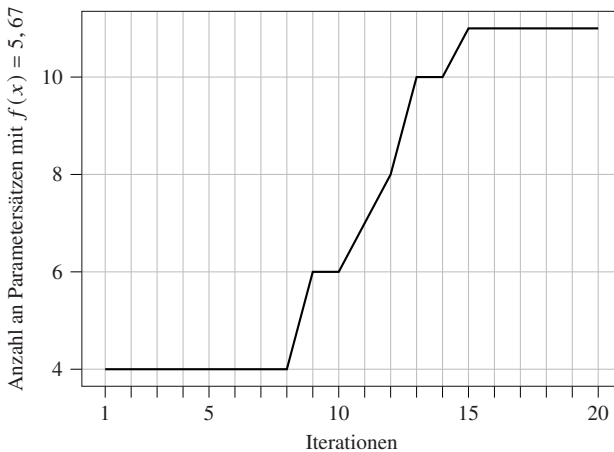


Abbildung 5.6: Anzahl gleichwertiger Parametersätze zu der besten Lösung über die Iterationen [FFFG20].

Der Kurvenverlauf offenbart eine kontinuierliche Erhöhung der Anzahl an Parametersätzen, die einen gleichwertigen Funktionswert zu der besten Lösung ermöglichen. Die Abbildung bestätigt somit, dass der GA das Finden alternativer Parametrierungen mit gleichem Funktionswert begünstigt. Da nach 15 Iterationen keine weiteren Individuen mit äquivalentem Ergebnis mehr ge-



funden werden, kann Konvergenz geschlussfolgert werden. Eine Diskussion möglicher Ursachen für den beobachteten Verlauf der Optimierung sowie die Bedeutung für den Applikationsprozess erfolgt in Abschnitt 5.3.

### 5.2.3 Robustheitsanalyse

Mit Hilfe der Anwendung des genetischen Algorithmus können elf Parametersätze gefunden werden, die im Hinblick auf die definierte Zielfunktion gleich bewertet werden. Ausgehend davon schließt sich entsprechend des Applikationsprozesses aus Abbildung 3.1 die manuelle Applikation im Fahrversuch an. Ist die Anzahl gleichwertiger, optimaler Datensätze deutlich größer als in diesem Anwendungsfall, müssten entweder einzelne Bedingungen für weitere Betrachtungen vernachlässigt oder alle Parametersätze im Realversuch erprobt werden. Da der damit verbundene Aufwand zu groß wäre, wird eine Robustheitsanalyse durchgeführt. Davon ausgehend können im Hinblick auf das Optimierungsergebnis gleichwertige Lösungen weiter klassifiziert werden. Für die Auswertung kommt der in Abschnitt 2.4.3 eingeführte *Degree-of-robustness*-Ansatz zum Einsatz.

Das Verfahren beruht auf einer schrittweisen Vergrößerung einer  $n_p$ -dimensionalen Hyperbox um die jeweilige Lösung so lange, bis die Robustheitsbedingung nicht mehr erreicht ist. Der minimale Hyperbox-Radius ist zu 0,01 definiert und leitet sich aus experimentellen Vorstudien ab:  $\delta_1 = 0,01$ . Eine Vergrößerung des Radius erfolgt in 10 Stufen:  $k \in [1, 2, \dots, 10]$ . Ist das Robustheitskriterium bei  $k = 10$  immer noch erfüllt, erfolgt keine weitere Erhöhung der Hyperbox-Größe. Ein Parametersatz ist dann robust in einer  $k\delta$ -Umgebung, wenn keiner der szenarienspezifischen Funktionswerte mehr als 1 von der jeweiligen Ausgangslösung abweicht. Die Robustheitsbedingung repräsentiert damit die minimal mögliche Abweichung, welche aufgrund der diskreten Komfortskala möglich ist (vgl. Tabelle 2.1).

Die Ergebnisse der Robustheitsanalyse sind in Abbildung 5.7 dargestellt. Das obere Diagramm zeigt die elf optimalen Parametersätze in Form paralleler Koordinaten (Ermöglicht die Darstellung höherdimensionale Parameter Räume im Zweidimensionalen). Im unteren Diagramm sind die den Parametrierungen zugehörigen Robustheitsgrade mit Hilfe von Linienstilen ablesbar. Die nor-

mierten Parameterwerte der dargestellten Parametrierungen und zugehörige Robustheiten sind in Tabelle A.7 zusammengetragen.

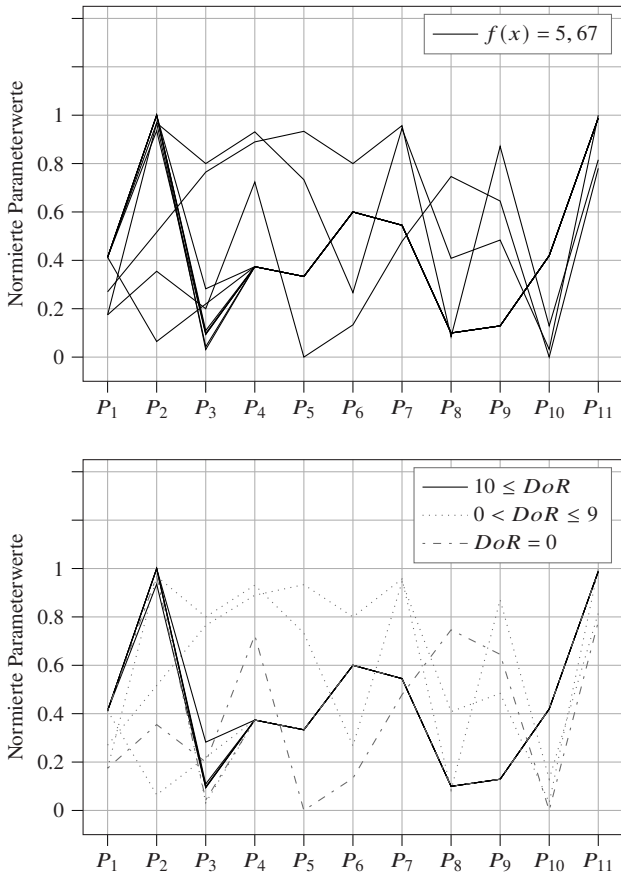


Abbildung 5.7: Oben: Parametersätze mit äquivalentem Funktionswert aus der Optimierung. Unten: Zugehörige Robustheitsinformationen. *DoR*: Degree of Robustness [FFFG20].

Beim Vergleich der Diagramme fällt auf, dass Individuen, die im Suchraum nah beieinander liegen, eine höhere Stabilität gegenüber parametrischen Ab-

weichungen haben als Parametersätze, welche sich an diskreten Positionen im Parameterraum befinden. Die im unteren Diagramm mit schwarzen Linien gekennzeichneten Parametersätze haben gleiche Werte für die Parameter  $P_5$  bis  $P_{11}$  und kleine Unterschiede für  $P_1$  bis  $P_4$ . Die übrigen sechs Individuen befinden sich allerdings an Punkten im Suchraum, in deren direkter Umgebung sich offensichtlich nur schlechter bewertete Lösungen befinden. Es kann folglich angenommen werden, dass sich die robusteren Lösungen an einem flacheren Peak im  $n_p$ -dimensionalen Raum befinden welcher von Samples umgeben ist, die einen vergleichbar hohen Funktionswert erlauben. Basierend auf diesen Erkenntnissen sollten die Parametersätze mit  $DoR \geq 10$  für weitere Applikationen im Fahrversuch verwendet werden. Die hohe Stabilität gegenüber kleinen parametrischen Änderungen lässt eine gesteigerte Robustheit auch bei sich ändernden Umgebungsbedingungen im Fahrversuch vermuten. Inwiefern diese Annahme gerechtfertigt ist, wird im Rahmen der Validierungsstudien in Kapitel 6 geprüft.

## 5.2.4 Rechenzeitbetrachtungen

Neben dem Optimierungsergebnis ist insbesondere die Rechenzeit sowie das Potential des eingesetzten Optimierungsverfahrens zur Reduktion von Szenariensimulationen relevant. Die kumulierte Anzahl der vom GA theoretisch geforderten Menge an Szenariensimulationen sowie die tatsächlich gerechnete Anzahl an Parametersätzen sind in Abbildung 5.8 dargestellt. Die Differenz aus beiden Kurven ergibt die eingesparte Menge an Simulationen und ist ebenfalls in Form der durchgezogenen Linie abgebildet. Es fällt auf, dass in der ersten Iteration 6000 Simulationen (1000 Individuen und 6 Szenarien) angefordert werden, diese Zahl aber in den folgenden Iterationen kleiner ist. Diese Beobachtung ist der Tatsache geschuldet, dass der GA die genetischen Operationen immer nur auf eine selektierte Submenge aus der Population anwendet und die verbliebenen Individuen unverändert in die neue Generation übernommen werden (sog. Elitismus, vgl. Abschnitt 2.4.2). Der benötigte Rechenaufwand ist in Folge dessen meistens kleiner als für vergleichbare Optimierer wie zum Beispiel die Partikelschwarmoptimierung (vgl. Abschnitt 4.3.1). Die Kurvenverläufe zeigen einen annähernd linearen Verlauf mit einer Gradientenänderung nach circa zehn Iterationen. Diese Charakteristik korreliert mit dem Verlauf der Kurven in den Abbildungen 5.5 und 5.6 und repräsentiert die Suchstrategie

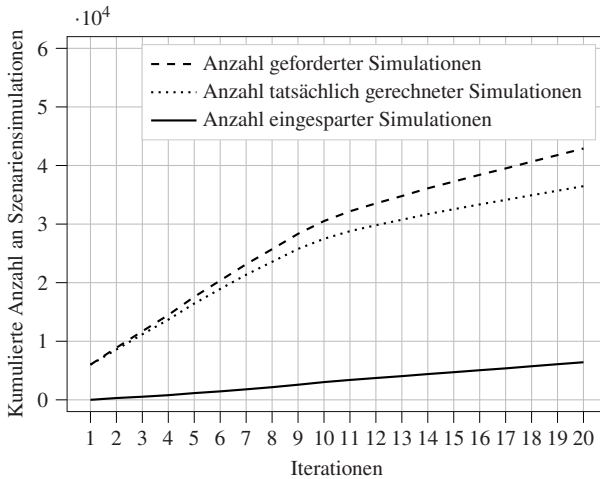


Abbildung 5.8: Kumulierte Anzahl angeforderter, tatsächlich gerechneter und eingesparter Szenariensimulationen [FFFG20].

des GA. Offensichtlich werden in den ersten acht Iterationen nicht viele gute Lösungen gefunden, sodass der Algorithmus den Parameterraum breit nach besseren Samples durchsucht (*exploration*). In der neunten Iteration können jedoch Individuen mit deutlich höherem Funktionswert gefunden werden, sodass der GA nun nach gleichwertigen oder besseren Lösungen in der direkten Umgebung dieser Punkte im Parameterraum sucht (*exploitation*). Im Anschluss daran werden in fast jedem Optimierungsschritt bis zur 15. Iteration neue gute Parametersätze gefunden. Der Algorithmus lässt als Konsequenz in dieser Zeit mehr Individuen unverändert in der Population und die Anzahl geforderter Simulationen pro Iterationsschritt sinkt.

Insgesamt ergeben sich 42000 geforderte Szenariensimulationen, was bei einer Simulationszeit von circa 10 Sekunden pro Szenario (vgl. Abschnitt 5.1) einer Rechenzeit von 116 Stunden entspricht. Durch die Anwendung der im Rahmen dieser Arbeit eingeführten Optimierungsmethodik können die Funktionswerte von 6416 Parametersätzen (ca. 17,82 Stunden) von gültigen Referenzsätzen übernommen werden, sodass dafür ein Simulationsdurchlauf vermieden wird. Die Nutzung von Sensitivitätsinformationen entsprechend des vorgestellten

Optimierungsverfahrens erlaubt somit eine Effizienzsteigerung von 15,36 % im Vergleich zur konventionellen Durchführung des GA.

## 5.3 Diskussion der Ergebnisse

Die Anwendung der Komplexitätsreduktion und anschließenden Optimierung auf ein repräsentatives Applikationsproblem verdeutlicht, dass eine Einschränkung des Suchraums und darauf aufbauende simulative Optimierung mit vertretbarem Rechenaufwand und gutem Optimierungsergebnis möglich ist. Neben initialen Parametersätzen können szenarienspezifische Einflussinformationen für eine zielgerichtete fahrzeugbasierte Applikation bereitgestellt werden.

Die Anwendung von Sensitivitätsanalysen am vorgestellten Beispiel bestätigt die Annahme, dass eine Identifikation relevanter Suchräume in Abhängigkeit des Fahrmanövers geschehen muss. Im Fall der Trajektorienplanung ist diese Beobachtung dadurch zu erklären, dass die Planung in Längs- und Querrichtung oft unabhängig voneinander mit zwei getrennten Optimierungsproblemen erfolgt. Auch für die Regelung kommen konzeptbedingt zumeist mindestens zwei Systeme zur Einhaltung der Sollvorgabe in Längsrichtung (z.B. durch die Beschleunigung) und Querrichtung (z.B. durch den Radlenkwinkel) zum Einsatz. Das ist auch dadurch begründet, dass die Bewegung in lateraler Richtung durch das Lenksystem und in longitudinaler Richtung durch das Vortriebssystem umgesetzt wird. In Folge dessen existieren viele Applikationsparameter, die sich fast ausschließlich auf eine der beiden Bewegungsrichtungen auswirken sollten. Weil die Fahrzeugdynamik vom Fahrer und damit vom Gütefunktional allerdings ganzheitlich wahrgenommen wird, existieren Wechselwirkungen, die sich in der Sensitivitätsanalyse widerspiegeln (vgl. Abbildung 5.2). Auch die Existenz von Softwaremodulen, welche die Längs- und Querplanung kombiniert vornehmen (wie z.B. die Fahrstrategie), kann dafür sorgen, dass eine unabhängige Einstellung von Parametern nicht vollständig möglich ist. Die Ergebnisse bekräftigen also das Potential der durchgeführten Einflussanalyse, die eine zielgerichtete Applikation in der Simulation und im Fahrversuch unter Einbeziehung aller relevanten Wechselwirkungen ermöglicht.

Darüber hinaus lässt sich, wie auch schon in vorherigen Studien gezeigt, ein reduzierter Einfluss einiger Parameter insbesondere im Bereich der Ober- und Untergrenzen des konservativ abgeschätzten Wertebereichs beobachten

[FFFG19a]. Zur Analyse dieser Charakteristik des Parameterraums soll näher auf die Trajektorienplanung eingegangen werden. In einer gängigen Implementierung wird eine Schar von Trajektorien generiert, die alle vorgegebene Anfangs- und Endbedingungen erfüllen. Je nach Anzahl der Bedingungen werden dafür Polynome unterschiedlichen Grades verwendet, welche die Bewegung in Längs- und Querrichtung für einen Prädiktionshorizont definieren. Ein erstes Ausschließen ungültiger Kandidaten kann durch eine Beschränkung des befahrbaren Korridors oder Dynamikgrenzwerte geschehen. Zur Auswahl der optimalen Trajektorie aus den theoretisch umsetzbaren Alternativen kommt eine Kostenfunktion zum Einsatz, die im Hinblick auf unterschiedliche Kriterien parametrisiert werden kann. Eine beispielhafte Umsetzung ist in [WZKT10] beschrieben:

$$C = k_J J + k_T T + k_d d^2. \quad (5.3)$$

Die Faktoren  $k_J$ ,  $k_T$  und  $k_d$  erlauben eine Gewichtung des Querrucks  $J$  (dritte Ableitung des Gierwinkels), der Manöverdauer  $T$  und der lateralen Spurbewegung  $d$  und damit eine Beeinflussung der Trajektorienauswahl. Wird in diesem Beispiel nun die Variable  $k_J$  so groß gewählt, dass die restlichen Kriterien überkompensiert werden und damit keinen Einfluss mehr haben, wird immer das Polynom aus der Trajektorienschar gewählt, das über den minimalsten Querruck verfügt. An diesem Punkt ändert auch eine weitere Erhöhung von  $k_J$  nichts an dieser Auswahl, da es keine weiteren Kandidaten mehr gibt, die einen noch kleineren Ruck bieten. In Folge dessen ändert sich bei sonst gleichen Rahmenbedingungen die Bewertung des Fahrverhaltens bei steigendem  $k_J$  auch nicht mehr und der Parametereinfluss wird in diesem Bereich als vernachlässigbar bewertet. Das gleiche Verhalten kann für sehr kleine Werte der Gewichtungsfaktoren auftreten, da diese in dem Fall von den restlichen Faktoren überkompensiert werden und der Einfluss abgewertet wird. Genauso ist eine solche Charakteristik bei der Applikation von Regelsystemen zu erwarten. Wird beispielweise ein proportionales Regelglied (P-Regler) so stark gewichtet, dass eine dauerhafte Regelabweichung resultiert, kann diese so groß werden, dass das Gütekriterium den Parametersatz mit dem schlechtesten verfügbaren Komfortwert belegt. Eine Steigerung des Regelfehlers über die Applikation würde somit zum gleichen Zielfunktionswert führen, sodass eine Schwelle erreicht ist, unterhalb derer Parameteränderungen keinen Einfluss mehr haben. Die analytisch herausgestellten Erkenntnisse können also mit Hilfe der Charakteristik der Zielfunktion und des Aufbaus applizierter Softwaremodule erklärt werden.

Die sich an die Komplexitätsreduktion anschließende Optimierung hat eine Auswahl optimaler Parametersätze mit gleichem Zielfunktionswert herausgestellt. Entgegen des üblichen Verlaufs der Optimierung wurde dabei die beste Lösung bereits in der ersten Iteration gefunden und diese lediglich mit gleichwertigen Samples erweitert. Die Robustheitsanalyse konnte darüber hinaus zeigen, dass Parametersätze an diskreten Punkten im Suchraum eine geringe Stabilität gegenüber parametrischen Änderungen aufweisen und Punkte in der Umgebung flacherer Peaks über höhere Robustheiten verfügen. Die Tatsache, dass gute Lösungen an diskreten Punkten zu finden sind, wird durch die verwendete Zielfunktion begünstigt, die lediglich acht verschiedene Komfortwerte zulässt. Als Folge ergeben sich steile Gradienten im Parameterraum, die eine Tiefensuche ausgehend von guten Lösungen erschweren. Die Anzahl möglicher Parameterwerte ist durch die Komfortskala ebenfalls stark limitiert, sodass die Wahrscheinlichkeit hoch ist, einen guten Parametersatz durch zufallsgesteuertes Erforschen (*exploration*) des Raums zu finden. Die Erkenntnis, dass es nur wenige Punkte im Suchraum zu geben scheint, die einen Funktionswert entsprechend der besten Lösung erreichen können, wird durch die Vielzahl an Wechselwirkungen verursacht. Dadurch, dass eine vollständig unabhängige Einstellung von Parametern nicht realisiert werden kann und stattdessen Abhängigkeiten zwischen fast allen Parametern bestehen (vgl. Abschnitt 5.2.1), entsteht nur ein sehr kleiner Bereich, der eine gleichzeitige Optimierung aller Use-Cases ermöglicht. Bei steigender Dimensionalität des Parameterraums und Komplexität der Wirkkette oder größerem Szenarienkatalog ist eine weitere Verkleinerung dieses Bereichs zu erwarten. Das Vorhandensein alternativer Lösungen mit gleichem Funktions- und unterschiedlichem Robustheitswert bietet somit auch vor dem Hintergrund anzunehmender Modellierungsungenauigkeiten eine höhere Flexibilität. Gerade von Parametersätzen die sich auf einem flacheren Peak im Suchraum befinden (und somit stabiler gegenüber parametrischen Abweichungen sind) ist eine zuverlässige Übertragbarkeit auf das Zielsystem zu erwarten. Eine Prüfung dieser Annahme und Diskussion einer Anwendbarkeit der Methodik auf vergleichbare Probleme sowie des Potentials zur Integration in den industriellen Entwicklungsprozess erfolgt in Kapitel 7.





