

---

---

**Lernen von Sensormesswerten zur Verbesserung  
von Fahrerassistenzsystemen am Beispiel eines  
adaptiven Kurvenwarners**

vorgelegt von  
Dipl.-Ing.  
Robert Heinrichs  
geb. in Berlin

von der Fakultät IV – Elektrotechnik und Informatik  
der Technischen Universität Berlin  
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften  
- Dr.-Ing. -

genehmigte Dissertation

**Promotionsausschuss:**

**Vorsitzender:** Prof. Dr. Andreas Vogelsang  
**Gutachter:** Prof. Dr.-Ing. Stefan Jähnichen  
Prof. Dr. Johann Marius Zöllner  
Prof. Dr. Peter Pepper

Tag der wissenschaftlichen Aussprache: 19. Juni 2018

---

Berlin 2018



# | Widmung

Diese Arbeit widme ich meinem unerwartet verstorbenen Vater:

*Reimar Manfred Karl Jurthe-Heinrichs*

*April 2017*

Ich vermisse dich!



# Danksagung

Zunächst möchte ich Ralf Guido Herrtwich danken, ohne ihn wäre das Forschungsprojekt zwischen Daimler und dem DCAITI nicht zustande gekommen auf dem meine Arbeit basiert. Ein großer Dank geht an Martin Fritzsche, der dieses Forschungsprojekt stets mit Rat und Tat seitens Daimler betreut hat.

Vielen Dank an Stefan Jähnichen für die herzliche Aufnahme in den Fachbereich Softwaretechnik und das zahlreiche und hilfreiche Feedback zu meiner Dissertation. Johann Marius Zöllner und Peter Pepper möchte ich in gleichem Maße danken, mit deren Feedback die Qualität der Arbeit weiter angehoben werden konnte.

Danke an alle Kollegen, speziell Gabi Ambach, Steffen Helke, Kristian Duske und Marcus Mews, für die zahlreichen Gespräche, die nicht nur thematisch geholfen haben, sondern auch beim generellen Verfassen der Dissertation.

Ich möchte meiner Frau Alexandra Mehlhase danken für die große kontinuierliche Unterstützung, die ein großer Baustein im Fundament zum Verfassen dieser Arbeit war. Sie hat mir stets den Rücken freigehalten und es mir so ermöglicht, mich stärker auf die Arbeit zu fokussieren. Vielen Dank!

Danke an David Rieck für das Lesen und Feedback geben. Er war ein exzellenter (Coop-)Partner beim Zocken und half somit den nötigen Ausgleich während des Schreibens zu bekommen.

Vielen Dank an meine Familie, Reimar Jurthe-Heinrichs, Ortrud Heinrichs und Johanna Gennrich, die mir einen starken Rückhalt gegeben, mich stetig motiviert und ohne zu zögern geholfen haben.

Papa, ich weiß du kannst dies hier nicht mehr lesen, aber ich möchte dir besonders Danken. Du hast oft mit wenigen aber gezielten Worten einen großen Effekt bei mir erzielt, wie auch bei dieser Arbeit. Diese Worte sind mir noch heute im Ohr und haben mir viel Energie und Antrieb für diese Arbeit gegeben. Danke! Ich wünschte du hättest bei meiner Verteidigung anwesend sein können. Du fehlst mir sehr!



# Kurzfassung

Fahrerassistenzsysteme sind heutzutage zentraler Bestandteil von Kraftfahrzeugen und verbessern die Sicherheit und den Komfort. Insbesondere im Bereich der Sicherheit haben Fahrerassistenzsysteme einen signifikanten positiven Einfluss, der in den Unfallstatistiken sichtbar ist.

Mit Sensoren und Aktoren können Fahrerassistenzsysteme den Zustand des Fahrzeugs und dessen Umgebung erfassen und gegebenenfalls eingreifen, um den Fahrer zu unterstützen. Bisherige Fahrerassistenzsysteme verworfen jedoch die Sensordaten nach der Auswertung und müssen Sensordaten jedes Mal von Grund auf neu auswerten.

Werden für Fahrerassistenzsysteme relevante Informationen im Fahrzeug gespeichert, können mit statistischen Verfahren neue Erkenntnisse und Informationen fahrtübergreifend gewonnen werden. Dies wird als *Lernen* bezeichnet. Mit diesen Informationen können Fahrerassistenzsysteme verbessert werden und sich an den individuellen Fahrstil eines Fahrers anpassen. Durch eine sich ständig verändernde Umgebung müssen die gelernten Informationen aktuell gehalten werden. Dazu müssen Änderungen erkannt und bei den bisher gelernten Informationen berücksichtigt werden.

Fahrerassistenzsysteme benötigen für ihre präzise und sicherheitskritische Funktion sehr umfangreiche Informationen. Bisherige lernende Systeme im Fahrzeug, die auf statistischen Verfahren basieren, können diese Informationen nicht bereitstellen. Ebenso wurden mögliche Änderungen unzureichend berücksichtigt, die bisher gelernte Informationen unter Umständen ungültig werden lassen.

Das entwickelte Verfahren namens *Messwertreihen-Transformation* ermöglicht es dem lernenden System, umfangreiche Informationen in Form von Wahrscheinlichkeitsverteilungen an Positionen auf der Straße zu generieren. Das dafür verwendete statistische Verfahren *Kerndichteschätzung*, ermöglicht das Lernen dieser Informationen mit konstantem Speicherplatzbedarf. Zur Erkennung von Änderungen wird das *t-Test-Vergessen* basierend auf Hypothesentests entwickelt. Es ist in der Lage, verschiedene Arten von Änderungen zu erkennen und ungültig gewordene Informationen aus dem System zu entfernen.

Diese Beiträge ermöglichen die Realisierung eines prototypischen lernenden Fahrerassistenzsystems, dem adaptiven Kurvenwarner, der in einem Versuchsträger demonstriert wird. Der adaptive Kurvenwarner lernt die Position und das Geschwindigkeitsprofil von Kurven selbstständig, aktualisiert das Profil mit jeder Fahrt durch diese Kurve und kann den Fahrer bei möglicher überhöhter Geschwindigkeit rechtzeitig warnen.

---

Das in der vorliegenden Arbeit erschaffene lernende System, ermöglicht es, beliebige Messgrößen entlang Straßen in Form von Wahrscheinlichkeitsverteilungen zu lernen. Mit der Änderungserkennung *t-Test-Vergessen* werden diese Informationen aktuell gehalten. Das lernende System stellt damit Fahrerassistenzsystemen aktuell gehaltene, umfangreiche Informationen zur Verfügung, mit denen die Funktion verbessert und für einen Fahrer individualisiert werden kann.

# Abstract

Today, Advanced Driver Assistance Systems are a central part of vehicles and improve safety and comfort. Especially in the safety area, Advanced Driver Assistance Systems have a significant positive impact, which is visible in accident statistics.

With sensors and actuators Advanced Driver Assistance Systems can capture the vehicles state and its surroundings, and can intervene if necessary, supporting the driver. Former Advanced Driver Assistance Systems discard sensor data after evaluation and must analyze them from scratch each time.

If relevant information is stored in the vehicle, new knowledge and new information can be gained across trips by using statistical algorithms. This is called *learning*. With this information, Advanced Driver Assistance Systems can be improved and can adapt to the individual driving style of a driver. Due to the always changing environment, the learned information has to be kept updated. To do this, changes have to be detected and incorporated into the previously learned information.

Advanced Driver Assistance Systems require extensive information for their precise and safety-critical function. Previous learning systems in vehicles based on statistical algorithms are not able to provide this information. Also, possible changes were not considered extensively, which could invalidate previously learned information.

The developed algorithm called *measurement series transformation* enables the learning system to generate extensive information consisting of probability distributions at positions along the road. For this, the statistical algorithm *kernel density estimation* is used and enables learning this information with a constant storage footprint. To detect changes, the *t-test-forgetting* is developed, which is based on hypothesis tests. It is able to detect different types of changes and remove invalidated information from the system.

These contributions allowed the implementation of a prototypical learning Advanced Driver Assistance System *Adaptive Curve Warning*, which is demonstrated in an experimental vehicle. The *Adaptive Curve Warning* system is able to learn the position and speed profile of curves autonomously, update the profile with every trip through that curve and timely warn the driver in case of excess speed.

The learning system created in this thesis enables learning any measurands as probability distributions along roads. With the *t-test-forgetting*,

---

this information is kept updated. Thus, the learning system provides Advanced Driver Assistance Systems with updated extensive information with which their function can be improved and individualized for a driver.

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Ziele der Arbeit . . . . .	3
1.3	Aufbau der Arbeit . . . . .	4
<b>2</b>	<b>Grundlagen Fahrerassistenzsysteme</b>	<b>5</b>
2.1	Fahrerassistenzsysteme . . . . .	5
2.1.1	Sensoren . . . . .	7
2.1.2	Aktoren . . . . .	11
2.1.3	Steuergeräte . . . . .	13
2.1.4	Bussysteme . . . . .	14
2.1.5	Digitale Karten . . . . .	15
2.2	Entwicklung der Fahrerassistenzsysteme . . . . .	18
2.3	Grenzen aktueller Fahrerassistenzsysteme . . . . .	24
<b>3</b>	<b>Lernende Systeme im Fahrzeug</b>	<b>27</b>
3.1	Lernendes System . . . . .	27
3.2	Annahmen und Einschränkungen . . . . .	29
3.3	Leitbeispiel: Adaptiver Kurvenwarner . . . . .	32
3.3.1	Beschreibung der Funktionalität . . . . .	33
3.3.2	Anforderungen an das lernende System . . . . .	36
3.4	Vorhandene Ansätze für lernende Systeme . . . . .	38
3.4.1	Ein lernendes System zur Korrektur der digitalen Karte . . . . .	39
3.4.2	Lernen von Routencharakteristiken für vorausschauende Fahrstrategien . . . . .	40
3.4.3	Reichweitenprädiktion von elektrischen Fahrzeugen über ein lernendes System . . . . .	41
3.4.4	Fazit vorhandener Ansätze . . . . .	43
3.4.5	Weitere Beiträge mit Relevanz für lernende Systeme . . . . .	44
3.5	Konkretisierte Ziele . . . . .	46
<b>4</b>	<b>Lernen von heterogenen Sensormesswertreihen</b>	<b>49</b>
4.1	Der Lernprozess . . . . .	49
4.1.1	Bisheriger Lernprozess basierend auf der Kerndichteschätzung . . . . .	50
4.1.2	Notwendigkeit einer Messwertreihen-Transformation . . . . .	52
4.1.3	Messwertreihen-Transformation auf ein positionsabhängiges Raster . . . . .	54

4.1.4	Zusammenführung von positionsabhängigen Messwertreihen mit der Kerndichteschätzung . . . . .	62
4.2	Evaluation des erweiterten Lernprozesses . . . . .	63
4.2.1	Anwendung des Lernprozesses mit echten Sensormesswertreihen . . . . .	64
4.2.2	Vorteile für den adaptiven Kurvenwarner durch gelernte Messwertreihen . . . . .	68
4.3	Zusammenfassung . . . . .	71
<b>5</b>	<b>Erkennung von ungültigen Informationen im lernenden System</b>	<b>73</b>
5.1	Änderungserkennung zur Detektion von ungültigen Informationen . . . . .	73
5.1.1	Bisherige Arbeiten zum Umgang mit Änderungen . . . . .	74
5.1.2	Verschiedene Arten von Änderungen . . . . .	77
5.1.3	Hypothesentest zum Erkennen von Änderungen . . . . .	78
5.1.4	Bilden von Stichproben für den Hypothesentest . . . . .	81
5.1.5	Erkennen von sprungartigen Änderungen . . . . .	84
5.1.6	Erkennen von schleichenden Änderungen . . . . .	84
5.1.7	Integration der Änderungserkennung in den Lernprozess . . . . .	87
5.2	Evaluation der Änderungserkennung t-Test-Vergessen . . . . .	91
5.2.1	Erstellung von Messesequenzen . . . . .	91
5.2.2	t-Test-Vergessen bei sprungartigen Änderungen . . . . .	97
5.2.3	t-Test-Vergessen bei schleichenden Änderungen: . . . . .	103
5.3	Zusammenfassung . . . . .	106
<b>6</b>	<b>Anwendung des lernenden Systems</b>	<b>109</b>
6.1	Benötigte Komponenten für das Gesamtsystem . . . . .	109
6.1.1	Persistente Speicherung gelernter Informationen . . . . .	110
6.1.2	Selektion relevanter Informationen . . . . .	118
6.1.3	Priorisierung zum Löschen von unwichtigen Informationen . . . . .	124
6.2	Der adaptive Kurvenwarner im Versuchsträger . . . . .	136
6.3	Zusammenfassung . . . . .	141
<b>7</b>	<b>Zusammenfassung</b>	<b>143</b>
7.1	Beiträge . . . . .	143
7.2	Weiteres Beispiel für die Anwendung des lernenden Systems . . . . .	145
7.3	Ausblick . . . . .	146

<b>Abbildungsverzeichnis</b>	<b>149</b>
<b>Abkürzungsverzeichnis</b>	<b>153</b>
<b>Literaturverzeichnis</b>	<b>155</b>



# Einführung

---

## 1.1 Motivation

Alleine in Deutschland sterben jährlich mehrere tausend Menschen durch Verkehrsunfälle, Tendenz fallend [102]. Diese Tendenz ist unter anderem den großen Fortschritten im Bereich der Fahrerassistenzsysteme (FAS) zu verdanken, die von den Automobilherstellern weiterentwickelt werden. Diese FAS unterstützen den Fahrer bei seiner Fahraufgabe und können in gefährlichen Situationen eingreifen. So kann ein Kontrollverlust über das Fahrzeug verhindert und damit die Schwere eines Unfalls gemindert oder ein Unfall vollständig vermieden werden. Die Automobilhersteller verbessern kontinuierlich bestehende FAS und entwickeln neue, unter anderem mit dem Ziel, die Zahl der Verkehrsunfälle weiter zu reduzieren.

Moderne FAS in Serienfahrzeugen werten Sensordaten über den Zustand des Fahrzeugs und der Umgebung aus und entscheiden, ob eine Assistenz erforderlich ist. Nach dieser Auswertung werden die Sensordaten verworfen. Durch das Verwerfen der Sensordaten muss ein FAS jede Situation von Grund auf neu auswerten, auch wenn eine Situation regelmäßig auftritt.

Wäre es möglich, Informationen im Fahrzeug zu speichern, könnten Informationen von bereits erkannten Situationen wiederverwendet werden. Dadurch kann beispielsweise das Eintreten einer bekannten Situation antizipiert werden, das dem Fahrzeug die Möglichkeit gibt, sich auf diese Situation einzustellen. Ebenso können Informationen über diese Situationen verfügbar sein, die für eine Assistenz relevant sind.

Der Ansatz, Sensordaten und deren Auswertung im Fahrzeug für zukünftige Fahrten zu speichern, wird in aktuellen Serienfahrzeugen nicht verwendet. Die Forschung in diesem Bereich zeigt jedoch das große Verbesserungspotential für FAS, die diesen Ansatz verwenden [47, 13, 75, 104, 21, 85] und ist deswegen auch Forschungsgegenstand der vorliegenden Arbeit.

Schlüsselement für das Nutzen von Informationen aus vergangenen Fahrten ist ein lernendes System, das relevante Informationen aus Ein-

gabewerten wie Sensordaten gewinnt und speichert. Der Begriff *Lernen* bedeutet in diesem Fall die Fähigkeit des Systems, neues Wissen aus den Eingabewerten zu generieren, dieses Wissen ständig zu aktualisieren und das Wissen auf relevante Umgebungseinflüsse anzupassen. Die Begriffsbedeutung von *Lernen* wird analog zu anderen Arbeiten in diesem Bereich [47, 75, 13] verwendet. Benutzen FAS die Informationen des lernenden Systems, führt die stetige Veränderung des Wissens zu einer Änderung der Arbeitsweise des FAS, sie werden zu adaptiven FAS.

Werden Messgrößen gelernt, die vom Fahrer beeinflusst werden, passen sich FAS auch an den Fahrstil des Fahrers an. Bisherige Ansätze mancher FAS scheiterten, da sie den Fahrer nicht berücksichtigten. Ein Beispiel dafür ist der Kurvenwarner, der bisher über statische Eigenschaften einer Kurve zu realisieren versucht wurde. Jedoch konnte keine allgemeingültige „richtige“ Geschwindigkeit bestimmt werden, die von allen Fahrern akzeptiert wird. Das liegt beispielsweise nicht nur an der unzureichenden Genauigkeit der Kurvengeometrie in Karten, sondern auch an den stark variierenden Fahrstilen und Präferenzen der Fahrer. [5, 75, 44, 101]

Die Regelung der Fahrzeuggeschwindigkeit oder eine Warnung vor überhöhter Geschwindigkeit in Kurven ist eine zeitkritische Assistenz. Verzögerungen dabei können zu kritischen Fahrsituationen führen, wenn der Fahrer zu spät gewarnt oder das Fahrzeug zu spät gebremst wird. Aus diesem Grund muss das lernende System die notwendigen fahrerspezifischen Messgrößen detailliert genug lernen, um eine präzise Assistenz zur rechten Zeit zu ermöglichen.

Ein weiterer wichtiger Aspekt bei lernenden Systemen ist die Aktualität des Wissens. Die fahrzeugrelevante Umgebung verändert sich ständig, wodurch auch fahrerspezifische Messgrößen beeinflusst werden können. Diese Änderungen können deswegen das bisher gelernte Wissen ungültig werden lassen [13, 47]. Werden diese Änderungen erkannt, kann das Wissen aktuell gehalten werden oder zumindest das Wissen vor der Änderung als ungültig eingestuft werden. Dies ist besonders wichtig, da ungültiges Wissen die Funktion eines FAS nicht nur beeinträchtigen kann, sondern auch zu kritischen Fahrsituationen führen kann. Aus diesem Grund ist eine möglichst schnelle Erkennung von Änderungen von großer Bedeutung.

### 1.2 Ziele der Arbeit

In der vorliegenden Arbeit soll ein lernendes System entworfen und umgesetzt werden, welches detaillierte Informationen während der Fahrt lernt, diese aktualisiert und dabei das Fahrverhalten des Fahrers berücksichtigt. Mit diesem neuen lernenden System sollen verbesserte FAS ermöglicht werden, welche den Fahrer fahrtübergreifend genauer, sicherer und individueller unterstützen. Dies soll am Beispiel eines adaptiven Kurvenwarners gezeigt werden, der sich an den Fahrstil des Fahrers anpasst. Dafür werden verschiedene Erweiterungen und Verbesserungen für ein lernendes System in der vorliegenden Arbeit entwickelt und implementiert.

#### 1. Lernen detaillierter Informationen über Fahrsituationen.

Moderne FAS stellen hohe Anforderungen an den Detailgrad der Sensordaten. Damit die gelernten Informationen einen Mehrwert für die FAS bieten, müssen sie ebenfalls einen hohen Detailgrad haben. Ein entsprechend hoher Detailgrad kann mit den bisherigen Arbeiten in diesem Bereich nicht erreicht werden.

Bestehende Lernprozesse sollen erweitert werden, um detaillierte Informationen lernen zu können. Das System soll auch fahrerspezifische Messgrößen verarbeiten und damit die Präferenzen und den Fahrstil des Fahrers lernen können.

#### 2. Erkennung von ungültigen Informationen im lernenden System.

Ungültige Informationen können entstehen, wenn Änderungen in der Fahrzeugumgebung stattfinden und die vorher gelernten Informationen nicht mehr der Realität entsprechen. Ungültige Informationen dürfen den FAS nicht bereitgestellt werden, weswegen sie schnellstmöglich identifiziert und von der Verwendung durch FAS ausgeschlossen werden müssen. Bisherige Ansätze im Bereich der Änderungserkennung haben Einschränkungen, so ist die Leistung der Erkennung von der Qualität der gelernten Daten abhängig.

In der vorliegenden Arbeit soll eine schnellere und vom Lernprozess unabhängige Änderungserkennung entwickelt und in das lernende System integriert werden.

### 3. Implementierung eines prototypischen Fahrerassistenzsystems im Versuchsträger basierend auf dem entwickelten lernenden System.

Es soll das lernende System realisiert und damit ein prototypisches FAS, ein adaptiver Kurvenwarner, umgesetzt werden. Weitere Komponenten, die für ein lauffähiges Gesamtsystem notwendig sind, sollen bereitgestellt werden. Anschließend soll das lernende System und der adaptive Kurvenwarner mit echten Sensormesswerten im Versuchsträger ausgeführt werden.

## 1.3 Aufbau der Arbeit

Kapitel 2 gibt einen Einblick in die grundlegenden Komponenten, der Entwicklung und den Grenzen aktueller FAS.

Kapitel 3 führt lernende Systeme in Fahrzeugen und dadurch ermöglichte Verbesserungen von FAS ein. Es wird der adaptive Kurvenwarner im Detail beschrieben und Anforderungen an das lernende System abgeleitet. Anschließend werden bisherige Arbeiten in diesem Bereich vorgestellt und konkretisierte Forschungsfragen für die vorliegende Arbeit formuliert.

Kapitel 4, 5 und 6 beschäftigen sich mit den drei oben vorgestellten Zielen dieser Arbeit.

In Kapitel 4 wird erarbeitet, wie das Lernen von Informationen mit dem von FAS benötigten Detailgrad möglich ist. Es wird eine entsprechende Erweiterung des Lernprozesses entwickelt, umgesetzt und evaluiert.

In Kapitel 5 wird eine schnelle und unabhängige Änderungserkennung entwickelt, umgesetzt und evaluiert. Diese dient dazu, ungültige, gelernte Informationen im lernenden System schneller als bisher zu identifizieren.

Kapitel 6 beschäftigt sich mit der Anwendung des in der vorliegenden Arbeit entwickelten lernenden Systems. Es werden Komponenten entwickelt, die für ein lauffähiges Gesamtsystem im Versuchsträger notwendig sind. Anschließend wird der adaptive Kurvenwarner mit Hilfe des lernenden Systems realisiert und im Versuchsträger mit echten Messwerten ausgeführt.

Kapitel 7 fasst den Inhalt dieser Arbeit zusammen und gibt einen Ausblick auf mögliche weitere Arbeiten im Bereich lernender Systeme für FAS.

# Grundlagen Fahrerassistenzsysteme

# 2

In diesem Kapitel werden die Grundlagen von Fahrerassistenzsystemen (FAS) beschrieben.

Dafür wird in Abschnitt 2.1 die Aufgabe und die allgemeine Arbeitsweise von FAS erläutert. Die dafür notwendigen Komponenten werden vorgestellt und deren Relevanz für ein lernendes System genannt. Es werden in Abschnitt 2.2 Beispiele für bisherige FAS, deren Aufgabe und konkrete Arbeitsweise gegeben. Zuletzt wird in Abschnitt 2.3 auf die Grenzen moderner FAS eingegangen und skizziert wie ein lernendes System weitere Verbesserungen möglich machen kann.

## 2.1 Fahrerassistenzsysteme

Ein *Fahrerassistenzsystem* bezeichnet ein elektronisches System in Kraftfahrzeugen, das dem Fahrer bei seiner Fahraufgabe unterstützt und damit die Sicherheit und dem Komfort erhöht [101, 3]. Dies trifft bereits für einfache Systeme zu, wie beispielsweise eine automatische Blinkerrückstellung oder ein elektrischer Starter für den Motor eines Fahrzeugs, die den Fahrer bei seiner Fahraufgabe unterstützen [101]. Doch die Systeme, die heute im üblichen Sprachgebrauch als FAS bezeichnet werden, sind komplexere Systeme wie zum Beispiel der Abstandsregeltempomat (engl. Adaptive Cruise Control (ACC)) oder das Antiblockiersystem (ABS) aus dem Jahr 1979 [35] bis hin zu modernen Systemen wie dem Totwinkel-Assistenten, der 2002 auf den Markt kam [101].

In [40] werden verschiedene Arten von Arbeitsteilungen zwischen Mensch und Maschine vorgestellt. Die *Assistenzfunktion* wird dabei als eine redundante und parallele Aufgabe beschrieben [40, 38]. Typischerweise ist ein FAS nur im Hintergrund aktiv, während der Fahrer seiner Fahraufgabe nachkommt. In dieser Zeit arbeitet das FAS redundant und parallel zum Fahrer. Erst wenn definierte Situationen eintreten, wird das FAS aktiv und greift ein. Diese Arbeitsteilung ist sinnvoll, da der Mensch nicht auf Dauer die volle Konzentration leisten kann, ein FAS hingegen schon [101].

## 2 Grundlagen Fahrerassistenzsysteme

---

Das FAS folgt beim Eingriff einer Assistenzstrategie, die üblicherweise ein automatischer Eingriff, eine automatische Regelung oder Warnung an den Fahrer ist.

Situationen, in denen der Fahrer nicht vollständig konzentriert oder überfordert ist, bergen ein großes Unfallpotential. In der Literatur wird auf diesen Umstand hingewiesen:

„Aus theoretischer Sicht wird die Entstehung eines Unfalls immer wahrscheinlicher, wenn die Anforderungen der Verkehrssituation die Leistungsmöglichkeiten des Fahrers übersteigen, d. h. die Aufgabenschwierigkeit ein für ihn bewältigbares Maß übersteigt.“ [23]

Zahlen aus Unfallstatistiken und deren Auswertung bestätigen dies: 95% aller Unfälle im Straßenverkehr sind mindestens anteilig auf menschliche Fehler zurückzuführen, wobei 75% der Unfälle als Alleinursache menschliches Fehlverhalten haben [9, 72]. Abschätzungen über das Unfallvermeidungspotenzial von FAS ergaben nach einer Analyse von über 2800 Unfallprotokollen aus dem Raum Braunschweig im Jahr 2002, dass circa 70% der protokollierten Unfälle durch FAS hätten vermieden werden können [96]. Diese Zahlen zeigen, wie wichtig FAS sind, warum die Automobilhersteller ein großes Interesse an der Weiterentwicklung von FAS haben, und warum sie nach wie vor Gegenstand aktueller Forschung sind.

Die grundlegende Arbeitsweise eines FAS wird im Folgenden am Beispiel des Antiblockiersystem (ABS) vorgestellt. Das ABS soll ein Blockieren der Räder verhindern, um den Kraftschluss zur Straße zu erhöhen und das Fahrzeug wieder in einen lenkbaren Zustand zu bringen:

1. *Sensoren* an jedem Rad messen die Raddrehzahlen und senden diese Messdaten digital über einen *Bus* weiter.
2. Das ABS *Steuergerät* empfängt die Raddrehzahlen über diesen *Bus* und wertet sie aus, um ein blockiertes Rad zu erkennen.
3. Nach dieser Auswertung entscheidet das *Steuergerät*, ob eine Assistenz notwendig ist und sendet bei einem erkannten, blockierten Rad die entsprechenden Steuersignale an den *Aktor* am betroffenen Rad über den *Bus*.
4. Der *Aktor* an diesem Rad empfängt die Steuersignale vom *Bus* und setzt diese um. Er reduziert den Bremsdruck bis sich das Rad wieder dreht. Die notwendige Assistenz wird damit ausgeführt.