## 6 Validierung

Die Anwendung der sensitivitätsbasierten Optimierung für ein realistisches Applikationsproblem hat gezeigt, dass die Ziele einer Steigerung des Systemverständnisses und Effizienzverbesserung erreicht werden können. Inwiefern das vorgestellte Verfahren zur Reduktion von Szenariensimulationen zulässig ist, wird im nun folgenden Kapitel geprüft. Auch für das in der Simulation verwendete hybride Fahrzeugmodell erfolgt ein Nachweis der Validität mit Hilfe von Realversuchsdaten und eine Ermittlung des Rechenzeitvorteils durch Gegenüberstellung mit alternativen Fahrzeugmodellen.

### 6.1 Validierungskonzept

Das Konzept zur Validierung der in Kapitel 4 vorgestellten Methodik und zugehöriger Einzelkomponenten erfolgt basierend auf dem in Abschnitt 3.3.3 bereits skizzierten Konzept. Dieses wird im Folgenden weiter spezifiziert.

#### Hybrides Fahrzeugmodell

Das Validierungskonzept des hybriden Fahrzeugmodells entspricht den vom Autor vorherveröffentlichten Ausführungen in [FFFG19b] und wird in dieser Arbeit aufgegriffen. Die Motivation zur Entwicklung des hybriden Modells war, die Fahrzeugdynamik realitätsgetreu abzubilden und gleichzeitig durch eine geringe Komplexität kurze Rechenzeiten zu ermöglichen (vgl. Komplexitäts-Genauigkeits-Trade-Off, Abbildung 2.12). Das etablierte Verfahren zum Nachweis der Generalisierungsfähigkeit neuronaler Netze sieht eine Auswertung des Modellfehlers bei Prädiktionen anhand eines Testdatensatzes vor. Da das LSTM-Netzwerk den größten Stellhebel zur Erreichung hoher Modellierungsgüten darstellt, wird dieser Ansatz für die Validierung des hybriden Modells

verwendet. Entsprechend Abbildung 4.2 kommen dafür Messdaten des realen Prozesses zum Einsatz. Der zur Verfügung stehende Datensatz besteht aus zwölf Szenarien, deren Zeitreihen im Realfahrzeug aufgenommen wurden. Dabei kam das gleiche Fahrzeug zum Einsatz, welches als Grundlage für die Parametrierung des HMs benutzt wurde. Die ersten zehn Datensätze (T1 - T10) haben eine Länge von circa 15 - 20 Sekunden und stellen repräsentative Manöver zum Testen und Applizieren automatisierter Fahrfunktionen dar. Der qualitative Verlauf vorgegebener Sollkrümmungen ist in Abbildung 6.1 dargestellt.

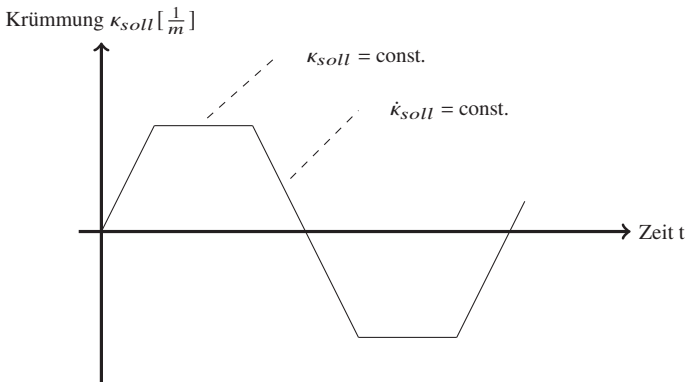


Abbildung 6.1: Qualitativer Verlauf der Sollkrümmung im Testdatensatz zur Validierung des HMs [FFFG19b].

Die Krümmungsvorgabe steigt zunächst für eine bestimmte Zeit mit konstantem Gradienten an bis das Maximum erreicht ist. Dieses wird für einige Zeit gehalten. Im Anschluss erfolgt eine Reduktion der Vorgabe mit betragsmäßig gleicher Krümmungs-Geschwindigkeit hin zur kleinsten Krümmung, welche ebenfalls kurz gehalten wird. Der Prozess wiederholt sich mehrere Male in jedem Testdatensatz. Für jedes Szenario gelten unterschiedliche Krümmungsamplituden und -geschwindigkeiten. Auch die Geschwindigkeit in Längsrichtung wird variiert. Die Parametrierungen der ersten zehn Szenarien realisieren damit für komfortbasierte Assistenzsysteme übliche Fahrprofile. Um aber auch Situationen zu erproben, welche dem nichtlinearen Dynamikbereich entsprechen werden im Szenario T11 Querbeschleunigungen oberhalb  $5 \frac{m}{s^2}$  angefordert. Auch die langfristige Prädiktionsgüte wird aufgrund der Gefahr kumulierter

Fehler (vgl. Abschnitt 4.1) mit Hilfe von Datensatz T12 geprüft. Die dem Satz T12 zugehörigen Zeitreihen wurden während einer Fahrt auf einer Teststrecke aufgenommen, die einem Rundkurs entspricht und Dynamiken im linearen Betriebsbereich wiedergibt. Die Charakteristika aller zwölf Szenarien können Tabelle A.8 entnommen werden. Als Vergleichsgrundlage zur Bewertung der Prädiktionsgüte dienen das komplexere ISAR-Modell, ein Einspurmodell sowie ein isoliertes LSTM-Fahrzeugmodell, welches nicht dem im hybriden Modell zum Einsatz kommenden entspricht. Mit Hilfe der Sollvorgaben erfolgt eine Resimulation aller Szenarien und eine Bestimmung des mittleren Abweichungsquadrats der Krümmungsverläufe (Root Mean Squared Error, RMSE).

Eine Minderung des Genauigkeits-Komplexitäts-Trade-Offs erfordert neben einer hohen Prädiktionsgüte auch einen Effizienzvorteil im Hinblick auf die benötigte Rechenzeit. Dafür erfolgt eine Messung der Simulationszeit bei Benutzung des ESMs, HMs und des ISAR-Modells. Die Auswertungen werden auf dem gleichen Computer (CPU 2,6 GHz, RAM 8 GB) durchgeführt, der bereits für die Optimierung in Kapitel 5 eingesetzt wurde. Während das ISAR-Modell und das ESM in MATLAB/Simulink umgesetzt sind, liegt das hybride Modell in Python vor. Obwohl sich die Simulationsumgebungen unterscheiden, können die Einflüsse dieser auf die Rechenzeitauswertung vernachlässigt werden. Der Grund dafür ist, dass die MATLAB/Simulink-Modelle in C-Code kompiliert werden, deren Ausführungsgeschwindigkeit der von Python-Modellen überlegen ist [Mar18, Pug16]. Sollte also das HM trotzdem eine geringere Rechenzeit aufweisen, ist der Effizienzvorteil bereits nachgewiesen. Würde in dem Fall eine Neu-Implementierung in C oder C++ erfolgen, wäre dieser sogar noch größer. Um stochastische Schwankungen bei der Simulation auszugleichen, wird die Resimulation für jedes Modell 30 Mal durchgeführt und der Mittelwert gegenübergestellt.

### **Sensitivitätsbasierte Optimierung**

Im zweiten Schritt der Validierung soll der Nachweis erbracht werden, dass die in Abschnitt 4.3.2 vorgestellte szenarienbasierte Optimierung zulässig ist. Das im Folgenden beschriebene konzeptionelle Vorgehen wurde in der Vorveröffentlichung [FFF20] bereits vorgestellt. Im Detail erfolgt eine Prüfung, ob der durch das Einsparen von Simulationen entstehende Fehler vernachlässigt werden kann. Damit wird implizit auch das Ergebnis der Komplexitätsredukti-

on validiert. Auf eine explizite Prüfung der Ergebnisse der Sensitivitätsanalyse wird verzichtet, da die Validierung des Verfahrens bereits mit einer varianzba-  
sierten SA durchgeführt wurde (vgl. [FFFG19a]). Die Methodik zur Einspa-  
rung von Simulationsdurchläufen gilt nur dann als zulässig, wenn der dadurch  
entstehende Fehler sich nicht negativ auf die Optimierung auswirkt. Das Opti-  
mierungsergebnis könnte somit mit dem eines Durchlaufs ohne Nutzung der  
Komplexitätsreduktion verglichen werden. Da der genetische Algorithmus ge-  
nau wie vergleichbare Optimierer aber nicht deterministisch arbeitet, ist ein  
solches Vorgehen nicht förderlich. Stattdessen werden neue Parametersätze  
gezielt innerhalb nicht sensitiver Bereiche gesampelt und die Streuung der  
Zielfunktionswerte analysiert.

Da die der Komplexitätsreduktion zu Grunde liegende Sensitivitätsanalyse auf  
einer Mittelung von Elementareffekten beruht (vgl. Abschnitt 2.3.1), besteht die  
Gefahr, dass einzelne Ausreißer in der Auswertung vernachlässigt werden. Als  
Folge könnten Wechselwirkungen oder Haupteffekte unberücksichtigt bleiben,  
die das Optimierungsergebnis unter Umständen negativ beeinflussen. Aus dem  
Grund werden alle für das untersuchte Problem relevanten Haupteffekte und  
Wechselwirkungen jeglicher Ordnung einzeln betrachtet. Eine Visualisierung  
dieser Effekte bietet Abbildung 6.2 für einen beispielhaften Parameterraum.  
In dem dargestellten Suchraum kann eine Reduktion der Wertebereiche für  
sechs Parameter ( $P_3, P_4, P_6, P_9, P_{10}, P_{11}$ ) erzielt werden. Daraus ergeben sich  
sechs Haupteffekte (Variation der Parametersätze entlang einer Dimension)  
und Wechselwirkungen (Gleichzeitige Variation entlang mehrerer Dimensio-  
nen) bis zur fünften Ordnung. Für eine unabhängige Prüfung aller Effekte wird  
nun ein initialer Versuchsplan erzeugt, wobei sich die vom jeweiligen Effekt  
betroffenen Parameter im nicht-sensitiven Bereich befinden. Beispielsweise  
müssten Samples zur Prüfung des Haupteffekts von  $P_3$  für den Parameter  $P_3$   
Werte oberhalb von 0,5 haben und dürften entlang der restlichen Dimensionen  
zufällig verteilt sein (durchgezogene Linie im ersten Plot in Abbildung 6.2).  
Die Sample-Anzahl dieses initialen Versuchsplans pro Effekt sei mit  $n_{Init}$  be-  
zeichnet. Für jeden der  $n_{Init}$  Parametersätze werden im Anschluss  $n_{Var}$  weitere  
Samples erzeugt, die sich von dem Basis-Datensatz lediglich innerhalb der für  
den Effekt relevanten nicht-sensitiven Regionen unterscheiden (z.B. die gestri-  
chelten und gepunkteten Samples im ersten Plot in Abbildung 6.2). Wenn die in  
Abschnitt 2.3 getroffenen Annahmen richtig sind, müssten die Unterschiede der  
Funktionswerte aller  $n_{Var}$  Parametersets und dem zugehörigen Basis-Sample  
vernachlässigbar klein sein. Um diesen Nachweis zu erbringen werden - genau

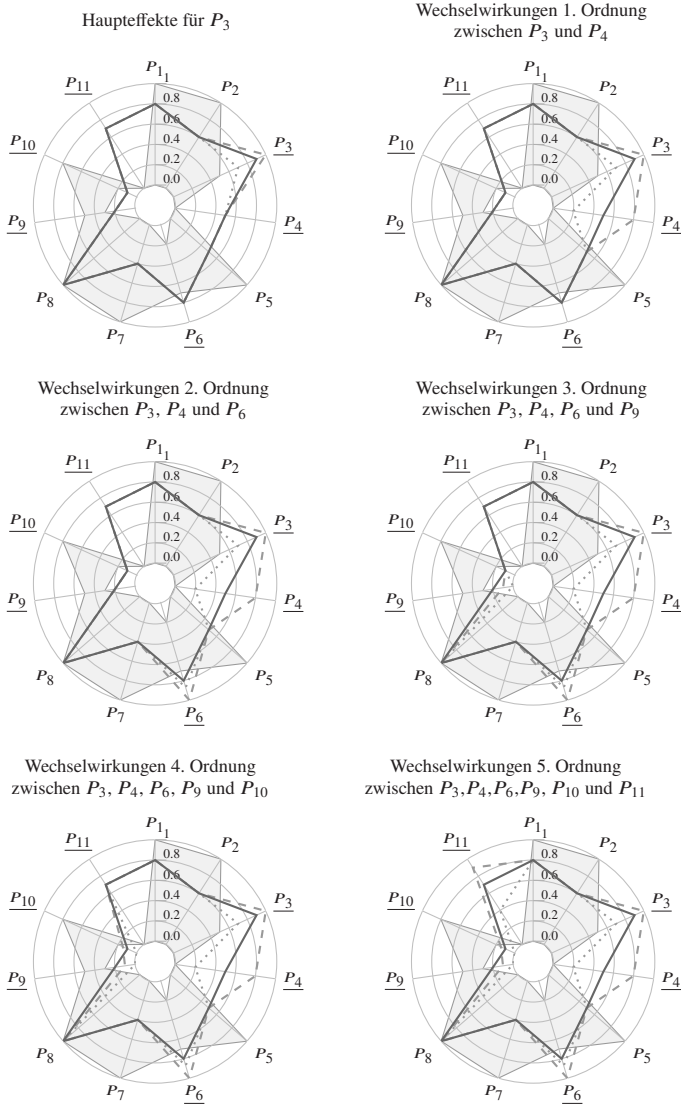


Abbildung 6.2: Visualisierung relevanter Haupteffekte und Wechselwirkungen für das Verfahren der sensitivitätsbasierten Optimierung [FFFG20].

wie bei der Robustheitsanalyse - Abweichungen des Gütekriteriums größer als eins als kritisch bewertet. Die Anzahl kritischer Parametersätze  $n_{\Delta f(x) > 1}$  wird dann der Gesamtanzahl ausgewerteter Samples ( $n_{\Delta f(x) > 1} + n_{\Delta f(x) \leq 1}$ ) gegenübergestellt. Als Metrik wird der Anteil kritischer Parametersätze  $R_{krit}$  pro Szenario definiert:

$$R_{krit} = \frac{n_{\Delta f(x) > 1}}{n_{\Delta f(x) > 1} + n_{\Delta f(x) \leq 1}} \quad (6.1)$$

Die für die Bestimmung der Metrik benötigte Anzahl an Simulationen pro Szenario  $n_{Validierung}$  ergibt sich aus der Anzahl relevanter Effekte  $n_{Effekte}$  (Haupteffekte und Wechselwirkungen). Diese leitet sich direkt aus den Ergebnissen der Komplexitätsreduktion ab und steigt exponentiell mit der Anzahl an Parametern  $n_{KR}$ , die hinsichtlich ihres relevanten Einflussbereichs durch die Komplexitätsreduktions-Methodik eingeschränkt wurden:

$$n_{Effekte} = 2^{n_{KR}} - 1. \quad (6.2)$$

Wie oben beschrieben, wird ein initialer Versuchsplan für jeden der  $n_{Effekte}$  Effekte bestimmt und jedes Sample dieses Plans wiederum variiert. Somit ergibt sich die Anzahl an Simulationen pro Szenario zu:

$$n_{Validierung} = n_{Effekte} \cdot n_{Init} \cdot (n_{Var} + 1) \quad (6.3)$$

## Übertragbarkeit der Ergebnisse auf den Fahrversuch

Aufgrund des weiterhin hohen Stellenwerts fahrzeugbasierter Applikationen im Anschluss an die simulativen Analysen soll final die Übertragbarkeit optimaler Parametersätze auf das Zielsystem getestet werden. Im Unterschied zur Simulation ergeben sich im Fahrversuch ständig wechselnde Rahmenbedingungen, insbesondere in Bezug auf die Manövercharakteristik und das dynamische Verhalten des Fahrzeugs. Im Fokus der Validierungsstudien steht die Robustheitsanalyse gegenüber Modellparametern des Fahrzeugmodells. Kann eine hohe Stabilität der Parametersätze in Bezug auf sich ändernde Übertragungsverhalten nachgewiesen werden, ist eine vergleichbare Performance bei unterschiedlichen Derivaten oder Umwelteinflüssen (z.B. Seitenwind, Gegenwind) zu erwarten. Dieses Vorgehen ist bereits in der Vorveröffentlichung [FFFG20] beschrieben und wird im Rahmen dieser Arbeit nochmals aufgegriffen.

Für die Robustheitsanalyse wird analog zu dem Vorgehen in Abschnitt 5.2.3 der *degree-of-robustness*-Ansatz verwendet. Im Unterschied zu der letzten Auswertung bleiben die Werte der Applikationsparameter jedoch gleich. Stattdessen erfolgt eine systematische Variation der ESM-Parameter des hybriden Modells in einer  $\delta$ -Umgebung um den jeweiligen Ausgangs-Parametersatz (vgl. Tabelle A.2). Der minimale Hyperbox-Radius  $\delta_1$  leitet sich - wie auch schon in Abschnitt 5.2.3 - aus experimentellen Vorstudien ab:  $\delta_1 = 0,01$ . Hyperbox-Vergrößerungen werden in 10 Schritten durchgeführt:  $k \in [1, 2, \dots, 10]$ . Die zu Grunde gelegten Ober- und Untergrenzen der ESM-Modellparameter bilden repräsentative Wertebereiche für Kraftfahrzeuge ab und sind in Tabelle A.9 angegeben. Die Robustheitsbedingung wird aus Abschnitt 5.2.3 übernommen und bewertet eine ESM-Parametrierung für einen Parametersatz als robust, wenn keiner der szenarienspezifischen Funktionswerte mehr als eins von dem ursprünglichen Ergebnis abweicht.

## 6.2 Ergebnisse

Entsprechend des im letzten Abschnitt beschriebenen Validierungskonzepts erfolgt nun eine Vorstellung und Analyse der Ergebnisse. Im sich daran anschließenden Abschnitt 6.3 werden Schlussfolgerungen aus den Erkenntnissen abgeleitet.

### 6.2.1 Validierung des hybriden Fahrzeugmodells

Die im Folgenden beschriebenen Ergebnisse stützen sich auf die in [FFFG19b] beschriebene Validierungsstudie. Die darin vom Autor vorveröffentlichten Erkenntnisse werden in diesem Abschnitt nochmals aufgegriffen. Wie beschrieben, wird die Generalisierungsfähigkeit des hybriden Modells anhand einer Resimulation von zwölf Testdatensätzen durchgeführt und der RMSE-Fehler als Vergleichsmetrik verwendet. Die Ergebnisse sind in Abbildung 6.3 gegenübergestellt. Die zugehörigen Zahlenwerte finden sich in Tabelle A.10. Das Diagramm zeigt, dass das hybride Modell eine bessere Modellierungsperformance bietet als das Einspurmodell und das isolierte LSTM-Modell. Es kann also bestätigt werden, dass eine Kombination beider Modelle zu einem kleineren Prädiktionsfehler führt als die individuelle Nutzung der Ansätze. Da

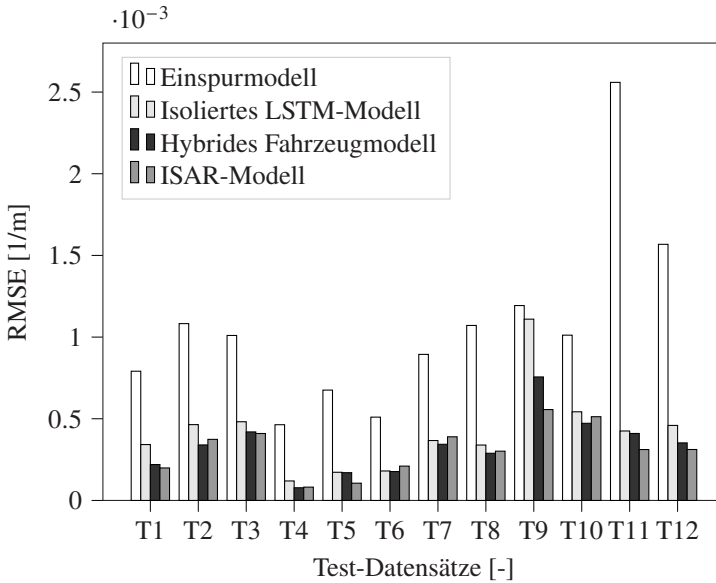


Abbildung 6.3: Prädiktionsgüte des ESMs, isoliertem LSTM-Fahrzeugmodell, ISAR-Modell und des hybriden Modells [FFFG19b].

für das Training des HMs Datensätze des ISAR-Modells verwendet werden, ist eine vergleichbare Performance zu erwarten. Ein Vergleich zugehöriger RMSE-Fehler bestätigt diese Annahme: Auch wenn das HM in einigen Fällen sogar genauer arbeitet als das ISAR-Modell (z.B. für T6, T7, T8, T10), sind die Unterschiede so gering, dass von einer gleichwertigen Modellierungsperformance ausgegangen werden kann. Der Nachweis, dass das hybride Modell trotz der deutlich reduzierten Komplexität eine ähnlich genaue Prädiktion ermöglicht wie das umfangreichere, analytische ISAR-Modell, kann somit erbracht werden. Auch die guten Ergebnisse im nichtlinearen Bereich (T11) und bei längeren Simulationszeiten (T12) belegen, dass eine sinnvolle Nutzung des HMs für virtuelle Applikationen möglich ist.

Zur detaillierten Analyse des Modellverhaltens von ESM, isoliertem LSTM und HM zeigt Abbildung 6.4 beispielhaft die Prädiktionen aller drei Modelle für den Test-Datensatz T5. Das Einspurmodell zeigt bei der Prädiktion fast keine Verzögerung im Vergleich zu den Fahrversuchsdaten, weicht aber ins-



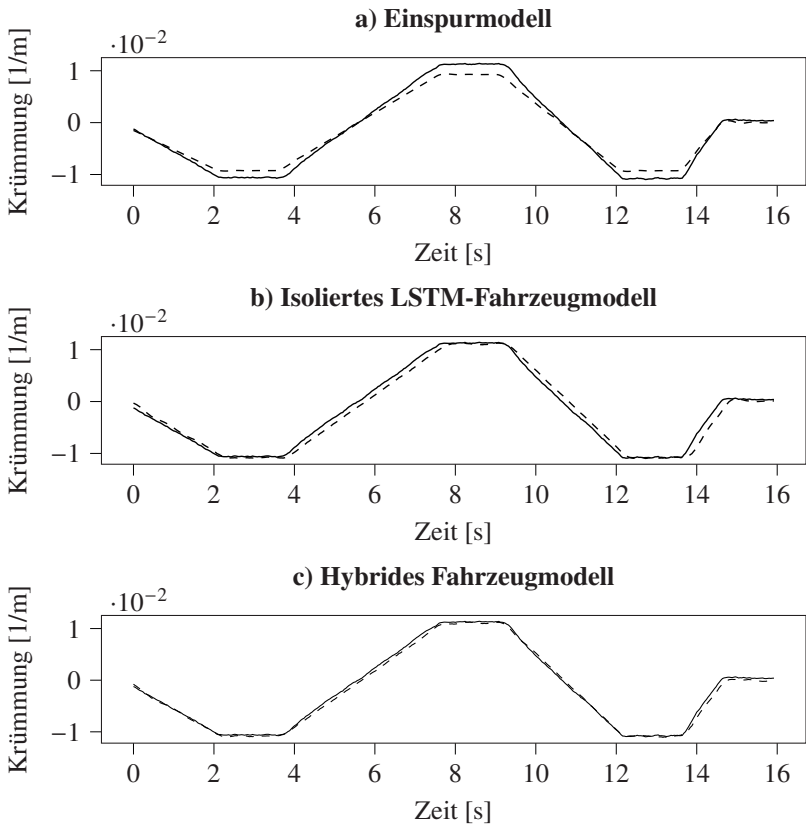


Abbildung 6.4: Vergleich der prädizierten Krümmungsverläufe (gestrichelte Linie) für das Einspurmodell, LSTM-Fahrzeugmodell und das hybride Modell im Vergleich zu Realversuchsdaten (durchgezogene Linie) für Szenario T5 [FFFG19b].

besondere im Bereich konstanter Krümmung von der Realität ab. Das isolierte LSTM-Modell zeigt hingegen einen kleineren absoluten Prädiktionsfehler, scheint die reale Krümmung aber nur mit einer zeitlichen Verzögerung abbilden zu können. Die Modellgenauigkeiten des ESMs sind auf die fehlerbehaftete Parametrierung zurückzuführen. Aufgrund der Fähigkeit des LSTMs, Prädiktionsfehler mit Hilfe genauerer ISAR-Trainingsdaten ausgleichen zu können, sind die Anforderungen an eine korrekte ESM-Parametrierung vergleichsweise

gering, sodass diese geschätzt wurde (vgl. Abschnitt 4.1.2). Die Abweichung von der realen Prozessdynamik zeigt sich insbesondere im nichtlinearen Bereich (vgl. Datensatz T11 in Abbildung 6.3), wobei zusätzlich konzeptbedingte Grenzen des Einspurmodells zum Tragen kommen (vgl. Abschnitt 2.5.1). Der entscheidende Unterschied zu Mehrkörpersystemen ist das vereinfachte Reifenmodell, welches gegenüber des im ISAR-Modell zum Einsatz kommenden Pacejka-Modells die Reifendynamik bei hohen Querbeschleunigungen nicht mehr modellieren kann. Da das HM für den Datensatz T11 einen vergleichsweise kleinen Fehler aufweist, während die Abweichung zwischen dem ESM und ISAR-Modell steigt, kann angenommen werden, dass das neuronale Netz die Modellschwächen des Einspurmodells lernen kann.

Der zeitliche Versatz des isolierten LSTM-Modells in Abbildung 6.4 b) ist ein verbreitetes Problem rekurrenter Netze, welches mit Hilfe von Hyperparameter-Tunings zwar abgeschwächt, aber nicht vollständig beseitigt werden kann. Die Krümmungsverläufe in Abbildung 6.4 c) belegen jedoch, dass durch eine Kombination des ESMs und LSTM-Netzwerks die individuellen Nachteile reduziert werden können und eine insgesamt verbesserte Prädiktion erreicht wird. Neben der Fähigkeit, nichtlineare Fahrdynamiken modellieren zu können, zeigen die Ergebnisse für Datensatz T12, dass auch das Risiko kumulierter Prädiktionsfehler durch das zweistufige Trainingsschema (vgl. Abbildung 4.3) behoben werden kann.

Neben einer hohen Modellierungsgüte ist der Anspruch an das hybride Fahrzeugmodell, kurze Simulationszeiten zu ermöglichen. Zur Überprüfung der Recheneffizienz zeigt Abbildung 6.5 deshalb die mittleren Rechenzeiten aller zwölf Datensätze. Die zugehörigen Werte können Tabelle A.11 entnommen werden.

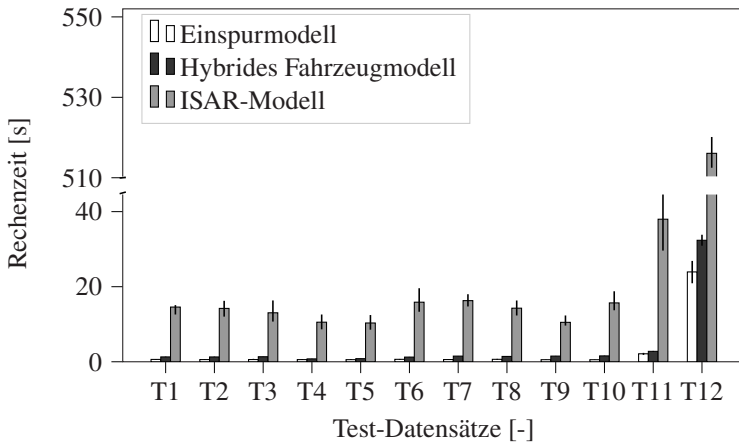


Abbildung 6.5: Vergleich der Rechenzeiten aller Modelle für die Test-Datensätze [FFFG19b]

Die Ergebnisse zeigen, dass Simulationen mit dem hybriden Modell circa 5 - 10 mal schneller durchgeführt werden können als mit dem ISAR-Modell. Die effizientere Performance ist durch die Komplexität beider Modelle zu erklären. Das HM besteht aus einem Einspurmodell, welches lediglich aus zwei analytisch lösbaren Differentialgleichungen besteht, deren Berechnung mit minimalem Ressourcenaufwand durchführbar ist (vgl. Abschnitt 2.5.1). Auch die Bestimmung von Ausgangswerten jeder LSTM-Unit ist deterministisch und verlangt keinen großen Aufwand. Weil die Netzstruktur des zum Einsatz kommenden KNNs über wenig Neuronen und Zwischenschichten verfügt (vgl. Tabelle A.3), lassen sich kurze Rechenzeiten realisieren. Dem gegenüber steht das ISAR-Modell, welches als Mehrkörpersystem die Auswertung aller teils komplexen Einzelkomponenten erfordert und somit deutlich mehr Rechenaufwand benötigt.

## 6.2.2 Validierung der sensitivitätsbasierten Optimierung

Die Validierung der Methodik zur Reduktion von Szenariensimulationen während der Optimierung erfolgt mit Hilfe des in Abschnitt 6.1 beschriebenen Sampling-Konzepts. Wie zuvor bereits beschrieben, wurde das konzeptionelle

Vorgehen erstmals in [FFFG20] vorgestellt, die Ergebnisse unterscheiden sich allerdings aufgrund der Tatsache, dass unterschiedliche Versuchsplangrößen zu Grunde liegen. Entscheidend für die Bestimmung der Versuchspläne ist die Anzahl an relevanten Haupteffekten und Wechselwirkungen, die sich direkt aus der Anzahl von der Komplexitätsreduktion reduzierter Wertebereiche und Dimensionen ableitet (vgl. Gleichung 6.2). Da der Simulationsaufwand exponentiell steigt, je mehr Parameter von der Komplexitätsreduktion betroffen sind, wird angestrebt, den Versuchsplan jedes Effekts ( $n_{Init} \cdot (n_{Var} + 1)$ ) so klein wie möglich zu halten, ohne dabei die Aussagekraft der Ergebnisse zu gefährden. In vorherigen Studien wurde eine Konvergenz lokal angewendeter SAs bereits mit 15 Elementareffekten pro Intervall nachgewiesen [FFFG19a] (vgl. Abschnitt 5.2.1). Da die EEM damit alle Effekte testet und im Fall der Validierungsstudie jeder Effekt einzeln untersucht wird, kann der durch  $n_{Init}$  und  $n_{Var}$  definierte Versuchsplan deutlich kleiner ausfallen. Um mindestens 30 Samples pro Effekt realisieren zu können, werden deshalb für Szenarien mit einer hohen Anzahl an Effekten fünf Basis-Samples erzeugt und diese jeweils fünf mal variiert ( $n_{Init} = n_{Var} = 5$ ). Für Szenarien, deren relevanter Suchraum weniger eingeschränkt ist, erfolgt eine Erhöhung der Sampling-Variablen. Tabelle 6.1 stellt die sich dadurch ergebenden Versuchsplangrößen zusammen. Die benötigte Rechenzeit kann mit einer Sample-Anzahl von insgesamt 132210 im realistischen Rahmen gehalten haben (entspricht ca. 367 Stunden).

Tabelle 6.1: Sample-Parameter für die Validierungsstudie

Szenario	$n_{KR}$	$n_{Effekte}$	$n_{Init}$	$n_{Var}$	$n_{Validierung}$
S1	3	7	25	25	4550
S2	3	7	25	25	4550
S3	3	7	25	25	4550
S4	6	63	20	20	26460
S5	11	2047	5	5	61410
S6	10	1023	5	5	30690

Zur Validierung der sensitivitätsbasierten Optimierung muss nachgewiesen werden, dass sich die Funktionswerte innerhalb jeder der  $n_{Init}$  Sampling-Gruppen aller Effekte nicht signifikant voneinander unterscheiden. Dafür wird

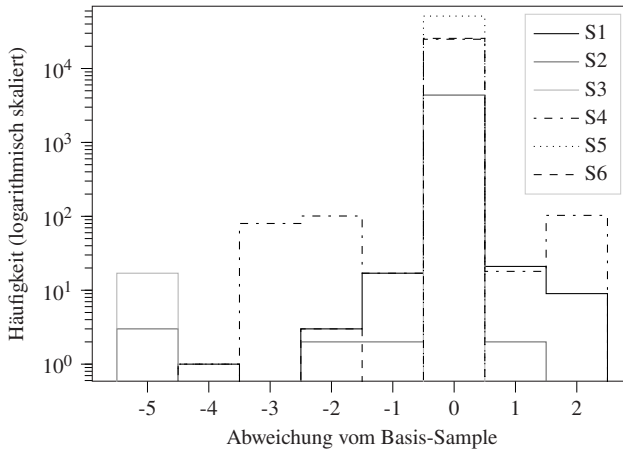


Abbildung 6.6: Häufigkeitsverteilung der Abweichung von Zielfunktionswerten der Sampling-Gruppen von dem jeweiligen Basis-Sample für alle Szenarien.

für jedes Szenario der Anteil kritischer Parametersätze  $R_{krit}$  entsprechend Gleichung 6.1 berechnet. Abbildung 6.6 stellt dafür zunächst die Häufigkeitsverteilungen der Abweichungen für alle Szenarien mit Hilfe gestapelter Histogramme dar. Es sei darauf hingewiesen, dass die Häufigkeit logarithmisch skaliert ist.

Das Diagramm lässt erkennen, dass die Mehrheit untersuchter Parametersätze in allen szenarienspezifischen Suchräumen eine Abweichung von null vom jeweiligen Basis-Datensatz aufweist. Abweichungen größer als 1 treten im Vergleich dazu nur selten auf (vgl. dazu die Zahlenwerte des Diagramms in Tabelle A.12). Die zugehörigen Anteile kritischer Szenarien  $R_{krit}$  zeigt Tabelle 6.2.

Tabelle 6.2: Häufigkeitsverteilung der Abweichung von Zielfunktionswerten vom jeweiligen Basis-Sample.

S1	S2	S3	S4	S5	S6
0,3 %	0,1 %	0,4 %	1,1 %	0,002 %	0,01 %

Das kritischste Ergebnis tritt für Szenario S4 auf, bei dem 1,1 % der für Simulationsreduktionen qualifizierten Parametersätze mit einem falschen Funktionswert belegt wurden. Wird eine Optimierung mit 1000 Individuen durchgeführt, würde also unter der Annahme, dass die Simulationen aller Parametersätze eingespart werden könnten, elf Lösungen fehlerbehaftet sein. Weil in der Praxis aber nicht zu erwarten ist, dass alle Parametersets für die Methodik qualifiziert sind (Im vorliegenden Beispiel waren es nur ca. 16 %, vgl. Abbildung 5.8), ist ein deutlich kleinerer Einfluss dieses Fehlers auf den Optimierungsverlauf zu erwarten. Auch die Tatsache, dass der genetische Algorithmus erneute Funktionsauswertungen nur für zur Rekombination und Mutation selektierte Individuen verlangt, reduziert den Einfluss. In dem Anwendungsbeispiel aus Kapitel 5 wurden bei einer Populationsgröße von 1000 Individuen und 20 Iterationen 6416 Szenariensimulationen auf Grundlage des vorgestellten Verfahrens eingespart. Unter der Annahme, dass der Anteil fehlerhafter Zielfunktionswerte für alle Szenarien 1,1 % entspricht, ergäben sich demnach maximal 65 falsch bewertete Parametersätze. Bei insgesamt 120000 während der Optimierung verwendeten Datensätzen wären also lediglich  $5,42 \cdot 10^{-2}$  % der Lösungen fehlerhaft. Dieser Anteil ist so gering, dass eine negative Beeinflussung des Optimierungsprozesses nicht zu erwarten ist. Eine Anwendung des vorgestellten Verfahrens zur Reduktion von Simulationszeiten kann vor dem Hintergrund gerechtfertigt werden. Die Ergebnisse bestätigen damit die in [FFFG20] bereits gewonnenen Erkenntnisse.

### **6.2.3 Robustheitsanalyse gegenüber Fahrzeugmodellparametern**

Im dritten Teil der Validierung wird die Übertragbarkeit virtuell gefundener Lösungen auf den Fahrversuch geprüft. Während die Studien in Abschnitt 6.2.1 bereits die Validität des verwendeten Fahrzeugmodells herausgestellt haben, soll die in diesem Abschnitt vorgestellte Robustheitsanalyse Aufschluss über die Stabilität der Parametersätze gegenüber sich änderndem Fahrzeugübertragungsverhalten unter realen Rahmenbedingungen geben. Dafür kommt ein *degree-of-robustness*-Ansatz zum Einsatz, welcher die elf in Kapitel 5 herausgestellten Lösungen im Hinblick auf deren Robustheit gegenüber veränderten ESM-Parametern prüft. Die im Folgenden vorgestellten Ergebnisse wurden vom Autor bereits in [FFFG20] veröffentlicht. Analog zum Vorgehen

in Abschnitt 5.2.3 erfolgt eine Darstellung in Parallelkoordinaten. Abbildung 6.7 stellt die Ergebnisse der Robustheitsanalyse mit Modellparametern denen im Hinblick auf parametrische Abweichungen durchgeführten Auswertungen gegenüber (vgl. Abbildung 5.7). Die zugehörigen Werte finden sich in Tabelle A.13.