5. Nach der Verarbeitung der Raddrehzahlen durch das ABS *Steuergerät*, werden diese Informationen verworfen und die Verarbeitung mit den darauf folgend gemessenen Raddrehzahlen fortgesetzt.

Zusammenfassend wird vom FAS ein oder mehrere **Sensoren** benötigt, die einen Fahrzeugzustand oder eine Umgebungssituation erkennen können. Ein **Steuergerät** bestimmt anhand der Sensordaten, ob eine Situation vorliegt in der eine Assistenz notwendig ist. Sofern eine Assistenz erforderlich ist, setzen **Aktoren** die Assistenz um. Ein **Bus** verbindet diese Komponenten und erlaubt eine Datenübertragung. Manche FAS verwenden neben den Sensordaten noch eine **digitale Karte**. Diese Karte stellt dem FAS weitere Informationen über die Umgebung zur Verfügung.

In Grafik 2.1 ist der Zusammenhang der benötigten Komponenten *Sensoren*, *Aktoren*, *Steuergeräte*, *digitale Karte* und deren Verbindung über ein oder mehrere *Bussysteme* dargestellt.

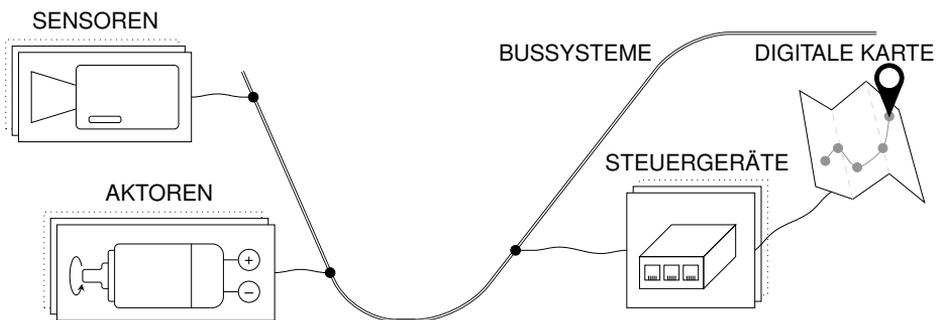


Abb. 2.1: Übersicht über die grundlegenden Komponenten von Fahrerassistenzsystemen und deren Zusammenspiel.

Im Folgenden wird einzeln auf die Komponenten *Sensoren*, *Aktoren*, *Steuergeräte*, *Bussysteme* und *digitale Karten* eingegangen.

### 2.1.1 Sensoren

Die Definition von Sensoren lautet:

„Sensoren sind Geräte, die physikalische Quantitäten wie Druck oder Beschleunigung (Messgrößen genannt) in Ausgangssignale (üblicherweise elektrisch) transformieren (oder umwandeln) und diese als Eingabe für Regelsysteme bereitstellen.“ (frei übersetzt aus [56])

## 2 Grundlagen Fahrerassistenzsysteme

---

Eine Unterscheidung der Sensoren kann im Bezug auf ihren Einsatzzweck vorgenommen werden. Im Automobilbereich werden Sensoren üblicherweise in zwei Gruppen eingeteilt: *Intrinsische Sensoren*, die Zustände des Fahrzeugs messen können, wie die Fahrdynamik, und *extrinsische Sensoren*, welche die Umgebung erfassen können. Extrinsische Sensoren werden auch umgebungserfassende Sensoren genannt.

### Übersicht der Sensoren

Im Folgenden wird eine Übersicht von ausgewählten in- und extrinsischen Sensoren und deren Anwendungsbereich im Automobil gegeben, die für die vorliegende Arbeit relevant sind.

#### Intrinsische Sensoren:

- **Geschwindigkeitssensoren und Timing-Sensoren:** Diese Sensoren können Drehgeschwindigkeiten wie die Radgeschwindigkeiten messen (daher manchmal auch *Raddrehzahlsensoren*) und sind damit Grundlage für die Anzeige der Fahrzeuggeschwindigkeit. Diese Werte sind essentiell für viele FAS wie das Antiblockiersystem (ABS), die Antischlupfregelung (ASR) und das elektronische Stabilitätsprogramm (ESP) (weitere Details folgen in Abschnitt 2.2).
- **Positionssensoren:** Hiermit können Pedalpositionen beispielsweise von Gas und Bremse gemessen werden. Auch ermöglichen es diese Sensoren den Lenkwinkel vom Lenkrad zu messen, der unter anderem für das ESP benötigt wird.
- **Beschleunigungssensoren:** Diese Sensoren messen Beschleunigungen durch die Trägheit einer Testmasse im Sensor. Diese Informationen sind für die Realisierung vom ESP notwendig, werden zum Konditionieren für aktive Fahrwerke benutzt, finden Verwendung in der Unfall- und Überrollerkennung sowie im Motormanagement.
- **Gyroskop:** Dieser Sensor wird benutzt, um die Gierrate des Fahrzeugs zu messen. Die wichtigste Anwendung dieses Sensors ist beim ESP und der Koppelnavigation. Eine Koppelnavigation wird eingesetzt, falls die Lokalisierung über das Global Positioning System (GPS) nicht möglich ist.

### Extrinsische Sensoren:

- **Ultraschallsensoren:** Dieser Sensor ist weit verbreitet und besitzt nur eine geringe Reichweite von wenigen Metern. Ultraschall wird vom Sensor ausgesendet und die Laufzeit der Schallreflexionen werden zur Bestimmung des Abstands zum reflektierenden Objekt verwendet. Er wird primär für die Einparkhilfe verwendet.
- **Radarsensoren:** Das Radio Detection and Ranging (RADAR) ist ein umgebungserfassender Sensor den eine genaue Entfernungsmessung und genaue Messung der relativen Geschwindigkeit von Objekten auszeichnet. Die Winkelposition und die Ausdehnung von Objekten kann jedoch nur ungenau bestimmt werden. Die Arbeitsweise ähnelt dem Ultraschallsensor, es werden jedoch elektromagnetische Wellen statt Ultraschall ausgesendet. Der Radarsensor wird beispielsweise für das ACC zur automatischen Einhaltung eines Abstands zum vorausfahrenden Fahrzeug verwendet.
- **Lidarsensoren:** Dieser Sensor ist neben der Bezeichnung Light Detection And Ranging (LiDAR) auch als *Laserscanner* bekannt. Ausgesendete Lichtstrahlen messen die Entfernung von Punkten in der Umgebung zum Sensor und können so eine dreidimensionale Punktwolke als Abbildung der Fahrzeugumgebung erstellen. Üblicherweise enthält ein Lidarsensor ein rotierendes Element, an dem die Licht aussendenden und empfangenden Bauteile des Sensors angebracht sind. Damit kann ein großer Öffnungswinkel bis hin zu einer Abdeckung von 360 Grad erreicht werden. Dieses rotierende mechanische Element ist in Bezug auf die Robustheit des Sensors die größte Schwäche und neben den vergleichsweise hohen Anschaffungskosten ein Hauptgrund für die geringe Verbreitung in aktuellen Serienfahrzeugen. Auch mit diesem Sensor können wie beim Radar andere Objekte erkannt werden, beispielsweise vorausfahrende Fahrzeuge für das ACC.
- **(Stereo-)Kameras:** Die Kamera ist ein umgebungserfassender Sensor und mit dem optische Eigenschaften und Muster in der gefilmten Umgebung erkannt werden können. Bei der Verwendung von zwei Kameras, Stereokamera genannt, können auch Tiefeninformationen der gefilmten Szene gewonnen werden. Die Genauigkeiten bezüglich der Vermessung eines Objekts sind komplementär zu den Genauigkeiten des Radarsensors: Die Winkelposition und Ausdehnung von

## 2 Grundlagen Fahrerassistenzsysteme

---

Objekten kann relativ genau erfasst werden. Die Bestimmung der Entfernung und der relativen Geschwindigkeit ist deutlich ungenauer. Folglich ergänzen sich Radar und Kamera in ihren Eigenschaften. Eine Fusion dieser Sensoren, um die jeweiligen Stärken zu vereinen, ist deswegen eine wichtige Grundlage für die Weiterentwicklung von FAS [94]. Kameras werden unter anderem zur Erkennung von Spurmarkierungen für einen Spurhalteassistenten verwendet.

Es gibt neben diesen Sensoren auch technisch deutlich umfangreichere Systeme, die als Sensorfunktionen betrachtet werden können, wie beispielsweise das *GPS*. Sie besitzen im Gegensatz zu normalen Sensoren noch weitere Komponenten, die sich auch außerhalb des Fahrzeugs befinden können, beispielsweise Satelliten im Falle des *GPS*:

**Global Positioning System:** Das *GPS* ermöglicht eine Positionsbestimmung über Satelliten mit bekannter Position im Orbit der Erde, die genaue Zeitsignale aussenden. Durch die Laufzeit der Signale entstehen Zeitdifferenzen zwischen dem Senden und Empfangen. Die absolute Position auf der Erde kann über die Triangulation dieser Differenzen bestimmt werden und besteht aus einem Längengrad, Breitengrad und einer Höhenangabe.

Die so berechnete Position ist mit einer Ungenauigkeit behaftet. Bisher wird deswegen im Automobilbereich üblicherweise eine Ungenauigkeit von circa  $\pm 10m$  angenommen [58]. Diese Ungenauigkeit erhöht sich weiter aufgrund von Verdeckungen von *GPS*-Satelliten, wie beispielsweise durch Gebäude, Berge oder Bäume [95]. Werden die *GPS*-Satelliten vollständig verdeckt, beispielsweise in Tunneln oder unter Brücken, wird mit Hilfe eines Gyroskops und Beschleunigungssensoren eine Koppelnavigation durchgeführt, um die Position (fehlerbehaftet) zu berechnen. Das *GPS* im Fahrzeug wird heutzutage hauptsächlich vom Navigationsgerät für die Routenführung verwendet.

Die Genauigkeit der *GPS* Positionierung ist schon seit Dekaden ein Thema [79] und selbst aktuelle Veröffentlichungen versuchen mit verschiedenen Ansätzen, die *GPS* Genauigkeit weiter zu verbessern. Diese Ansätze reichen von der Anwendung von Partikelfiltern auf *GPS* Navigationssysteme [95, 97] bis hin zu kooperativen Systemen, die relevante Informationen für die Positionierung über Fahrzeug-zu-Fahrzeug Kommunikation austauschen [51].

Das Differential Global Positioning System (*DGPS*) ist durch zusätzliche Informationen (Phasenverschiebung der Trägerwelle, Einfallwinkel

des Satellitensignals, Referenzposition einer Empfangsstation) deutlich genauer. Diese Informationen werden von einer festen Referenzstation mit bekannter Position gemessen und per Funk an einen DGPS-Empfänger übermittelt. So kann die Genauigkeit der Positionierung bis auf wenige Millimeter ( $\pm 1 - 10\text{mm}$ ) verbessert werden. [65]

Die anhaltenden Bemühungen in der Forschung lassen weitere Verbesserungen der Positionsbestimmung in naher Zukunft erwarten.

**Zykluszeiten der Sensoren:** Üblicherweise erheben Sensoren in einer festen Zykluszeit ihre Messwerte. Diese Zeit ist abhängig vom Sensor und liegt üblicherweise zwischen wenigen Millisekunden bis zu einer Sekunde. Beispielsweise wird beim GPS nur jede Sekunde eine neue Position gemessen. Ein Beschleunigungssensor kann hingegen jede  $20\text{ms}$  einen neuen Messwert erheben. Damit liegen beispielsweise zwischen zwei gemessenen Positionen 50 Messungen der Beschleunigung vor.

Diese variierenden Zykluszeiten führen dazu, dass die Sensormesswerte in den meisten Fällen nicht zum gleichen Zeitpunkt erhoben werden. Diese Heterogenität muss bei FAS und lernenden Systemen berücksichtigt werden.

Die Auswahl der vorgestellten Sensoren zeigen die Vielfalt der im Fahrzeug verwendeten Sensoren auf und wie deren Signale in FAS Verwendung finden. Die hier genannten Sensoren stellen natürlich nur einen Teil der verbauten Sensoren in heutigen Fahrzeugen dar. Es wird damit bereits deutlich, wie wichtig die Sensorik in den heutigen Fahrzeugen für moderne FAS ist. Auch für lernende Systeme ist ein Zugriff auf die Fahrzeugsensorik essentiell, um dessen Messwerte lernen zu können.

### 2.1.2 Aktoren

Damit ein FAS nach der Auswertung der Sensordaten die Möglichkeit hat, den Fahrer zu unterstützen, muss das FAS das Fahrzeug beeinflussen können. Aktoren setzen elektrische Signale in mechanische Bewegungen oder andere physikalische Größen um, wie zum Beispiel Druck, Temperatur, Schall oder Licht. Damit kann beispielsweise mit einem Lautsprecher ein Warnton ausgegeben, mit einem elektrischen Motor das Fenster geöffnet, mit einer Glühbirne ein Signal im Cockpit angeschaltet oder mit einem elektrischen Ventil die Fahrzeugbremse aktiviert werden.

## 2 Grundlagen Fahrerassistenzsysteme

---

Aktoren können im Automobilbereich in zwei Gruppen unterteilt werden:

1. **Aktoren für Fahrdynamik:** Aktoren, die in der Lage sind, direkten Einfluss auf die Steuerung des Fahrzeugs zu nehmen.
2. **Aktoren für HMI und Komfortfunktionen:** Aktoren, die dem Fahrer Informationen zur Verfügung stellen also der Mensch-Maschine-Schnittstelle (engl. Human Machine Interface (HMI)). Und Aktoren, die dem Fahrer Komfortfunktionen bereitstellen.

### Aktoren für Fahrdynamik

Diese Gruppe der Aktoren kann direkt die Fahrdynamik des Fahrzeugs beeinflussen.

Dazu gehören die Steuerung der Lenkung, der Bremse und der Motorleistung. Systeme, die dabei die mechanische Kopplung vollständig ersetzen, werden „X-by-Wire“-Systeme genannt. Bei der Lenkung wird von „Steer-by-Wire“ gesprochen, wenn die Kopplung über einen Elektromotor realisiert wird. Das „Brake-by-Wire“-System kann über ein elektrisches Hydraulikventil bremsen, indem der hydraulische Druck in der Bremse erhöht wird. Die Motorleistung kann über einen Elektromotor oder ein elektrisches Ventil durch das „Gas-by-Wire“-System gesteuert werden.

Die gängigsten Aktoren im Fahrdynamik Bereich sind Elektromotoren und elektrische Ventile für Flüssigkeiten oder Gase. Mit diesen Aktoren können auch Gänge eingelegt und aktive Fahrwerke eingestellt werden.

### Aktoren für HMI und Komfortfunktionen

Das HMI verwendet Aktoren in drei Fahrzeug-typischen Signalarten *akustisch/auditiv*, *haptisch* und *visuell* [90, 73]:

- **Akustisches Feedback:** Ein Schallwandler (Magnet + Spule [64, 34]) wandelt Strom in Bewegung und über eine Membran in Schallwellen um. Mit ihm können beispielsweise Warntöne an den Fahrer ausgegeben werden.
- **Haptisches Feedback:** Ein Elektromotor mit Unwucht erzeugt aus Strom Vibrationen z.B. im Lenkrad oder im Sitz [101]. Aber auch Aktoren wie Schalter und Pedale geben ein haptisches Feedback durch einen Widerstand beim Betätigen.

- **Visuelles Feedback:** Eine LED oder Glühbirne wandelt Strom in Licht und kann damit die Betriebszustände des Fahrzeugs anzeigen, wie Blinkindikatoren, Abblend- und Fernlicht. Auch Stellmotoren für den Zeiger der Drehzahl und der Geschwindigkeit im Tacho können in das visuelle Feedback eingeordnet werden [101].

Aktoren können auch den Komfort erhöhen, beispielsweise durch elektrische Fensterheber, verstellbare Spiegel, Scheibenwischer und das elektrische Starten des Motors. Diese Komfortsysteme nutzen meist einen Elektromotor als Aktor. Diese Systeme haben jedoch nur eine untergeordnete Relevanz für FAS und werden deswegen in der vorliegenden Arbeit nicht weiter betrachtet.

Die Aktoren für die Fahrdynamik und das HMI stellen wichtige Komponenten für FAS in heutigen Fahrzeugen dar, da ohne sie keine Rückmeldung an den Fahrer und auch kein Eingriff in die Fahrdynamik möglich wäre. Ein heutiges Fahrzeug verfügt über eine Vielzahl solcher Aktoren, welche auf Basis von Sensordaten und Steuergeräten Aktionen ausführen und damit den Fahrer unterstützen können. Aus diesem Grund ist der Zugriff auf die Aktoren auch für das lernende System notwendig.

### 2.1.3 Steuergeräte

Die zentrale Komponente eines FAS ist das Steuergerät. Es erhält die Daten aller relevanten Sensoren, führt die benötigten Auswertungen durch, entscheidet über die Notwendigkeit einer Assistenz und führt diese anhand verbundener Aktoren aus.

Im Fahrzeug werden keine Computer verbaut, wie sie an einem üblichen Arbeitsplatz verwendet werden, sondern auf die Anforderungen von FAS optimierte und nach umfangreichen Anforderungen für den Einsatz in einem Fahrzeug gefertigte Steuergeräte.

Steuergeräte werden für spezielle Aufgaben eines FAS entworfen und gebaut. Assistenzsysteme dürfen nicht ausfallen und deswegen müssen die dafür gebauten Steuergeräte robust sein und eine Fehlertoleranz, wie beispielsweise eine Toleranz gegenüber Ungenauigkeiten von Sensoren, aufweisen. Umfangreiche Testverfahren versuchen, die Funktionen von Steuergeräten zu verifizieren.

Die harten Betriebsbedingungen in Fahrzeugen stellen hohe Anforderungen an die Belastbarkeit von Steuergeräten. Extreme Temperaturen,

Erschütterungen, elektromagnetische Einflüsse und chemische Belastungen, wie Streusalz, Schmutz und aggressive Betriebsmittel des Fahrzeugs, dürfen das Steuergerät nicht beeinträchtigen.

Moderne Oberklassefahrzeuge waren schon 2009 mit typischerweise über 60 Steuergeräten von unterschiedlichen Zulieferern ausgestattet, die zusammen die Gesamtfunktionalität bereitstellten [101]. In den Jahren 2012 bis 2014 besaßen Fahrzeuge der Oberklasse nach Abschätzungen bereits 80 bis 100 Steuergeräte, Tendenz steigend [22, 36].

Ein lernendes System im Fahrzeug würde ebenfalls über ein Steuergerät realisiert werden. Dafür benötigte Algorithmen wie ein Lernprozess würden in diesem Steuergerät ausgeführt werden. Jedoch hat ein lernendes System andere Anforderungen an die Leistung eines Steuergeräts. Beispielsweise sind die Berechnungen aufwändiger und es wird ein persistenter Datenspeicher benötigt. Dies kann dazu führen, dass die Leistung eines klassischen Steuergeräts für ein lernendes System nicht ausreicht.

### 2.1.4 Bussysteme

Anfangs waren die Steuergeräte im Fahrzeug sehr gering oder gar nicht mit anderen Steuergeräten vernetzt [101]. Ebenfalls wurden Sensoren und Aktoren direkt analog mit den Steuergeräten verkabelt, was eine hohe Ausfallgefahr mit sich brachte, da analoge Signale leicht gestört werden können. Erst mit der Einführung von Bussystemen konnten diese Probleme gelöst werden. Heutzutage besitzen nahezu alle Steuergeräte einen oder mehrere Busanschlüsse. Bussysteme ermöglichen es, digitale Signale mit einer Fehlerkorrektur an angeschlossene Komponenten zu übertragen, womit die Störsicherheit deutlich erhöht wurde.

Es existieren vier stark verbreitete Bussysteme im Fahrzeug mit jeweils verschiedenen Anwendungsgebieten:

1. **Local Interconnect Network (LIN):** Dieser Bus wird primär für einen kleinen Verbund von Komponenten verwendet. Beispielsweise wird der LIN-Bus für die Verbindung von Tür-Komponenten benutzt, wie dem elektrischen Fensterheber, elektrisch verstellbaren Spiegel, elektrischen Türschloss und jeweils deren Bedienschalter an der Tür.
2. **Controller Area Network (CAN):** Über diesen Bus werden Informationen für nicht zeitkritische Anwendungen zwischen den Komponenten im Fahrzeug verteilt. Der CAN-Bus kann keine zeitliche

Garantie für die Übertragung der Informationen geben. Beispielsweise werden so Anzeigeinformationen für das Cockpit übermittelt. Dazu gehören Sensordaten wie die Geschwindigkeit, Drehzahl des Motors und eingelegter Gang.

3. **FlexRay:** Neben nicht-zeitkritischen Informationen können hier auch zeitkritische Informationen für FAS übertragen werden. Dies ist möglich, da der FlexRay-Bus eine deterministische Übertragung von Informationen ermöglicht.
4. **Media Oriented Systems Transport (MOST):** Dieser Bus wurde für Multimediaanwendungen entwickelt und besitzt für die Übertragung von beispielsweise Videosignalen eine vergleichsweise hohe Bandbreite. Er kann jedoch im Gegensatz zum FlexRay-Bus keine Daten deterministisch übertragen.

Für FAS und ein lernendes System ist eine schnelle und zuverlässige Übertragung von Informationen wie den Sensormesswerten oder Steuersignale essentiell. Aus diesem Grund sollten diese Komponenten über einen Bussystem verbunden werden, das eine deterministische Übertragung beherrscht. Der FlexRay-Bus wäre deswegen eine geeignete Wahl für FAS und ein lernendes System. Weiterführende Informationen zu Bussystemen in Fahrzeugen können in [105] nachgelesen werden.

### 2.1.5 Digitale Karten

Mit der Einführung des ersten europäischen Seriennavigationsgeräts im Fahrzeug 1989 [101] wurde die Idee einer elektronischen Zielführung für Kraftfahrzeuge aus dem Jahre 1968 für viele PKW-Käufer zugänglich [71].

Primäre Aufgabe eines Navigationsgeräts ist die Führung des Nutzers zu einem geographischen Ziel. Dafür wird neben der Ortung des Fahrzeugs, der Routenberechnung und der Zielführung auch eine digitale Karte benötigt. Diese Karte enthält das Straßennetz und kann weitere Informationen enthalten, wie beispielsweise die Richtung von Einbahnstraßen, Art der Straße oder auch Geschwindigkeitsbegrenzungen.

Die größten Kartenanbieter für Navigationssysteme sind TomTom (ehemals Tele Atlas), HERE (ehemals Nokia Maps und davor Navteq) und OpenStreetMaps (OSM). OSM ist im Gegensatz zu den anderen Anbietern ein Open-Source-Projekt und kann kostenlos genutzt werden. Aus diesen Gründen wird in der vorliegenden Arbeit das OSM Kartenmaterial verwendet.

## 2 Grundlagen Fahrerassistenzsysteme

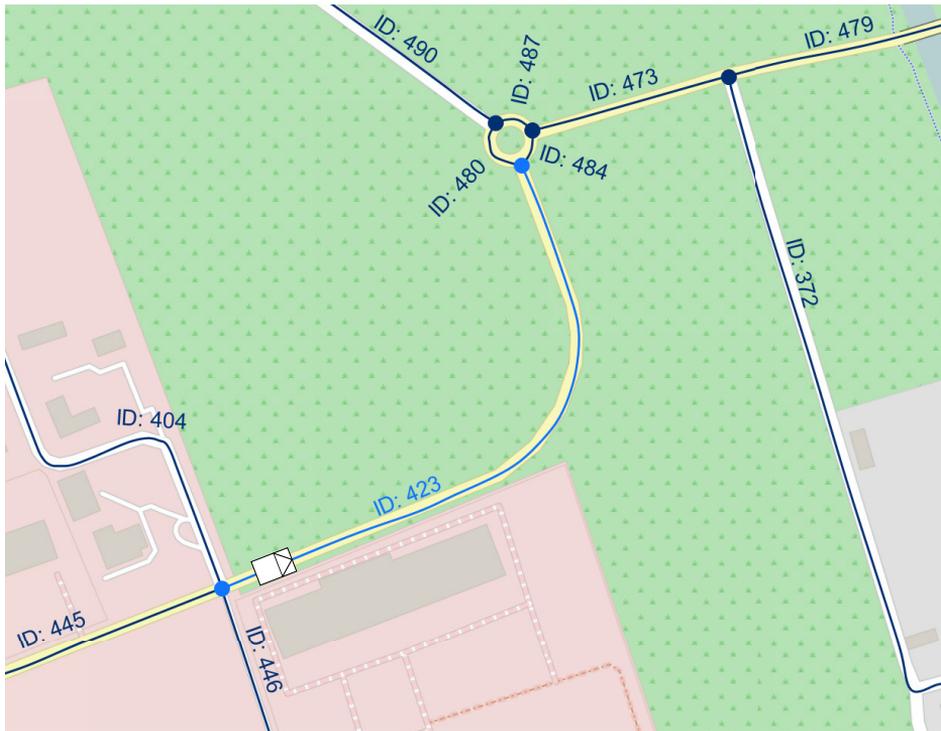


Abb. 2.2: Beispielhafte Darstellung eines Kartengraphs mit einzigartigen Identifikationsnummern. (OSM Kartenmaterial)

Das Datenmodell der OSM Karte besteht aus sogenannten Elementen [61]. Das Straßennetz wird durch die Elemente *node* und *way* abgebildet, die zusammen einen Graph bilden. Dabei entsprechen die Knoten des Graphen dem OSM-Element *node* und werden verwendet, wenn Straßen aufeinandertreffen, beispielsweise Kreuzungen [60]. Die Kanten des Graphen entsprechen dem OSM-Element *way* [59] und repräsentieren die Straßenstücke zwischen zwei Knoten, wie beispielsweise zwei Kreuzungen. Diese beiden Elemente besitzen jeweils eine einzigartige Identifikationsnummer, die eine eindeutige Referenz von Kreuzungen oder Straßenstücken in der Karte ermöglicht. Ein Kartenausschnitt mit eingezeichnetem Kartengraph und beispielhaften Identifikationsnummern der Kanten ist in Abbildung 2.2 dargestellt.

Um ein Routing durchzuführen, muss die Position des Fahrzeugs in der digitalen Karte bekannt sein. Diese kann mit einem Mapmatching berechnet werden.

**Mapmatching:** Die Fahrzeugposition innerhalb der digitalen Karte wird mit dem Mapmatching bestimmt. Dazu wird die von der Sensorik ermittelte Fahrzeugposition zu einer Straßenkante in der digitalen Karte zugeordnet [101].

Für die Navigation und für FAS ist eine genaue Bestimmung der Fahrzeugposition unabdingbar. Wird nur eine reine GPS-Ortung durchgeführt, liegt die Ungenauigkeit bei  $\pm 10$  Metern. Diese Ungenauigkeit kann durch die Verdeckung von Satelliten, beispielsweise durch hohe Gebäude in einer Stadt, noch größer ausfallen (siehe Abschnitt 2.1.1). Bei einem dichten Straßennetz einer Stadt kann es deswegen unmöglich sein die Fahrzeugposition korrekt zu bestimmen. Aus diesem Grund werden beim Mapmatching zusätzlich Informationen über die bisherige Bewegung des Fahrzeugs berücksichtigt. Durch den Abgleich dieser Bewegung mit dem Kartenmaterial, kann in den meisten Fällen genau bestimmt werden, auf welcher Straße und welchem Element im Kartengraph sich das Fahrzeug befindet [101].