Ein Vergleich der Robustheitsgrade zeigt, dass die im Hinblick auf Applikationsparameter als sehr robust ( $DoR \geq 10$ ) klassifizierten Parametersätze ebenfalls weitgehend stabil gegenüber variierenden Fahrzeugdynamiken sind. Eine Abweichung um mehr als ein Robustheitslevel ist lediglich für drei der elf Parametersätze erkennbar. Die Erkenntnisse zeigen, dass ein Zusammenhang zwischen der Stabilität eines Datensatzes gegenüber Applikations- und Modellparametern zu existieren scheint. Lösungen, die sich im Suchraum an flacheren Peaks befinden, sollten deshalb aufgrund ihrer hohen Robustheit bevorzugt werden und als Grundbedingung für weitere Applikationen auf der Zielhardware dienen. Eine Koppelung von Simulation und Fahrversuch gemäß Abbildung 3.1 kann somit ohne Inkaufnahme großer Differenzen umgesetzt werden. Während die in Abschnitt 6.2.1 vorgestellten Ergebnisse die Genauigkeit des Simulationsmodells belegen, weist die in diesem Abschnitt beschriebene Robustheitsanalyse die Stabilität virtuell gefundener Lösungen hinsichtlich unter realen Bedingungen veränderter Dynamik der Wirkkette nach. Auch eine Abschätzung der Performance virtuell optimaler Bedingungen auf anderen Zielerivaten ist möglich, wenn sich dessen ESM-Parametrierung im ausgewerteten Bereich befindet.

## 6.3 Fazit

Das Ziel der Validierungsstudien war, die in Kapitel 4 vorgestellten Applikationsverfahren und -modelle sowie die in Kapitel 5 präsentierten Ergebnisse hinsichtlich ihrer Gültigkeit zu bewerten. Analog zu dem dreiteiligen Aufbau der Methodik erfolgte zunächst eine Prüfung der Genauigkeit und Effizienz des hybriden Fahrzeugmodells, bevor im Anschluss die sensitivitätsbasierte Optimierung validiert wurde.

Entsprechend des üblichen Vorgehens für KNN-basierte Modelle erfolgte eine Validierung der Modellierungsperformance des HMs aufgrund von Daten des realen Prozesses und eine Auswertung der Recheneffizienz im Vergleich

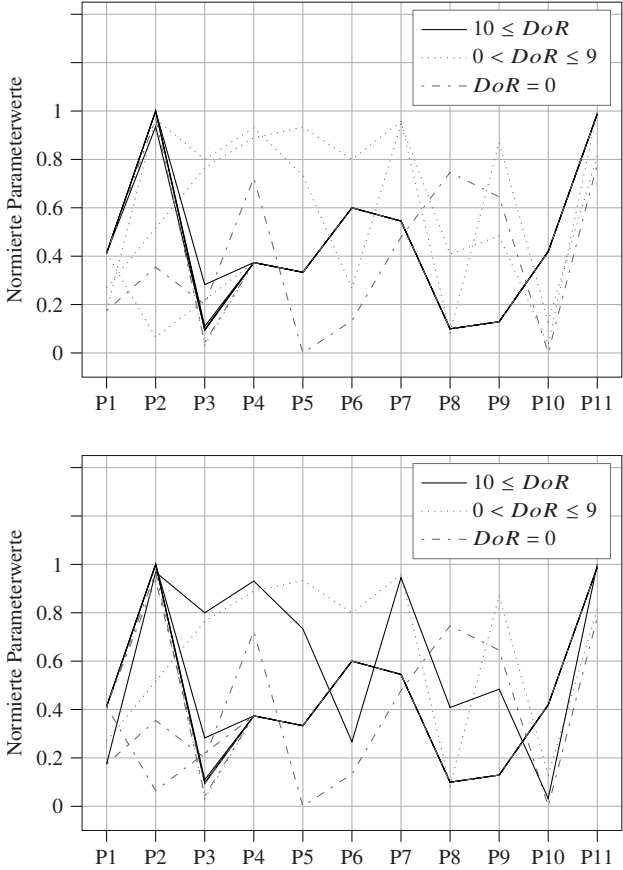


Abbildung 6.7: Oben: Parametersätze und Robustheitsinformationen in Bezug auf Applikationsparameter (vgl. 5.7. Unten: Ergebnisse der Robustheitsanalyse gegenüber ESM-Modellparametern [FFFG20])



zu alternativen, analytischen Ansätzen. Das hybride Modell ist mit dem Ziel entstanden, den in Abbildung 2.12 beschriebenen Komplexitäts-Genauigkeits-Trade-Off zu lösen und eine gleichzeitig gesteigerte Nachvollziehbarkeit gegenüber reinen KNNs zu erreichen. Die Auswertung des Prädiktionsfehlers stellt eine dem komplexeren ISAR-Modell gleichwertige Modellierungsperformance heraus. Die Schwächen des deutlich einfacheren Einspurmodells können offensichtlich durch eine Einbeziehung des LSTM-Netzes im hybriden Modell ausgeglichen werden. Dies umfasst sowohl parametrische Ungenauigkeiten als auch konzeptbedingte Grenzen des ESMs (z.B. im nichtlinearen Dynamikbereich). Die zweite Anforderung einer hohen Recheneffizienz lässt sich anhand des experimentellen Vergleichs mit dem ESM und ISAR-Modell nachweisen. Wegen der 5-10 mal schnelleren Simulation können schnelle Feedback-Schleifen für einen global optimalen Applikationsprozess realisiert werden. Durch die Verwendung einer parallelen Architektur zur Kombination von ESM und KNN in dieser Arbeit lässt sich ebenfalls der Black-Box-Charakter reiner neuronaler Netze auflösen. Abbildung 6.8 gibt die in Abschnitt 6.2.1 vorgestellten Ergebnisse qualitativ im Komplexitäts-Genauigkeits-Diagramm wieder und repräsentiert damit die Vorteile des entwickelten Modellierungsansatzes.

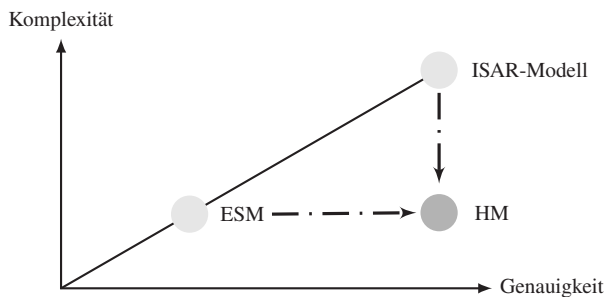


Abbildung 6.8: Einordnung des hybriden Modells im Komplexitäts-Genauigkeits-Trade-Off-Diagramm (vgl. Abbildung 2.12).

Das Verfahren zur Reduktion von Szenariensimulationen wurde mit Hilfe eines umfangreichen Resamplings validiert. Im Fokus stand dabei die Frage, ob sich eine Ersetzung von Funktionswerten basierend auf Sensitivitätsinformationen negativ auf den Optimierungfortschritt auswirkt. Eine Auswertung

von insgesamt 132210 Parametersätzen konnte zeigen, dass der Anteil fehlerhaft bewerteter Lösungen für fast alle Szenarien vernachlässigbar klein ist. Bei dem kritischsten Szenario wurden circa 1 % der Samples mit falschen Funktionswerten belegt. Weil der genetische Algorithmus, so wie andere populationsbasierte Optimierer, allerdings über eine relativ stark ausgeprägte Breitensuche verfügt, kann die Suche in Richtung falsch bewerteter Parametersätze durch ein zufälliges Erforschen (z.B. mit Mutationen) neuer Bereiche leicht wieder verworfen werden. Darüber hinaus ist davon auszugehen, dass niemals alle Individuen einer Population für das eingeführte Verfahren qualifiziert sind, sodass sich die Anzahl fehlerhafter Funktionswerte weiter reduziert. Beim genetischen Algorithmus kommt noch hinzu, dass dieser lediglich Systemevaluationen selektierter Parametersets fordert. Für den in Kapitel 5 beschriebenen Anwendungsfall wurde herausgestellt, dass maximal  $5,42 \cdot 10^{-2}$  % der betrachteten Individuen fehlerbehaftet sein könnten. Darauf basierend lässt sich schlussfolgern, dass der Optimierungsprozess durch eine Nutzung von Einflussinformationen zur Erhöhung der Effizienz nicht gestört wird. Darüber hinaus kann das Verfahren zur Komplexitätsreduktion des Suchraums als valide bewertet werden, da sich die dadurch ergebenden nichtsensitiven Bereiche auch basierend auf den neu gesampelten Versuchsplänen als nicht relevant herausgestellt haben. Die Einflussinformationen können somit ebenfalls zu einer Erhöhung des Systemverständnisses beitragen und sinnvoll für eine manuelle, zielgerichtete Applikation eingesetzt werden.

Zuletzt erfolgte eine Untersuchung der Übertragbarkeit virtuell gefundener Lösungen auf den Fahrversuch. Weil sich unter realen Bedingungen insbesondere die Fahrzeugdynamik verändert, wurde eine Robustheitsanalyse mit ESM-Modellparametern durchgeführt. Dabei konnte herausgestellt werden, dass die Stabilitäten der Parametersätze bezüglich Änderungen von Applikations- und Modellparametern korrelieren. Für den im Rahmen dieser Arbeit untersuchten Anwendungsfall konnten hohe Stabilitäten für mehrere virtuell optimale Parametersets nachgewiesen werden, die sich an flacheren Peaks im Suchraum befinden. Lösungen, welche sich an diskreten Punkten im Raum befinden, waren hingegen nicht robust gegenüber sich änderndem Fahrzeugübertragungsverhalten. Die Ergebnisse zeigen, dass sich auf Grundlage des hybriden Fahrzeugmodells und der Optimierungsmethodik Bedatungen finden lassen, die auch unter realen Bedingungen ein gutes Fahrverhalten erwarten lassen. Eine Robustheitsanalyse sollte immer durchgeführt werden, um virtuell identifizierte, gleichwertige Applikationsdatensätze bezüglich ihrer Robustheit im

Fahrversuch weiter einschränken zu können. Das für die Applikationsmethodik entscheidende Zusammenspiel zwischen Simulation und Fahrversuch (vgl. Abbildung 3.1) kann somit gewährleistet werden.



## 7 Bewertung und Diskussion der Methodik

Im Rahmen dieser Arbeit wurden Anforderungen an eine Applikationsmethodik für automatisierte Fahrfunktionen aus dem aktuellen Stand der Technik und Wissenschaft abgeleitet. Eine Bewertung existierender Ansätze in Bezug auf deren Potential zur Erfüllung dieser Anforderungen erlaubte im Anschluss die Herleitung des Forschungsbedarfs und die Formulierung von drei Forschungsfragen (vgl. Kapitel 3). Daraufhin konnten ein Forschungskonzept und Verfahren zur effizienten Applikation entwickelt werden (Kapitel 4), welche an einem Beispiel angewendet (Kapitel 5) und experimentell validiert wurden (Kapitel 6). Im nun folgenden Kapitel erfolgt basierend auf den Erkenntnissen eine Beantwortung der Frage, inwiefern der Forschungsbedarf adressiert werden konnte. Außerdem wird der Ansatz kritisch diskutiert und dabei insbesondere die Anwendbarkeit für den industriellen Entwicklungsprozess sowie die Übertragbarkeit auf vergleichbare Probleme hinterfragt.

Im ersten Schritt erfolgt eine Bewertung der Applikationsmethodik hinsichtlich der Forschungsfragen (vgl. Abschnitt 3.3.1):

**1. Forschungsfrage:** Wie können Simulationsumgebungen sinnvoll in den Applikationsprozess eingebunden werden und mit welchem Konzept lässt sich eine hohe Genauigkeit, Effizienz und Adaptivität von Fahrzeugmodellen erreichen?

Die Simulation nimmt in dem vorgeschlagenen Applikationsprozess (vgl. Abbildung 3.1) eine besondere Rolle ein. Zum Einen werden Vorstudien zur Komplexitätsreduktion und Optimierung virtuell durchgeführt und zum Anderen dient sie als Werkzeug zur kontinuierlichen Absicherung von Applikationen im Fahrversuch. In dem Zusammenhang entstand die Forderung nach einem hochgenauen, recheneffizienten und adaptiven Fahrzeugmodell, um kurze Feedback-Schleifen, eine hohe Prädiktionsgüte und Anpassbarkeit an unter-

schiedliche Derivate sicherzustellen. Basierend auf den Defiziten bestehender Ansätze wurde ein hybrides Modell vorgeschlagen, welches aufgrund seines Aufbaus die Chance bietet, hohe Genauigkeiten bei gleichzeitig minimalen Rechenaufwand zu erreichen. Die Adaptivität wird durch die Verwendung eines neuronalen Netzes realisiert, das basierend auf Messdaten trainiert werden kann. Das theoretisch erarbeitete Konzept wurde mit Testdaten repräsentativer Szenarien validiert, indem ein Vergleich des Modellfehlers und der Rechenzeit mit alternativen Ansätzen erfolgte (vgl. Abschnitt 6.2.1). Die Studien belegen, dass durch Kombination eines einfachen analytischen Modells (Einspurmodell) mit einem LSTM-Netzwerk die Schwächen beider Modelle ausgeglichen werden können und eine verbesserte Modellgüte und Effizienz erzielt wird. Aufgrund der Ergebnisse ist eine Realisierung des vorgeschlagenen Konzepts aus Abbildung 3.1 möglich. Durch das hybride Fahrzeugmodell kann eine Simulationsumgebung geschaffen werden, die eine Übertragbarkeit virtuell erzielter Ergebnisse auf den Fahrversuch ermöglicht und gleichzeitig ein effizientes Absicherungswerkzeug für die fahrzeuggestützte Applikation darstellt. Im Gegensatz zum konventionellen Applikationsprozess ist mit einem verringerten Ressourcenaufwand zu rechnen, da Grundapplikationen rein simulativ durchgeführt werden können und das manuelle Vorgehen zur Parametrierung durch kontinuierliches Feedback systematisch zum globalen Optimum gesteuert wird. Durch ein rechtzeitiges Aufdecken von Wechselwirkungen und der Erweiterung des Fahrversuchs mit der Simulation können unvorhergesehene Effekte früher aufgezeigt und eine zeitgemäße Berücksichtigung ermöglicht werden. Die Integration des menschlichen Fahrers in die Applikationsschleife bietet weiterhin die Möglichkeit, das subjektive Fahrgefühl mit einzubeziehen, was bei rein simulativer Durchführung nicht gegeben wäre.

**2. Forschungsfrage:** Gibt es eine Methode, mit der einflussreiche Parameter und Wertebereiche im Suchraum verlässlich mit vertretbarem Rechenaufwand bestimmt werden können?

Die Komplexität und Dimensionalität des Applikationsproblems führt im konventionellen Applikationsprozess zu einem hohen Ressourceneinsatz. Um eine Parametrierung mit vertretbarem Aufwand zu ermöglichen, werden deshalb oft Abschätzungen gemacht oder Annahmen getroffen, welche die Güte der gefundenen Lösung reduzieren können. Beispiele dafür sind die relevanten Wertebereiche jedes Parameters sowie Einflussinformationen im Suchraum. In der fahrzeuggestützten Applikation ergeben sich diese entweder aus Erfahrungs-

---

Expertenwissen oder werden unter hohem Aufwand experimentell bestimmt. Da die Zuverlässigkeit der Informationen in diesen Fällen nicht gegeben ist oder deren Bestimmung mit nicht vertretbaren Kosten verbunden wäre, wurde in Abschnitt 3.3.1 der Bedarf nach einem Verfahren zur Bestimmung einflussreicher Parameter und Wertebereiche abgeleitet. Aufgrund des hohen Effizienzvorteils kommen dazu ebenfalls Simulationsumgebungen zum Einsatz.

Das vorgestellte Verfahren verwendet die weit verbreitete Elementareffektmethode als Sensitivitätsanalyse zur lokalen Anwendung im Suchraum mit dem Ziel der Identifikation sensitiver Regionen. Dazu wird jeder Parameter entlang initialer, konservativ geschätzter Grenzen in Intervalle aufgeteilt und die Parameterwirkungen in jedem Abschnitt berechnet. Nehmen die Sensitivitäten in den äußeren Bereichen ab, können basierend auf einer Relevanzschwelle reduzierte Wertebereiche abgeleitet werden. Die im Anschluss durchgeführte globale Sensitivitätsanalyse erlaubt darüber hinaus einen quantitativen Vergleich der Parametereinflüsse innerhalb der vorher bestimmten relevanten Grenzen. Da eine Szenarienabhängigkeit angenommen wird, erfolgt eine individuelle Anwendung der Methode für jedes Szenario.

Das Verfahren gibt damit nicht nur Auskunft über relevante Wechselwirkungen zwischen Parametern, sondern auch zwischen den Szenarien. Desweiteren stellt es die Wichtigkeit aller Parameter heraus und erlaubt eine Einschränkung des Suchraums basierend auf einflussreichen Wertebereichen. Aufgrund des dadurch gesteigerten Systemverständnisses kann eine zielgerichtete Applikation im Fahrversuch ermöglicht werden. Durch die Verwendung genauer Simulationsumgebungen werden hohe Modellierungsgüten erreicht, sodass eine Nutzung der qualitativen Sensitivitätsinformationen im Fahrversuch möglich ist. Auch der Ressourcenaufwand ist durch die Effizienz der Methode und der Möglichkeit zur Parallelisierung vergleichsweise gering. Die in Kapitel 6 beschriebene umfangreiche Validierungsstudie bestätigt die Gültigkeit des Verfahrens.

**3. Forschungsfrage:** Welcher Optimierungsalgorithmus eignet sich am besten für eine virtuelle automatisierte Applikation und wie können szenarienspezifische Einflussinformationen zur Reduktion der Rechenzeit genutzt werden?

Basierend auf einer hochgenauen, effizienten Simulationsumgebung und den Ergebnissen der Komplexitätsreduktion sieht das Applikationskonzept aus Ab-

bildung 3.1 eine Parameteroptimierung vor. Die dadurch bereit gestellten Lösungen sollen als Grundapplikation dienen und den Anteil von Versuchsfahrten reduzieren. Die fahrzeugbasierte Applikation dient somit nur noch als Werkzeug zur Feinabstimmung unter Einbeziehung des menschlichen Fahrgefühls und läuft damit deutlich ressourcenschonender ab.

Im ersten Teil der Forschungsfrage geht es darum, einen für das beschriebene Problem geeigneten Optimierungsalgorithmus zu finden. Dieser soll eine globale Optimierung und schnelle Konvergenz ermöglichen. Basierend auf Literaturrecherchen erfolgte eine erste Einschränkung in Frage kommender Verfahren. Durch die Eigenschaft des Problems Systemevaluationen durch Simulationsdurchläufe abbilden zu müssen, können gradientenbasierte Verfahren ausgeschlossen werden. Auch Ansätze, die eine lokale Suche durchführen, kommen nicht in Frage. Aufgrund der weiten Verbreitung populationsbasierter Optimierer für vergleichbare Probleme und der guten Performance genetischer Algorithmen und Partikelschwarmoptimierungs-Algorithmen erfolgte ein experimenteller Vergleich beider Verfahren. Dafür wurde ein vereinfachtes, repräsentatives Applikationsproblem definiert und die Ansätze hinsichtlich Optimierungsergebnis und Rechenzeit gegenübergestellt. Obwohl beide Alternativen gute Ergebnisse lieferten und sich prinzipiell zur Lösung des Optimierungsproblems eignen, wurde der genetische Algorithmus aufgrund seines Effizienzvorteils weiter verwendet.

Im zweiten Teil der Frage wird eine Nutzung von Ergebnissen der Komplexitätsreduktion gefordert, um einen effizienten Optimierungsprozess zu gewährleisten. Während die szenarienspezifischen Einflüsse auf der einen Seite ein gesteigertes Systemverständnis für manuelle Applikationen bieten, lassen sich daraus ebenfalls Bereiche ableiten, innerhalb derer Parameteränderungen ohne relevanten Einfluss bleiben. Darauf basierend ergibt sich ein Verfahren, welches die Notwendigkeit von Simulationen jedes vom Optimierer neu generierten Parametersatzes prüft. Wird eine Parametrierung erzeugt, welche von bereits ausgewerteten Parametersätzen lediglich innerhalb nicht sensitiver Bereiche abweicht, kann der Funktionswert von einer Referenzlösung kopiert werden und die zeitaufwändige Simulation entfällt. Die Anwendung der beschriebenen Methodik verspricht eine deutliche Reduktion der Rechenzeit ohne Einschränkung des Optimierungsergebnisses (vgl. Kapitel 5). Auch die in Kapitel 6 vorgestellte Validierung zeigt, dass der Anteil falsch bewerteter Lösungen vernachlässigbar klein ist. Die Ergebnisse bestätigen damit, dass eine



---

sinnvolle Einbeziehung von Sensitivitätsinformationen während der automatisierten Applikation den Ressourcenaufwand weiter senken kann. Obwohl der genetische Algorithmus durch ein vergleichbares populationsbasiertes Verfahren ersetzt werden könnte, zeigt er für die untersuchten Probleme insbesondere in Bezug auf die Recheneffizienz gute Ergebnisse. Die Studien in dieser Arbeit belegen darüber hinaus, dass Robustheitsanalysen zur weiteren Klassifizierung gleich bewerteter Lösungen dienen und Aufschluss über die zu erwartende Performance unter realen Umgebungsbedingungen geben können.

Die Anwendung der Applikationsmethodik auf realitätsnahe Probleme zeigt, dass ein Einsatz im industriellen Entwicklungsprozess grundsätzlich vielversprechend ist. Eine Implementierung des Verfahrens sollte allerdings immer vor dem Hintergrund in der Praxis zu erwartender gesteigerter Dimensionalitäten diskutiert werden. Obwohl sich alle Anwendungsbeispiele auf das in Abbildung 2.2 dargestellte Applikationsproblem zurückführen lassen, ist für die meisten Fahrfunktionen ein deutlich komplexerer Szenarienkatalog und eine gesteigerte Anzahl genauerer Gütefunktionale nötig, um alle relevanten Use-Cases abbilden zu können. Während der Manöverkatalog vergleichsweise einfach erweitert werden kann, lassen sich Objektivierungsmodelle nur mit hohem Aufwand bestimmen (vgl. Abschnitt 2.6). Aufgrund der Komplexität und schwierigen Messbarkeit des menschlichen Fahrgefühls ist deshalb eine Diskrepanz zwischen Simulation und Fahrversuch zu erwarten, deren Kritikalität gesondert bewertet werden muss. Mit größerer Vielfalt des Applikationsproblems steigt ebenfalls die benötigte Rechenzeit simulativer Parameterstudien und -optimierungen, sodass die Effizienz der Methodik mit entsprechend performanterer Simulationshardware sichergestellt werden muss. Auch wenn das hybride Fahrzeugmodell bereits einen ersten Beitrag dazu leistet, erfordert die Anwendung in der Praxis weitere Effizienzsteigerungen der Simulationsmodelle (z.B. durch rechenzeitoptimierte Implementierungen), um kurze Feedback-Schleifen und schnelle Optimierungen weiterhin gewährleisten zu können. Die vom HM bereitgestellte Adaptivität ermöglicht eine kontinuierliche Verbesserung des Modells aufgrund von Messdaten. Als Applikationswerkzeug in der industriellen Entwicklung entsteht dadurch aber auch ein Risiko, dass eine Konvergenz niemals eintritt. Durch die gleichzeitige Weiterentwicklung der Modell- und Applikationsgüte existieren zwei Optimierungsschleifen, deren Fortschritt ständig überwacht werden muss, um eine Konvergenz beider Prozesse sicherzustellen.

Obwohl die vorgestellte Methodik mit der Motivation zur Optimierung fahrzeugbasierter Applikationen von Fahrerassistenzsystemen entwickelt wurde, ist eine Übertragbarkeit auf andere Bereiche der Fahrzeugtechnik problemlos möglich. Der Grund dafür ist, dass die Rahmenbedingungen sich nicht stark voneinander unterscheiden: Das Ziel ist immer eine Optimierung freier Parameter in verschiedenen Situationen im Hinblick auf eine oder mehrere Gütefunktionale. Eine Komplexitätsreduktion anhand der vorgestellten Methode ist allerdings nur unter der Prämisse vielversprechend, dass sich der Suchraum basierend auf Sensitivitätsinformationen einschränken lässt und abfallende Einflüsse in den äußeren Abschnitten des Wertebereichs auftreten. Da sich die Applikation allerdings oft auf Regelsysteme oder Softwaremodule mit den Output begrenzenden Mechanismen bezieht (vgl. Abschnitt 5.3), ist eine ähnliche Charakteristik auch in anderen Bereichen zu erwarten. Darüber hinaus ist der Ansatz einfach erweiterbar, sodass auch Intervalle in der Mitte des Wertebereichs als nicht relevant deklariert werden können.

Eine Anwendung auf beliebige Parameterräume ist auch grundsätzlich möglich, aber unter Umständen nicht zielführend. Insbesondere für Probleme, bei denen keine verringerten Sensitivitäten in Teilbereichen des Suchraums zu erwarten sind oder deren Systemevaluationen keinen großen Rechenaufwand bedeuten, ist eine Nutzung der Methodik nicht zu empfehlen. Von einer allgemeingültigen Anwendbarkeit des Optimierers kann ebenso nicht ausgegangen werden, da die Eignung problemabhängig ist (*No-Free-Lunch-Theorem*, vgl. Abschnitt 2.4.2).

Neben Applikationsproblemen kann die Absicherung automatisierter Fahrfunktionen hinsichtlich Zuverlässigkeit und Sicherheit als vielversprechendes Anwendungsbeispiel genannt werden. Die Rahmenbedingungen sind ähnlich mit dem Unterschied, dass der Fokus in der Absicherung auf der Identifikation von Szenariencharakteristika liegt, welche kritische Use-Cases abbilden. Das hybride Modell kann auch in dem Zusammenhang den geforderten hohen Testaufwand reduzieren. Einen weiteren sinnvollen Anwendungsfall stellt die Parametrierung und Absicherung in der Luftfahrt dar. Durch die im Vergleich zur Fahrzeugtechnik höheren Kosten der Erprobung auf der Zielhardware liegen in dem Bereich bereits zuverlässige Simulationumgebungen vor und eine virtuelle Produktentwicklung ist lange etabliert. Die Rahmenbedingungen für eine Anwendung der vorgestellten Methodik sind also geschaffen, sodass diese im Bereich der Reglerauslegung, Absicherung sicherheitskritischer Situationen

---

oder zur Erhöhung des Systemverständnisses sinnvoll zum Einsatz kommen könnte.



## **8 Zusammenfassung und Ausblick**

Abschließend werden die bisherigen Ausführungen zusammengefasst. Neben einer Wiederholung der Inhalte vorausgegangener Kapitel erfolgt eine Herausstellung des wissenschaftlichen Mehrwerts. In Abschnitt 8.2 werden offene Fragestellungen diskutiert und Anregungen für weitere Forschungsfelder beschrieben.

### **8.1 Zusammenfassung der Arbeit**

In der vorliegenden Arbeit wurde eine Applikationsmethodik zur virtuellen und ressourcenschonenden Applikation automatisierter Fahrfunktionen vorgestellt. Die Analyse des etablierten Prozesses und Bewertung vorliegender Ansätze aus dem Bereich der Antriebsstrangentwicklung führte zur Ableitung des Forschungsbedarfs und Formulierung von drei Forschungsfragen. Diese zielen zum Einen darauf ab, die fahrzeugbasierte Applikation durch eine sinnvolle Einbeziehung geeigneter Simulationsmodelle ressourcenschonender zu gestalten und eine globale Optimierung sicherzustellen. Zum Anderen wurde gefordert, eine Methodik zu entwickeln, welche basierend auf Sensitivitätsanalysen relevante Parameter und Wertebereiche herausstellt und das Systemverständnis damit steigert. Zuletzt sollte ein Konzept zur effizienten simulativen Optimierung und Bereitstellung einer Initialbedatung hergeleitet werden, welches die Ergebnisse der Einflussanalyse nutzt.

Das erarbeitete Konzept sieht vor, ein hochgenaues, effizientes und adaptives sogenanntes hybrides Fahrzeugmodell (HM) zu nutzen, um simulative Komplexitätsreduktionen des Parameterraums und Optimierungen durchzuführen. Die automatisch generierten Ergebnisse werden dann dem Applikateur übergeben, der eine Feinbedatung im Fahrversuch vornimmt. Die fahrzeugbasierte Applikation wird somit nicht vollständig ersetzt, läuft aber ressourcenschonender ab da die übergebene Startapplikation bereits einen hohen Reifegrad

besitzt. Um den begrenzten Kapazitäten im Fahrversuch Rechnung zu tragen, wird jeder neue Applikationsstand simulativ in einem umfangreichen Szenarienkatalog abgesichert und damit die Möglichkeit geschaffen, im Zielsystem nicht beobachtete Effekte frühzeitig aufzudecken. Die im Fahrzeug gewonnenen Messdaten können zur stetigen Verbesserung des Fahrzeugmodells eingesetzt werden, sodass sich mit fortschreitender Entwicklungszeit sowohl der Applikationsstand als auch die Modellgüte verbessert.

Zur Realisierung des Konzepts wurde im ersten Schritt das hybride Fahrzeugmodell als Komposition aus einem Einspurmodell und neuronalem Netz zum Ausgleich von Modellierungsfehlern vorgestellt. Der Ansatz ermöglicht zuverlässige Prädiktionen bei parametrischen Fehlern des ESMs oder konzeptbedingten Grenzen (z.B. im nichtlinearen Dynamikbereich). Eine Validierung des HMs mit repräsentativen Messdaten aus Applikationsfahrten konnte neben einer hohen Generalisierungsfähigkeit deutlich reduzierte Rechenzeiten im Vergleich zu einem ähnlich genauen Mehrkörpersystem herausstellen. Ein sinnvoller Einsatz als Simulationswerkzeug für virtuelle Applikationen konnte somit nachgewiesen werden.

Für die ebenfalls geforderte Einschränkung des Parameterraums wurde ein Verfahren vorgestellt, welches basierend auf einer lokalen Anwendung von Sensitivitätsanalysen einflussreiche Wertebereiche und Parameter in Abhängigkeit von Szenarien herausstellt. Aufgrund der vergleichsweise geringen Anzahl an Systemevaluationen kam dafür die Elementareffektmethode zum Einsatz. Die Anwendung des Verfahrens und Validierung mit einem umfangreichen Resampling konnten das Potential zur Komplexitätsreduktion herausstellen. Die Methodik deckte im ausgewerteten Beispiel Wechselwirkungen zwischen Parametern und Szenarien auf und erhöhte damit das Systemverständnis zur anschließenden (manuellen und automatisierten) Optimierung. Außerdem konnte der relevante Suchraum durch eine Reduktion von Wertebereichen und Ausschluss einzelner Parameter deutlich verkleinert werden, was sich in einer Effizienzsteigerung zur virtuellen Applikation einer Level-3-Fahrfunktion widerspiegelte.

Um die angesprochene Parameteroptimierung unter Nutzung von Sensitivitätsinformationen durchführen zu können, wurde im letzten Teil der Methodik ein Optimierungsansatz vorgestellt, welcher eine Reduktion von Simulationsdurchläufen ohne Einschränkung des Applikationsergebnisses ermöglicht. Dafür kam ein genetischer Algorithmus zum Einsatz, welcher für jeden neu

kreierten Parametersatz prüft ob eine Simulation notwendig ist oder die Lösung von einer zuvor ausgewerteten Referenz übernommen werden kann. Die Anwendung des Verfahrens hat eine deutliche Effizienzsteigerung bei der simulativen Applikation verursacht. Als Ergebnis konnten mehrere gleich bewertete Parametrierungen identifiziert werden, die als Startbedatung zur weiteren Verwendung im Fahrversuch in Frage kommen. Um die Auswahl aber weiter einschränken zu können, wurden die Lösungen noch im Hinblick auf ihre Stabilität gegenüber sich ändernder Fahrzeugdynamik ausgewertet. Die dabei zum Einsatz gekommene Robustheitsanalyse konnte eine weitere Klassifizierung der Lösungen ermöglichen und gleichzeitig die Übertragbarkeit robuster Parametersätze auf das Zielsystem sicherstellen. Im Rahmen der Validierung wurde darüber hinaus nachgewiesen, dass durch die sensitivitätsbasierte Ersetzung von Simulationsdurchläufen entstehende Fehler vernachlässigbar klein sind und keine negative Beeinflussung des Optimierungsfortschritts zu erwarten ist.

Entgegen bestehender Applikationsmethoden wird mit dem in dieser Arbeit beschriebenen Ansatz ein Konzept zur Kombination der simulativen Optimierung und manuellen Parametrierung im Fahrversuch vorgestellt. Aufgrund der im Bereich der Fahrerassistenzsystementwicklung notwendigen Betrachtung der gesamten Wirkkette kann eine isolierte Applikation auf Prüfständen oder in X-in-the-Loop-Umgebungen nur mit Inkaufnahme großer Ungenauigkeiten durchgeführt werden. Mit dem vorgestellten hybriden Modell ist hingegen ein hochgenaues Simulationswerkzeug entstanden, um umfangreiche Optimierungen und Parameterstudien mit vertretbarem Rechenaufwand durchführen zu können. Entgegen bestehender Modellierungsansätze vereint es die Vorteile einfacher analytischer Modelle und neuronaler Netze. Es bietet somit die Möglichkeit, den Trade-Off zwischen der Komplexität und Genauigkeit von Fahrzeugmodellen abzumildern. Auch die im Vergleich zur Motor- und Getriebeapplikation hohe Relevanz fahrzeuggesteuerter Applikationen und der Einbezug des menschlichen Empfindens zur Bewertung des Fahrkomforts finden in der vorgestellten Methode Berücksichtigung. Durch kontinuierliche simulative Absicherung vom Applikateur herausgefahrener Bedatungen mit dem HM, Bereitstellung von Sensitivitätsinformationen und Unterstützung des manuellen Prozesses durch virtuelle Optimierungen kann eine gezieltere Applikation als mit konventionellen Verfahren sichergestellt werden.

Die ebenfalls vorgestellten Methoden zur Komplexitätsreduktion und Optimierung leiten sich zwar aus bestehenden Arbeiten ab, mussten aber für die Applikation automatisierter Fahrfunktionen angepasst werden. Obwohl die Elementareffektmethode als DoE-Verfahren bereits breite Anwendung findet, stellt die Herleitung gültiger Parametergrenzen aufgrund des Verfahrens einen Neuheitswert dar. Im Vergleich zu bestehenden Ansätzen zur ersatzmodellgestützten algebraischen Herleitung von Wertebereichen hat das beschriebene Konzept den Vorteil, auf beliebig komplexe Probleme anwendbar zu sein und vergleichsweise wenig Systemevaluationen zu benötigen. Durch eine Verwendung szenarienspezifischer Sensitivitätsinformationen kann das erhöhte Systemverständnis sowohl im Fahrversuch als auch in der Simulation für schnellere Applikationen und ein verbessertes Konvergenzverhalten sorgen. Die Klassifizierung gleich bewerteter Parametersätze auf Grundlage von Robustheitsanalysen, welche die Stabilität im Hinblick auf sich unter Realbedingungen veränderte Fahrzeugdynamik auswerten, stellt einen weiteren Beitrag der Arbeit dar.

## 8.2 Ausblick

Zur weiteren Vertiefung der behandelten Themen lassen sich mehrere Handlungsfelder ableiten. Das hybride Fahrzeugmodell hat bewiesen, dass durch Kombination eines einfachen Einspurmodells und dynamischen neuronalen Netzes hohe Prädiktionsgüten und geringe Rechenzeiten entstehen. Außerdem bleibt der Modellausgang nachvollziehbar und verhindert den Black-Box-Charakter reiner neuronaler Netze. Es ist zu erwarten, dass der Ansatz auf andere Bereiche übertragbar ist. Insbesondere im Kontext dieser Arbeit könnte ein ähnliches Konzept für die Modellierung der Längsdynamik zum Einsatz kommen und somit den Ressourcenaufwand zur Absicherung fahrzeugbasierter Applikationen weiter senken.

Das Verfahren zur Komplexitätsreduktion geht für den untersuchten Anwendungsfall von reduzierten Einflüssen der Parameter in den äußeren Bereichen des Wertebereichs aus. Für andere Parameterräume ist diese Prämisse möglicherweise nicht zutreffend, sodass eine Erweiterung des Konzepts zu empfehlen ist. Zukünftige Forschungsbemühungen könnten die Methodik so anpassen, dass ein Ausschluss von Intervallen an beliebiger Stelle entlang des



Wertebereichs möglich sind. Die im Rahmen der Definition einflussreicher Bereiche verwendete Relevanzschwelle ist in dieser Arbeit konservativ geschätzt worden. Diese könnte jedoch auch analytisch oder experimentell hergeleitet werden, um den Einfluss des dadurch entstehenden maximalen Fehlers bereits vor Durchführung der Komplexitätsreduktion zu berücksichtigen. Die szenarienabhängigen einflussreichen Regionen werden bereits zur Einsparung von Simulationen im Rahmen der Optimierung verwendet. Es ist aber auch eine weitergehende Verwendung denkbar, indem die Suchstrategie zur Optimierung ebenfalls sensitivitätsbasiert abläuft. Somit könnten die wichtigsten Parameter im Sinne eines hierarchischen Vorgehens unter Berücksichtigung von Wechselwirkungen als Erstes eingestellt werden, bevor die Untersuchung weniger sensibler Faktoren erfolgt. Der Ansatz bietet möglicherweise eine schnellere Konvergenz. Auch im Rahmen der Absicherung fahrzeuggesteuerter Applikationen könnten die Einflussinformationen dazu genutzt werden, um die Anzahl auszuwertender Szenarien zu reduzieren, indem nur die aufgrund von Bedarfsänderungen relevanten Use-Cases evaluiert werden.