**Aktuelle Karten:** Die Kartenhersteller sind bemüht, die Daten ihrer digitalen Karten korrekt zu halten. Die Aktualität von digitalen Karten wird als wichtiger Aspekt genannt [63, 62, 88]. Je älter die Karte ist, desto mehr Abweichungen sind von der Realität vorhanden [101].

Ansätze, die Karten automatisch aktuell zu halten, wurden bereits in EU-Förderprojekten in den Jahren 2005 bis 2008 untersucht [63, 62, 88]. Für Funktionen und insbesondere sicherheitsrelevante FAS, die von Daten aus digitalen Karten abhängig sind, wird es immer wichtiger eine vom Fahrer unabhängige Aktualisierung des Kartenmaterials zu haben [101]. Dennoch ist es bis heute üblich, dass der Fahrer manuell das Kartenmaterial aktualisiert, beispielsweise durch einen Datenträger mit aktuellem Kartenmaterial.

Die Unsicherheit, ob die Informationen der digitalen Karte aktuell sind, ist ein Grund weswegen digitale Karten bisher nicht eng in die Funktion von FAS eingebunden werden.

Mit zunehmender Genauigkeit der digitalen Karten steigt auch deren Bedeutung für den Einsatz bei FAS. Digitale Karten sind auch für lernende Systeme relevant, so könnten Informationen eines lernenden Systems in der Karte an die entsprechenden Elemente der Straßen und Kreuzungen gespeichert werden.

### 2.2 Entwicklung der Fahrerassistenzsysteme

Üblicherweise wird zwischen konventionellen und fortschrittlichen FAS unterschieden. Fortschrittliche FAS besitzen umgebungserfassende (extrinsische) Sensoren und je nach Assistenzfunktion auch eine komplexe Signalverarbeitung [18]. In dieser Arbeit wird die Unterscheidung zwischen den beiden Gruppen anhand der verwendeten Sensorik durchgeführt: Verwendet ein FAS nur intrinsische Sensoren, wird es als konventionelles FAS klassifiziert. Werden zusätzlich extrinsische Sensoren für die FAS Funktion benötigt, wird es als fortschrittliches FAS klassifiziert.

#### Konventionelle Fahrerassistenzsysteme

In [35] wird eine kompakte Übersicht über die Entwicklung von konventionellen FAS gegeben, die hier auszugsweise vorgestellt wird:

- Tempomat (1979): Die Geschwindigkeitsregelung mittels Geschwindigkeitssensor und elektromotorischem Seilzug-Aktor.
- ABS (1979): Das ABS dient zur Blockierverhinderung einzelner Räder beim Bremsen, besonders um die Lenkfähigkeit durch Erhalt von Seitenkräften zu ermöglichen. Ermöglicht wird dies durch Raddrehzahlsensoren an allen Rädern und einem magnetischen Ventil zur Regelung des Bremsdrucks.
- Antriebsschlupfregelung (ASR) (1986): Die ASR der Antriebsräder kann das einseitige und zweiseitige Durchdrehen von Rädern durch Bremsen eines Rades oder Reduzierung des Motordrehmomentes vermeiden. Ebenfalls messen Geschwindigkeitssensoren die Raddrehzahlen und magnetische Ventile bremsen einzelne Räder oder das Motordrehmoment wird geregelt.
- Elektronisches Stabilitätsprogramm (ESP) (1995): Das ESP dämpft Schleuderbewegungen durch Giermomentenerzeugung über das Bremsen einzelner Räder und hält damit das Fahrzeug auf Kurs. Beschleunigungssensoren, Gyroskop, Raddrehzahlsensoren und der Lenkwinkel können verwendet werden, um Magnetventile für den Bremsdruck anzusteuern. Dieses System ist seit 2009 Pflicht in den USA [53].
- Active Body Control (ABC) (1999): Das ABC kann mit hydraulischen Aktoren aktiv Vertikalkräfte an Stahlfedern erzeugen, die zur

besseren Stoßdämpfung eingesetzt werden und auch eine Wankregelung ermöglicht. Unter anderem werden dafür die Beschleunigungssensoren als Eingabewerte benutzt.

### Fortschrittliche Fahrerassistenzsysteme

Im Folgenden wird eine Übersicht über die Entwicklung von fortschrittlichen FAS gegeben.

Eines der ersten fortschrittlichen FAS wurde für eine Anwendung entwickelt, bei der nur geringe Geschwindigkeiten gefahren werden, dem Einparken:

**Einparkassistent:** Dieses FAS benutzt Ultraschallsensoren als umgebungs-erfassende Sensoren. Der Einparkassistent kann dem Fahrer helfen, schneller einen passenden Parkplatz zu finden und sicher in diesen einzuparken [37]. Diese Assistenzsysteme reichen von rein informierenden Systemen, die Abstände zu Objekten mitteilen, über geführte Einparkassistenten bis hin zu semi- und vollautomatischen Einparksystemen, welche die gesamte Fahrzeugführung zum Einparken übernehmen [101].

Üblicherweise werden Ultraschallsensoren verwendet, um im nahen Umfeld des Fahrzeugs die Abstände zu Objekten zu messen. Für semi- und vollautomatische Einparksysteme werden meist andere oder weitere Sensoren wie eine Kamera verwendet.

Später folgte ein neues FAS, das eine erweiterte und verbesserte Version des zu diesem Zeitpunkt über 15 Jahre alten Tempomats war:

**Abstandsregeltempomat:** Der ACC ist eine Weiterentwicklung des einfachen Tempomats. Dieses FAS kann neben dem Halten einer eingestellten Geschwindigkeit auch eine automatische Anpassung der Geschwindigkeit vornehmen. Über einen Radar- oder Lidarsensor können vorausfahrende Fahrzeuge erkannt werden, womit die Geschwindigkeit so reduziert wird, dass ein fester Abstand zu ihnen gehalten wird.

Die erste Idee eines Prototyps des ACC wurde 1980 dokumentiert [1]. Aus dem europäischen Projekt PROMETHEUS (PROgramMe for a European Traffic with Highest Efficiency and Unprecedented Safety), das von 1987 bis 1995 lief, entsprang die Bezeichnung AICC (Autonomous In-

## 2 Grundlagen Fahrerassistenzsysteme

---

telligent Cruise Control) [20], aus der später ACC wurde. Im Jahr 1995 präsentierte Mitsubishi das erste Fahrzeugmodell mit ACC [98].

Manche weiterentwickelte Versionen dieses FAS können auch eine Kollisionswarnung ausgeben, falls der Fahrer ungebremst auf ein erkanntes Fahrzeug oder Hindernis zufährt. Diese Kollisionswarnung kann in entsprechenden Situationen zu einer Unfallvermeidung von 60% bis 90% führen [46].

Zwischen 2005 und 2008 führten die großen deutschen Fahrzeughersteller ein sogenanntes Full-Speed-Range-ACC ein, das ein Fahrzeug bis in den Stand bremsen und in Stausituationen vollautomatisch dem vorausfahrenden Fahrzeug während einem Stop&Go folgen konnte. [101]

Um die Ausmaße von Unfällen zu reduzieren, ist das frühzeitige Verlangsamten des Fahrzeugs ein vielversprechender Ansatz:

**Bremsassistent:** Der Bremsassistent (BAS) ist seit 1997 auf dem Markt verfügbar. Er unterstützt den Fahrer bei starken Bremsungen und Notbremsungen, in dem das System die Bremskraft verstärkt und schnellstmöglich die maximale Verzögerung aufbaut, auch wenn der Fahrer den Bremsdruck unwissentlich zurücknimmt. Mit diesem Assistenzsystem kann der Bremsweg einer Notbremsung bei 100 km/h im Mittel um circa 20% verkürzt werden [99].

Weiterentwicklungen des BAS verwenden Sensoren wie zum Beispiel den Radarsensor aus einem ACC, um Notbremsituationen besser zu erkennen. In Situationen, in denen ein Aufprall nicht mehr zu vermeiden ist, führt das System einen starken Bremseingriff durch, um damit die Unfallfolgen zu verringern. Der BAS wird kontinuierlich verbessert und mit immer mehr Funktionen ausgestattet, wie beispielsweise einen Schutz der Hörorgane von Insassen durch das Abspielen von *rosa Rauschen* kurz vor dem Zünden der Airbags [49].

Die intensive Verwendung einer Kamera im Fahrzeug ermöglicht es, die Umwelt noch detaillierter zu erfassen, als es beispielsweise mit Radarsensoren möglich war. Damit konnten neue FAS erschaffen werden:

**Spurhalteassistent:** Der Spurhalteassistent versucht eine Unfallart zu vermeiden, die für mehr als ein Drittel aller tödlichen Unfälle verantwortlich ist: Der Spurwechsel oder das unbeabsichtigte Verlassen der Spur oder der Straße [101, 100]. In den USA sind sogar 55% der tödlichen Unfälle auf

das Verlassen der Straße zurückzuführen [101]. Typischerweise erkennt der Spurhalteassistent über eine Kamera die Spur, auf der sich das Fahrzeug befindet und kann so ein Verlassen der Spur erkennen. Es gibt mehrere Varianten dieses FAS, welche von einer reinen Warnung beim versehentlichen Verlassen der Spur (Lane Departure Warning) bis hin zum vollautomatischen Halten der Spur über Brems- oder Lenkeingriffe (Lane Keeping, Lane Departure Prevention) reichen.

Schon 1943 wurde eine ähnliche Idee in einem US Patent festgehalten, bei der ein Fahrzeug automatisch anhand eines Kabels in oder auf der Straße und einer Spule am Fahrzeug gelenkt werden sollte [66]. Die ersten kommerziellen Spurhalteassistenten-Systeme wurden gegen Ende der 90er Jahre entwickelt und fanden ihren Ursprung in den Forschungsergebnissen des PROMETHEUS-Projekts der EU [20]. Von 2000 bis 2005 wurde der Spurhalteassistent erst für Nutzfahrzeuge und später auch für PKWs verfügbar [101].

Nach einem FAS, das die eigene Spur des Fahrzeugs erfassen konnte, folgte bald ein weiteres, das die benachbarten Spuren berücksichtigte:

**Totwinkel-Assistent:** Der Totwinkel-Assistent oder auch Spurwechsel-Assistent, hilft dem Fahrer gefährliche Situationen beim Spurwechsel zu vermeiden. Dafür überwachen Kamera- oder Radarsensoren den seitlich hinteren Bereich des Fahrzeugs [101]. So kann erkannt werden, ob sich Fahrzeuge im toten Winkel befinden oder sich andere Fahrzeuge schnell von hinten auf der benachbarten Spur nähern. Versucht der Fahrer in einem solchen Moment die Spur zu wechseln, wird er durch Warnungen und je nach System auch mit einem leichten Gegenmoment am Lenkrad auf diesen Umstand aufmerksam gemacht.

Die ersten Totwinkel-Assistenten wurden 2002 von Peugeot (Kamera), 2005 von Volvo (Kamera), 2005 von Audi (Radarsensor) und 2007 von Mercedes-Benz (Radarsensor) angeboten [101].

Die Hersteller von Totwinkel-Assistenten weisen auf Schwächen der Systeme hin [101] und auch der ADAC bestätigte 2008 in einem Test der verfügbaren Assistenten ein vorhandenes Verbesserungspotential [2].

## 2 Grundlagen Fahrerassistenzsysteme

---

Eines der neusten fortschrittlichen FAS kann Hilfe in komplexen und unübersichtlichen Situationen leisten:

**Kreuzungsassistent:** Mit der Einführung des Kreuzungsassistenten im Jahr 2009 [92, 91] ist eines der neusten Assistenzsysteme auf den Markt gebracht worden, um Unfälle im Bereich von Kreuzungen und Einmündungen zu vermeiden. Hauptunfallursache ist dort ein Fehlverhalten des Fahrers durch eine Fehlinterpretation der Situation, Unaufmerksamkeit und Sichtbehinderungen, die nicht berücksichtigt wurden [33, 70].

Innerhalb geschlossener Ortschaften in Deutschland ist der *Einbiegen-/Kreuzen-Unfall* mit 26% aller Unfälle der häufigste Unfalltyp [82]. Damit ist der Kreuzungsassistent aus sicherheitstechnischer Sicht ein sehr wichtiges Assistenzsystem.

Zu unterscheiden sind zwei verschiedene Ansätze des Kreuzungsassistenten: Unfälle indirekt verringern, indem auf STOP-Schilder oder (rote) Ampeln hingewiesen wird [101], und Unfälle direkt vermeiden, indem bei drohender Kollision eine automatisierte Bremsung durchgeführt wird. Dafür werden mit Radarsensor und Stereokamera andere Verkehrsteilnehmer erkannt [48], deren Trajektorien vorhergesagt und bei Kollisionsgefahr eingegriffen. Derartig eingreifende Systeme sind seit 2013 auf dem Markt verfügbar [48].

Es existieren aber auch fortschrittliche FAS, die es bisher nicht in die Serienfahrzeuge geschafft haben:

**Kurvenwarner:** Der Kurvenwarner ist ein System, das vom Abkommen von der Fahrbahn aufgrund zu hoher Geschwindigkeit schützen soll. Die Europäische Kommission ermittelte im Rahmen ihrer eSafety-Initiative in 2005 eine Prioritätenliste, die besagt, dass eine allgemeine Geschwindigkeitswarnung zu einer effizienten Reduzierung der Zahl der Verkehrstoten führen würde [101, 42]. Untersuchungen zeigten, dass viele Fahrer die eigene Geschwindigkeit in Kurven unterschätzen und dies das Risiko von Unfällen in Kurven erhöhen kann [50, 15]. Kann eine ungefährliche Geschwindigkeit für eine Kurve ermittelt werden, ist eine rechtzeitige Warnung bei zu hoher Geschwindigkeit möglich. Auf diese Art kann der Fahrer vor der Kurve die Geschwindigkeit auf einen ungefährlichen Wert reduzieren. Dieser Wert ist jedoch stark vom Fahrer und seinem Fahrstil abhängig.

Es wurden positive Tests mit interaktiven Schildern vor Kurven gemacht, um die Geschwindigkeit in Kurven zu reduzieren [28]. Auch wurden Systeme entwickelt, welche die Kurvengeometrie über RFID-Tags in der Straße übermittelt bekommen und damit die Fahrzeuggeschwindigkeit limitieren [26]. Der Kurvenwarner hat das gleiche Ziel, soll jedoch bei jeder Kurve, unabhängig von derartiger Verkehrsinfrastruktur, funktionieren.

Auf Fahrzeugebene wurden Ansätze für eine Kurvenwarnung anhand der Kurvengeometrie aus einer digitalen Karte vorgeschlagen [25, 44]. Dabei wurde in [44] gleichzeitig versucht, die Kurvengeometrie in Echtzeit genauer zu messen, da bisherige Karten dafür meist zu ungenau oder veraltet waren. Im Jahr 2010 hat Bosch eine Navigations-App für Smartphones mit Kurvenwarner veröffentlicht, die eine Warnung basierend auf dem Kurvenradius in einer digitalen Karte ausgibt [25, 55]. Jedoch basierten diese Ansätze nur auf den statische Eigenschaften von Kurven und berücksichtigen den Fahrer nicht.

Die Auswertung eines ähnlichen Warnsystems in [75] zeigte, dass Fahrer die empfohlene Geschwindigkeit für Kurven, welche aus den Kurvenradien nach [41] abgeleitet wurde, zu 50% als „nicht transparent“ einstufen. Für diese Warnung wurde auch der Fahrstil des Fahrers in einfacher Form mitberücksichtigt. Gründe für das kontroverse Empfinden dieses Systems waren unter anderem unübersichtliche Kurvenverläufe, die den Fahrer zu einem defensiveren Fahrstil bewegten [75]. In diesen Fällen wurde die empfohlene Geschwindigkeit als zu hoch empfunden.

Wird der Fahrer nicht ausreichend berücksichtigt, können diese Systeme keine größtenteils akzeptierten Warnungen ausgeben. Darüber hinaus können die vorgestellten Ansätze nur eine Warnung bei bekannten Kurven ausgeben, da zu unbekannt Kurven nicht die notwendigen Informationen vorliegen. Ändern sich die Gegebenheiten einer Kurve, müssen zudem die Informationen bei vielen Systemen erst manuell aktualisiert werden.

Die genannten Einschränkungen könnten der Grund sein, warum in heutigen Serienfahrzeugen der Kurvenwarner nicht als FAS verfügbar ist.

### **Rechtliche Einschränkungen von Fahrerassistenzsystemen**

Im Jahre 1968 wurde das sogenannte *Wiener Übereinkommen über Straßenverkehr* [10, 77] erarbeitet. Es wurde mit der Absicht umgesetzt, den Straßenverkehr durch die Standardisierung von Verkehrsregeln sicherer zu machen. Insgesamt haben 73 Länder dieses Übereinkommen anerkannt, unter anderem auch Deutschland. Die Vertragsstaaten sind demnach ver-

pflichtet, im Bereich des Straßenverkehrs die nationalen Vorschriften im Einklang mit dem Wiener Übereinkommen zu gestalten [17].

Das Wiener Übereinkommen hat einen direkten Einfluss auf die Entwicklung von FAS und auch auf autonome Funktionen und autonome Fahrzeuge [4]. Das Übereinkommen erlegt ausschließlich dem Fahrer die Beherrschung eines Fahrzeugs auf, wonach ein FAS nicht die vollständige Kontrolle über das Fahrzeug haben darf [10]. Auch schließen diese Regelungen den Einsatz von autonomen Fahrzeugen aus.

Neuste Entwicklungen nehmen sich genau dieser Hürden an [27]. So wurde 2014 eine Änderung des Wiener Übereinkommen vorgeschlagen, welche die entsprechenden Artikel für vorschriftenkonforme „Fahrzeugsysteme“ (womit FAS gemeint sind) und Systeme für autonome Fahrzeuge, außer Kraft setzen.

Das Wiener Übereinkommen mit dieser Änderung ist seit dem 23. März 2016 in Kraft getreten und ebnet damit eine wichtige rechtliche Hürde für die nächste Generation von FAS und autonome Fahrzeuge im öffentlichen Straßenverkehr [76].

Damit ist auch der Einsatz eines lernenden Systems in Serienfahrzeugen realistischer geworden, wodurch die Relevanz der Forschung im Bereich von lernenden Systemen steigt.

Im weiteren Verlauf dieser Arbeit werden vorrangig fortschrittliche FAS betrachtet. Aus diesem Grund sind im Folgenden bei der Nennung von FAS die fortschrittlichen FAS gemeint, wenn nichts anderes spezifiziert wird.

### 2.3 Grenzen aktueller Fahrerassistenzsysteme

Derzeit werden die meisten Sensordaten nach deren Auswertung verworfen [31]. Bis auf wenige Informationen, wie beispielsweise Fehlercodes im Fehlerspeicher des Fahrzeugs [78], werden keine erfassten Sensordaten in Serienfahrzeugen gespeichert und für die Funktion von FAS wiederverwendet. Aus diesem Grund müssen FAS die Sensordaten in jeder Situation von Grund auf neu auswerten, auch wenn diese sich wiederholen. So können FAS erst nach der Auswertung und dem Eintreten einer relevanten Situation reagieren und eingreifen.

Gäbe es eine Möglichkeit Sensormesswerte zu speichern, die während der normalen Fahrt generiert werden, könnten diese Informationen als Grundlage für vielfältige Verbesserungen von FAS dienen.

## 2.3 Grenzen aktueller Fahrerassistenzsysteme

---

Der ACC hält in Abhängigkeit von der Geschwindigkeit einen Abstand zum vorausfahrenden Fahrzeug. Dieser Abstand kann der Fahrer manuell an seine Präferenz in einem festgelegten Rahmen anpassen. Ein System könnte bei deaktiviertem ACC den typischen Abstand zum vorausfahrenden Fahrzeug speichern und diesen für den ACC nutzen.

Manche Parkassistenten erfassen die Größe von Parklücken, um dem Fahrer eine Empfehlung zu geben, ob das Fahrzeug hineinpasst. Parkt der Fahrer in eine Parklücke ein, obwohl keine Empfehlung ausgesprochen wurde, kann diese Information gespeichert werden und der Parkassistent seine Empfehlung auf kleinere Parklücken erweitern.

Werden Informationen vom Fahrzeug gemessen, die in Karten vorkommen (Straßenverlauf, Steigungen), könnten die Karteninformationen automatisiert überprüft und gegebenenfalls korrigiert werden [75, 13].

Neuartige FAS können realisiert werden, die beispielsweise den Fahrzeugzustand mit einbeziehen. Mit dem Ladezustand der Batterie eines elektrischen Fahrzeugs und Routeninformationen kann eine Reichweitenprädiktion durchgeführt werden [47].

Wird das Verhalten des Fahrers gespeichert, könnten FAS ihre Funktion an dieses Verhalten anpassen. Zu diesem typischen Verhalten des Fahrers könnten Abweichungen festgestellt und der Fahrer darauf hingewiesen werden. Beispielsweise kann ein adaptiver Kurvenwarner basierend auf dem gespeicherten Geschwindigkeitsprofil des Fahrers in Kurven umgesetzt werden [32]. Der Fahrer wird dann gewarnt, wenn er dieses gespeicherte Geschwindigkeitsprofil überschreitet. Damit würde der Fahrer eine individuelle Warnung und eine persönliche Geschwindigkeitsempfehlung vom adaptiven Kurvenwarner erhalten.

Die Geschwindigkeit kann auch entlang ganzer Routen gespeichert werden, um auf der gesamten Route eine Geschwindigkeitswarnung auszugeben oder sogar eine automatische Regelung der Geschwindigkeit zu realisieren.

Auch kann die Kombination aus Stecken-, Fahrzeug- und Fahrerinformationen zur Verbesserung von Systemen wie einer Automatikschaltung genutzt werden. Befindet sich das Fahrzeug vor einer Steigung in einem gewissen Gang und der Fahrer fährt diese üblicherweise mit einer typischen Beschleunigung oder Geschwindigkeit hoch, kann das Fahrzeug mit diesen Informationen entscheiden, welcher Gang für die kommende Steigung am besten geeignet ist.

Um solche Verbesserungen und Erweiterungen von FAS umzusetzen, müssten Fahrzeuge nicht nur mit einem Speicher ausgestattet werden.

## 2 Grundlagen Fahrerassistenzsysteme

---

Zusätzlich muss ein System Regelmäßigkeiten, wie sich wiederholende Situationen, zwischen den Fahrten erkennen können. Während dieser Situationen muss das System die relevanten Informationen detailliert lernen können. Darüber hinaus müssen die so gelernten Informationen immer überprüft und bei Änderungen gegebenenfalls korrigiert werden. Diese umfangreichen Aufgaben soll das lernende System erfüllen, das in der vorliegenden Arbeit entworfen werden soll.

# Lernende Systeme im Fahrzeug

---

Das folgende Kapitel beschäftigt sich mit lernenden Systemen im Fahrzeug.

Abschnitt 3.1 erklärt was in der vorliegenden Arbeit unter dem Begriff *lernendes System* verstanden wird. In Abschnitt 3.2 werden die Annahmen und Einschränkungen für die vorliegende Arbeit begründet und genannt. Weiter werden Beispiele für FAS gegeben, die durch die Anwendung eines lernenden Systems verbessert werden können. Im darauf folgenden Abschnitt 3.3 wird der adaptive Kurvenwarner als Leitbeispiel eingeführt. In Abschnitt 3.4 werden vorhandene Ansätze in diesem Bereich vorgestellt und schließlich in Abschnitt 3.5 konkretisierte Ziele für die vorliegende Arbeit aufgestellt.

## 3.1 Lernendes System

Unter einem *lernenden System* wird in dieser Arbeit - analog zu anderen Arbeiten in diesem Bereich [47, 75, 13] - ein System verstanden, welches:

- sich wiederholende Situationen in Sensordaten verschiedener Fahrten erkennt,
- diese Sensordaten zusammenführen kann und damit detaillierte Informationen über diese Situation erzeugt,
- die so erzeugten Informationen persistent speichert und den FAS zur Verfügung stellt
- und die gelernten Informationen auf Gültigkeit überprüft und bei Änderungen eine Aktualisierung durchführt.

Ein lernendes System kann in drei wichtige Komponenten gegliedert werden:

1. Ein *Datenspeicher*, der für das persistente Speichern der gelernten Informationen notwendig ist.

### 3 Lernende Systeme im Fahrzeug

---

2. Ein *Lernprozess*, der Sensordaten von Situationen aus verschiedenen Fahrten zusammenführt und damit das „Wissen“ erzeugt.
3. Eine Komponente, die sich mit *Änderungen* beschäftigt, da aufgrund von Änderungen die gelernten Informationen ungültig werden können und daher korrigiert werden müssen.

Zusammenfassend wird die folgende Definition formuliert:

**Lernendes System:** *Ein System das fahrtübergreifend neue Informationen aus Sensordaten generiert, speichert und diese mit jeder Fahrt präzisiert und aktualisiert.*

Ziel eines solchen lernenden Systems ist es, die Funktionen von bestehenden FAS zu verbessern und die Umsetzung neuer FAS zu ermöglichen. Mit einem lernenden System könnten vielfältige Anwendungen im Bereich von FAS realisiert werden, wie beispielsweise:

- **Adaptive Automatikschaltung:** Heutige Automatikschaltungen wählen den Gang hauptsächlich in Abhängigkeit der Stellung des Gaspedals [83]. Umwelteinflüsse, wie Steigungen, werden wenn nur indirekt berücksichtigt. Dies kann zu Situationen führen in denen die Automatikschaltung vor einer Steigung einen dafür ungeeigneten Gang einlegt und beim Erreichen der Steigung in den vorherigen Gang zurückschaltet. In solchen Situationen beschleunigt das Fahrzeug erst nach einer Verzögerung durch erneute Gangwechsel.

Können die Steigungen und die vom Fahrer geforderte typische Fahrzeuglängsbeschleunigung an dieser Steigung gelernt werden, kann die Automatikschaltung so beeinflusst werden, dass ein passender Gang für die Steigung gewählt wird. Damit können unnötige Schaltvorgänge reduziert werden und das Fahrzeug würde schneller wie angefordert ansprechen. Positiver Nebeneffekt bei der Vermeidung von unnötigen Schaltvorgängen ist ein geringerer Verschleiß des Automatikgetriebes.

- **Adaptiver Kurvenwarner:** Mit einer Erkennung von Kurven kann ein Geschwindigkeitsprofil für jede Kurve gelernt werden. So kann der Fahrer gewarnt werden, falls dieser zu schnell in eine Kurve einfahren wird.