Zur automatisierten Applikation wurde eine sensitivitätsbasierte Optimierung mit Hilfe des genetischen Algorithmus (GA) vorgeschlagen. Wie zuvor erwähnt begünstigt der GA die Anwendung der Methode aufgrund einer Diskretisierung des Wertebereichs und damit Beschränkung der theoretisch möglichen Punkte im Suchraum. Für weitere Probleme könnte jedoch ein anderer Algorithmus die beste Performance liefern (*No-Free-Lunch-Theorem*), der eine kontinuierliche Skalierung verwendet. In diesem Fall wäre es ratsam, das vorgestellte Verfahren so zu erweitern, dass zur Ersetzung von Parametersätzen mit Referenzsätzen die Abweichung im sensitiven Bereich unterhalb einer Schwelle und nicht, wie in der jetzigen Ausführung gefordert, null sein muss. Für die Ableitung eines maximal erlaubten Hyperbox-Radiuses müsste ein Ansatz entwickelt werden, der einer negativen Beeinflussung des Optimierungsergebnisses vorbeugt. In manchen Fällen ist es außerdem nicht möglich oder ratsam, multikriterielle in einkriterielle Probleme umzuwandeln. Damit entsteht ein weiterer Forschungsbedarf in der Anpassung des Verfahrens, sodass auch Pareto-Optimierungen möglich sind.



# A Anhang

## A.1 Dreistufige Applikationsmethodik

### A.1.1 Hybrides Fahrzeugmodell

Tabelle A.1: Spezifikation des Trainingsdatensatzes [FFFG19b]

Parameter	Wert
Laterale Beschleunigungen $\hat{a}_y [m/s^2]$	[0,5; 1,5; 2,5; 3,5; 4,5; 5,5]
Frequenzen $f [1/s]$	[0,1; 0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; 0,9; 1]
Geschwindigkeiten $v [m/s]$	[8,33; 13,89; 22,22; 27,78; 33,33; 41,67]
Sampling-Rate der Zeitreihen $[Hz]$	25
Länge jeder Zeitreihe $[s]$	10

Tabelle A.2: Parametrierung des Einspurmodells [FFFG19b]

Parameter	Wert
Masse $m$	2060 kg
Schräglaufsteifigkeit Vorderachse $c_{\alpha,f}$	75000 N/rad
Schräglaufsteifigkeit Hinterachse $c_{\alpha,r}$	150000 N/rad
Abstand Vorderachse - Schwerpunkt $l_f$	1,66 m
Abstand Hinterachse - Schwerpunkt $l_r$	1,55 m
Gierträgheit $\theta_z$	3832 $kgm^2$

Tabelle A.3: Hyperparameter des LSTM-Netzes [FFFG19b]

Parameter	Wert
Anzahl verdeckter Schichten	1
Anzahl Neuronen	32
Aktivierungsfunktion	Sigmoid
Optimierer	Stochastic gradient descent (SGD)
Verlustfunktion	Mean absolute error (MAE)
Lernrate	0,01
Frequenz des Krümmungssignals	25
Anzahl an Zeitschritten, die zur Prädiktion genutzt werden	25 (entspricht 1 s)
Anzahl an Trainingsepochen im Open-Loop-Setup	300
Anzahl an Trainingsepochen im Closed-Loop-Setup	200

### A.1.2 Simulative Optimierung von Applikationsparametern

Tabelle A.4: Parametrierung des GA und der PSO für den experimentellen Vergleich beider Verfahren (abgeleitet aus [Ped10, Tre03, HM90, BS04, PCM14])

Parameter		Wert
PSO	$c_1$	0,5
	$c_2$	0,5
GA	$p_s$	0,8
	$p_r$	0,7
	$p_m$	0,2

## A.2 Anwendung der Methodik für die Applikation einer automatisierten Fahrfunktion

### A.2.1 Komplexitätsreduktion

Tabelle A.5: Initiale Parametergrenzen für die Anwendung der Komplexitätsreduktion [FFFG20]

Parameter	Untere Grenze	Obere Grenze
$P_1$	-15	14
$P_2$	0	2
$P_3$	0	200
$P_4$	0	2000
$P_5$	0	1500
$P_6$	0	1
$P_7$	0	200
$P_8$	0	2000
$P_9$	0	10
$P_{10}$	0	20
$P_{11}$	0	200

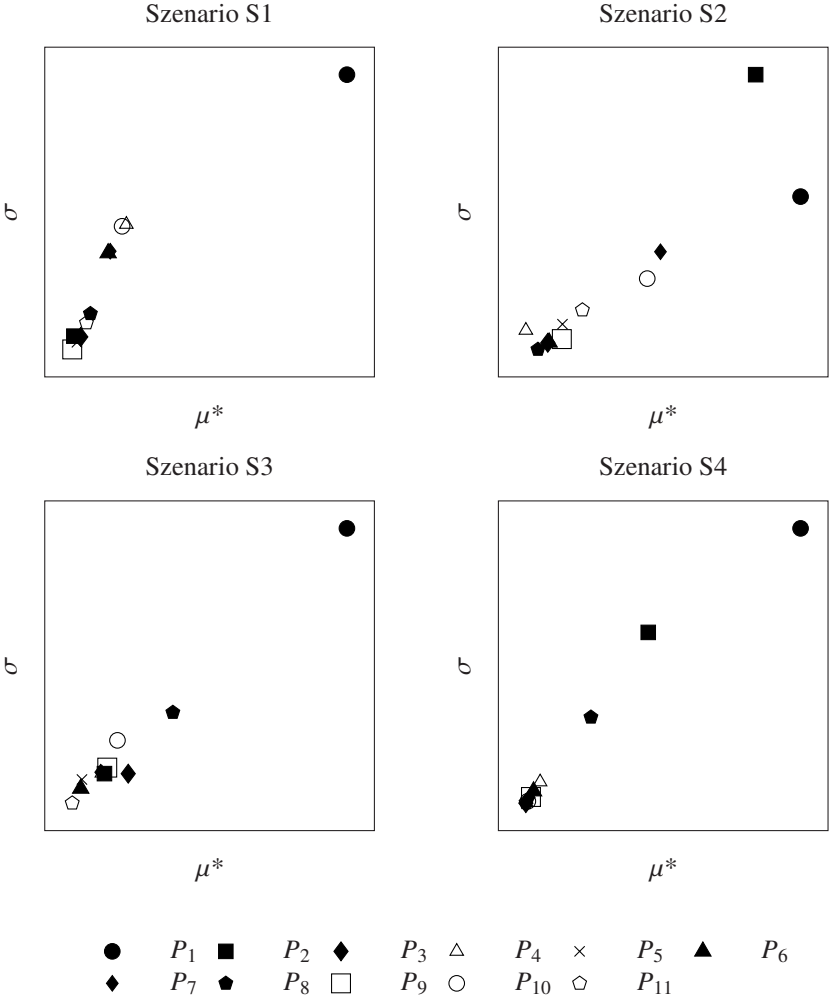


Abbildung A.1:  $\mu^*$  -  $\sigma$ -Diagramme für Szenarien 1-4

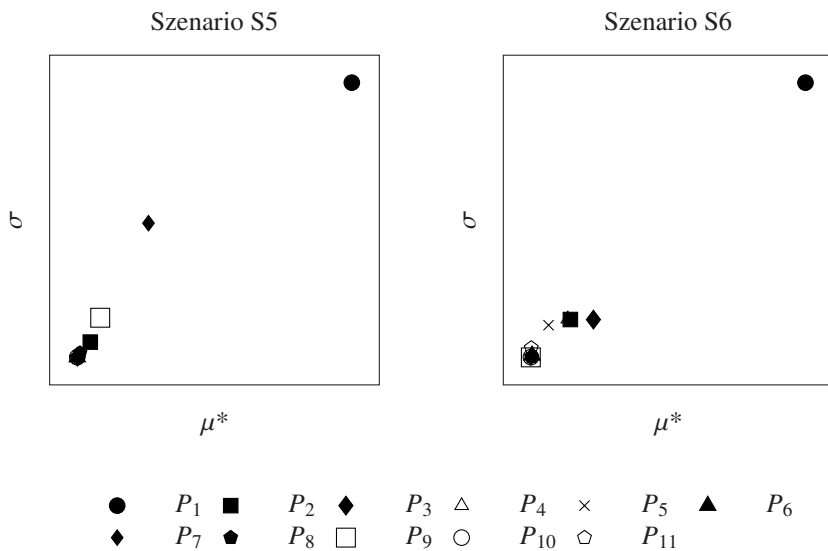


Abbildung A.2:  $\mu^* - \sigma$ -Diagramme für Szenarien 5 und 6

## A.2.2 Optimierung

Tabelle A.6: Diskretisierung der Wertebereiche für die Darstellung als Binärstrang zur Anwendung des GA [FFFG20]

Parameter	Diskretisierung	$n_{P_i}^{min}$	$k_i$
$P_1$	0,5	58	2
$P_2$	0,01	200	8
$P_3$	1	200	8
$P_4$	1	2000	11
$P_5$	1	1500	11
$P_6$	0,01	100	7
$P_7$	0	200	8
$P_8$	0	2000	11
$P_9$	0	100	7
$P_{10}$	0	200	8
$P_{11}$	0	200	8

## A.2.3 Robustheitsanalyse



Tabelle A.7: Normierte Parameterwerte der elf besten Parametersätze und zugehörige Robustheitsgrade [FFFG20]

Parametersatz	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$	$P_8$	$P_9$	$P_{10}$	$P_{11}$	$DoR$
1	0,175	0,355	0,2	0,724	0	0,133	0,478	0,746	0,645	0	0,78	0
2	0,413	0,968	0,043	0,374	0,333	0,6	0,545	0,1	0,129	0,419	0,988	0
3	0,27	0,516	0,765	0,89	0,933	0,8	0,957	0,084	0,871	0,129	0,816	1
4	0,413	0,064	0,22	0,374	0,333	0,6	0,545	0,1	0,129	0,419	0,988	2
5	0,175	0,968	0,8	0,931	0,733	0,267	0,945	0,408	0,484	0,032	1	2
6	0,413	1	0,031	0,374	0,333	0,6	0,545	0,1	0,129	0,419	0,988	4
7	0,413	1	0,094	0,374	0,333	0,6	0,545	0,1	0,129	0,419	0,988	$\geq 10$
8	0,413	1	0,098	0,374	0,333	0,6	0,545	0,1	0,129	0,419	0,988	$\geq 10$
9	0,413	0,935	0,094	0,374	0,333	0,6	0,545	0,1	0,129	0,419	0,988	$\geq 10$
10	0,413	1	0,282	0,374	0,333	0,6	0,545	0,1	0,129	0,419	0,988	$\geq 10$
11	0,413	1	0,11	0,374	0,333	0,6	0,545	0,1	0,129	0,419	0,988	$\geq 10$

## **A.3 Validierung**

### **A.3.1 Validierungskonzept**

Tabelle A.8: Charakteristiken des Testdatensatzes zur Validierung des hybriden Fahrzeugmodells [FFFG19b]

	Maximale Krümmung [ $\frac{1}{m}$ ]	Maximale Querbesehleumigung [ $\frac{1}{s^2}$ ]	Geschwindigkeitsprofil [ $\frac{m}{s}$ ]	Dauer [s]
T1	0,005	2,293	21,58; konstant	18
T2	0,0114	2,018	13,385; konstant	16
T3	0,0034	2,3	27,11; konstant	14
T4	0,0033	2,418	Verzögerung von 27 auf 16	13
T5	0,0014	1,647	34,5; konstant	12
T6	0,0018	1,655	Verzögerung von 41 auf 25	20
T7	0,0013	0,983	Verzögerung von 42 auf 22	22
T8	0,0047	2,2	21,66; konstant	18
T9	0,0048	2,16	21,48; konstant	16
T10	0,0044	2,07	21,66; konstant	19
T11	0,0145	7,018	22; konstant	53
T12	0,0091	2,33	20 in Geraden und ca. 16 in Kurven	600

Tabelle A.9: Grenzwerte für die Variation von ESM-Modellparametern zur Robustheitsanalyse abgeleitet aus [MW14] [FFFG20]

Parameter	Minimum	Maximum
Masse $m$	1200 kg	2600 kg
Schräglaufsteifigkeit Vorderachse $c_{\alpha,f}$	55000 N/rad	90000 N/rad
Schräglaufsteifigkeit Hinterachse $c_{\alpha,r}$	120000 N/rad	180000 N/rad
Abstand Vorderachse - Schwerpunkt $l_f$	1 m	2 m
Abstand Hinterachse - Schwerpunkt $l_r$	1 m	2 m
Gierträgheit $\theta_z$	2000 $kgm^2$	4500 $kgm^2$

## A.3.2 Ergebnisse

### Validierung des hybriden Fahrzeugmodells

Tabelle A.10: RMSE-Werte der Prädiktion aller vier betrachteten Fahrzeugmodelle [FFFG19b]

Test-Datensatz	RMSE ESM	RMSE iso- liertes LSTM	RMSE ISAR- Modell	RMSE HM
T1	0,00079	0,00034	0,00021	0,00019
T2	0,00107	0,00046	0,00034	0,00037
T3	0,00089	0,00048	0,00041	0,00041
T4	0,00046	0,00012	0,00008	0,00008
T5	0,00068	0,00017	0,00017	0,00011
T6	0,00051	0,00018	0,00018	0,00021
T7	0,00089	0,00037	0,00034	0,00039
T8	0,00108	0,00035	0,00029	0,0003
T9	0,00119	0,00111	0,00076	0,00056
T10	0,00101	0,00054	0,00047	0,00051
T11	0,00256	0,00042	0,00041	0,00031
T12	0,00157	0,00046	0,00035	0,00031

Tabelle A.11: Vergleich der Rechenzeiten von ESM, ISAR-Modell und HM bei 30-maliger Resimulation jedes Szenarios.

Test-Datensatz	ESM			ISAR			HM		
	Min.	Max.	Mittelw.	Min.	Max.	Mittelw.	Min.	Max.	Mittelw.
T1	0,355	0,993	0,626	9,127	17,515	14,574	1,203	1,417	1,293
T2	0,385	1,003	0,581	8,981	20,156	14,218	1,187	1,617	1,283
T3	0,375	1,028	0,59	8,487	17,7	13,048	1,241	1,501	1,341
T4	0,361	1,321	0,579	6,015	15,314	10,544	0,688	0,837	0,767
T5	0,407	1,048	0,561	6,139	15,172	10,331	0,725	0,96	0,827
T6	0,35	0,889	0,65	9,697	20,845	15,871	1,163	1,324	1,246
T7	0,408	0,933	0,59	10,474	22,358	16,288	1,408	1,64	1,508
T8	0,401	0,909	0,651	9,268	19,173	14,276	1,315	1,574	1,421
T9	0,418	1,265	0,551	7,703	14,069	10,504	1,428	1,8	1,512
T10	0,426	1,055	0,546	10,889	21,55	15,681	1,452	1,732	1,561
T11	1,536	2,471	2,127	21,167	56,861	37,981	2,457	3,051	2,776
T12	18,382	29,403	23,938	510,857	521,676	516,073	29,518	37,03	32,367

## Validierung der sensitivitätsbasierten Optimierung

Tabelle A.12: Häufigkeitsverteilung der Abweichung von Zielfunktionswerten vom jeweiligen Basis-Sample.

	$\Delta f(x)$ = -5	$\Delta f(x)$ = -4	$\Delta f(x)$ = -3	$\Delta f(x)$ = -2	$\Delta f(x)$ = -1	$\Delta f(x)$ = 0	$\Delta f(x)$ = 1	$\Delta f(x)$ = 2
S1	0	1	0	3	17	4325	21	9
S2	3	0	0	2	2	4366	2	0
S3	17	0	0	0	0	4358	0	0
S4	0	1	80	101	17	24880	18	103
S5	0	1	0	0	0	51174	0	0
S6	0	0	0	3	0	25572	0	0

## Robustheitsanalyse gegenüber Fahrzeugmodellparametern

Tabelle A.13: Vergleich der Robustheitsgrade der Robustheitsanalysen gegenüber Applikationsparametern  $DoR_{AP}$  und Fahrzeugmodellparametern  $DoR_{RESM}$  [FFFG20]

Parametersatz	$DoR_{AP}$	$DoR_{RESM}$
1	0	0
2	0	0
3	1	2
4	2	0
5	2	$\geq 10$
6	4	4
7	$\geq 10$	0
8	$\geq 10$	$\geq 10$
9	$\geq 10$	$\geq 10$
10	$\geq 10$	$\geq 10$
11	$\geq 10$	$\geq 10$





# Abbildungsverzeichnis

1.1	Markteinführung von Fahrerassistenzsystemen . . . . .	2
1.2	Aufbau der Arbeit. . . . .	5
2.1	V-Modell für die Entwicklung von Fahrzeugsystemen . . . . .	9
2.2	Applikationsprozess für automatisierte Fahrfunktionen . . . . .	12
2.3	Wirkkette für automatisierte Fahrfunktionen . . . . .	13
2.4	Prozesskette für die modellgestützte Motorapplikation . . . . .	18
2.5	Überblick X-in-the-Loop-Verfahren . . . . .	22
2.6	Sampling-Schema der Elementareffektmethode . . . . .	28
2.7	$\mu^* - \sigma$ -Diagramm der Elementareffektmethode . . . . .	30
2.8	Optimierungsproblem der Applikation von FAS . . . . .	37
2.9	Partikelbewegung bei der Partikelschwarmoptimierung . . . . .	41
2.10	Evoluationäre Operationen des genetische Algorithmus . . . . .	44
2.11	Schematischer Ablauf von Robustheitsanalysen . . . . .	45
2.12	Komplexitäts-Genauigkeits-Trade-Off von Fahrzeugmodellen . . . . .	48
2.13	Geometrie des Einspurmodells . . . . .	50
2.14	Aufbau des ISAR-Modells . . . . .	52
2.15	Aufbau einer LSTM-Zelle . . . . .	57
3.1	Konzept der Applikationsmethodik . . . . .	72
4.1	Aufbau des hybriden Fahrzeugmodells . . . . .	77
4.2	Trainings- und Validierungskonzept des hybriden Modells . . . . .	80
4.3	Zweistufiges Trainingsschema des hybriden Modells . . . . .	82
4.4	Beispielhafter Versuchsplan zur Ableitung gültiger Wertebereiche . . . . .	87
4.5	Berechnung der kombinierten Sensitivität im $\mu^* - \sigma$ -Diagramm . . . . .	88
4.6	Methodik zur Identifikation einflussreicher Parametergrenzen . . . . .	90
4.7	Optimierungsergebnisse der Partikelschwarmoptimierung . . . . .	99