Der Fahrer wählt eine Kurvengeschwindigkeit mit Berücksichtigung verschiedener Parameter, wie beispielsweise Kurvenradius und -länge,

## 3.2 Annahmen und Einschränkungen

die Einsehbarkeit der Kurve, sowie auch seinen eigenen Fahrstil. Durch das Lernen der Geschwindigkeit, die der Fahrer für eine Kurve wählt, sind diese Einflussparameter bereits enthalten. Dies ermöglicht eine Kurvenwarnung, die der Fahrer als transparent einstuft, da das System nur bei Abweichungen von seinem typischen Verhalten Warnungen ausgibt.

Die gelernten Informationen müssen mit jeder Fahrt präzisiert und gegebenenfalls korrigiert werden. Tritt eine Änderung auf, muss das lernende System die bisher gelernten Informationen auf die neue Gegebenheit anpassen. Damit verändern FAS ihr Verhalten und sind damit adaptiv.

In Abbildung 3.1 ist dargestellt, wie ein lernendes System in die bestehende IT-Infrastruktur von heutigen Fahrzeugen integriert werden könnte. Über die Anbindung an die Busse im Fahrzeug hat das lernende System Zugriff auf die Sensoren, Aktoren, Steuergeräte der FAS und die digitale Karte, womit die Voraussetzungen eines lernenden Systems auf Hardwareebene erfüllt wären.

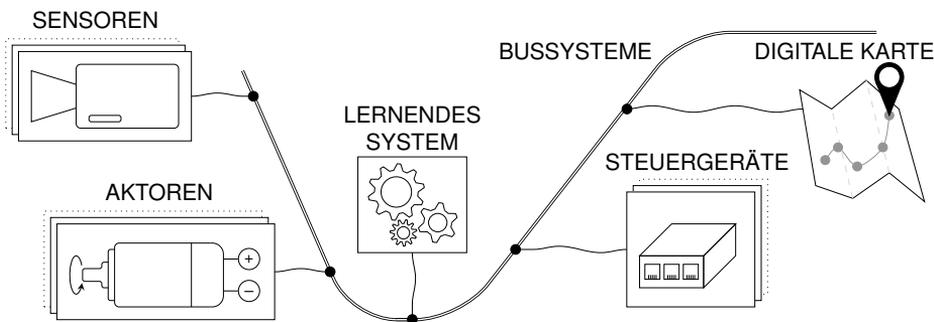


Abb. 3.1: Integration eines lernenden Systems in die bestehende IT-Infrastruktur von heutigen Fahrzeugen.

## 3.2 Annahmen und Einschränkungen

In diesem Abschnitt werden Annahmen und Einschränkungen für die vorliegenden Arbeit getroffen und erläutert.

Grundsätzlich wird davon ausgegangen, dass das lernende System einen Zugriff auf sämtliche Sensoren, Aktoren, Steuergeräte, Bussysteme und eine digitale Karte hat.

Damit die übliche Leistung von aktuellen Steuergeräten das lernende System nicht limitiert, wird in der vorliegenden Arbeit das Steuergerät des lernenden Systems mit einem handelsüblichen Computer ersetzt, wodurch ausreichend Leistung verfügbar ist.

Der verwendete Versuchsträger verfügt nicht über die Möglichkeit die verschiedenen Fahrspuren einer Straße, beispielsweise mit einer Stereokamera, auseinanderzuhalten. Aus diesem Grund wird jede Straße als einspurige Straße betrachtet und damit Sensormesswerte, die auf unterschiedlichen Spuren erhoben wurden, als gleich behandelt.

Durch die bereits hohe Genauigkeit in der Positionsbestimmung durch das DGPS und zu erwartenden weiteren Verbesserungen in naher Zukunft, wird die Position des Fahrzeugs immer als fehlerfrei angenommen. Ebenfalls wird das Vorhandensein einer aktuellen digitalen Karte angenommen.

Das lernende System in der vorliegenden Arbeit soll nur skalare Messgrößen verarbeiten können. Komplexere Messwerte wie Reflexionen eines Radars, (Stereo-)Kamerabilder oder Punktwolken von LiDAR Sensoren sollen nicht direkt verarbeitet werden.

Im Bereich der Ingenieurwissenschaft und Statistik werden seit Jahrzehnten Werkzeuge aus der Wahrscheinlichkeitstheorie verwendet, damit Systeme aus Daten lernen können [52]. Ein Gebiet davon ist das maschinelle Lernen, das unter anderem versucht Gesetzmäßigkeiten oder Muster in den Eingabewerten herauszufinden und Eingabewerte in diese Muster zu klassifizieren [89]. Werden bei den Eingabewerten jedoch keine verschiedenen Muster erwartet oder soll keine Klassifikation durchgeführt werden, können andere statistische Werkzeuge benutzt werden. Dies kann eine Schätzung der Wahrscheinlichkeitsverteilung eines Eingabewerts mit Hilfe von Schätzverfahren sein [54, 19]. Die geschätzte Wahrscheinlichkeitsverteilung stellt eine neutrale Repräsentation des Eingabewerts dar ohne eine Interpretation des Eingabewerts durchzuführen. Diese interpretationsfreie Repräsentation von Eingabewerten wird in Forschungsarbeiten für vielfältige Anwendungen im Automobilkontext verwendet. Sie ermöglicht es neue Informationen aus Messgrößen zu generieren [104, 93] und wird auch für lernende Systeme im Fahrzeug betrachtet [47]. In der vorliegenden Arbeit soll eine neutrale, interpretationsfreie Repräsentation der Eingabewerte gelernt werden.

Verschiedene Fahrstile und Fahrzeugcharakteristiken können sich auf die Daten auswirken, die das lernende System erfasst. Deutliche Unterschiede in Sensordaten können zwischen einem sportlichen und defensiven Fahrer ausgemacht werden [75]. Auch die Fahrzeugart bestimmt die

Limitierungen und den Wertebereich von Sensorwerten, wie beispielsweise die Beschleunigung und die Höchstgeschwindigkeit eines Sportwagens im Vergleich zu einem Lastkraftwagen. Das direkte Zusammenführen der Sensordaten verschiedener Fahrzeuge und Fahrer würde zu einer allgemeingültigen Repräsentation führen. Diese Repräsentation würde jedoch einen individuellen Fahrer in seinem Fahrzeug ungenauer beschreiben, als wenn eine explizite Kombination von Fahrer und Fahrzeug als Quelle für das lernende System verwendet wird. Aus diesem Grund wird das Erhalten von Informationen aus externen Quellen, wie beispielsweise durch Kommunikation mit anderen Fahrzeugen oder Verkehrsinfrastruktur, nicht betrachtet. Das Fahrzeug soll autark arbeiten, wird als geschlossenes System betrachtet und nur von einem Fahrer geführt.

Viele Faktoren können die Sensormesswerte direkt oder indirekt beeinflussen. Temperaturen, Helligkeit und Verkehr sind Beispiele dafür. Das Fahrzeug wird üblicherweise langsamer als normal geführt, wenn die Straßen durch niedrige Temperaturen und Witterung glatt sind, es dunkel ist oder der Verkehr dicht ist. Es existieren viele Faktoren und noch mehr Kombinationen dieser Faktoren, die Modi genannt werden. Das lernende System muss Sensordaten verschiedener Modi getrennt verarbeiten, was eine Erkennung des aktuellen Modus voraussetzt. Diese Modus-Erkennung liegt außerhalb des Fokus der vorliegenden Arbeit und es wird deswegen ein stets gleicher Modus angenommen.

Zusammenfassend gelten für das lernende System der vorliegenden Arbeit folgende Annahmen und Einschränkungen:

- Das lernende System hat Zugriff auf alle Sensormesswerte im Fahrzeug.
- Das lernende System kann Fahrdynamik- und HMI-Aktoren verwenden, direkt oder indirekt über ein FAS.
- Die Komponenten des lernenden Systems sind durch einen deterministischen Bus miteinander verbunden und Nachrichten werden ohne Verzögerung übermittelt.
- Das Steuergerät des lernenden Systems wird beim Prototyp durch einen handelsüblichen Computer ersetzt.
- Mehrspurige Straßen werden als einspurig betrachtet.
- Die GPS Position des Fahrzeugs wird als fehlerfrei angenommen.

- Es wird das Vorhandensein einer aktuellen digitalen Karte angenommen.
- Der Lernprozess soll skalare Messgrößen verarbeiten können.
- Die Informationen sollen interpretationsfrei gelernt werden.
- Das Fahrzeug wird als geschlossenes System betrachtet.
- Das Fahrzeug wird nur von einem Fahrer geführt.
- Umgebungsbedingungen des Fahrzeugs werden als konstant angenommen und befindet sich damit stets im gleichen Modus.

### 3.3 Leitbeispiel: Adaptiver Kurvenwarner

Für die vorliegende Arbeit wird der *adaptive Kurvenwarner* als Beispiel verwendet, das als Assistenzstrategie eine Warnung des Fahrers verwendet. Der warnende Charakter dieses Systems stellt von den vorgestellten Anwendungen die höchsten Anforderungen an das lernende System:

- Die Warnung des adaptiven Kurvenwarners muss zum richtigen Zeitpunkt ausgegeben werden.

Das veränderte Schaltverhalten der adaptiven Automatikschaltung hat ebenfalls Zeitschranken, deren Überschreitung jedoch nicht zum Versagen des Systems oder zu einer kritischen Fahrsituation führt.

- Für den adaptiven Kurvenwarner muss ein detailliertes Geschwindigkeitsprofil einer Kurve gelernt werden. Dazu müssen Messwertreihen entlang einer Strecke vom System verarbeitet und zusammengeführt werden können.

Statt der Messgröße *Geschwindigkeit* kann jede beliebige skalare Messgröße gelernt werden, so kann leicht auch die Steigung der Straße und ein Beschleunigungsprofil für die adaptive Automatikschaltung gelernt werden.

Der adaptive Kurvenwarner stellt aufgrund seiner zeitkritischen Assistenz sehr hohe Anforderungen an das lernende System und deckt die Anforderungen der adaptiven Automatikschaltung ab. Aus diesen Gründen wird der adaptive Kurvenwarner in der vorliegenden Arbeit als Leitbeispiel verwendet.

### 3.3.1 Beschreibung der Funktionalität

Im Folgenden wird das Konzept eines adaptiven Kurvenwarners, der auf einem lernenden System basiert, vorgestellt und seine Funktion und Arbeitsweise im Detail beschrieben.

Die Funktion des adaptiven Kurvenwarners lässt sich in drei Bereiche unterteilen:

1. **Erkennung von Kurven:** Die Erkennung von Kurven kann über passend gewählte Schwellwerte für die Querbearbeitung und Geschwindigkeit realisiert werden. Womit ebenso die Position des Kurvengangs und -ausgangs bestimmt werden kann. Kurven können so unabhängig von bereits vorhandenen Informationen, wie die aus der digitalen Karte, erkannt werden.

Weitere Verbesserungen dieser Kurvenerkennung, beispielsweise über die Auswertung des GPS Tracks des Fahrzeug, sind möglich, stehen jedoch nicht im Mittelpunkt der vorliegenden Arbeit und werden deswegen nicht weiter betrachtet.

2. **Lernen der Geschwindigkeit:** Wird eine Kurve erkannt, können Sensordaten (wie die Geschwindigkeit) von Kurvengang bis -ausgang gelernt werden. Mit jeder Fahrt durch diese Kurve kann das System genauer vorhersagen, wie hoch die wahrscheinlichste Geschwindigkeit ist, die der Fahrer dort fahren wird.

Ändern sich Gegebenheiten der Kurve, können diese Änderungen vom lernenden System berücksichtigt werden. Eine solche Änderung könnte beispielsweise ein neues Schlagloch sein, weswegen der Fahrer diese Kurve langsamer durchfährt. Das lernende System kann diese reduzierte Geschwindigkeit erkennen und die Informationen über diese Kurve aktualisieren und korrigieren.

3. **Warnung vor zu hoher Geschwindigkeit:** Mit Kenntnis über die Position einer Kurve und das typische Verhalten des Fahrers (Kurvengeschwindigkeit, Bremsung vor der Kurve) kann eine Warnung vor zu hoher Geschwindigkeit vor der Kurve ausgegeben werden. Dazu wird die Position und die aktuelle Geschwindigkeit des Fahrzeugs verwendet und mit der typischen Bremsung des Fahrers berechnet, wie schnell das Fahrzeug in der Kurve wäre. Überschreitet die Kurvengeschwindigkeit das gelernte Geschwindigkeitsprofil dieser Kurve, wird eine Warnung ausgegeben. So hat der Fahrer ausreichend

Zeit die Geschwindigkeit des Fahrzeugs vor der Kurve zu reduzieren.

Mit einem lernenden System können die in Abschnitt 2.2 genannten Einschränkungen bisheriger Kurvenwarner vermieden werden:

- Der Fahrstil des Fahrers in einer Kurve kann über die gefahrene Geschwindigkeit gelernt werden. Der Fahrer berücksichtigt alle Eigenschaften der Kurve, welche für ihn die Geschwindigkeit beeinflussen. Damit erlernt das System indirekt auch diese Eigenschaften der Kurve. Die so berechneten Warnungen und Geschwindigkeitsempfehlungen sind dadurch individuell auf den Fahrer angepasst und entsprechen seinem Fahrstil.
- Unbekannte Kurven (beispielsweise nicht in der digitalen Karte verzeichnete) können vom lernenden System erkannt und zum Datenspeicher hinzugefügt werden. Auf diese Weise ist auch für diese Kurven eine Warnung möglich.
- Änderungen von Gegebenheiten einer Kurve können vom lernenden System erfasst und die entsprechenden Informationen korrigiert werden. Dies macht beispielsweise eine manuelle Aktualisierung der digitalen Karte für den adaptiven Kurvenwarner überflüssig.

**Berechnung der Warnung:** Die Warnung wird mit Hilfe der *gleichmäßig beschleunigten Bewegung* berechnet. Dabei wird die aktuelle Fahrzeuggeschwindigkeit  $v_0$ , die typische Verzögerung des Fahrers  $a$  und ein korrigierter Abstand zur Kurve  $s_{korr}$  genommen, um die Fahrzeuggeschwindigkeit in der Kurve zu bestimmen:

$$v_{Kurve} = -\sqrt{2 \cdot a \cdot s_{korr}} + v_0 \quad (3.1)$$

Überschreitet diese berechnete Geschwindigkeit den gelernten Wert, wird eine Warnung ausgegeben. Der korrigierte Abstand  $s_{korr}$  in Gleichung 3.1 berechnet sich aus dem Abstand zur Kurve  $s$  und der Verluststrecke durch die Dauer verschiedener Reaktionen (nach [12]) in Abhängigkeit der Geschwindigkeit  $v$ :

- $s_{Reakt}(v)$  ist die verlorene Strecke durch die Reaktion des Fahrers (Dauer:  $300ms$ ).
- $s_{Umsetz}(v)$  ist die verlorene Strecke durch das Umsetzen vom Fuß des Fahrers vom Gas- zum Bremspedal (Dauer:  $200ms$ ).

### 3.3 Leitbeispiel: Adaptiver Kurvenwarner

- $s_{Ansprech}(v)$  ist die verlorene Strecke durch die Ansprechzeit der hydraulischen Bremse (Dauer:  $50ms$ ).
- $s_{Aufbau}(v)$  ist die verlorene Strecke durch die Schwellzeit der hydraulischen Bremse bis zur maximalen Bremswirkung (Dauer:  $200ms$ ).

Damit ergibt sich für den korrigierten Kurvenabstand, der die Kurvenwarnung früher auslösen lässt:

$$s_{korrr} = s - s_{Reakt}(v) - s_{Umsetz}(v) - s_{Ansprech}(v) - s_{Aufbau}(v) \quad (3.2)$$

**Unterschiede durch den Detailgrad gelernter Informationen:** Die Warnung unterscheidet sich abhängig vom Detailgrad der gelernten Informationen über die Kurvengeschwindigkeit. Ein einzelner, über die gesamte Kurve gemittelter Messwert kann nicht alle Eigenschaften einer Kurve berücksichtigen. Eine Messwertreihe von Kurveneingang bis -ausgang hingegen bildet beispielsweise die Länge einer Kurve oder die Dynamik der Geschwindigkeit innerhalb der Kurve ab.

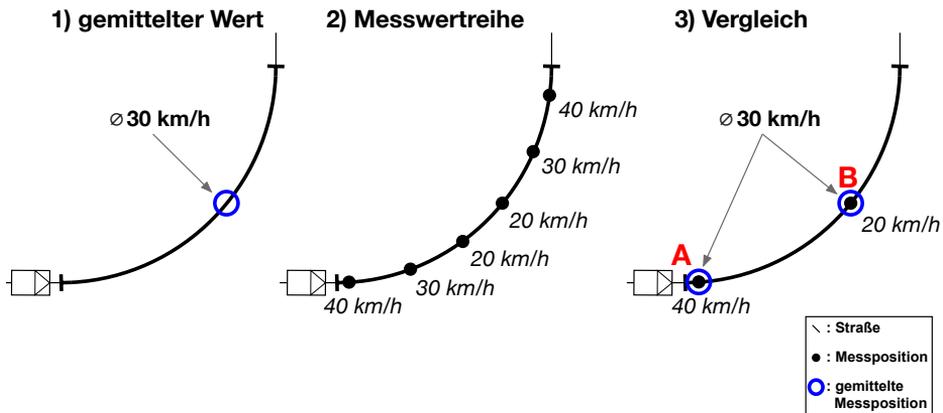


Abb. 3.2: Detailgrad gelernter Informationen wirkt sich auf die Warnung des adaptiven Kurvenwarners aus.

Abbildung 3.2 zeigt diesen Unterschied: In 1) ist eine gemittelte Geschwindigkeit von  $30 \frac{km}{h}$  dargestellt, welche für die gesamte Kurve gilt. In 2) ist die dazugehörige Messwertreihe dargestellt, aus der das Mittel berechnet wurde. Zu dieser Messwertreihe gehören die gezeigten sechs Messwerte. In 3) ist ein Vergleich von A und B dargestellt:

- A:** Die gemittelte Geschwindigkeit von  $30 \frac{km}{h}$  wird am Kurveneingang angenommen. In Wirklichkeit wurde an dieser Position jedoch laut Messwertreihe eine Geschwindigkeit von  $40 \frac{km}{h}$  gemessen. Der Kurvenwarner würde somit bei der gemittelten Geschwindigkeit am Kurveneingang möglicherweise eine falsche Warnung aussprechen, da die Geschwindigkeit um  $10 \frac{km}{h}$  zu niedrig angenommen wurde.
- B:** Die gemittelte Geschwindigkeit von  $30 \frac{km}{h}$  wird hier in der Mitte der Kurve angenommen. Die korrekte Geschwindigkeit laut Messwertreihe wäre jedoch  $20 \frac{km}{h}$ . Der Kurvenwarner würde mit der gemittelten Geschwindigkeit in diesem Fall möglicherweise fälschlich keine Warnung ausgeben, da die Geschwindigkeit um  $10 \frac{km}{h}$  zu hoch angenommen wurde.

Mit einer Messwertreihe kann eine korrekte Kurvenwarnung ausgegeben werden, die mit einem gemittelten Messwert in einer solchen Situation nicht möglich wäre.

Darüber hinaus enthält die Messwertreihe der Geschwindigkeit auch weitere Informationen, welche die Warnung beeinflussen kann. Beispielsweise ein variierender Kurvenradius: Wird die Kurve im Verlauf enger, muss die Warnung möglicherweise später erfolgen als bei einer Kurve, die im Verlauf weiter wird. Der gemittelte Geschwindigkeitswert der sich verengenden und sich öffnenden Kurve kann der gleiche sein. Wird nur ein gemittelter Wert verwendet, kann der Kurvenwarner keine Unterscheidung dieser beiden Kurven durchführen.

Dieser Vergleich zeigt, dass der Detailgrad der gelernten Informationen wichtig für die Funktion des adaptiven Kurvenwarners und allgemein für FAS ist. Eine **Messwertreihe**, die im Verlauf einer ganzen Situation aufgenommen wurde, ermöglicht die Realisierung des adaptiven Kurvenwarners in der beschriebenen Form und soll deshalb vom lernenden System unterstützt werden.

#### 3.3.2 Anforderungen an das lernende System

Dieser Abschnitt ist in die drei Komponenten eines lernenden Systems (*Datenspeicher*, *Lernprozess* und *Änderungen*) gegliedert und beschreibt darin eingeordnet die Anforderungen des adaptiven Kurvenwarners an das lernende System.

**Datenspeicher:** Das lernende System muss die gelernten Informationen in einem Datenspeicher für zukünftige Fahrten persistent speichern können. Damit können erkannte, bis dahin unbekannte Kurven für den adaptiven Kurvenwarner gespeichert werden und ermöglichen eine Warnung bei darauf folgenden Fahrten. Ein dafür benötigter Datenspeicher ist in heutigen Serienfahrzeugen nicht vorhanden.

Die Leistung des Datenspeichers und die Adressierung der Daten ist für das rechtzeitige Laden und Speichern der gelernten Informationen entscheidend. Werden die Daten nicht rechtzeitig vor dem benötigten Warnzeitpunkt geladen, kann der adaptive Kurvenwarner eine Warnung nur verspätet oder gar nicht ausgeben. Das Bereitstellen der gelernten Informationen ist damit zeitkritisch.

**Anforderung:** Für die zeitkritische Warnung des adaptiven Kurvenwarners, ist eine Bereitstellung der Informationen durch den Datenspeicher vor dem Warnzeitpunkt notwendig.

**Lernprozess:** Ein modernes Fahrzeug kann während der Fahrt mit seinen Bussystemen Datenmengen in der Größe von  $0.875\text{MByte}$  pro Sekunde ( $3.15\text{GByte}$  pro Stunde) übertragen. Damit diese Daten für FAS nutzbar werden, müssen sie verarbeitet und mit zuvor gelernten Informationen zusammengeführt werden. Dieser Vorgang wird als *Lernprozess* bezeichnet. Mit den neuen Sensordaten jeder weiteren Fahrt, können die gelernten Informationen präzisiert werden und ermöglichen genauere Vorhersagen. Beispielsweise kann so das lernende System mit jeder Fahrt durch eine Kurve genauer vorhersagen, mit welcher Geschwindigkeit der Fahrer bei der nächsten Fahrt durch diese Kurve fahren wird.

Der Lernprozess wird durch einen Algorithmus repräsentiert, der die Sensordaten verschiedener Fahrten zusammenführt. Die gelernten Informationen sollen mit hohem Detailgrad und möglichst platzsparend zusammengeführt werden, wie die Geschwindigkeit in Kurven für den adaptiven Kurvenwarner. Darüber hinaus sollen nur für die FAS relevante Daten gelernt werden.

**Anforderung:** Der Lernprozess muss Messwertreihen aus verschiedenen Fahrten platzsparend zusammenführen können.

**Anforderung:** Es sollen benötigte Sensordaten in relevanten Situationen selektiert werden können und nur diese aufgrund limitierter Ressourcen gelernt werden.

### 3 Lernende Systeme im Fahrzeug

---

Ein lernendes System ist auf freien Speicher angewiesen, um neue Informationen zu lernen. Ist der Speicher voll, müssen Informationen gelöscht werden, um die Funktion des Systems nicht einzuschränken. Für eine gezielte Löschung, muss eine Bewertung der gelernten Informationen möglich sein.

**Anforderung:** Für eine gezielte Löschung bei vollem Speicher, muss eine Bewertung von gelernten Informationen möglich sein.

**Änderungen:** Die Umgebung des Fahrzeugs verändert sich, wodurch gelernte Informationen ungültig werden können, da sie nicht mehr der Realität entsprechen. Ungültige Informationen können die Funktion von FAS einschränken oder das System unwirksam werden lassen. Verringert sich beispielsweise die sichere Geschwindigkeit durch eine Kurve durch ein neues Schlagloch, würde der Kurvenwarner ohne Berücksichtigung dieser geänderten Geschwindigkeit fälschlich keine Warnung ausgeben.

Aus diesem Grund ist es notwendig im lernenden System eine Erkennung von Änderungen durchzuführen, um ungültig gewordene Informationen zu korrigieren. Damit können die gelernten Informationen immer aktuell gehalten werden. Je besser und schneller Änderungen erkannt werden, desto schneller kann die uneingeschränkte Funktion von FAS wiederhergestellt werden.

**Anforderung:** Änderungen müssen möglichst schnell und zuverlässig erkannt werden, um dem adaptiven Kurvenwarner möglichst keine ungültigen Daten zur Verfügung zu stellen.

#### 3.4 Vorhandene Ansätze für lernende Systeme

Der Einsatz von lernenden Systemen im Fahrzeug war bereits mehrfach Gegenstand der Forschung. Relevante Arbeiten aus diesem Bereich werden im Folgenden vorgestellt. Dabei wird hervorgehoben, welche der aufgestellten Anforderungen für den adaptiven Kurvenwarner nicht erfüllt werden.

### 3.4.1 Ein lernendes System zur Korrektur der digitalen Karte

Die Dissertation mit dem Titel:

„Umgebungserfassung auf Basis lernender digitaler Karten zur vorausschauenden Konditionierung von Fahrerassistenzsystemen“

von Michael Schraut [75] wird im weiteren Verlauf dieses Abschnitts mit *Schraut* abgekürzt.

Die Arbeit von *Schraut* beschäftigt sich primär mit einem lernenden System zur geometrischen Korrektur von Straßen und Kartendaten anhand gemessener Fahrzeugbewegungen.

**Datenspeicher:** In *Schraut* wird ein relationales Datenbankmodell für die Speicherung von Informationen verwendet. Die zu speichernden Daten werden in Verhältnis zueinander gesetzt (Attribute, wie Brücken und Straßennamen, gehören zu Straßensegmenten) und können dadurch als Relationen beschrieben und organisiert werden.

Die Informationen werden über Elemente aus der digitalen Karte, sogenannten Segmenten und Features, in der Datenbank organisiert. Anhand dieser Elemente, die einen Bezeichner in der Karte besitzen, wird eine Verbindung zwischen der digitalen Karte und der Datenbank für die gelernten Informationen hergestellt.

Der verwendete Datenspeicher wurde in der Arbeit nicht auf eine schnelle Bereitstellung der Daten überprüft.

**Lernprozess:** In der Arbeit von *Schraut* werden die gelernten Messwerte für zwei Anwendungen eingesetzt:

1. Korrektur von Straßengeometrien in der digitalen Karte.
2. Ein prototypisches FAS namens „NavSpeed“, welches eine empfohlene Geschwindigkeit für eine bekannte Route ausgibt, die aus unterschiedlichen Eigenschaften berechnet wird. Zu diesen Eigenschaften gehören unter anderem drei grobe Kategorien für die Sportlichkeit des Fahrers und die Geometrie der Kurven auf der gefahrenen Route. Die Akzeptanz dieser Geschwindigkeitsempfehlung in Kurven war durchwachsen, 50% der Fahrer stufen sie als „nicht transparent“ ein (siehe auch Abschnitt 2.2).

Für diese Anwendungen werden ausschließlich einzelne Messwerte gelernt. Das Lernen von Messwertreihen wurde in *Schraut* nicht betrachtet. Ebenfalls wurde nicht beschrieben, wie der Lernprozess die gemessenen Straßenattribute aus verschiedenen Fahrten zusammenführt.

Es wird angesprochen, dass korrigierte Straßengeometrien überflüssig und in Bezug auf den Speicherplatz problematisch werden können. Der Zustand eines vollen Speichers wird jedoch nicht direkt betrachtet und ebenfalls kein Verfahren zur gezielten Löschung von gelernten Informationen vorgeschlagen.

**Änderungen:** Eine Erkennung von Änderungen wird in *Schraut* nicht vorgestellt. Änderungen werden zwar betrachtet, jedoch wird als Lösung lediglich das manuelle Einpflegen dieser Änderungen vorgeschlagen. Eine automatisierte Korrektur von ungültig gewordenen Informationen im lernenden System ist nicht vorgesehen.

#### 3.4.2 Lernen von Routencharakteristiken für vorausschauende Fahrstrategien

Die Dissertation mit dem Titel:

„A System for the Provision and Management of Route Characteristic Information to Facilitate Predictive Driving Strategies”

von Anne Carlsson [13] wird im weiteren Verlauf dieses Abschnitts mit *Carlsson* abgekürzt.

Die Arbeit von *Carlsson* beschäftigt sich mit dem Erlernen von Charakteristiken, wie beispielsweise Geschwindigkeitslimits, entlang gefahrener Routen, um den Fahrer frühzeitig über diese zu informieren.

**Datenspeicher:** Es werden in *Carlsson* keine konkreten Informationen bezüglich des eingesetzten Datenspeichers gegeben.

Die Adressierung der Daten erfolgt über die GPS Position, an der sie erhoben wurden. Sollen Daten geladen werden, wird über die aktuelle GPS Position des Fahrzeugs eine Umkreissuche durchgeführt. Es werden keine Kontextinformationen, wie das Straßennetz, für das Speichern und Laden von Daten verwendet.

Die Eignung des verwendeten Datenspeichers für den Einsatz im Fahrzeug wurde nicht näher untersucht. Auch wurden keine Betrachtungen bezüglich der Zugriffszeiten durchgeführt.

**Lernprozess:** In *Carlsson* werden sogenannte „Situationen“ gelernt, die Geschwindigkeitsprofile, Verläufe von Kurven und Steigungen sein können. Dazu werden gefahrene „Routen“ gelernt und ihnen die erkannten Situationen zugeordnet. Die Routen selbst enthalten keine Informationen, sondern nur Verweise auf die gelernten Situationen auf dieser Route.

Die gelernten Informationen in *Carlsson* sind meist einzelne Messwerte, für die ausführlich beschrieben wird, wie eine Zusammenführung funktioniert. Es wird auch das Zusammenführen von Messwertreihen erwähnt. Wie genau diese Zusammenführung funktioniert, ist jedoch nicht näher beschrieben. Mit den dazu gegebenen Informationen in *Carlsson* kann keine Evaluation oder Implementierung des Lernprozesses für Messwertreihen erfolgen.

Ein voller Speicher wird in *Carlsson* nicht betrachtet. Es werden deswegen auch keine Maßnahmen vorgestellt, um Informationen gezielt zu löschen.

**Änderungen:** Für jede gelernte „Situation“ wird in *Carlsson* ein Verhältnis von der Anzahl der Erkennungen zu der Anzahl der Vorbeifahrten dieser Situation berechnet. Unterschreitet dieses Verhältnis einen Schwellwert, werden die dazugehörigen Daten gelöscht. Bei einer großen Anzahl an Vorbeifahrten und erfolgreichen Erkennungen einer Situation kann dieses Verhältnis sehr stabil sein und deshalb bei Änderungen nicht ausreichend schnell zur Löschung führen. In *Carlsson* wird deswegen vorgeschlagen, nur die letzten fünf Messungen für die Berechnung dieses Verhältnisses zu verwenden. Dies führt zu einer schnellen Anpassung bei Änderungen, jedoch auch zu einer höheren Instabilität gelernter Informationen und unter Umständen zu einer verfrühten Löschung. Eine Erkennung von Änderungen wird nicht betrachtet.

#### 3.4.3 Reichweitenprädiktion von elektrischen Fahrzeugen über ein lernendes System

Die Dissertation mit dem Titel:

„Selbstlernende, zuverlässigkeitsorientierte Prädiktion energetisch relevanter Größen im Kraftfahrzeug“

von Tobias Mauk [47] wird im weiteren Verlauf dieses Abschnitts mit *Mauk* abgekürzt.

Das Hauptaugenmerk der Arbeit von *Mauk* liegt in der Verwendung eines lernenden Systems zur Prädiktion der verbleibenden Reichweite von elektrischen Fahrzeugen mit Berücksichtigung von Routeneigenschaften, wie Steigungen, und einer vorausschauenden Ladestrategie der im Fahrzeug verbauten Akkumulatoren.

**Datenspeicher:** In *Mauk* wird von der Notwendigkeit eines Datenspeichers gesprochen, jedoch nicht näher spezifiziert, ob eine Datenbank verwendet wird.

Genaue Informationen zur Adressierungsart der gelernten Daten, sind in *Mauk* nicht zu finden. Es wird jedoch die Speicherung der gelernten Daten mit Hilfe eines Graphen erwähnt (mehr Details dazu im folgenden Abschnitt *Lernprozess*).

Auch in *Mauk* wurde der Datenspeicher nicht auf Eignung für den Einsatz im Fahrzeug oder seine Zugriffszeiten untersucht.

**Lernprozess:** In *Mauk* werden „gewünschte Strecken-Attribute“ gelernt. Dabei wird der Fokus auf wertkontinuierliche Größen gelegt. Größen mit diskreten Werten (beispielsweise die Gangstufe) werden nicht näher betrachtet. Es werden zwar Messwertreihen betrachtet, jedoch wird die Zusammenführung nur mit dem neusten Messwert an ausgewählten Positionen, den sogenannten „Trackpoints“, durchgeführt. So werden nur einzelne Messwerte zusammengeführt. Messwertreihen können mit dem vorgestellten Ansatz nicht zusammengeführt werden.

Die Speicherung der gelernten Informationen wird in Form eines Graphen durchgeführt. Die Abstände der „Trackpoints“, den Knoten im Graph, betragen in Abhängigkeit zur Geschwindigkeit zwischen 10 – 100m und bilden die Kante zwischen zwei Knoten. Diese Abstände zwischen den gelernten Messwerten genügen nicht dem geforderten Detailgrad.

In *Mauk* wird ein Verfahren vorgestellt, mit dem Sensordaten im Fahrzeug iterativ zusammengeführt werden können. In diesem Fall bezeichnet *iterativ* die Möglichkeit, Messwerte einzeln nacheinander zusammenzuführen. Dies wird mit dem Schätzverfahren namens *Kerndichteschätzung* durchgeführt, welches die Wahrscheinlichkeitsverteilung von Messungen schätzen kann, ohne Annahmen über die Verteilung treffen zu müssen (mehr Details dazu folgen in Abschnitt 3.4.5). In *Mauk* wird anstatt sämtlicher Messwerte nur die Verteilung dieser Messwerte gespeichert, was bei

einer „geeigneten“ Diskretisierung zu einem vergleichsweise geringen, konstanten Speicherbedarf führt (auch für  $n \rightarrow \infty$  Messungen).

Ein voller Speicher wird in dieser Arbeit nicht betrachtet. Es wurden deswegen auch keine Maßnahmen vorgestellt, um Informationen gezielt zu löschen.

**Änderungen:** In *Mauk* wird eine Erkennung von Änderungen vorgestellt, die auf der geschätzten Verteilung der Kerndichteschätzung basiert. Die Erkennung, „Eskalationsstufe“ genannt, ermöglicht die Löschung durch Änderungen ungültig gewordener Informationen. Nachteil der vorgestellten Erkennung ist die Abhängigkeit zur geschätzten Verteilung. Wurde die Verteilung (beispielsweise durch eine ungünstige Konfiguration der Kerndichteschätzung) ungenau oder falsch geschätzt, könnte auch die Erkennung von Änderungen negativ beeinflusst werden.

#### 3.4.4 Fazit vorhandener Ansätze

Ein zusammenfassendes Fazit für die vorgestellten vorhandenen Ansätze für lernende Systeme im Fahrzeug:

- **Datenspeicher:** Alle betrachteten Arbeiten beschäftigen sich nur am Rande mit dem Datenspeicher des lernenden Systems oder der dafür benötigten Adressierung der Informationen. In keiner Arbeit wird die verwendete Datenbank auf dessen Zugriffszeiten getestet oder eine Analyse für ein passendes Datenbankmodell durchgeführt. Die Wahl des jeweiligen Datenbankmodells ist in keiner Arbeit begründet worden.

Keine Arbeit hat den Fall betrachtet, wie mit einem vollen Datenspeicher umgegangen wird. Jedes vorgestellte lernende System funktioniert nur, wenn Daten gespeichert werden können. Das Entfernen von Daten bei einem vollen Speicher ist in den vorgestellten Arbeiten nur am Rand und in Sonderfällen betrachtet worden.

- **Lernprozess:**

Es wird in keiner Arbeit ein Verfahren vorgestellt, das in der Lage ist, Messwertreihen mit konstantem Speicherbedarf iterativ zusammenzuführen.

Eine Selektion von relevanten Daten wird in jeder Arbeit durchgeführt. Eine detaillierte Erläuterung über die Arbeitsweise der Selektion wird jedoch nicht gegeben.

- **Änderungen:** Eine Erkennung von Änderungen und der Umgang mit veralteten Informationen wird lediglich in *Mauk* näher betrachtet. Die dort vorgestellte Änderungserkennung basiert auf der geschätzten Verteilung der Kerndichteschätzung. Damit ist diese Änderungserkennung direkt von der Qualität der geschätzten Verteilung abhängig. Die Leistungsfähigkeit der Erkennung kann deswegen bei einer ungünstig konfigurierten Kerndichteschätzung eingeschränkt sein. Eine qualitative Verbesserung wäre eine Erkennung von Änderungen unabhängig und losgelöst vom Lernprozess.

In Abbildung 3.3 sind die gemeinsamen Komponenten der lernenden Systeme aus den vorgestellten Arbeiten dargestellt. Jedes System führt eine Selektion von relevanten Sensormesswerten durch und übergibt diese dem Lernprozess, welcher einzelne Messwerte mittelt oder dessen Wahrscheinlichkeitsverteilung schätzt. Die so zusammengeführten Informationen werden in einem Speicher persistent abgelegt und den FAS zur Verfügung gestellt. Nur in *Mauk* wird eine Änderungserkennung basierend auf den Ergebnissen des Lernprozesses durchgeführt.

#### 3.4.5 Weitere Beiträge mit Relevanz für lernende Systeme

In diesem Abschnitt werden weitere Beiträge und Informationen vorgestellt, die eine Relevanz für das lernende System der vorliegenden Arbeit haben.

**Lernprozess:** In der Statistik kann zwischen zwei Bereichen unterschieden werden: Die parametrische Statistik und die *nicht*parametrische Statistik. Bei parametrischen Schätzverfahren muss die Verteilung bekannt sein oder eine Verteilung angenommen werden. Bei der nichtparametrischen Statistik muss keine Annahme über die Art der Verteilung getroffen werden, die geschätzt werden soll [11]. So kann ohne Vorwissen über die zu erwartenden Messwerte deren Wahrscheinlichkeitsverteilung geschätzt werden.

Für die Wahrscheinlichkeitsverteilung der Sensormesswerte können in der Regel keine allgemeingültigen Verteilungen angenommen oder bestimmt werden. Aus diesem Grund bietet sich die nichtparametrische Statistik für

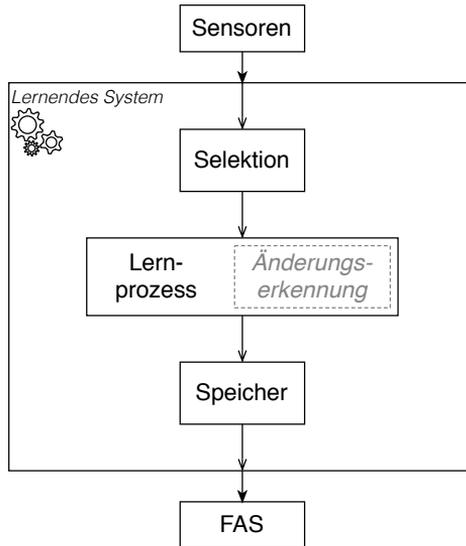


Abb. 3.3: Gemeinsame Komponenten der lernenden Systeme aus vorgestellten Arbeiten.

diese Aufgabe an. Die Kerndichteschätzung ist ein bekanntes und das einzige Schätzverfahren der nichtparametrischen Statistik. Sie wird in vielen Bereichen der Forschung aufgrund ihrer besonderen Eigenschaften erfolgreich eingesetzt.

Im Bereich des Verkehrswesen wird sie verwendet, um Gebiete mit einer hohen Unfalldichte zu identifizieren [7, 103, 29]. Mit ihr ist es auch möglich, die Anzahl und Breite von Spuren auf einer Autobahn zu lernen [93], ebenso kann sie auch für eine Klassifikation angewendet werden [24]. Beispielsweise kann die Trajektorie von Fahrzeugen gelernt werden und für eine Klassifikation zur Vorhersage der Fahrzeugbewegung verwendet werden [104].

In der vorliegenden Arbeit wird für den Lernprozess wie in *Mauk* ebenfalls die Kerndichteschätzung für das Zusammenführen von Sensormesswerten verwendet. Damit können beliebige Messgrößen im Fahrzeug gelernt werden ohne deren Verteilung zu kennen.

**Änderungen:** Die Forschung im Bereich der Erkennung von Änderungen ist seit Dekaden aktiv [6] und noch Gegenstand aktueller Forschung [45, 86]. Änderungserkennungen können in zwei Kategorien eingeteilt werden: echtzeitfähige Erkennung und rückblickende Erkennung. Für die Er-

kennungen von Änderungen im Fahrzeug werden echtzeitfähige Verfahren benötigt, um schnellstmöglich Änderungen zu erkennen und darauf zu reagieren.

Durch die Verwendung von Verfahren zur Änderungserkennung in Fahrzeugen entstehen besondere Anforderungen. Die Komplexität und damit auch der Berechnungsaufwand des Verfahrens darf aufgrund der limitierten Ressourcen im Fahrzeug nicht zu groß sein. Das Verfahren muss in der Lage sein, Messwerte von Fahrzeugsensoren zu verarbeiten, und in Kombination mit der für die vorliegende Arbeit gewählten Kerndichteschätzung funktionieren.

## 3.5 Konkretisierte Ziele

Nach dem Evaluieren von vorhandenen Ansätzen können die Ziele der vorliegenden Arbeit genauer formuliert werden. Dafür werden die Ziele der vorliegenden Arbeit aus Abschnitt 1.2 aufgegriffen und im Folgenden konkretisiert:

### 1. Lernen von heterogenen Sensormesswertreihen.

Im Lernprozess der vorliegenden Arbeit soll für das Zusammenführen von Sensordaten die Kerndichteschätzung verwendet werden. Das Zusammenführen von Messwertreihen wurde nicht betrachtet und deswegen wurde auch die Heterogenität von Sensormesswertreihen durch beispielsweise deren Zykluszeit nicht betrachtet. Eine Zusammenführung von heterogenen Sensormesswertreihen ist mit den bisherigen Ansätzen nicht möglich.

**Ziel:** Der Lernprozess soll erweitert werden, damit heterogene Sensormesswertreihen mit der Kerndichteschätzung gelernt werden können.

### 2. Erkennung von ungültigen Informationen im lernenden System.

Die Änderungserkennung aus *Mauk*, die Eskalationsstufe, ist abhängig vom Lernprozess. Eine falsche oder ungenau geschätzte Wahrscheinlichkeitsverteilung der Kerndichteschätzung kann sich negativ auf diese Änderungserkennung auswirken. Eine vom Lernprozess unabhängige Änderungserkennung ist in bisherigen Arbeiten nicht vorgestellt worden.

**Ziel:** Es soll eine unabhängige und schnellere Änderungserkennung zur Detektion ungültiger Informationen im lernenden System erarbeitet werden.

### 3. Anwendung des lernenden Systems.

Bisher vorgestellte lernende Systeme im Fahrzeug genügen nicht den Anforderungen der vorgestellten verbesserten FAS. Der erweiterte Lernprozess der vorliegenden Arbeit in Verbindung mit der Änderungserkennung soll die Realisierung des prototypischen adaptiven Kurvenwarners ermöglichen.

Für das lernende System sind noch weitere periphere Komponenten erforderlich: Ein persistenter Datenspeicher, eine Selektion relevanter Sensordaten und eine Bewertung gelernter Informationen, um die unwichtigsten Informationen bei vollem Speicher löschen zu können. Der erweiterte Lernprozess zum Lernen von heterogenen Messwertreihen, die Änderungserkennung und die genannten Komponenten sollen implementiert, in ein lauffähiges Gesamtsystem integriert und im Versuchsträger demonstriert werden.

**Ziel:** Das lernende System soll als lauffähiges Gesamtsystem im Versuchsträger integriert und der adaptive Kurvenwarner realisiert werden. Dazu soll der für das Lernen von Messwertreihen erweiterte Lernprozess, die neu entwickelte Änderungserkennung, der Datenspeicher, die Selektion relevanter Daten und die Priorisierung von Informationen implementiert und als ein lernendes System zusammengefügt werden.

Analog zur Abbildung 3.3 werden in der Abbildung 3.4 die zu entwickelnden Komponenten der vorliegenden Arbeit dargestellt und in ein Gesamtsystem des lernenden Systems eingeordnet. Für den Lernprozess wird die Kerndichteschätzung (in der Abbildung mit *KDE* abgekürzt) verwendet.

Kapitel 4 beschäftigt sich mit der Erweiterung des Lernprozesses zum *Lernen von heterogenen Sensormesswertreihen*. Dazu werden bisherige Ansätze und deren Grenzen im Detail vorgestellt, eine Erweiterung des Lernprozesses entwickelt und anschließend das Lernen von heterogenen Sensormesswertreihen evaluiert.

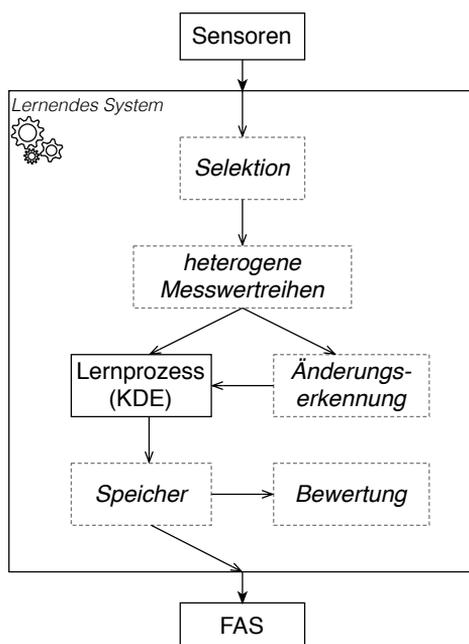


Abb. 3.4: Komponenten des lernenden Systems die in der vorliegenden Arbeit entwickelt werden sollen.

Kapitel 5 beschäftigt sich mit der *Erkennung von ungültigen Informationen im lernenden System*. Bisherige Arbeiten in diesem Bereich werden vorgestellt und eine verbesserte, schnellere Änderungserkennung entwickelt, die anschließend evaluiert und mit anderen Verfahren verglichen wird.

Kapitel 6 beschäftigt sich mit der *Anwendung des lernenden Systems*. Es wird ein lauffähiges Gesamtsystem des lernenden Systems im Versuchsträger implementiert. Die dazu benötigten peripheren Komponenten werden zuerst entwickelt und implementiert. Anschließend wird der adaptive Kurvenwarner mit Hilfe des lernenden Systems umgesetzt und im Versuchsträger mit echten Sensormesswerten demonstriert.

Kapitel 7 fasst diese Arbeit zusammen und gibt einen Ausblick.

# Lernen von heterogenen Sensormesswertreihen

---

Im folgenden Kapitel wird untersucht, wie ein Lernprozess basierend auf der Kerndichteschätzung auch für Messwertreihen verwendet werden kann, um detaillierte Verläufe von Sensormesswerten entlang einer Strecke zur Anwendung in FAS zu lernen.

Dazu wird zunächst in Abschnitt 4.1 detailliert auf die Kerndichteschätzung und deren Verwendung in einem Lernprozess eingegangen. Anschließend wird betrachtet, wie die Heterogenität von Messwertreihen den Lernprozess erschwert und die Notwendigkeit einer Messwertreihen-Transformation erklärt. Danach wird die Messwertreihen-Transformation entwickelt, um Messwertreihen zu homogenisieren und das Lernen von Messwertreihen mit der Kerndichteschätzung zu ermöglichen.

Abschließend wird in Abschnitt 4.2 die Funktion der Messwertreihen-Transformation mit generierten und echten Sensormesswertreihen aus dem Versuchsträger nachgewiesen und die gelernten Informationen präsentiert, die aus dem Lernen von Messwertreihen resultieren.

## 4.1 Der Lernprozess

In Arbeiten zu lernenden Systemen im Fahrzeug wurden mehrere Ansätze für einen Lernprozess vorgestellt. Dabei werden Sensordaten aus verschiedenen Fahrten zusammengeführt, beispielsweise durch die Schätzung der Wahrscheinlichkeitsverteilung von einzelnen Messwerten mit der Kerndichteschätzung [47].

*Messwertreihen* enthalten im Gegensatz zu einem *einzelnen Messwert* mehr Informationen und sind deshalb für die meisten FAS, wie dem adaptiven Kurvenwarner, von größerem Nutzen. In Abschnitt 3.3.1 und Abbildung 3.2 wurde der bedeutende Unterschied dieser beiden Datenformen für den adaptiven Kurvenwarner im Detail vorgestellt. Der Lernprozess muss deswegen auch Messwertreihen zusammenführen können. Der in [47]

vorgestellte Lernprozess basierend auf der Kerndichteschätzung ist bisher jedoch nur für einzelne Messwerte verwendbar.

Die Kerndichteschätzung ist für die vorliegende Arbeit aufgrund zwei besonderer Eigenschaften interessant:

1. Es ist ein nichtparametrisches Schätzverfahren und benötigt deswegen kein Vorwissen über die Verteilung der Messgröße.
2. Es ermöglicht das Zusammenführen von beliebig vielen Messwerten mit konstantem Speicherbedarf.

Aufgrund der Heterogenität der Messwertreihen ist eine Zusammenführung nicht eindeutig möglich. Deshalb wird in diesem Abschnitt eine neue Messwertreihen-Transformation entwickelt, die eine Zusammenführung über die Kerndichteschätzung möglich macht.

### 4.1.1 Bisheriger Lernprozess basierend auf der Kerndichteschätzung

In [47] wird bei der Zusammenführung im Lernprozess die Wahrscheinlichkeitsverteilung von zusammengehörenden Sensormesswerten geschätzt. Diese Verteilung wird anstatt der einzelnen Messwerte gespeichert, um Speicherplatz einzusparen. Die Schätzung der Wahrscheinlichkeitsverteilung der Messwerte wird mit der Kerndichteschätzung  $\hat{f}$  nach folgender Gleichung durchgeführt:

$$\hat{f}(v) = \frac{1}{nb} \sum_{i=1}^n K\left(\frac{v - v_i}{b}\right) \quad (4.1)$$

Dabei ist  $n$  die Anzahl der Messwerte und  $v_i$  mit  $i = 1 \dots n$  sind die Messwerte selbst.  $K()$  ist die Kernfunktion, für die in [47] der Epanechnikov-Kern [87] gewählt wird. Die Breite der Kernfunktion wird durch die Bandbreite  $b$  gewählt, die (wie in [47] vorgeschlagen) für jede Messgröße beziehungsweise jeden Sensor einmalig am Anfang per Hand (*Smoothing-by-Eye*) angepasst werden muss.  $\hat{f}$  berechnet den Wahrscheinlichkeitswert für  $v$ .

**Beispiel 4.1 (Geschätzte Verteilungen mit der Kerndichteschätzung):** *Abbildung 4.1 zeigt zwei Beispiele für mit der Kerndichteschätzung  $\hat{f}$  geschätzte Verteilungen (A und B) der Geschwindigkeit  $v$ . Die Verteilung A wurde aus verteilten Messwerten geschätzt und*

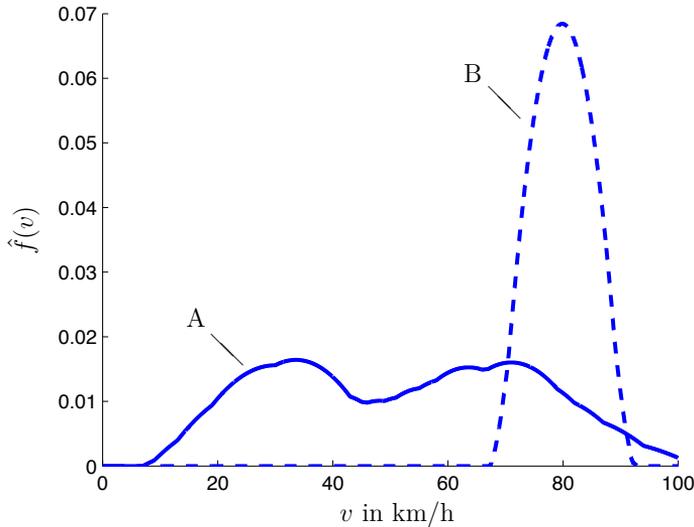


Abb. 4.1: Zwei unterschiedliche Wahrscheinlichkeitsverteilungen A und B der Geschwindigkeit  $v$  die mit der Kerndichteschätzung  $\hat{f}$  geschätzt wurden. (Grafik aus [47])

ist damit relativ breit (die einzelnen Messwerte werden in [47] nicht angeben). Es sind zwei (lokale) Maxima bei circa  $30 \frac{km}{h}$  und  $70 \frac{km}{h}$  zu sehen. Diese Geschwindigkeiten sind nach dieser Verteilung die wahrscheinlichsten, aber auch Geschwindigkeiten im Bereich  $10 - 100 \frac{km}{h}$  sind nach dieser Verteilung möglich. Die Verteilung B besteht nur aus Messungen im Bereich  $80 \frac{km}{h}$ , ist damit deutlich schmäler als A und ähnelt damit der Kernfunktion  $K()$  (Gleichung 4.1). Es existiert nur ein Maximum bei circa  $80 \frac{km}{h}$  und stellt damit die wahrscheinlichste Geschwindigkeit dar.