4.8	Optimierungsergebnisse für den genetischen Algorithmus . . . . .	100
4.9	Vergleich der Reproduzierbarkeit der Anwendung des GA . . . . .	101
4.10	Beispiel für Parametersätze die für eine Ersetzung von Simulationen qualifiziert sind . . . . .	104
4.11	Ablauf der Methode zur sensitivitätsbasierten Optimierung . . . . .	105
5.1	Sensitive Wertebereiche jedes Parameters . . . . .	112
5.2	Relative Sensitivitäten jedes Parameters . . . . .	114
5.3	Einflussreiche Parameterräume (grau) der Szenarien S1 bis S4 . . . . .	115
5.4	Einflussreiche Parameterräume (grau) der Szenarien S1 bis S4 . . . . .	116
5.5	Minimaler, maximaler und gemittelter Funktionswert der Individuen-Population über die Iterationen . . . . .	117
5.6	Anzahl gleichwertiger Parametersätze zu der besten Lösung . . . . .	118
5.7	Ergebnisse der Robustheitsanalyse optimaler Lösungen . . . . .	120
5.8	Rechenzeitbetrachtungen der sensitivitätsbasierten Optimierung . . . . .	122
6.1	Qualitativer Verlauf der Sollkrümmung im Testdatensatz . . . . .	128
6.2	Beispielhafte Effekte zur Berücksichtigung in der Validierung . . . . .	131
6.3	Vergleich der Prädiktionsgüten verschiedener Fahrzeugmodelle . . . . .	134
6.4	Vergleich der Krümmungsverläufe des ESM, LSTM und HM . . . . .	135
6.5	Vergleich der Rechenzeiten aller Modelle für die Test-Datensätze . . . . .	137
6.6	Abweichungen der Zielfunktionswerte für die Validierung . . . . .	139
6.7	Robustheitsanalyse gegenüber ESM-Modellparametern . . . . .	142
6.8	HM im Komplexitäts-Genauigkeits-Trade-Off-Diagramm . . . . .	143
A.1	$\mu^* - \sigma$ -Diagramme für Szenarien 1-4 . . . . .	164
A.2	$\mu^* - \sigma$ -Diagramme für Szenarien 5 und 6 . . . . .	165

# Tabellenverzeichnis

2.1	Komfortskala für das <i>Dynamic-Time-Warping</i> -Modell . . . . .	60
6.1	Sample-Parameter für die Validierungsstudie . . . . .	138
6.2	Häufigkeitsverteilung von Zielfunktionsabweichungen der Validierungsstudie . . . . .	139
A.1	Spezifikation des Trainingsdatensatzes . . . . .	161
A.2	Parametrierung des Einspurmodells . . . . .	161
A.3	Hyperparameter des LSTM-Netzes . . . . .	162
A.4	Parametrierung des GA und der PSO für den experimentellen Vergleich beider Verfahren . . . . .	162
A.5	Initiale Parametergrenzen für die Komplexitätsreduktion . . . . .	163
A.6	Diskretisierung der Wertebereiche für die Anwendung des GA . . . . .	166
A.7	Parameterwerte und Robustheitsgrade der elf besten Parametersätze . . . . .	167
A.8	Eigenschaften des Testdatensatzes zur Validierung des HMs . . . . .	169
A.9	Grenzen der ESM-Modellparameter zur Robustheitsanalyse . . . . .	170
A.10	RMSE-Werte aller vier betrachteten Fahrzeugmodelle . . . . .	171
A.11	Vergleich der Rechenzeiten von ESM, ISAR-Modell und HM . . . . .	172
A.12	Zielfunktionsabweichungen der Validierungsstudie . . . . .	173
A.13	Robustheitsgrade gegenüber ESM-Parametern . . . . .	173



# Literaturverzeichnis

- [AEK09] ALADAG, Cagdas H. ; EGRIOGLU, Erol ; KADILAR, Cem: Forecasting nonlinear time series with a hybrid methodology. In: *Applied Mathematics Letters* 22 (2009), Nr. 9, S. 1467–1470
- [Aga97] AGARWAL, Mukul: Combining neural and conventional paradigms for modelling, prediction and control. In: *International Journal of Systems Science* 28 (1997), Nr. 1, S. 65–81
- [AGR<sup>+</sup>16] ALAHI, Alexandre ; GOEL, Krarthar ; RAMANATHAN, Vignesh ; ROBICQUET, Alexandre ; FEI-FEI, Li ; SAVARESE, Silvio: Social lstm: Human trajectory prediction in crowded spaces. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, S. 961–971
- [Ahl19] AHLWEDE, Andreas: Statistiken zu autonomen Fahrzeugen <https://de.statista.com/themen/2561/autonome-fahrzeuge/>, 2019
- [ALF17] ALTCHÉ, Florent ; LA FORTELLE, Arnaud de: An LSTM network for highway trajectory prediction. In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC) IEEE*, 2017, S. 353–359
- [BA06] BARRICO, Carlos ; ANTUNES, Carlos H.: Robustness analysis in multi-objective optimization using a degree of robustness concept. In: *2006 IEEE International Conference on Evolutionary Computation IEEE*, 2006, S. 1887–1892
- [BABPA18] BIJAN, Mahmud G. ; AL-BADRI, Maher ; PILLAY, Pragasen ; ANGERS, Pierre: Induction machine parameter range constraints in genetic algorithm based efficiency estimation techniques. In: *IEEE Transactions on Industry Applications* 54 (2018), Nr. 5, S. 4186–4197
- [Bau15] BAUR, Mathias: *Modellierungsrahmen für Intelligente Verkehrssysteme zur simulationsbasierten Analyse ihrer Wirkun-*

- gen auf den Straßenverkehr*. München, Technische Universität München, Diss., 2015
- [BBB+06] BARTELT, AC ; BAUER, O ; BENEKEN, G ; BERGNER, K ; BIROWICZ, U ; BLIß, T ; BAYERN, F: V-Modell XT Das deutsche Referenzmodell für Systementwicklungsprojekte. (2006)
- [BBVB+01] BECK, Kent ; BEEDLE, Mike ; VAN BENNEKUM, Arie ; COCKBURN, Alistair ; CUNNINGHAM, Ward ; FOWLER, Martin ; GRENNING, James ; HIGHSMITH, Jim ; HUNT, Andrew ; JEFFRIES, Ron u. a.: Manifesto for agile software development. (2001)
- [BDVCB09] BRAHMI, El H. ; DENIS-VIDAL, Liliane ; CHERFI, Zohra ; BOUDAUD, Nassim: Statistical modeling and optimization for diesel engine calibration. In: *2009 35th Annual Conference of IEEE Industrial Electronics IEEE*, 2009, S. 1770–1775
- [Bec03] BECK, Kent: *Test-driven development: by example*. Addison-Wesley Professional, 2003
- [Ber14] BERG, Guy: *Das Vehicle in the Loop: Ein Werkzeug für die Entwicklung und Evaluation von sicherheitskritischen Fahrerassistenzsystemen*. München, Universitätsbibliothek der Universität der Bundeswehr München, Diss., 2014
- [Boc08] BOCK, Thomas: *Vehicle in the loop: Test-und Simulationsumgebung für Fahrerassistenzsysteme*. Cuvillier Verlag, 2008
- [Bog13] BOGON, Tjorben: *Agentenbasierte Schwarmintelligenz*. Springer-Verlag, 2013
- [BS04] BOYABATLI, Onur ; SABUNCUOGLU, Ihsan: Parameter selection in genetic algorithms. In: *Journal of Systemics, Cybernetics and Informatics* 4 (2004), Nr. 2, S. 78
- [BS05] BRANKE, Jürgen ; SCHMIDT, Christian: Faster convergence by means of fitness estimation. In: *Soft Computing* 9 (2005), Nr. 1, S. 13–20
- [BZ17] BREUER, Claus ; ZIMA, Stefan: Geschichtlicher Rückblick. In: *Handbuch Verbrennungsmotor*. Springer, 2017, S. 1–8
- [C+14] COMMITTEE, SAE On-Road Automated Vehicle S. u. a.: Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems. In: *SAE Standard J 3016* (2014), S. 1–16

- [Cao08] CAO, Ming: Applications of artificial neural networks on ground vehicle systems-modeling and diagnostics: A literature survey. In: *ASME 2008 International Mechanical Engineering Congress and Exposition* American Society of Mechanical Engineers, 2008, S. 407–425
- [CB92] CHEN, SABS ; BILLINGS, SA: Neural networks for nonlinear dynamic system modelling and identification. In: *International journal of control* 56 (1992), Nr. 2, S. 319–346
- [CCS07] CAMPOLONGO, Francesca ; CARIBONI, Jessica ; SALTELLI, Andrea: An effective screening design for sensitivity analysis of large models. In: *Environmental modelling & software* 22 (2007), Nr. 10, S. 1509–1518
- [CLVV<sup>+</sup>07] COELLO, Carlos A C. ; LAMONT, Gary B. ; VAN VELDHIJZEN, David A. u. a.: *Evolutionary algorithms for solving multi-objective problems*. Bd. 5. Springer, 2007
- [CR03] CERONE, Vito ; REGRUTO, Diego: Parameter bounds for discrete-time Hammerstein models with bounded output errors. In: *IEEE transactions on Automatic Control* 48 (2003), Nr. 10, S. 1855–1860
- [CR06] CERONE, Vito ; REGRUTO, Diego: Parameter bounds evaluation of Wiener models with noninvertible polynomial nonlinearities. In: *Automatica* 42 (2006), Nr. 10, S. 1775–1781
- [CRW10] CANDITT, Sabine ; RAUH, Doris ; WITTMANN, Marion: Brückenschlag: Das V-Modell XT mit Scrum inside. In: *Objekt Spektrum* (2010), Nr. 5, S. 54
- [CS09] CHIANG, Chia-Jui ; STEFANOPOULOU, Anna G.: Sensitivity analysis of combustion timing of homogeneous charge compression ignition gasoline engines. In: *Journal of Dynamic Systems, Measurement, and Control* 131 (2009), Nr. 1, S. 014506
- [CSC11] CAMPOLONGO, Francesca ; SALTELLI, Andrea ; CARIBONI, Jessica: From screening to quantitative sensitivity analysis. A unified approach. In: *Computer Physics Communications* 182 (2011), Nr. 4, S. 978–988
- [CSV09] CONN, Andrew R. ; SCHEINBERG, Katya ; VICENTE, Luis N.: *Introduction to derivative-free optimization*. Bd. 8. Siam, 2009

- [CZD15] CHEN, Kai ; ZHOU, Yi ; DAI, Fangyan: A LSTM-based method for stock returns prediction: A case study of China stock market. In: *2015 IEEE International Conference on Big Data (Big Data)* IEEE, 2015, S. 2823–2824
- [CZL<sup>+</sup>15] CHAI, Zhi-gang ; ZHANG, Fu-jun ; LIU, Bo-lan ; HUANG, Ying ; FU, Geng: Application Study of Kriging Model in Common Rail Diesel Engine Calibration. In: *Transactions of Beijing Institute of Technology* (2015), Nr. 3, S. 5
- [Deb01] DEB, Kalyanmoy: *Multi-objective optimization using evolutionary algorithms*. Bd. 16. John Wiley & Sons, 2001
- [Deb14] DEB, Kalyanmoy: Multi-objective optimization. In: *Search methodologies*. Springer, 2014, S. 403–449
- [Dec09] DECKER, Medon: *Zur Beurteilung der Querdynamik von Personenkraftwagen*. München, Technische Universität München, Diss., 2009
- [DG05] DEB, Kalyanmoy ; GUPTA, Himanshu: Searching for robust Pareto-optimal solutions in multi-objective optimization. In: *International Conference on Evolutionary Multi-Criterion Optimization* Springer, 2005, S. 150–164
- [Die] DIETMAYER, Klaus: *Automatisiertes Fahren und Fahrerassistenz*. <https://www.uni-ulm.de/in/mrm/forschung/anwendungsfelder/automatisiertes-fahren-fahrerassistenz/>. – abgerufen am 16.08.2019
- [DK01] DURALI, Mohammad ; KASSAIEZADEH, Ali R.: A Neural Network approximation of nonlinear car model using ADAMS simulation results / SAE Technical Paper. 2001. – Forschungsbericht
- [Dor18] DORRER, Claus: Automated driving at BMW–Solutions for today and tomorrow. In: *18. Internationales Stuttgarter Symposium* Springer, 2018, S. 859–871
- [Dou95] DOUGHERTY, Mark: A review of neural networks applied to transport. In: *Transportation Research Part C: Emerging Technologies* 3 (1995), Nr. 4, S. 247–260
- [EGP93] EL-GINDY, M ; PALKOVICS, L: Possible application of artificial neural networks to vehicle dynamics and control: a literature



- review. In: *International Journal of Vehicle Design* 14 (1993), Nr. 5-6, S. 592–614
- [EHG05] ELBELTAGI, Emad ; HEGAZY, Tarek ; GRIERSON, Donald: Comparison among five evolutionary-based optimization algorithms. In: *Advanced engineering informatics* 19 (2005), Nr. 1, S. 43–53
- [EK95a] EBERHART, Russell ; KENNEDY, James: A new optimizer using particle swarm theory. In: *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science Ieee*, 1995, S. 39–43
- [EK95b] EBERHART, Russell ; KENNEDY, James: Particle swarm optimization. In: *Proceedings of the IEEE international conference on neural networks* Bd. 4 Citeseer, 1995, S. 1942–1948
- [ESJ15] ELBANHAWI, Mohamed ; SIMIC, Milan ; JAZAR, Reza: In the passenger seat: investigating ride comfort measures in autonomous cars. In: *IEEE Intelligent Transportation Systems Magazine* 7 (2015), Nr. 3, S. 4–17
- [F<sup>+</sup>15] FU, Michael C. u. a.: *Handbook of simulation optimization*. Bd. 216. Springer, 2015
- [FMB<sup>+</sup>09] FÜRST, Simon ; MÖSSINGER, Jürgen ; BUNZEL, Stefan ; WEBER, Thomas ; KIRSCHKE-BILLER, Frank ; HEITKÄMPER, Peter ; KINKELIN, Gerulf ; NISHIKAWA, Kenji ; LANGE, Klaus: AUTOSAR—A Worldwide Standard is on the Road. In: *14th International VDI Congress Electronic Systems for Vehicles, Baden-Baden* Bd. 62, 2009, S. 5
- [GD91] GOLDBERG, David E. ; DEB, Kalyanmoy: A comparative analysis of selection schemes used in genetic algorithms. In: *Foundations of genetic algorithms* Bd. 1. Elsevier, 1991, S. 69–93
- [GDS14] GARG, Anant B. ; DIWAN, Parag ; SAXENA, Mukesh: Artificial neural networks for internal combustion engine performance and emission analysis. In: *International Journal of Computer Applications* 87 (2014), Nr. 6
- [GKF<sup>+</sup>17] GROH, Korbinian ; KUEHBECK, Thomas ; FLEISCHMANN, Benjamin ; SCHIEMENTZ, Mark ; CHIBELUSHI, Claude: Towards a Scenario-Based Assessment Method for Highly Automated Driving Functions. (2017)

- [GKK13] GERDES, Ingrid ; KLAWONN, Frank ; KRUSE, Rudolf: *Evolutionäre Algorithmen: Genetische Algorithmen-Strategien und Optimierungsverfahren- Beispielanwendungen*. Springer-Verlag, 2013
- [GLF<sup>+</sup>08] GRAVES, Alex ; LIWICKI, Marcus ; FERNÁNDEZ, Santiago ; BERTOLAMI, Roman ; BUNKE, Horst ; SCHMIDHUBER, Jürgen: A novel connectionist system for unconstrained handwriting recognition. In: *IEEE transactions on pattern analysis and machine intelligence* 31 (2008), Nr. 5, S. 855–868
- [GSK<sup>+</sup>16] GREFF, Klaus ; SRIVASTAVA, Rupesh K. ; KOUTNÍK, Jan ; STEUNEBRINK, Bas R. ; SCHMIDHUBER, Jürgen: LSTM: A search space odyssey. In: *IEEE transactions on neural networks and learning systems* 28 (2016), Nr. 10, S. 2222–2232
- [GVRP17] GIFFI, C ; VITALE, J ; ROBINSON, R ; PINGITORE, G: What's ahead for fully autonomous driving: Consumer opinions on advanced vehicle technology. In: *Deloitte's Global Automotive Consumer Study* (2017)
- [H<sup>+</sup>92] HOLLAND, John H. u. a.: *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992
- [H<sup>+</sup>09] HAYKIN, Simon S. u. a.: *Neural networks and learning machines/Simon Haykin*. New York: Prentice Hall,, 2009
- [Haa04] HAASE, Dirk: Ein neues Verfahren zur modellbasierten Prozessoptimierung auf der Grundlage der statistischen Versuchsplanung am Beispiel eines Ottomotors mit elektromagnetischer Ventilsteuerung (EMVS). (2004)
- [Haf02] HAFNER, Michael: Modellbasierte stationäre und dynamische Optimierung von Verbrennungsmotoren am Motorenprüfstand unter Verwendung neuronaler Netze. In: *at-Automatisierungstechnik Methoden und Anwendungen der Steuerungs-, Regelungs-und Informationstechnik* 50 (2002), Nr. 4/2002, S. 192
- [HCDWV05] HASSAN, Rania ; COHANIM, Babak ; DE WECK, Olivier ; VENTER, Gerhard: A comparison of particle swarm optimization and the genetic algorithm. In: *46th AIAA/ASME/ASCE/AHS/ASC*

- structures, structural dynamics and materials conference*, 2005, S. 1897
- [Hen15] HENN, T: Methodenentwicklung zur modellbasierten Applikation von Fahrdynamikregelsystemen am Beispiel der adaptiven Dämpfung. In: *Entscheidungen beim Übergang in die Elektromobilität*. Springer, 2015, S. 239–251
- [Hen18] HENN, Timo: *Modellbasierte Applikation von Fahrdynamiksystemen am Beispiel der adaptiven Dämpfung*. Duisburg, Universität Duisburg-Essen, Diss., 2018
- [HKRW13] HERMAN, JD ; KOLLAT, JB ; REED, PM ; WAGENER, T: Method of Morris effectively reduces the computational demands of global sensitivity analysis for distributed watershed models. In: *Hydrology and Earth System Sciences* 17 (2013), Nr. 7, S. 2893–2903
- [HLT<sup>+</sup>19] HORVÁTH, Márton T. ; LU, Qiong ; TETTAMANTI, Tamás ; TÖRÖK, Árpád ; SZALAY, Zsolt: Vehicle-In-The-Loop (VIL) and Scenario-In-The-Loop (SCIL) Automotive Simulation Concepts from the Perspectives of Traffic Simulation and Traffic Control. In: *Transport and Telecommunication Journal* 20 (2019), Nr. 2, S. 153–161
- [HLV98] HERRERA, FRANCISCO ; LOZANO, Manuel ; VERDEGAY, Jose L.: Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. In: *Artificial intelligence review* 12 (1998), Nr. 4, S. 265–319
- [HM90] HESSER, Jürgen ; MÄNNER, Reinhard: Towards an optimal mutation probability for genetic algorithms. In: *International Conference on Parallel Problem Solving from Nature* Springer, 1990, S. 23–32
- [HNHI99] HOLZMANN, Henning ; NELLES, O ; HALFMANN, C ; ISERMANN, R: Vehicle dynamics simulation based on hybrid modeling. In: *1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (Cat. No. 99TH8399)* IEEE, 1999, S. 1014–1019
- [HNR19] HIMANEN, Veli ; NIJKAMP, Peter ; REGGIANI, Aura: *Neural networks in transport applications*. Routledge, 2019