Um die Verteilung zu berechnen, setzt die Gleichung 4.1 das Vorhandensein aller Messwerte voraus, die berücksichtigt werden sollen. Bei einem lernenden System im Fahrzeug werden jedoch mit jeder einzelnen Fahrt neue Messwerte erhoben, die einzeln zu den bereits gelernten Werten hinzugefügt werden müssen. In [47] wird dafür eine *Lernfunktion* auf Basis der Kerndichteschätzung vorgestellt, welche mit jedem neuen Messwert  $v_{\kappa+1}$  eine korrigierte Verteilung  $\hat{f}_{korr}$  berechnet:

$$\hat{f}_{korr}(v) = \frac{\kappa}{\kappa + 1} \hat{f}(v) + \frac{1}{\kappa + 1} \cdot \frac{1}{b(\kappa + 1)} K\left(\frac{v - v_{\kappa+1}}{b(\kappa + 1)}\right) \quad (4.2)$$

Dabei ist  $\kappa$  die Anzahl der bisher gelernten Messwerte,  $v_i$  mit  $i = 1 \dots \kappa$  sind die bisher gelernten Messwerte (in  $\hat{f}$  benötigt),  $K()$  der Epanechnikov-Kern als Kernfunktion und  $b()$  die in [47] per Hand angepasste Bandbreite in Abhängigkeit von  $\kappa$ . In der Gleichung 4.2 repräsentiert der erste Term ( $\hat{f}(v)$ ) die Verteilung der bisher gelernten Messwerte und der zweite Term ( $K(\frac{v-v_{\kappa+1}}{b(\kappa+1)})$ ) die Änderung der Verteilung durch den neuen Messwert. Die beiden Terme werden gleichgewichtet über die Anzahl der Messwerte  $\kappa$  addiert ( $\frac{\kappa}{\kappa+1}$  und  $\frac{1}{\kappa+1}$ ). Durch die Normierung ( $\frac{1}{b(\kappa+1)}$ ) ist der Flächeninhalt der resultierenden Verteilung  $\hat{f}_{korrr}$  immer 1. Diese Lernfunktion stellt den Lernprozess des lernenden Systems dar.

Im Datenspeicher wird lediglich der „geeignet diskretisierte“ [47] Verlauf der Wahrscheinlichkeitsverteilung  $\hat{f}_{korrr}$  zusammen mit der Anzahl der Messwerte  $\kappa$  abgelegt. Die einzelnen Messwerte werden nicht gespeichert. Damit ist der Speicherbedarf unabhängig von der Anzahl der eingeflossenen Messwerte. Wird nur ein begrenztes Intervall der Verteilung (zum Beispiel  $[-50, 300]$  für die Geschwindigkeit) gespeichert, ist der Speicherbedarf sogar konstant. Die Eigenschaft des konstanten Speicherbedarfs des Lernprozesses ist von Bedeutung für ein lernendes System im Fahrzeug mit limitierten Speicherplatz. Für den Kurvenwarner wäre eine mögliche geeignete Diskretisierung der Geschwindigkeit eine Schrittgröße von  $1 \frac{km}{h}$ .

Die Kerndichteschätzung soll auch für den Lernprozess in der vorliegenden Arbeit verwendet werden. In der vorliegenden Arbeit soll der Lernprozess jedoch auch ganze Messwertreihen zusammenführen können.

### 4.1.2 Notwendigkeit einer Messwertreihen-Transformation

Für die Zusammenführung von ganzen Messwertreihen mit der Kerndichteschätzung muss eindeutig sein, welche Messwerte der Reihen zusammengehören. Eigenschaften von Messwertreihen eines Sensors auf einer Strecke variieren zwischen den Fahrten und sind damit heterogen. Deshalb können deren Messwerte nicht eindeutig einander zugeordnet werden.

Vier mögliche Unterschiede zwischen den Messwertreihen konnten festgestellt werden:

1. Verschiebung der Messposition zwischen Messwertreihen
2. Unterschiedliche Messwertdichten
3. Variierende Start- und Endpunkte der Messwertreihe
4. Variierende Länge der Messwertreihe

**Unterschied 1. & 2.) Messpositionsverschiebung und Messwertdichte:**

Sensoren haben meist eine feste Zykluszeit und erfassen die Messwerte dadurch in festen Zeitintervallen, die unabhängig vom Ort des Fahrzeugs sind; die Sensormesswerte sind also zeitabhängig. Dadurch werden die Sensormesswerte an beliebigen Positionen entlang der gefahrenen Strecke erhoben, was zu einer Verschiebung der Messpositionen führt. Diese Verschiebung ist beispielhaft in Abbildung 4.2 zwischen 1) und 2) dargestellt.

Hinzu kommt, dass ein Fahrzeug mit unterschiedlichen Geschwindigkeiten fährt, und dadurch Messwertreihen eines Sensors unterschiedlich dicht in Bezug auf die Messposition sind. Dies ist ebenfalls in Abbildung 4.2 bei 1), 3) und 4) dargestellt.

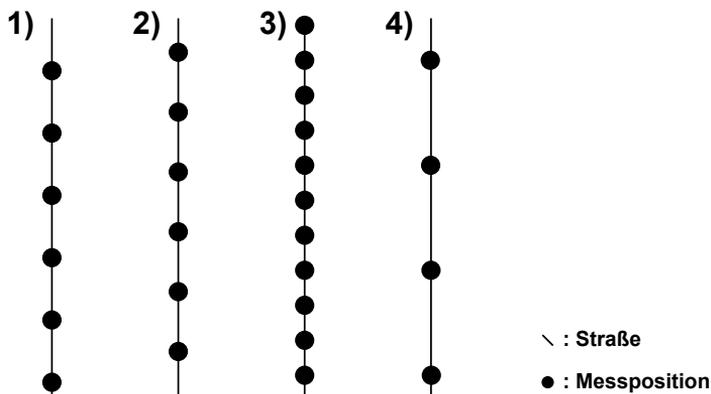


Abb. 4.2: Beispiele der Heterogenität von Messwertreihen: Verschiebung und variierende Dichte.

**Unterschied 3. & 4.) Variierende Start-, Endpunkte und Länge:** Neben der Verschiebung von Messpositionen zwischen Messwertreihen und der unterschiedlichen Messwertreihendichte kann auch die Länge der Messwertreihe und die Position des Start- und des Endpunkts einer Messwertreihe variieren. Dies wird durch die Erkennung von Situationen hervorgerufen, welche die Aufzeichnung der Messwertreihen steuert. So kann beispielsweise durch den Fahrer und abhängig von der Implementierung der Erkennung eine Situation früher, später, kürzer oder länger detektiert werden.

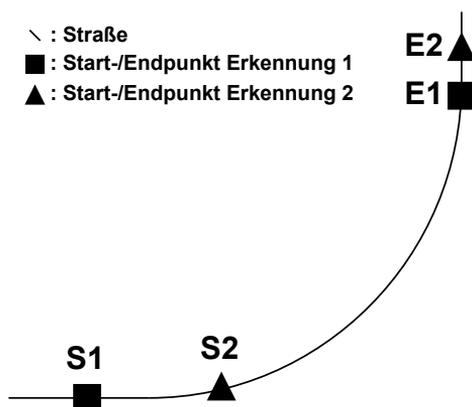


Abb. 4.3: Zwei Fahrten durch eine Kurve mit unterschiedlichen Kurveneingängen ( $S1$ ,  $S2$ ) und Kurvenausgängen ( $E1$ ,  $E2$ ) aufgrund der Arbeitsweise der Kurvenerkennung.

In Abbildung 4.3 ist eine Situation mit zwei abweichenden Erkennungen dargestellt. Eine Messwertreihe, die während einer detektierten Kurve aufgezeichnet wird, hat je nach Fahrt möglicherweise eine unterschiedliche Länge oder unterschiedliche Start- und Endpunkte. Die Ursache dafür ist im Beispiel der Kurvenerkennung die Variabilität im Fahrstil des Fahrers, wodurch zum Beispiel die Schwellwerte einer Erkennung früher oder später über-/unterschritten und damit die Aufzeichnung an anderen Positionen startet oder stoppt.

Die vorgestellten Heterogenitäten von Messwertreihen verhindern die einfache Zusammenführung, da nicht eindeutig ist, welche Werte der verschiedenen Messwertreihen zusammengehören.

### 4.1.3 Messwertreihen-Transformation auf ein positionsabhängiges Raster

Um eine eindeutige Zuordnung von Messwerten herzustellen, müssen heterogene Messwertreihen homogenisiert werden.

Dies soll mit einer Messwertreihen-Transformation durchgeführt werden, indem die zeitabhängigen Messwertreihen der Sensoren auf einheitliche, gemeinsame Positionen abgebildet werden. Alle auf eine Position transformierten Messwerte bilden eine Sequenz und gehören zusammen. Diese Messwertsequenz kann anschließend mit der Kerndichteschätzung

zusammengeführt werden. Ein gemeinsames Raster definiert die festen Positionen, auf die alle Messwertreihen abgebildet werden sollen. Mit dieser Transformation wird die Zykluszeit der Sensoren für die Zusammenführung im Lernprozess irrelevant. So liegen für jeden Rasterpunkt alle gelernten Messgrößen vor, unabhängig von der jeweiligen Zykluszeit des Sensors.

Im Folgenden wird das gemeinsame Raster und dessen Funktion vorgestellt. Anschließend wird beschrieben wie Messwerte auf das Raster abgebildet werden und wie die dafür benötigte Transformation von der Zeitabhängigkeit zur Positionsabhängigkeit funktioniert. Zuletzt wird beschrieben, wie die daraus resultierenden Messesequenzen mit der Kerndichteschätzung zusammengeführt werden können.

### **Gemeinsames positionsabhängiges Raster**

Bei der Abbildung auf dieses positionsabhängige Raster wird eine mögliche Messpositionsverschiebung und eine variierende Messwertdichte kompensiert. Das Raster wird initial entlang einer Strecke festgelegt, zum Beispiel als äquidistantes Raster. Ein Rasterpunkt steht dabei für die gesamte Breite der Straße. Die Abstände zwischen zwei Rasterpunkten ist dabei frei wählbar und kann entsprechend der Anforderungen von FAS angepasst werden.

In Abbildung 4.4, 1) ist das gewählte äquidistante Raster über den Messpositionen der Messwertreihe einer Situation gelegt. Das Raster ist als gestrichelte, horizontale Linie dargestellt. Die einzelnen Messwerte entlang des Straßenverlaufs sind als schwarze Kreise dargestellt. In 2) wird eine benötigte Zwischenwertberechnung als Bogen dargestellt. Ziel ist es, die Zwischenwerte exakt auf dem Raster wie in 3) zu erhalten, die als schwarze Vierecke dargestellt sind.

Wird zusätzlich eine Verwaltung hinzugefügt, die das Raster dynamisch an die Größe der Messwertreihen anpassen kann, ist auch eine eindeutige Zuordnung von Messwertreihen mit unterschiedlichen Start- und Endpunkten oder Längen möglich.

Überschreitet ein Messwert einer Messwertreihe das bisher bestehende Raster um mindestens die Länge seines Rasterabstands, wird das Raster an dieser Stelle erweitert. In diesem Fall kann für diesen neuen Rasterpunkt ein Zwischenwert berechnet werden.

## 4 Lernen von heterogenen Sensormesswertreihen

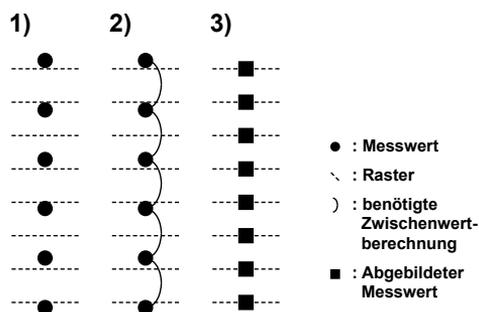


Abb. 4.4: Messwertreihen mit beliebigen Messpositionen 1) können über eine Zwischenwertberechnung 2) auf ein Raster 3) abgebildet werden.

**Beispiel 4.2 (Dynamische Erweiterung des Rasters):** *Der Vorgang der dynamischen Erweiterung wird in Abbildung 4.5 gezeigt. Im ersten Schritt 1) der Abbildung 4.5 ist links eine bereits gelernte Messwertreihe als vertikale Folge von Messwerten (Vierecke) auf dem initial festgelegten Raster zu sehen. Ebenfalls in 1) ist rechts davon eine neue, noch nicht vorverarbeitete Messwertreihe (Punkte) zu sehen, die über die bisherigen Rastergrenzen hinausgeht (einen Rasterschritt weiter nach unten). Werden die in 2) markierten Zwischenwertberechnungen durchgeführt, resultieren die abgebildeten Zwischenwerte auf dem Raster in 3). Die zweite Messwertreihe der Situation enthält keine Messwerte für die oberen drei Rasterpunkte, deswegen ist keine Zwischenwertberechnung notwendig. Für die darauf folgenden 5 Rasterpunkte können Zwischenwerte berechnet werden, wodurch an diesen Rasterpunkten jeweils zwei Messwerte verfügbar sind. Der unterste Rasterpunkt wurde neu hinzugefügt, da die zweite Messwertreihe das bisherige Raster weit genug überschritten hat, um einen Messwert für diesen neuen Rasterpunkt zu liefern.*

Mit der Abbildung auf ein gemeinsames, positionsabhängiges, dynamisches Raster, können heterogene Messwertreihen homogenisiert werden und eine eindeutige Zuordnung zwischen ihnen hergestellt werden. Verschiebungen der Messpositionen zwischen Messwertreihen und unterschiedliche Messwertdichten von Messwertreihen stellen so kein Problem mehr dar. Die dynamische Vergrößerung des Rasters ermöglicht es, alle Messwerte von Messwertreihen mit variierenden Start- und Endpunkten und auch variierender Länge zu berücksichtigen.

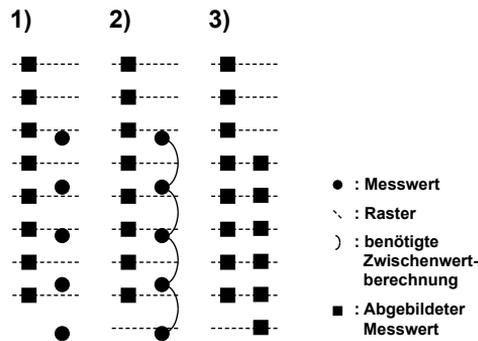


Abb. 4.5: Das gemeinsame Raster wird dynamisch erweitert, wenn neue Messwertreihen das bisherige überschreiten.

### Zwischenwertberechnung im Zeitbereich

Für die Berechnung der Zwischenwerte auf dem gemeinsamen Raster, müssen die vorhandenen, zeitabhängigen Messwerte interpoliert werden. Die Interpolation ist eine Standardmethode zum Approximieren von Werten [57]. Es wird eine Funktion anhand bekannter Stützstellen durch einfachere Funktionen näherungsweise dargestellt [74].

Es existieren verschiedene Interpolationsverfahren mit unterschiedlichen Eigenschaften. Die dabei entstehende, vereinfachte Funktion enthält in den meisten Fällen die Stützstellen, ist stetig und stellt die Werte zwischen Stützstellen näherungsweise dar. Die Nähe der Approximation zur echten Funktion wird maßgeblich vom eingesetzten Verfahren bestimmt.

In Abbildung 4.6 ist ein Vergleich der linearen Interpolation und der Interpolation mit kubischen Splines zu sehen. Dieser Vergleich wurde mit einer Geschwindigkeitsmesswertreihe aus dem Versuchsträger durchgeführt. Die Geschwindigkeitsmessung hat in diesem Beispiel eine Zykluszeit von  $1000\text{ms}$ . Die lineare Interpolation wurde jeweils zwischen zwei aufeinander folgenden Messwerten durchgeführt. Für die kubischen Splines wurden zwischen zwei Stützpunkten die Zwischenwerte  $.25$ ,  $.5$  und  $.75$  berechnet. Bei Sekunde  $14.5$  tritt die größte Differenz ( $0.04 \frac{\text{km}}{\text{h}}$ ) zwischen den Interpolationsverfahren auf (lineare Interpolation  $72.2 \frac{\text{km}}{\text{h}}$ , kubischen Splines  $72.24 \frac{\text{km}}{\text{h}}$ ), die jedoch verhältnismäßig klein ist. Da die echte Geschwindigkeit für die Zeitspannen zwischen zwei Messzeitpunkten nicht vorliegt, kann keine Aussage darüber getroffen werden, welche der beiden Verfahren näher an der Realität liegt.

Die *lineare Interpolation* ist das einfachste Interpolationsverfahren und

## 4 Lernen von heterogenen Sensormesswertreihen

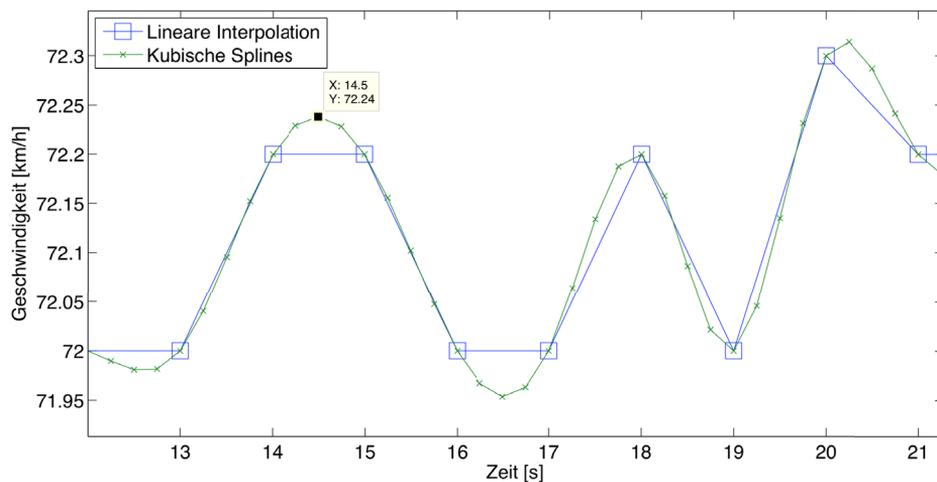


Abb. 4.6: Interpolationsverfahren im Vergleich für reale Geschwindigkeitsmesswerte: Linear und kubische Splines.

hat den geringsten Berechnungsaufwand. Die dabei entstehenden Fehler sind durch die relativ geringen Zykluszeiten der Sensoren (20 – 1000ms) im Fahrzeug klein (siehe Abbildung 4.6). Dadurch fällt der Mehrwert einer möglicherweise genaueren Approximation durch beispielsweise kubische Splines sehr gering aus, der den erhöhten Berechnungsaufwand nicht rechtfertigt. Auch eine Differenzierbarkeit der resultierenden Gesamtfunktion wird für die Anwendung in der vorliegenden Arbeit nicht benötigt.

Bei der linearen Interpolation werden, anders als bei beispielsweise der *Polynominterpolation*, nur zwei benachbarte Stützstellen beziehungsweise Messwerte verwendet. Dies ermöglicht eine selektive Interpolation, die den Berechnungsaufwand reduziert (siehe Beispiel 4.3). Nur wenn sich ein Rasterpunkt zwischen zwei Messwerten befindet, muss interpoliert werden. Dies ist aufgrund der limitierten Ressourcen im Fahrzeug erwünscht. Mit Interpolationsverfahren wie der Polynominterpolation ist eine derartige Einsparung nicht möglich, da alle Stützstellen betrachtet werden.

**Beispiel 4.3 (Selektive Interpolation):** In der Abbildung 4.7 sind zwei Beispiele mit unterschiedlichen Dichten der Messpunkte dargestellt. Auf der linken Seite in jeweils 1), 2) und 3) in vertikaler Anordnung ist eine Messwertreihe mit geringer Dichte zu sehen. Im Gegensatz dazu ist auf der rechten Seite jeweils eine Messwertreihe mit hoher Dichte zu sehen. In 2) sind die benötigten Interpolationen zwi-

schen zwei Messwerten aufgezeigt. Bei der linken Reihe werden bei der Interpolation zwischen dem obersten und dem darunter liegenden Messwert gleich zwei Rasterpunkte abgedeckt und entsprechend Werte berechnet, wie in 3) zu sehen. Aufgrund der hohen Dichte der zweiten, rechts dargestellten Messwertreihe können manche Interpolationen zwischen Messwertepaaren ausgelassen werden. Wie in 2) zu sehen, müssen zum Beispiel keine Interpolationen für den von oben gezählten zweiten, dritten und vierten Messwert durchgeführt werden, da sich keine Rasterpunkte dazwischen befinden.

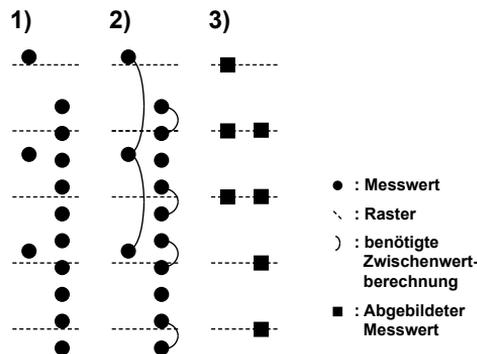


Abb. 4.7: Selektive Interpolation zur Abbildung auf das Raster in Abhängigkeit zur Dichte der Messwertreihe.

Aus den genannten Gründen eignet sich die lineare Interpolation für die vorliegende Arbeit und wird für die Messwertreihen-Transformation verwendet. Bei der Umsetzung der Messwertreihen-Transformation wird eine Austauschbarkeit des Interpolationsverfahrens ermöglicht.

### Transformation von Zeitabhängigkeit zur Positionsabhängigkeit

Es sollen bei der Messwertreihen-Transformation die Zwischenwerte berechnet werden, die exakt auf den Positionen der Rasterpunkte liegen.

Sensormesswerte und damit auch die Messwertreihen sind von der Zeit abhängig, da die Messungen der Sensoren in einem festen Zyklus durchgeführt werden. Die Interpolation der Messwerte wird deswegen im Zeitbereich durchgeführt. Das gemeinsame Raster ist jedoch positionsabhängig, weswegen für die Interpolation die Zeitpunkte bekannt sein müssen, bei denen das Fahrzeug einen Rasterpunkt überfährt. Erst mit diesen Zeitpunkten können die entsprechenden Zwischenwerte auf den Rasterpunk-

## 4 Lernen von heterogenen Sensormesswertreihen

---

ten berechnet werden. Diese Transformation von der Zeitabhängigkeit zur Positionsabhängigkeit wird im Folgenden beschrieben.

Für die Berechnung der Zeitpunkte, in denen sich das Fahrzeug auf einem Rasterpunkt befindet, muss die Bewegung des Fahrzeugs interpoliert werden. Dafür wird die lineare Interpolation verwendet, da eine konstante Beschleunigung  $a$  zwischen zwei Messwerten der Geschwindigkeit angenommen wird. Das Weg-Zeit-Gesetz beschreibt die Bewegung des Fahrzeugs:

$$s(t) = \frac{a}{2}t^2 + v_0t + s_0 \quad (4.3)$$

Dabei ist  $s(t)$  die Position des Fahrzeugs zum Zeitpunkt  $t$ ,  $a$  die aktuelle Beschleunigung,  $v_0$  die aktuelle Geschwindigkeit und  $s_0$  die aktuelle Position des Fahrzeugs.

Es existieren zwei Fälle, für die jeweils eine andere Gleichung gilt. Im ersten Fall ist die Beschleunigung ungleich Null, weswegen die Gleichung 4.3 nach der Zeit umgestellt wird:

$$t_{1/2} = -\frac{v_0}{a} \pm \sqrt{\left(\frac{v_0}{a}\right)^2 + \frac{2(s_t - s_0)}{a}} \quad (4.4)$$

$s_t$  ist eine Position des Fahrzeugs, zu dem der Zeitpunkt  $t$  des Überfahrens dieser Position berechnet werden kann. Es werden nur positive Zeitwerte aus der Wurzel betrachtet, weswegen  $t_2$  verworfen wird:

$$t_1 = -\frac{v_0}{a} + \sqrt{\left(\frac{v_0}{a}\right)^2 + \frac{2(s_t - s_0)}{a}} \quad (4.5)$$

Im zweiten Fall ist die Beschleunigung Null und es ergibt sich mit  $a = 0$  aus der Gleichung 4.3:

$$s(t) = v_0t + s_0 \quad (4.6)$$

Umgestellt nach der Zeit ergibt sich:

$$t = \frac{s_t - s_0}{v_0} \quad (4.7)$$

Aus den Gleichungen 4.5 und 4.7 leitet sich folgende Berechnungsvorschrift für die Zeitpunkte des Überfahrens eines Rasterpunktes ab:

$$t_R(s_R) = \begin{cases} \sqrt{\left(\frac{v_0}{a}\right)^2 + \frac{2(s_R - s_0)}{a}} - \frac{v_0}{a} & , a \neq 0 \\ \frac{s_R - s_0}{v_0} & , a = 0 \end{cases} \quad (4.8)$$

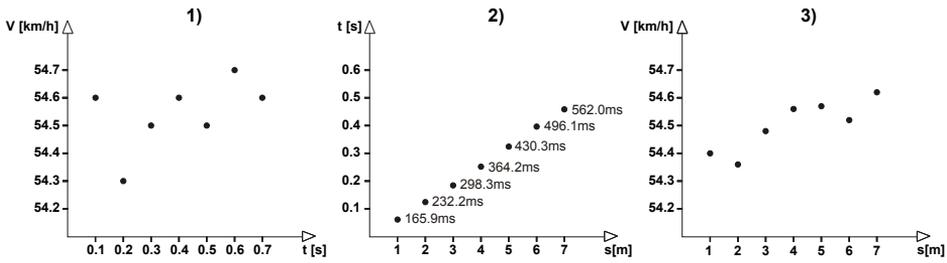


Abb. 4.8: Messwertreihen-Transformation einer zeitabhängigen Messwertreihe auf ein positionsabhängige Raster.

Mit dieser Gleichung lassen sich die Zeitpunkte  $t_R$  berechnen, bei denen das Fahrzeug einen Rasterpunkt  $R$  an Position  $s_R$  überfährt.

Mit den Zeitpunkten  $t_R$  können anschließend die Interpolationen der Messwerte auf das gemeinsame Raster durchgeführt werden. Dies ist in Abbildung 4.8 dargestellt:

- Graph 1): Eine Messwertreihe dargestellt über die Zeit.
- Graph 2): Die Zeitpunkte  $t_R$  (y-Achse), in denen das Fahrzeug die 1 Meter Rasterpunkte  $R$  an Position  $s_R$  überfährt (x-Achse), wurde anhand der zuvor beschriebenen Transformation mit Hilfe der Interpolation der Fahrzeugbewegung berechnet.
- Graph 3): Mit den berechneten Zeitwerten  $t_R$  wurden mit der linearen Interpolation sämtliche Zwischenwerte aller Sensormesswerte berechnet, hier ist die Geschwindigkeit dargestellt.

In Abbildung 4.9 sind die Messwerte und die auf das gemeinsame Raster abgebildeten Werte dargestellt.

Es wurde die Messwertreihen-Transformation von zeitabhängigen Messwertreihen auf das positionsabhängige Raster mit der linearen Interpolation beschrieben. Sie ermöglicht die eindeutige Zuordnung von Messwerten aus verschiedenen, heterogenen Messwertreihen, um diese mit der Kern-dichteschätzung zusammenzuführen.

## 4 Lernen von heterogenen Sensormesswertreihen

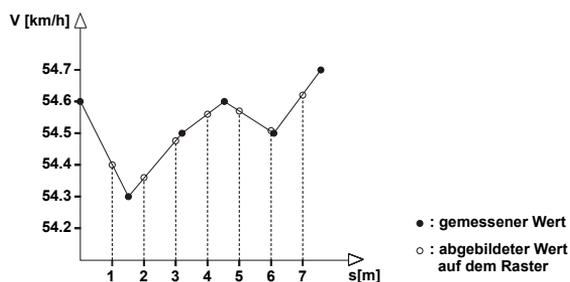


Abb. 4.9: Darstellung der gemessenen und der interpolierten Werte einer Sensormesswertreihe.