- [Hol92] HOLLAND, John H.: Genetic algorithms. In: *Scientific american* 267 (1992), Nr. 1, S. 66–73
- [HS97] HOCHREITER, Sepp ; SCHMIDHUBER, Jürgen: Long short-term memory. In: *Neural computation* 9 (1997), Nr. 8, S. 1735–1780
- [HSHK15] HOFFMANN, Stefan ; SCHROTT, Michael ; HUBER, Thorsten ; KRUSE, Thomas: Modellbasierte Methoden zur Applikation moderner Verbrennungsmotoren. In: *MTZ-Motortechnische Zeitschrift* 76 (2015), Nr. 4, S. 46–51
- [HSMS18] HOHL, Philipp ; STUPPERICH, Michael ; MÜNCH, Jürgen ; SCHNEIDER, Kurt: Combining agile development and software product lines in automotive: challenges and recommendations. In: *2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)* IEEE, 2018, S. 1–10
- [Hug01] HUGHES, Evan J.: Evolutionary multi-objective ranking with uncertainty and noise. In: *International Conference on Evolutionary Multi-Criterion Optimization* Springer, 2001, S. 329–343
- [IL15] IOOSS, Bertrand ; LEMAÎTRE, Paul: A review on global sensitivity analysis methods. In: *Uncertainty management in simulation-optimization of complex systems*. Springer, 2015, S. 101–122
- [Ins19] INSTITUTE, Capgemini R.: The Autonomous Car: A Consumer Perspective. (2019)
- [Jan99] JANSEN, Michiel J.: Analysis of variance designs for model output. In: *Computer Physics Communications* 117 (1999), Nr. 1-2, S. 35–43
- [Jan08] JANKOV, Kristian: *Beitrag zur automatisierten Steuerkennfeld-Applikation bei Fahrzeug-Dieselmotoren*. Berlin, TU Berlin, Diss., 2008
- [JLY07] JIANG, Ming ; LUO, Yupin P. ; YANG, Shiyuan Y.: Particle swarm optimization-stochastic trajectory analysis and parameter selection. In: *Swarm intelligence, Focus on ant and particle swarm optimization*. IntechOpen, 2007
- [Jos17] JOSHI, Adit: Real-Time Implementation and Validation for Automated Path Following Lateral Control Using Hardware-in-the-Loop (HIL) Simulation / SAE Technical Paper. 2017. – Forschungsbericht

- [JS03] JIN, Yaochu ; SENDHOFF, Bernhard: Trade-off between performance and robustness: an evolutionary multiobjective approach. In: *international conference on Evolutionary Multi-Criterion Optimization* Springer, 2003, S. 237–251
- [Kah13] KAHLBAU, S: *Mehrkriterielle Optimierung des Schaltablaufs von Automatikgetrieben*. Cottbus, BTU Cottbus-Senftenberg, Diss., 2013
- [KAV15] KHANDELWAL, Ina ; ADHIKARI, Ratnadip ; VERMA, Ghanshyam: Time series forecasting using hybrid ARIMA and ANN models based on DWT decomposition. In: *Procedia Computer Science* 48 (2015), S. 173–179
- [KD03] KUHNERT, Klaus-Dieter ; DONG, Wenjie: Über die lernende Regelung autonomer Fahrzeuge mit neuronalen Netzen. In: *Autonome Mobile Systeme 2003*. Springer, 2003, S. 106–116
- [KD13] KVASNICKA, Pavel ; DICK, Peter: Integrated development of vehicle dynamics demonstrated on the new BMW 3 series. In: *Proceedings of the FISITA 2012 World Automotive Congress* Springer, 2013, S. 379–389
- [Ken10] KENNEDY, James: Particle swarm optimization. In: *Encyclopedia of machine learning* (2010), S. 760–766
- [KN12] KUIPERS, Lauwerens ; NIEDERREITER, Harald: *Uniform distribution of sequences*. Courier Corporation, 2012
- [KPD<sup>+</sup>06] KVASNICKA, P ; PROKOP, G ; DORLE, M ; RETTINGER, A ; STAHL, H: Durchgangige Simulationsumgebung zur Entwicklung und Absicherung von Fahrdynamischen Regelsystemen. In: *VDI BERICHTE* 1967 (2006), Nr. 1, S. 387
- [Lie17] LIESNER, Louisa: *Automatisierte Funktionsoptimierung von Adaptive Cruise Control*. Shaker Verlag, 2017
- [LZC<sup>+</sup>17] LIU, Jia ; ZHANG, Liyan ; CHEN, Qihong ; QUAN, Shuhai ; LONG, Rong: Hardware-in-the-loop test bench for vehicle ACC system. In: *2017 Chinese Automation Congress (CAC) IEEE*, 2017, S. 1006–1011
- [Mac15] MACH, F: Reduction of optimization problem by combination of optimization algorithm and sensitivity analysis. In: *IEEE Transactions on Magnetics* 52 (2015), Nr. 3, S. 1–4

- [Mar18] MAROWKA, Ami: Python accelerators for high-performance computing. In: *The Journal of Supercomputing* 74 (2018), Nr. 4, S. 1449–1460
- [MDF13] MARTINUS, Marcus ; DEICKE, Markus ; FOLIE, Michael: Virtual test driving hardware-independent integration of series software. In: *ATZelextronik worldwide* 8 (2013), Nr. 5, S. 16–21
- [MHS19] MOSER, Uwe ; HARMENING, Nick ; SCHRAMM, Dieter: A new method for the objective assessment of ADAS based on multivariate time series classification. In: *19. Internationales Stuttgarter Symposium* Springer, 2019, S. 636–651
- [Mie12] MIETTINEN, Kaisa: *Nonlinear multiobjective optimization*. Bd. 12. Springer Science & Business Media, 2012
- [Mit00] MITTERER, Alexander: *Optimierung vielparametrischer Systeme in der Kfz-Antriebsentwicklung: statistische Versuchsplanung und Künstliche Neuronale Netze in der Steuergeräteauslegung zur Motorabstimmung*. VDI-Verlag, 2000
- [MNPLW13] MILANESE, Mario ; NORTON, John ; PIET-LAHANIER, H el ene ; WALTER,  Eric: *Bounding approaches to system identification*. Springer Science & Business Media, 2013
- [Mon17] MONTGOMERY, Douglas C.: *Design and analysis of experiments*. John Wiley & Sons, 2017
- [Moo98] MOORE, Gordon E.: Cramming more components onto integrated circuits. In: *Proceedings of the IEEE* 86 (1998), Nr. 1, S. 82–85
- [Mor91] MORRIS, Max D.: Factorial sampling plans for preliminary computational experiments. In: *Technometrics* 33 (1991), Nr. 2, S. 161–174
- [Mos20] MOSER, Uwe D.: Multivariate Time Series Clustering and Classification for Objective Assessment of Automated Driving Functions. (2020)
- [MP43] McCULLOCH, Warren S. ; PITTS, Walter: A logical calculus of the ideas immanent in nervous activity. In: *The bulletin of mathematical biophysics* 5 (1943), Nr. 4, S. 115–133
- [MW14] MITSCHKE, Manfred ; WALLENTOWITZ, Henning: Fahrerassistenzsysteme. In: *Dynamik der Kraftfahrzeuge*. Springer, 2014, S. 761–768

- [Nau02] NAUMANN, Tino: Wissensbasierte Optimierungsstrategien für elektronische Steuergeräte an Common-Rail-Dieselmotoren. (2002)
- [OEB00] OMAR, TA ; ESKANDARIAN, A ; BEDEWI, NE: Artificial neural networks for modelling dynamics of impacting bodies and vehicles. In: *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics* 214 (2000), Nr. 3, S. 133–142
- [Osc19] OSCHLIES, Hendrik: *Komfortorientierte Regelung für die automatisierte Fahrzeugquerführung*. Springer, 2019
- [Pac05] PACEJKA, Hans: *Tire and vehicle dynamics*. Elsevier, 2005
- [Par64] PARETO, Vilfredo: *Cours d'économie politique*. Bd. 1. Librairie Droz, 1964
- [PCM14] PANDEY, Hari M. ; CHAUDHARY, Ankit ; MEHROTRA, Deepti: A comparative review of approaches to prevent premature convergence in GA. In: *Applied Soft Computing* 24 (2014), S. 1047–1077
- [PD11] PEDRO, Jimoh ; DAHUNSI, Olurotimi: Neural network based feedback linearization control of a servo-hydraulic vehicle suspension system. In: *International Journal of Applied Mathematics and Computer Science* 21 (2011), Nr. 1, S. 137–147
- [PDPS15] PEI, Yuanjiang ; DAVIS, Michael J. ; PICKETT, Lyle M. ; SOM, Sibendu: Engine Combustion Network (ECN): Global sensitivity analysis of Spray A for different combustion vessels. In: *Combustion and Flame* 162 (2015), Nr. 6, S. 2337–2347
- [Ped10] PEDERSEN, Magnus Erik H.: Good parameters for particle swarm optimization. In: *Hvass Lab., Copenhagen, Denmark, Tech. Rep. HL1001* (2010), S. 1551–3203
- [Pfe19] PFEIL, Raphael: *Methodischer Ansatz zur Optimierung von Energieladestrategien für elektrisch angetriebene Fahrzeuge*. Springer, 2019
- [Pil17] PILLAS, Julien: *Modellbasierte Optimierung dynamischer Fahrmanöver mittels Prüfständen*. Technische Universität, 2017
- [PU92] PSICHOGIOS, Dimitris C. ; UNGAR, Lyle H.: A hybrid neural network-first principles approach to process modeling. In: *AI-ChE Journal* 38 (1992), Nr. 10, S. 1499–1511

- [Pug16] PUGET, JF: A speed comparison of C, Julia, Python, Numba, and Cython on LU factorization. In: *IBM Community, January* (2016)
- [Puk06] PUKELSHEIM, Friedrich: *Optimal design of experiments*. SIAM, 2006
- [PZE17] PÜTZ, Andreas ; ZLOCKI, Adrian ; ECKSTEIN, Lutz: Absicherung hochautomatisierter Fahrfunktionen mithilfe einer Datenbank relevanter Szenarien. In: *Workshop Fahrassistenzsysteme und automatisiertes Fahren*, 2017, S. 161–168
- [Raj11] RAJAMANI, Rajesh: *Vehicle dynamics and control*. Springer Science & Business Media, 2011
- [Rec94] RECHENBERG, Ingo: Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. frommann-holzboog, Stuttgart, 1973. In: *Step-Size Adaptation Based on Non-Local Use of Selection Information*. In *Parallel Problem Solving from Nature (PPSN3)* (1994)
- [Roj13] ROJAS, Raul: *Theorie der neuronalen Netze: eine systematische Einführung*. Springer-Verlag, 2013
- [RRGK04] RAKOPOULOS, CD ; RAKOPOULOS, DC ; GIAKOURIS, EG ; KYRITSIS, DC: Validation and sensitivity analysis of a two zone Diesel engine model for combustion and emissions prediction. In: *Energy conversion and management* 45 (2004), Nr. 9-10, S. 1471–1495
- [RS40] RIEKERT, Paul ; SCHUNCK, Theo-Ernst: Zur fahrmechanik des gummibereiften kraftfahrzeugs. In: *Ingenieur-Archiv* 11 (1940), Nr. 3, S. 210–224
- [RS13] RIOS, Luis M. ; SAHINIDIS, Nikolaos V.: Derivative-free optimization: a review of algorithms and comparison of software implementations. In: *Journal of Global Optimization* 56 (2013), Nr. 3, S. 1247–1293
- [SAA<sup>+</sup>10] SALTELLI, Andrea ; ANNONI, Paola ; AZZINI, Ivano ; CAMPOLONGO, Francesca ; RATTO, Marco ; TARANTOLA, Stefano: Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. In: *Computer Physics Communications* 181 (2010), Nr. 2, S. 259–270



- [Sch10] SCHMIDT, Christian: *Hardware-in-the-Loop gestützte Entwicklungsplattform für Fahrerassistenzsysteme. Analyse und Generierung kritischer Verkehrsszenarien*. kassel university press GmbH, 2010
- [Sch15] SCHMIDHUBER, Jürgen: Deep learning in neural networks: An overview. In: *Neural networks* 61 (2015), S. 85–117
- [Sch18] SCHOLZ, Daniel: Multikriterielle Optimierung. In: *Optimierung interaktiv*. Springer, 2018, S. 169–187
- [SHB10] SIEBERTZ, David van Bebber K. ; HOCHKIRCHEN, Thomas ; BEBBER, D van: *Statistische versuchsplanung*. Springer, 2010
- [SHI00] SCHÜLER, Matthias ; HAFNER, Michael ; ISERMANN, Rolf: Einsatz schneller neuronaler Netze zur modellbasierten Optimierung von Verbrennungsmotoren. In: *MTZ-Motortechnische Zeitschrift* 61 (2000), Nr. 10, S. 704–711
- [Sit14] SITTIG, Andy: *Optimierung und Applikation von Betriebsstrategien in Hybridfahrzeugen*. München, Technische Universität München, Diss., 2014
- [SLKK11] SEO, Sung-Il ; LEE, Moon K. ; KIM, Sang-Jo ; KIM, Naksoo: Robust optimum design of a bearingless rotation motor using the Kriging model. In: *International Journal of Precision Engineering and Manufacturing* 12 (2011), Nr. 6, S. 1043–1050
- [Som11] SOMMERVILLE, Ian: Software engineering 9th Edition. In: *ISBN-10137035152* (2011)
- [SPW16] SARRAZIN, Fanny ; PIANOSI, Francesca ; WAGENER, Thorsten: Global Sensitivity Analysis of environmental models: Convergence and validation. In: *Environmental Modelling & Software* 79 (2016), S. 135–152
- [SR08] SEIFFERT, Ulrich ; RAINER, Gotthard P.: *Virtuelle Produktentwicklung für Fahrzeug und Antrieb im KFZ: Prozesse, Komponenten, Beispiele aus der Praxis*. Springer, 2008
- [SRA+08] SALTELLI, Andrea ; RATTO, Marco ; ANDRES, Terry ; CAMPOLONGO, Francesca ; CARIBONI, Jessica ; GATELLI, Debora ; SAISANA, Michaela ; TARANTOLA, Stefano: *Global sensitivity analysis: the primer*. John Wiley & Sons, 2008

- [SSB14] SAK, Haşim ; SENIOR, Andrew ; BEAUFAYS, Françoise: Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In: *Fifteenth annual conference of the international speech communication association*, 2014
- [SSR<sup>+</sup>18] SPIEGEL, L ; SCHÜRMAN, T ; RAUNER, I ; STACHE, M ; GÖHRING, M ; NEUSSER, HJ: Das Antriebssystem des neuen Cayenne S Hybrid. (2018)
- [Sta98] STAGGE, Peter: Averaging efficiently in the presence of noise. In: *International Conference on Parallel Problem Solving from Nature* Springer, 1998, S. 188–197
- [STCR04] SALTELLI, Andrea ; TARANTOLA, Stefano ; CAMPOLONGO, Francesca ; RATTO, Marco: Sensitivity analysis in practice: a guide to assessing scientific models. In: *Chichester, England* (2004)
- [STCR13] SALTELLI, A ; TARANTOLA, S ; CAMPOLONGO, F ; RATTO, M: Sensitivity analysis in practice: A guide to assessing scientific models. 2004. In: *Chichester, England: John Wiley & Sons, Ltd. Steven Batill (2) Dept. of Aerospace & Mechanical Engr. University of Notre Dame Notre Dame, IN 46556* (2013), S. 48090–9055
- [TCMA17] TÖRNGREN, Martin ; CHEN, DeJiu ; MALVIUS, Diana ; AXELSSON, Jakob: Model-based development of automotive embedded systems. (2017)
- [Tre03] TRELEA, Ioan C.: The particle swarm optimization algorithm: convergence analysis and parameter selection. In: *Information processing letters* 85 (2003), Nr. 6, S. 317–325
- [TSVT18] TETTAMANTI, Tamás ; SZALAI, Mátyás ; VASS, Sándor ; TIHANYI, Viktor: Vehicle-in-the-loop test environment for autonomous driving with microscopic traffic simulation. In: *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)* IEEE, 2018, S. 1–6
- [Unt13] UNTERREINER, Michael: *Modellbildung und simulation von fahrzeugmodellen unterschiedlicher komplexität*. Duisburg, Universitätsbibliothek Duisburg-Essen, Diss., 2013
- [VDA15] VDA: Automatisierung–von Fahrerassistenzsystemen zum automatisierten Fahren. In: *Verband der Automobilindustrie, Berlin* (2015)

- [VMCN15] VANROLLEGHEM, Peter A. ; MANNINA, Giorgio ; COSENZA, Alida ; NEUMANN, Marc B.: Global sensitivity analysis for urban water quality modelling: Terminology, convergence and comparison of different methods. In: *Journal of Hydrology* 522 (2015), S. 339–352
- [Wei09] WEISE, Thomas: Global optimization algorithms-theory and application. In: *Self-Published Thomas Weise* (2009)
- [WHLS15] WINNER, Hermann ; HAKULI, Stephan ; LOTZ, Felix ; SINGER, Christina: *Handbuch Fahrerassistenzsysteme: Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort*. Springer-Verlag, 2015
- [WMF<sup>+</sup>00] WEICKER, Karsten ; MITTERER, Alexander ; FLEISCHHAUER, Th ; ZUBER-GOOS, Frank ; ZELL, Andreas: Einsatz von Softcomputing-Techniken zur Kennfeldoptimierung elektronischer Motorsteuergeräte (Using Softcomputing Techniques for Map Optimization of Electronic Control Units for Combustion Engines). In: *at-Automatisierungstechnik Methoden und Anwendungen der Steuerungs-, Regelungs- und Informationstechnik* 48 (2000), Nr. 11/2000, S. 529
- [WPM18] WINNER, Hermann ; PROKOP, Günther ; MAURER, Markus: *Automotive Systems Engineering II*. Bd. 1. Springer, 2018
- [WWJ18] WINNER, Hermann ; WACHENFELD, Walther ; JUNIETZ, Phillip: Validation and introduction of automated driving. In: *Automotive Systems Engineering II*. Springer, 2018, S. 177–196
- [WZKT10] WERLING, Moritz ; ZIEGLER, Julius ; KAMMEL, Sören ; THRUN, Sebastian: Optimal trajectory generation for dynamic street scenarios in a frenet frame. In: *2010 IEEE International Conference on Robotics and Automation IEEE*, 2010, S. 987–993
- [YO04] YIM, Young U. ; OH, Se-Young: Modeling of vehicle dynamics from real vehicle measurements using a neural network with two-stage hybrid learning for accurate long-term prediction. In: *IEEE Transactions on Vehicular Technology* 53 (2004), Nr. 4, S. 1076–1084



## Eigene Veröffentlichungen

- [FFFG19a] FRAIKIN, Nicolas ; FUNK, Kilian ; FREY, Michael ; GAUTERIN, Frank: Dimensionality reduction and identification of valid parameter bounds for the efficient calibration of automated driving functions. In: *Automotive and Engine Technology* 4 (2019), Nr. 1-2, S. 75–91
- [FFFG19b] FRAIKIN, Nicolas ; FUNK, Kilian ; FREY, Michael ; GAUTERIN, Frank: A fast and accurate hybrid simulation model for the large-scale testing of automated driving functions. In: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* (2019), S. 0954407019861245
- [FFFG20] FRAIKIN, Nicolas ; FUNK, Kilian ; FREY, Michael ; GAUTERIN, Frank: Efficient simulation based calibration of automated driving functions based on sensitivity based optimization. In: *IEEE Open Journal of Intelligent Transportation Systems* 1 (2020), S. 63–79
- [Fra21] FRAIKIN, Nicolas: *Verfahren zur effizienten, simulativen Applikation automatisierter Fahrfunktionen*. Veröffentlichung der Patentanmeldung DE102019126195A1, 2021.