### 4.1.4 Zusammenführung von positionsabhängigen Messwertreihen mit der Kerndichteschätzung

Alle Messwerte, die auf den gleichen Rasterpunkt abgebildet wurden, gehören zusammen und bilden eine Messsequenz. Diese Messsequenz wird mit der Kerndichteschätzung (wie in Abschnitt 4.1.1 beschrieben) zusammengeführt.

**Beispiel 4.4 (Zusammenführung der Messsequenzen):** In Abbildung 4.10 ist dieser Prozess dargestellt: In 1) wurden sechs Messwertreihen mit der entwickelten Messwertreihen-Transformation auf das gemeinsame Raster abgebildet. Jeder Rasterpunkt repräsentiert eine Sequenz von Messwerten wie in 2) mit Kästen dargestellt. Diese Messsequenzen können anschließend mit der Kerndichteschätzung zusammengefasst werden, um die Verteilung der Messwerte an der Position des Rasterpunktes zu berechnen. 3): Die zusammengefassten Messwerte in Form einer geschätzten Verteilung sind dargestellt.

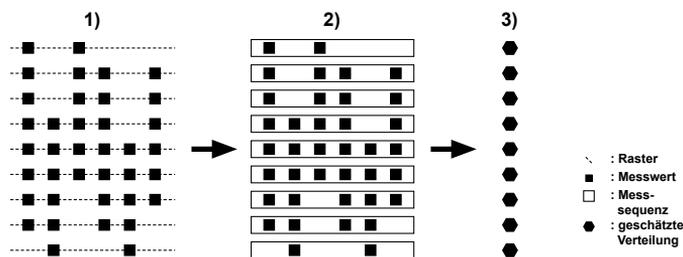


Abb. 4.10: Messwertreihen wurden über die Messwertreihen-Transformation auf das Raster abgebildet und können zusammengeführt werden.

Wurde die Zusammenführung der Messequenzen durchgeführt, liegt für jeden Rasterpunkt eine Wahrscheinlichkeitsverteilung des entsprechenden Sensormesswerts vor. So erhalten die FAS Zugriff auf einen Verlauf von Wahrscheinlichkeiten beliebiger Sensorsignale über die gesamte Situation hinweg und damit entlang einer Strecke. Für das Beispiel des Kurvenwarners bedeutet dies einen detaillierten Geschwindigkeitsverlauf entlang einer Kurve. Für jeden Rasterpunkt der Kurve kann eine Aussage über die Geschwindigkeit getroffen werden, die der Fahrer am wahrscheinlichsten fahren wird. So können sich verengende Kurven von sich öffnenden Kurven unterscheiden und für jede Kurve eine angepasste und zutreffende Warnung ausgegeben werden.

Mit der in diesem Abschnitt entwickelten Messwertreihen-Transformation, können heterogene Messwertreihen eindeutig einander zugeordnet werden und dadurch mit der Kerndichteschätzung im Lernprozess zusammengeführt werden. Dazu wird ein gemeinsames, positionsabhängiges Raster definiert, auf das die zeitabhängigen Messwertreihen mit der linearen Interpolation abgebildet werden. Mit der beschriebenen Transformation, die zeitabhängige Messwerte in positionsabhängige überführt, ist dies möglich. Abschließend werden die dabei resultierenden Messequenzen an jedem Rasterpunkt mit der Kerndichteschätzung zusammengeführt. So liegt für jeden Rasterpunkt die Wahrscheinlichkeitsverteilung aller gelernten Messgrößen vor und es lässt sich der wahrscheinlichste Verlauf jeder Messgröße entlang der Strecke extrahieren.

## 4.2 Evaluation des erweiterten Lernprozesses

Die im vorherigen Abschnitt vorgestellte Messwertreihen-Transformation ermöglicht ein positionsabhängiges Lernen von Messwertreihen mit hohem Detailgrad.

In diesem Abschnitt werden die einzelnen Schritte der Verarbeitung und die Ergebnisse des Lernprozesses mit der Messwertreihen-Transformation präsentiert. Anschließend wird am adaptiven Kurvenwarner konkret gezeigt, welche Vorteile durch die gelernten Messwertreihen entstehen.

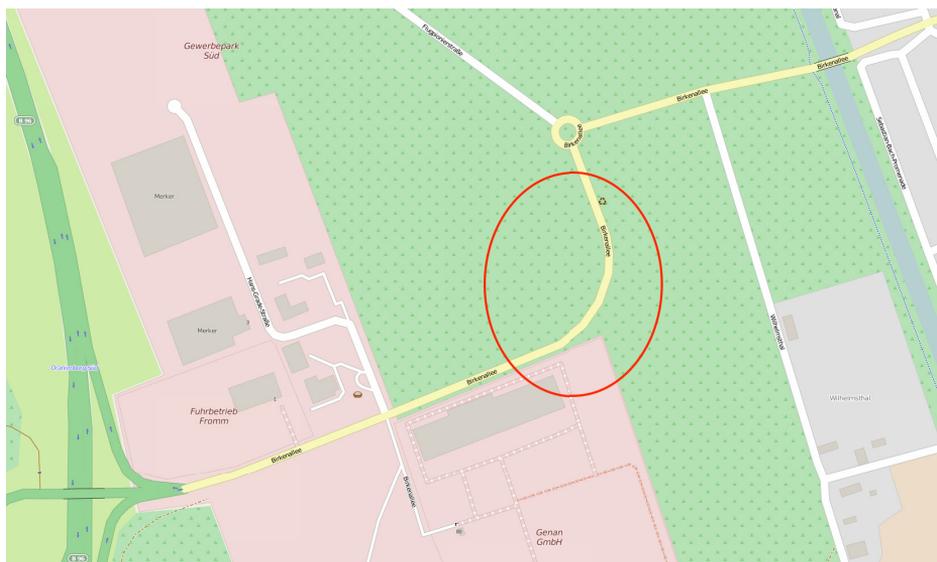


Abb. 4.11: Kurve auf der Birkenallee in Oranienburg nahe der B96 (Quelle: OSM Kartenmaterial).

### 4.2.1 Anwendung des Lernprozesses mit echten Sensormesswertreihen

Aus mehreren Fahrten kann der wahrscheinlichste Verlauf einer Messgröße entlang der Strecke extrahiert werden. Der Rasterabstand wird dafür auf  $1m$  festgelegt, da dieser Detailgrad in Versuchen für den adaptiven Kurvenwarner ausreichte und Umrechnungen und Darstellungen vereinfacht. Für die Evaluation der Messwertreihen-Transformation wird als Beispiel, mit Hinblick auf den adaptiven Kurvenwarner, der Geschwindigkeitsverlauf durch eine Kurve gelernt. In Abbildung 4.11 ist die verwendete Kurve auf der Birkenallee in Oranienburg (in der Nähe der B96) zu sehen.

In Abbildung 4.12 sind GPS Spuren von 5 Fahrten durch die *Birkenallee* Kurve abgebildet. Die Heterogenität der Messreihen ist deutlich erkennbar: die GPS Spuren unterscheiden sich zwischen den einzelnen Fahrten durch variierende Anfangs- und Endpunkte, die Dichte der Messpunkte variiert ebenfalls und bei ähnlichen Dichten sind die Messpunkte verschoben. Dies zeigt die Notwendigkeit der in der vorliegenden Arbeit entwickelten Messwertreihen-Transformation, damit eine weitere Verarbeitung mit der Kerndichteschätzung möglich ist.

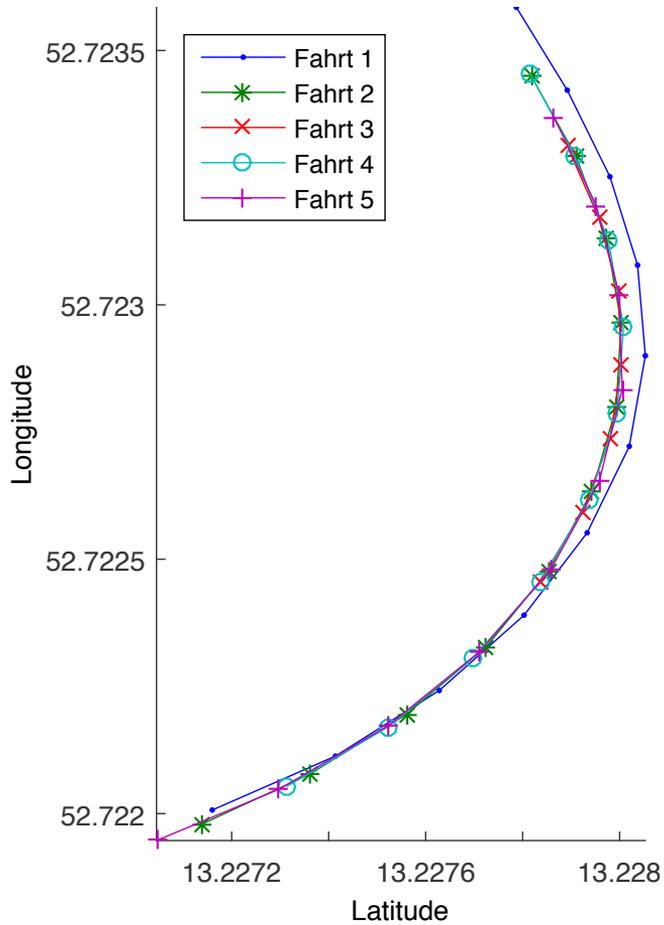


Abb. 4.12: GPS Spuren, die bei 5 Fahrten durch die *Birkenallee* Kurve aufgenommen wurden.

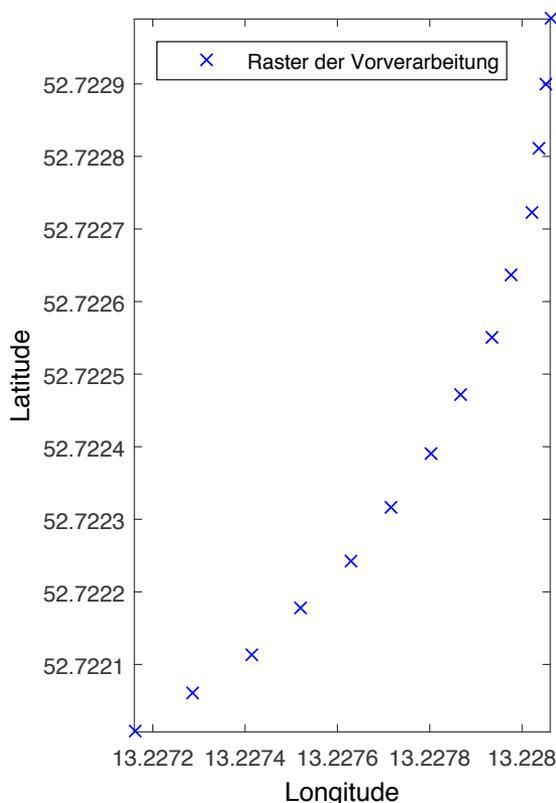


Abb. 4.13: Ausschnitt der Rasterpunkte nach der Messwertreihen-Transformation der Fahrten durch die Kurve der *Birkenallee*.

Durch die Messwertreihen-Transformation wurden die Sensormesswerte auf ein einheitliches Raster abgebildet. Für diesen Test wurde ein Rasterabstand von einem Meter gewählt. In Abbildung 4.13 ist ein Ausschnitt des resultierenden Rasters zu sehen. Aus Darstellungsgründen wurde nur jeder zehnte Rasterpunkt abgebildet. Für jeden dieser Rasterpunkte liegen die Geschwindigkeiten aus allen Fahrten durch die Kurve vor.

In der Abbildung 4.14 ist beispielhaft die Wahrscheinlichkeitsverteilung eines Rasterpunkts nach diesen Fahrten zu sehen. Die wahrscheinlichste Geschwindigkeit nach dieser Verteilung beträgt  $70.92 \frac{\text{km}}{\text{h}}$ . Darüber hinaus ist erkennbar, dass auch im Bereich  $60 \frac{\text{km}}{\text{h}}$  eine oder mehrere Messungen erhoben wurden. Deswegen hat die Wahrscheinlichkeit dort ein Plateau. Eine derartige Verteilung wird für jeden Rasterpunkt berechnet.

Liegt für jeden Rasterpunkt der Kurve diese Wahrscheinlichkeitsvertei-

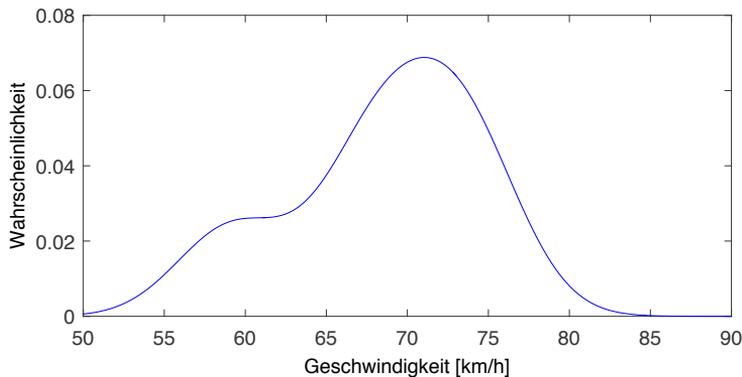


Abb. 4.14: Verteilung der Geschwindigkeit an einem Rasterpunkt der Kurve.

lung der Geschwindigkeit vor, kann die wahrscheinlichste Geschwindigkeit aus jeder Verteilung entnommen werden. Damit lässt sich das wahrscheinlichste Geschwindigkeitsprofil für die Kurve bilden, wie in Abbildung 4.15 für die *Birkenallee* dargestellt. Auch wenn sich die Geschwindigkeit innerhalb der Kurve nur gering ändert, lassen sich dennoch drei markante Punkte im Verlauf erkennen:

1. Höhere Geschwindigkeit am Kurveneingang beim Einfahren in die Kurve (Rasterpunkt 1 bis ~70)
2. Niedrigste Geschwindigkeit um den Scheitelpunkt der Kurve herum (Rasterpunkt ~70 bis ~160)
3. Höhere Geschwindigkeit am Kurvenausgang beim Rausbeschleunigen aus der Kurve (Rasterpunkt ~160 bis ~180)

Das Abfallen der Geschwindigkeit bei den Rasterpunkten um Punkt 180 ist auf das Abbremsen für den folgenden Kreisverkehr an dem nördlichen Ende der *Birkenallee* (siehe Abbildung 4.11) zurückzuführen.

Derartige Verläufe entlang Strecken können für jede beliebige skalare Sensormessgröße gelernt werden und den FAS zur Verfügung gestellt werden.

## 4 Lernen von heterogenen Sensormesswertreihen

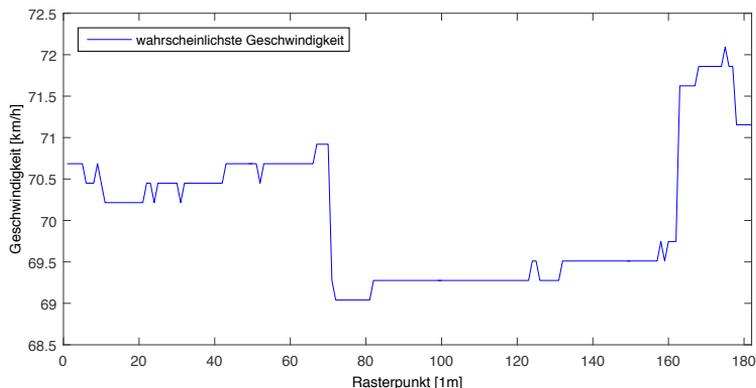


Abb. 4.15: Wahrscheinlichstes Geschwindigkeitsprofil durch die *Birkenallee* Kurve abgebildet über ein  $1m$  Raster.

### 4.2.2 Vorteile für den adaptiven Kurvenwarner durch gelernte Messwertreihen

Wurde ein Geschwindigkeitsprofil durch eine Kurve gelernt, ist eine fein abgestimmte Warnung vom adaptiven Kurvenwarner möglich: Mit diesem Profil ist eine Vorhersage über den wahrscheinlichsten Verlauf der Geschwindigkeit in dieser Kurve möglich. In Abbildung 4.16 werden die Distanzen der Warnung für jeden zehnten Rasterpunkt (aus Darstellungsgründen) in Relation zum Kurveneingang gezeigt, wenn das Fahrzeug mit  $150 \frac{km}{h}$  auf die Kurve zufährt. Durch die geringe Variation der gelernten Geschwindigkeit (siehe Abbildung 4.15) wird die früheste Warnung durch den ersten Rasterpunkt bei  $150.1m$  vor dem Kurveneingang ausgegeben werden. Alle folgenden Rasterpunkte würden erst später eine Warnung auslösen.

Wird eine Kurve schneller und dynamischer gefahren, als es mit dem Versuchsträger auf öffentlichen Straßen getan wurde, können die gelernten Geschwindigkeiten innerhalb der Kurve stärker variieren. Abbildung 4.17 zeigt ein gelerntes wahrscheinlichstes Geschwindigkeitsprofil einer Kurvenfahrt, das für diesen Vergleich künstlich erzeugt wurde. Der Geschwindigkeitsunterschied zwischen Kurveneingang und Kurvenscheitelpunkt beträgt in diesem Verlauf  $27.4 \frac{km}{h}$  und ist damit deutlich größer als bei der Kurvenfahrt mit dem Versuchsträger.

## 4.2 Evaluation des erweiterten Lernprozesses

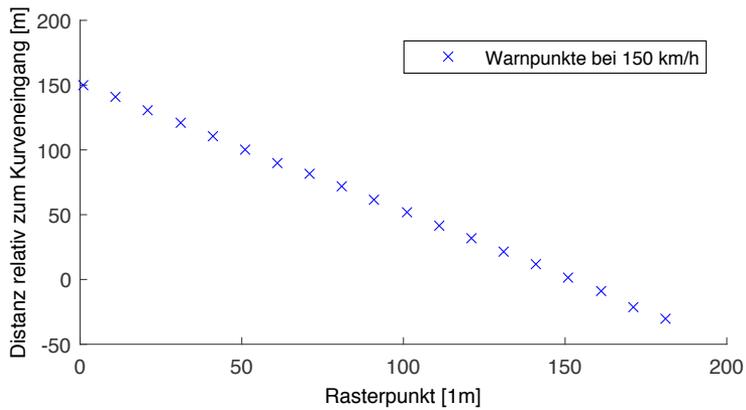


Abb. 4.16: Distanz der Warnungen jedes Rasterpunkts in Relation zum Kurveneingang der gelernten Geschwindigkeiten.

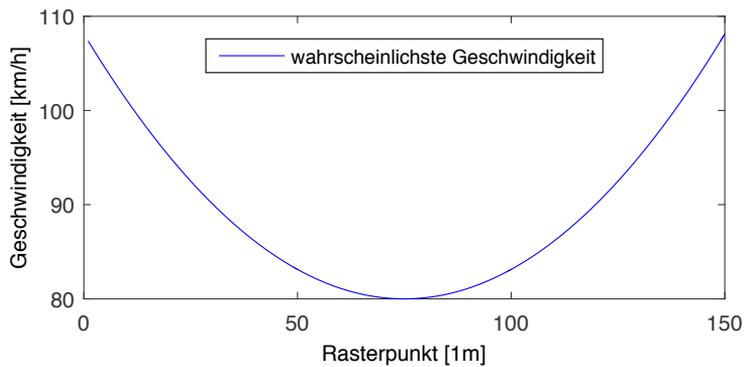


Abb. 4.17: Wahrscheinlichstes Geschwindigkeitsprofil einer künstlichen Kurvenfahrt abgebildet über ein 1m Raster.

## 4 Lernen von heterogenen Sensormesswertreihen

In Abbildung 4.18 sind die berechneten Warnungen der künstlich erzeugten Kurvenfahrt für jeden zehnten Rasterpunkt zu sehen, wenn das Fahrzeug mit  $150 \frac{km}{h}$  auf die Kurve zufährt. In diesem Fall löst der 11. Rasterpunkt (zweiter dargestellter Rasterpunkt) und der 21. Rasterpunkt (dritter dargestellter Rasterpunkt) früher eine Warnung aus als der erste Rasterpunkt. Bei  $96.56m$  vor der Kurve wird die früheste Warnung durch den 15. Rasterpunkt (nicht in der Abbildung dargestellt) ausgelöst. Im Gegensatz zum ersten Rasterpunkt mit  $94.05m$ , löst er  $2.51m$  früher eine Warnung aus und gibt damit dem Fahrer mehr Zeit und Strecke auf die Warnung zu reagieren.

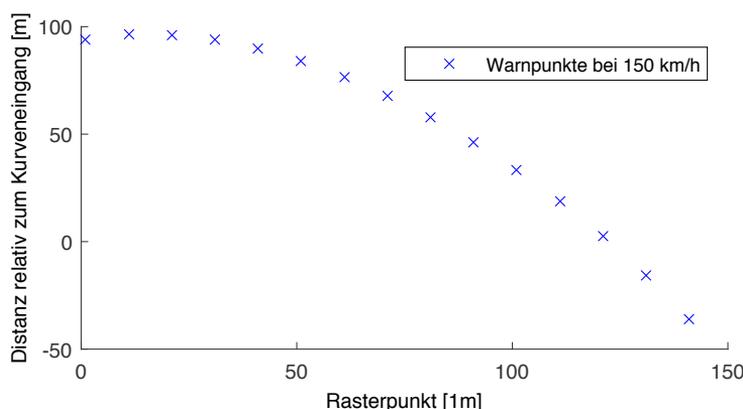


Abb. 4.18: Distanz der Warnungen jedes Rasterpunkts in Relation zum Kurveneingang der künstlich erzeugten Kurvenfahrt.

Dies zeigt, dass mit dem Lernen einer Messwertreihe der adaptive Kurvenwarner präzisere Warnungen an den Fahrer ausgeben kann. Noch deutlicher wird dies, wenn die Warnzeitpunkte mit einem lernenden System verglichen werden das nur einzelne Messwerte lernen kann und daher nur die mittlere Kurvengeschwindigkeit berücksichtigen kann.

### Vergleich einer Warnung mit der mittleren Kurvengeschwindigkeit:

Würde statt des wahrscheinlichsten Geschwindigkeitsverlaufs nur eine mittlere Kurvengeschwindigkeit verwendet werden, würden die Warnungen zu früh oder zu spät ausgelöst werden:

- Der Mittelwert der Kurvenfahrten mit dem Versuchsträger (siehe Abbildung 4.15) beträgt  $69.79 \frac{km}{h}$ . Bei Annahme dieser Geschwindigkeit am Anfang der Kurve würde bei einer Fahrzeuggeschwin-

digkeit von  $150 \frac{km}{h}$  eine Warnung  $151.14m$  vor der Kurve ausgelöst werden. Dies wäre  $1.04m$  früher als eine Kurvenwarnung basierend auf einer Messwertreihe. Bei Annahme dieser Geschwindigkeit in der Kurvenmitte, würde erst  $59.14m$  vor der Kurve eine Warnung ausgelöst werden. Somit fehlen dem Fahrer über  $90m$  Strecke, um die Geschwindigkeit zu reduzieren.

- Bei den künstlich erzeugten Messfahrten (siehe Abbildung 4.17) fallen die Abweichungen zur Warnung basierend auf einer Messwertreihe drastischer aus. Der Mittelwert der Kurvengeschwindigkeit liegt in diesem Fall bei  $89.38 \frac{km}{h}$ . Bei Annahme dieses Mittelwerts am Kurveneingang, würde  $124.42m$  vor der Kurve und damit fast  $28m$  früher eine Warnung ausgelöst werden, die möglicherweise nicht notwendig wäre. Bei Annahme in der Kurvenmitte, würde die Warnung nur  $32.42m$  vor der Kurve und damit dem Fahrer rund  $64m$  weniger Strecke geben, um die Geschwindigkeit zu reduzieren.

Durch diesen Vergleich ist deutlich, dass eine präzise Warnung vom adaptiven Kurvenwarner nur möglich ist, wenn Messwertreihen gelernt werden können. Es können damit überflüssige Warnungen vermieden werden und korrekte Warnungen früher ausgegeben werden, wodurch der Fahrer mehr Zeit und Strecke zum Reagieren hat. Das Lernen dieser Messwertreihen mit der Kerndichteschätzung ist durch die in der vorliegenden Arbeit entwickelten und implementierten Messwertreihen-Transformation möglich geworden.

## 4.3 Zusammenfassung

In diesem Kapitel wurde eine Messwertreihen-Transformation erarbeitet, die es dem Lernprozess ermöglicht Messwertreihen mit der Kerndichteschätzung zusammenzuführen. Die Messwertreihen-Transformation bildet heterogene, zeitabhängige Messwertreihen auf ein gemeinsames, positionsabhängiges Raster mit beliebigem Rasterabstand ab und stellt damit eine eindeutige Zuordnung über die Rasterpunkte her. Auf einen Rasterpunkt abgebildete Messwerte bilden eine Messsequenz, die mit der Kerndichteschätzung im Lernprozess weiterverarbeitet werden kann. So sind entlang von Strecken die Wahrscheinlichkeitsverteilungen von Messgrößen für jeden Rasterpunkt verfügbar und ermöglichen eine detaillierte Vorhersage der wahrscheinlichsten Sensormesswerte.

## 4 Lernen von heterogenen Sensormesswertreihen

Die Resultate von der Anwendung der Messwertreihen-Transformation im Lernprozess zeigen deutlich die Vorteile gegenüber bisheriger Ansätze, die nur einzelne Messwerte zusammenführen. Der adaptive Kurvenwarner konnte durch das Lernen von Messwertreihen, im Vergleich zu einzelnen Messwerten, deutlich präzisere Warnungen ausgeben. Damit konnte eine Warnung zum korrekten Warnzeitpunkt ausgegeben werden, das dem Fahrer in manchen Situationen eine zusätzliche Strecke von 90m zur Verfügung stellte, um seine Geschwindigkeit anzupassen.

Abbildung 4.19 zeigt die Einbettung der Messwertreihen-Transformation in das lernende System der vorliegenden Arbeit.

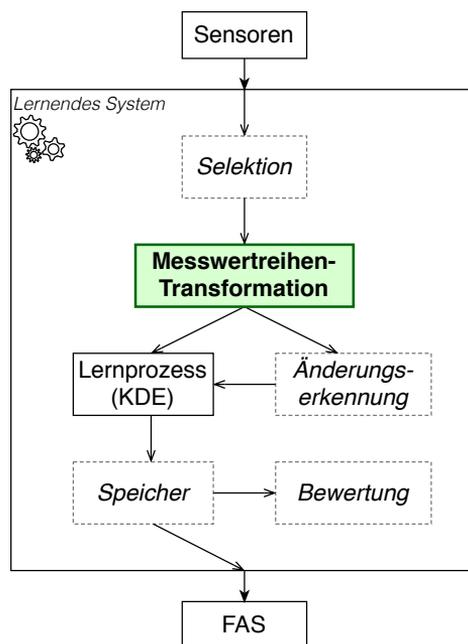


Abb. 4.19: Komponenten des lernenden Systems der vorliegenden Arbeit. Beiträge des 4. Kapitels sind grün hervorgehoben.

# Erkennung von ungültigen Informationen im lernenden System

---

Im folgenden Kapitel wird betrachtet, wie gelernte Informationen durch Änderungen in der Fahrzeugumgebung veralten und damit ungültig werden können. Um negative Einflüsse auf die Funktion von FAS durch ungültige Informationen zu vermeiden, wird untersucht wie ungültige Informationen schnell erkannt werden können.

Am Anfang von Abschnitt 5.1 werden bisherige Arbeiten vorgestellt, die sich mit veralteten und ungültigen Informationen beschäftigt haben. Anschließend wird eine verbesserte Änderungserkennung erarbeitet, die unabhängig vom Lernprozess ist und verschiedene Änderungstypen detektieren kann. Es folgt eine Erläuterung, wie mit dieser Erkennung der Lernprozess verbessert wird.

In Abschnitt 5.2 wird die in der vorliegenden Arbeit entwickelte Änderungserkennung mit künstlichen und simulierten Sensormesswerten getestet und mit bisherigen Verfahren verglichen.

## 5.1 Änderungserkennung zur Detektion von ungültigen Informationen

Die Informationen im lernenden System bilden oft direkt oder indirekt eine Repräsentation der Fahrzeugumgebung und des Fahrstils des Fahrers. Diese Informationen werden durch Sensoren erfasst und stellen beispielsweise die Position und Länge einer Kurve, die gefahrene Geschwindigkeit oder auch die Steigung der Straße dar. Ändern sich relevante Aspekte in der Fahrzeugumgebung, stimmen die gespeicherten Informationen nicht mehr mit der Realität überein und werden damit ungültig. Durch Änderungen ungültig gewordener Informationen können die Funktion von FAS einschränken oder vollständig verhindern und sollten deswegen schnell identifiziert und im lernenden System korrigiert werden. Wird beispielsweise

die Spurbreite in einer Kurve durch eine neue Baustelle verengt oder entsteht ein Schlagloch, ändert sich höchstwahrscheinlich auch die gefahrene Geschwindigkeit in dieser Kurve. Damit der adaptive Kurvenwarner eine korrekte Warnung ausgibt, sollte das Geschwindigkeitsprofil dieser Kurve mit Baustelle schnell korrigiert werden.

In Arbeiten über ein lernendes System im Fahrzeug wurde eine Erkennung von Änderungen in Sensormesswerten nicht immer betrachtet. Ein Ansatz für eine Änderungserkennung basierend auf dem Lernprozess wurde in [47] vorgestellt. Diese Änderungserkennung ist jedoch von der Qualität der Verteilung abhängig, die mit der Kerndichteschätzung im Lernprozess berechnet wurde. Wird eine ungünstige oder falsche Konfiguration für die Kerndichteschätzung gewählt, kann die Änderungserkennung negativ beeinflusst werden.

Es wird in diesem Abschnitt eine vom Lernprozess unabhängige Änderungserkennung für verschiedene Änderungsarten vorgestellt. Mit einer solchen Änderungserkennung kann eine konstante Leistungsfähigkeit gewährleistet werden. Darüber hinaus wird im Detail beschrieben, wie diese Änderungserkennung den Lernprozess der vorliegenden Arbeit erweitern kann, um ungültiges Wissen zu korrigieren.

### 5.1.1 Bisherige Arbeiten zum Umgang mit Änderungen

In diesem Abschnitt werden bisherige Arbeiten zur Erkennung von Änderungen und damit veralteten Informationen vorgestellt. Dafür wird zunächst auf allgemeine Änderungserkennungen in Informationssystemen eingegangen. Anschließend wird das Verfahren zur Änderungserkennung im Kontext von Fahrzeugsensoren aus [47] vorgestellt.

#### Erkennung von Änderungen in der Statistik

In [6, 86] wird eine große Anzahl an theoretischen Änderungserkennungen vorgestellt. Die Anwendungsgebiete beschränken sich auf die Bereiche der Signalverarbeitung von Sprache, Seismik und Biomedizin, sowie die Überwachung von Vibrationen in mechanischen Systemen oder eine Fehlererkennung in chemischen Prozessen.

Die dort vorgeschlagenen Änderungserkennungen passen nicht auf die Anforderungen und das Anwendungsgebiet der vorliegenden Arbeit. Es ist deswegen notwendig, eine eigene Änderungserkennung für den Kontext dieser Arbeit zu entwickeln.

## 5.1 Änderungserkennung zur Detektion von ungültigen Informationen

### Änderungserkennung über die Wahrscheinlichkeitsverteilung

In [47] wird ein Ansatz beschrieben, wie mit veralteten gelernten Informationen umgegangen werden kann. Dieser Ansatz besteht aus zwei Komponenten:

Eine Komponente erzeugt ein generelles, langsames Altern von Messwerten, welches die Gewichtung älterer Messungen in der Lernfunktion begrenzt. Damit haben neue Messungen einen größeren Einfluss auf die geschätzte Wahrscheinlichkeitsverteilung. Konkret wird dafür  $\kappa$  aus der Gleichung 4.2 auf einen maximalen Wert limitiert, welcher unterhalb des maximalen Werts der Anzahl bisher gelernter Messwerte entspricht.

Eine weitere Komponente dieses Ansatzes setzt ein beschleunigtes Altern um, welches die Gewichtung älterer Messungen in der Lernfunktion kurzzeitig drastisch verringert und diese somit schnell aus den gelernten Informationen entfernt. Diese Komponenten wird im Folgenden im Detail beschrieben.

**Eskalationsstufe:** Basierend auf einer Eskalationsstufe wird die Gewichtung der bisher gelernten Informationen in der Lernfunktion (Gleichung 4.2) gesteuert. Die Eskalationsstufe kann als Hinweis auf eine Änderung angesehen werden. Welche Eskalationsstufe aktiv ist, wird durch die *Passgenauigkeit* neuer Messwerte bestimmt. Eine Messung *passt*, wenn sie im Bereich der Verteilung liegt, welche die wahrscheinlichsten Werte enthält. Liegt eine Messung hingegen im unwahrscheinlichen Randbereich der Verteilung, passt diese nicht und eine *Eskalationsstufe* wird erhöht. Ob eine neue Messung  $v_{neu}$  zur bisher geschätzten Verteilung passt, wird anhand des Bereichs der Verteilung entschieden, in der die neue Messung eingeordnet wird. Dazu wird die Wahrscheinlichkeitsverteilung in fünf Bereiche, anhand von Flächeninhalten mit den Grenzen der Interdezil- (die oberen und unteren 10% der Verteilung) und Interquartilbereiche (die oberen und unteren 25% der Verteilung) unterteilt. Daraus entstehen die fünf Bereiche, wie sie in Abbildung 5.1 dargestellt sind.  $F()$  ist das Integral und berechnet für einen Messwert den absoluten Flächeninhalt der Verteilung von negativ unendlich bis zum gegebenen Messwert hin. Dieser Flächeninhalt liegt im Intervall  $[0..1]$ , da die Verteilung durch die Normierung immer eine Fläche von 1 hat.

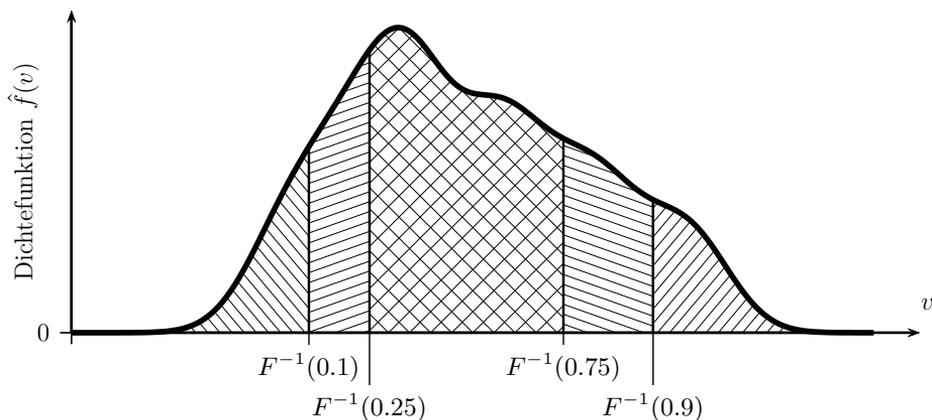


Abb. 5.1: Einteilung der Wahrscheinlichkeitsverteilung in fünf Bereiche über Flächeninhalte. (Grafik aus [47])

	$\eta < 0$	$\eta = 0$	$\eta > 0$
$F(v_{neu}) < 0.1$	-1	-1	-1
$0.1 \leq F(v_{neu}) < 0.25$	$\pm 0$	$\pm 0$	-1
$0.25 \leq F(v_{neu}) \leq 0.75$	+1	$\pm 0$	-1
$0.75 < F(v_{neu}) \leq 0.9$	+1	$\pm 0$	$\pm 0$
$F(v_{neu}) > 0.9$	+1	+1	+1

Tabelle 5.1: Tabelle für die Fallunterscheidung zur Berechnung der Eskalationsstufe aus [47]

Die Eskalationsstufe  $\eta$  wird nach jeder neu erfassten Messung aktualisiert und über folgende Gleichung mit Hilfe der Funktion  $\Delta\eta(F(v_{neu}), \eta)$  bestimmt:

$$\eta := \eta + \Delta\eta(F(v_{neu}), \eta) \tag{5.1}$$

Die Funktion  $\Delta\eta(F(v_{neu}), \eta)$  ist in Tabelle 5.1 beschrieben. Je nach aktueller Stufe  $\eta$  und der *Passgenauigkeit* einer neuen Messung  $v_{neu}$  wird über  $F()$  die neue Eskalationsstufe bestimmt, wobei die Stufe auf ein Maximum von  $\eta = 3$  beschränkt wird. Mit dieser Eskalationsstufe wird anschließend die Gewichtung  $\phi(\eta)$  der bis zu diesem Zeitpunkt gelernten Informationen

## 5.1 Änderungserkennung zur Detektion von ungültigen Informationen

bestimmt:

$$\phi(\eta) = \begin{cases} 1.0 & , |\eta| = 0 \\ 0.9 & , |\eta| = 1 \\ 0.6 & , |\eta| = 2 \\ 0.4 & , |\eta| = 3 \end{cases} \quad (5.2)$$

Je höher die Eskalationsstufe, desto kleiner ist der Faktor  $\phi(\eta)$ , welcher mit der Anzahl der Messungen  $\kappa$  der Lernfunktion (Gleichung 4.2) multipliziert wird. Dies führt dazu, dass in der Lernfunktion alte Informationen geringer gewichtet und damit schneller entfernt werden, wenn eine Änderung (unpassende Messwerte und damit eine höhere Eskalationsstufe) stattgefunden hat.

Es wurde in [47] eine Änderungserkennung vorgestellt. Wie zu sehen hängt die Erkennung direkt von der geschätzten Verteilung des Lernprozesses ab. Liegt eine ungünstig oder falsch geschätzte Verteilung vor, kann die Leistungsfähigkeit dieser Erkennung negativ beeinflusst werden. Bereits einzelne Messwerte im Randbereich der Verteilung lösen ein Entfernen von gelernten Informationen aus. So können einzelne Ausreißer zu einer fälschlichen Löschung von gültigen Daten führen.

### 5.1.2 Verschiedene Arten von Änderungen

Änderungen können sich auf unterschiedliche Art und Weise auf Sensormesswerte auswirken. In Versuchen zeigten sich zwei charakteristische Änderungen: Sprungartige und schleichende Änderungen.

**Sprungartige Änderungen:** Die sprungartige Änderung wirkt sich zwischen zwei Messungen eines Sensors aus. Der Verlauf der dabei erhobenen Werte ähnelt deswegen einer Sprungfunktion.

Beispiele für eine solche Änderung sind Baustellen, die in kurzer Zeit errichtet werden und dadurch schlagartig einen Einfluss auf die Sensormesswerte haben. Wird beispielsweise eine Baustelle in oder in der Nähe einer Kurve errichtet, ändert sich wahrscheinlich die Kurvengeschwindigkeit und damit sollte sich das Verhalten des Kurvenwarners auch ändern.

**Sleichende Änderungen:** Wirkt sich eine Änderung über mehrere Messwerte hinweg aus, ähnelt der Verlauf einer Rampenfunktion. Dies wird in der vorliegenden Arbeit als schleichende Änderung bezeichnet.

Dies tritt im Automobilumfeld beispielsweise auf, wenn die Straßenoberfläche im Winter über einen längeren Zeitraum kontinuierlich schlechter wird. Es können Risse in der Fahrbahn über einen Zeitraum stetig größer, tiefer und schließlich zu Schlaglöchern werden. Findet eine solche kontinuierliche Verschlechterung des Asphalts in einer Kurve statt, wird sich wahrscheinlich auch die Kurvengeschwindigkeit verringern. Das Ausbessern des Straßenbelags und von Schlaglöchern wäre wiederum eine sprungartige Änderung.

Die Änderungserkennung aus [47] betrachtet nicht explizit eine Erkennung von schleichenden Änderungen.

### 5.1.3 Hypothesentest zum Erkennen von Änderungen

Die genannten Änderungstypen sollen erkannt werden, um eine Reaktion darauf zu ermöglichen. Änderungen wirken sich auf die Messwerte der Sensoren aus, womit auch die Verteilung dieser Messwerte beeinflusst wird. Das führt zu zwei verschiedenen Verteilungen: Eine Verteilung der Messwerte vor einer Änderung und eine andere Verteilung nach einer Änderung.

Mit Hypothesentests ist es möglich zu überprüfen, ob sich zwei Verteilungen ähneln. Liegt eine Änderung vor, unterscheiden sich die Verteilungen und ein Hypothesentest kann dies erkennen, wodurch auch die Änderung erkannt wird.

**Hypothesentests:** In der Statistik werden Hypothesentests verwendet um zu überprüfen, ob die Verteilung einer Gruppe von Messwerten (Stichprobe genannt) einer anderen „Referenz“-Verteilung entspricht. Diese andere Verteilung wird entweder vorgegeben oder kann aus weiteren Stichproben berechnet werden. Abhängig davon, woher diese „Referenz“-Verteilung für den Hypothesentest kommt, werden die Hypothesentests unterschieden:

1. *Einstichproben-Tests:* Diese Tests berechnen eine Hypothese, ob eine Stichprobe zu einer definierten Verteilung gehört.
2. *Zweistichproben-Tests:* Diese Tests berechnen eine Hypothese, ob zwei gegebene Stichproben die gleiche Verteilung haben.
3. *Mehrstichproben-Tests:* Diese Tests berechnen eine Hypothese, ob drei oder mehr Stichproben z.B. einer gemeinsamen Population entstammen.

## 5.1 Änderungserkennung zur Detektion von ungültigen Informationen

In der vorliegenden Arbeit sollen Änderungen erkannt werden, die sich auf die Verteilung der Messwerte auswirken. Können zwei Stichproben gebildet werden, von denen eine aus Messwerten vor und eine aus Messwerten nach der Änderung besteht, kann ein Hypothesentest diesen Unterschied feststellen. Dafür muss der Test zwei Stichproben vergleichen können, weswegen für diese Anwendung die Zweistichproben-Tests geeignet sind.

Ein Hypothesentest geht grundsätzlich davon aus, dass die Verteilungen gleich sind. Diese Hypothese wird *Nullhypothese*  $H_0$  genannt. Erst wenn genug Hinweise auf unterschiedliche Verteilungen hindeuten, wird die gegenteilige *Alternativhypothese*  $H_1$  als wahr eingestuft. Wie diese Berechnung durchgeführt wird, ist vom Test abhängig. So wird bei manchen Hypothesentests der Median, der Mittelwert oder die Häufigkeitsverteilung der Stichprobe berechnet und daraus abgeleitet, ob die Messwerte der Stichproben die gleiche Verteilung haben.

In [6, 86] wird unter anderem auch die Möglichkeit angesprochen, eine Änderungserkennung über einen Hypothesentest umzusetzen, jedoch keine Änderungserkennung in Sensormesswerten im Fahrzeug vorgestellt. Der Hypothesentest gehört bezüglich des Berechnungsaufwands zu den einfacheren Algorithmen und eignet sich aus diesem Grund für den Einsatz im Fahrzeug, da für Berechnungen nur limitierte Ressourcen vorhanden sind. In [45] wird zwar der Einsatz eines Hypothesentests für eine Änderungserkennung in Sensormesswerten angesprochen, jedoch nicht weiter untersucht.

**Wahl eines Zweistichproben-Hypothesentests:** Für die Erkennung von Änderungen soll in der vorliegenden Arbeit ein Zweistichproben-Hypothesentest verwendet werden. Zwei bekannte Vertreter sind der *Zweistichproben-Kolmogorow-Smirnow-Test* und der *Zweistichproben-t-Test*.

- Der *Zweistichproben-Kolmogorow-Smirnow-Test* berechnet seine Hypothesen mit Hilfe einer relativen Häufigkeitsverteilung [16]. Diese gibt für jeden vorkommenden Wert einer Stichprobe an, wie oft dieser Wert vorkommt. Anschließend werden die beiden berechneten Häufigkeitsverteilungen der Stichproben auf Ähnlichkeit überprüft und eine entsprechende Hypothese abgeleitet. Der Kolmogorow-Smirnow-Test kann ohne Annahmen über die Verteilungen verwendet werden, gehört damit zur nichtparametrischen Statistik und passt in das Konzept der vorliegenden Arbeit.

## 5 Erkennung von ungültigen Informationen im lernenden System

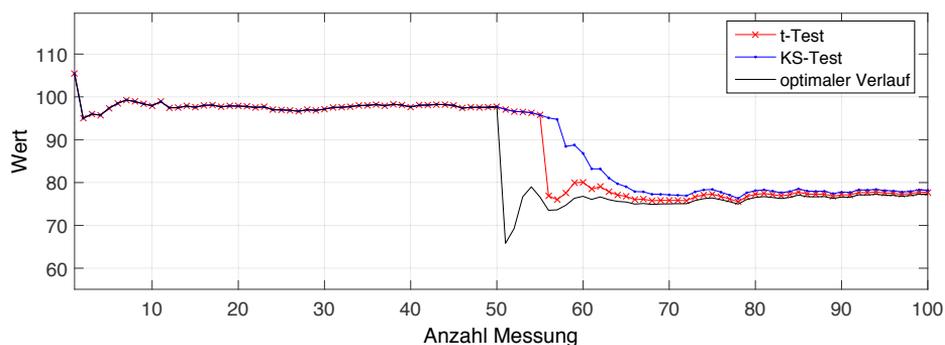


Abb. 5.2: Eine Messsequenz mit einer Änderung nach Messung 50. Gezeigt ist 1) der optimale Verlauf und die Mittelwertverläufe von 2) Änderungserkennung mit KS-Test und 3) Änderungserkennung mit t-Test.

- Der *Zweistichproben-t-Test* entscheidet anhand der Mittelwerte von zwei unabhängigen Stichproben, ob die Verteilungen gleich sind [30]. Voraussetzung dafür ist eine gleiche Varianz und eine Normalverteilung der Stichproben. Der Zweistichproben-t-Test gehört damit zur parametrischen Statistik und passt damit eher nicht in das Konzept der vorliegenden Arbeit, da die Voraussetzungen nicht mit Sicherheit erfüllt werden können.

Der folgende Vergleich zeigt die Schnelligkeit und Zuverlässigkeit der beiden Zweistichproben-Hypothesentests, eine Änderung zu erkennen. Dafür werden künstlich generierte Messsequenzen mit sprunghaften Änderungen verwendet.

In Abbildung 5.2 ist eine repräsentative Messsequenz aus über 40 durchgeführten Tests verwendet worden, um den KS-Test und den t-Test zu vergleichen. Die Messsequenz hat eine Standardabweichung von 10, bis zur Messung 49 einen Mittelwert von 100 und ab Messung 50 einen Mittelwert von 80. Der optimale Verlauf ist als schwarze Linie dargestellt, der nur mit Kenntnis über die Änderung erreicht werden kann. Die Änderungserkennung mit dem KS-Test ist als blaue Linie dargestellt. Es ist deutlich sichtbar, dass die Änderungserkennung mit dem t-Test (rote Linie) dem optimalen Verlauf besser folgen kann, da der t-Test die Änderung früher erkannt hat.

Obwohl bei der Anwendung in der vorliegenden Arbeit nicht immer eine gleiche Varianz beider Stichproben und eine Normalverteilung vorhan-

## 5.1 Änderungserkennung zur Detektion von ungültigen Informationen

den war, konnten wider Erwarten mit dem Zweistichproben-t-Test bessere Ergebnisse als mit dem Zweistichproben-Kolmogorow-Smirnow-Test erreicht werden. Aus diesem Grund wird für die vorliegende Arbeit der Zweistichproben-t-Test verwendet.

Weiterführende Informationen über statistische Tests und (Zweistichproben-) Hypothesentests können in folgender Fachliteratur gefunden werden: [80, 8].

### 5.1.4 Bilden von Stichproben für den Hypothesentest

Mit Kenntnis über den Änderungszeitpunkt können die Messwerte vor und nach der Änderung leicht getrennt werden. Messwerte vor dieser Änderung werden ungültig und das lernende System kann die neue Gegebenheit durch die Messwerte nach dieser Änderung abbilden.

Da der Änderungszeitpunkt nicht bekannt ist, soll dieser erkannt werden. Um dies zu ermöglichen, wird im Folgenden erklärt, wie die Stichproben für den Hypothesentest in der vorliegenden Arbeit gewählt werden: Die erste Stichprobe soll die Messwerte *vor* der Änderung und die zweite Stichprobe soll die Messwerte *nach* der Änderung enthalten. Werden die Stichproben so gewählt, kann der Hypothesentest diese Änderung erkennen. Im Folgenden wird beschrieben, wie die beiden Stichproben genau gewählt werden, um eine sprungartige Änderung erkennen zu können.

Werden alle Messwerte eines Rasterpunkts (siehe Abschnitt 4.1.3) zeitlich sortiert in eine Messsequenz  $M$  mit  $n$  Messungen überführt,

$$M = [M_0, \dots, M_n] \quad (5.3)$$

dann lassen sich zwei statische *Stichprobenfenster* definieren, die jeweils eine bestimmte Anzahl an Messungen umfassen und damit je eine Stichprobe für den Hypothesentest bilden. Das *Analysefenster*  $M_{Analyse}$  umfasst die aktuellsten Messungen mit einem Umfang von  $a$  und enthält die Messwerte

$$M_{Analyse} = [M_{n+1-a}, \dots, M_n] \quad (5.4)$$

aus der Messsequenz  $M$ . Das *Gedächtnisfenster*  $M_{Gedächtnis}$  enthält die älteren Messungen mit einem Umfang von  $g$  Messwerten und enthält die Messwerte

$$M_{Gedächtnis} = [M_{n+1-g-a}, \dots, M_{n-a}] \quad (5.5)$$

## 5 Erkennung von ungültigen Informationen im lernenden System

aus der Messsequenz  $M$ . In Abbildung 5.3 sind diese Fenster über einer beispielhaften Messsequenz eines Rasterpunkts dargestellt.

Sobald ein neuer Messwert zu der Messsequenz  $M$  hinzugefügt wird und anschließend  $n + 1$  Messungen enthält, werden die Fenster so verschoben, dass immer die aktuellsten Messungen mit den beiden Fenstern abgedeckt werden. Das Analysefenster wird dafür beim Hinzufügen eines neuen Messwerts nach  $n + 2 - a$  bis  $n + 1$  verschoben und das Gedächtnisfenster nach  $n + 2 - g - a$  bis  $n + 1 - a$ :

$$\begin{aligned} M &= [M_0, \dots, M_{n+1}] \\ M_{Analyse} &= [M_{n+2-a}, \dots, M_{n+1}] \\ M_{Gedächtnis} &= [M_{n+2-g-a}, \dots, M_{n+1-a}] \end{aligned} \quad (5.6)$$

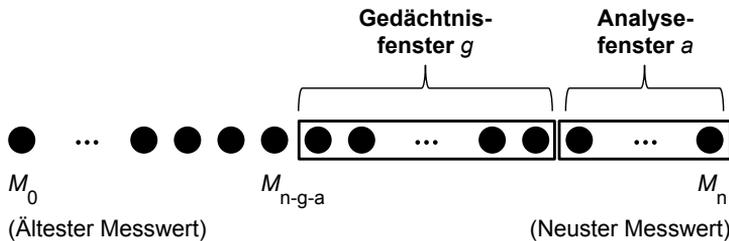


Abb. 5.3: Zwei Fenster bilden die Stichproben für den Zweistichproben-Hypothesentest.

Damit die Stichproben über die Fenster gebildet werden können, ist es notwendig, die innerhalb der Fenster liegenden Messwerte zusätzlich zur Messwertverteilung zu speichern. Damit erhöht sich der Speicheraufwand und es ist erforderlich, die Größe der Fenster mit Bedacht zu wählen, um den Speicherbedarf so gering wie möglich zu halten.

**Stichprobenfenstergröße:** Die Größe des Analyse- und Gedächtnisfensters hat einen Einfluss auf die Zuverlässigkeit und Geschwindigkeit der Erkennung von Änderungen. Je größer der jeweilige Stichprobenumfang ist, desto zuverlässiger ist die Hypothese des verwendeten Tests. Die so erhöhte Zuverlässigkeit führt jedoch zu einer langsameren Erkennung von Änderungen. [68]

Damit Änderungen möglichst schnell erkannt werden, sollte das Analysefenster möglichst klein sein. Nur so können die neuen Messwerte nach der Änderung die Stichprobe schnell dominieren und der Test schnell eine  $H_1$  Hypothese (unterschiedliche Verteilungen) ausgeben und damit die

## 5.1 Änderungserkennung zur Detektion von ungültigen Informationen

Änderung erkennen. Der Umfang des Gedächtnisfensters spielt in Bezug auf die Geschwindigkeit eine untergeordnete Rolle, da die Messwerte nach der Änderung lediglich das Analysefenster beeinflussen bis eine Änderung erkannt wird. Die Motivation, das Gedächtnisfenster zu limitieren, ist die Minimierung des Speicherplatzbedarfs der zu speichernden Daten. Werden alle bisher erhobenen Messwerte im Gedächtnisfenster gespeichert, würde der Speicherplatzbedarf mit jeder Fahrt linear ansteigen. Eine Begrenzung des Gedächtnisfensters führt zu einem konstanten Platzbedarf.

Es muss eine Balance zwischen den Kriterien *Zuverlässigkeit*, *Geschwindigkeit* und *Speicherplatzbedarf* gefunden werden. Dafür wird sich in der vorliegenden Arbeit an folgende Empfehlungen und Richtlinien gehalten:

1. **Zuverlässigkeit:** Beide Stichprobenfenster sollten so groß wie möglich sein, damit Änderungen zuverlässig erkannt werden.
2. **Geschwindigkeit:** Das Analysefenster sollte so groß wie nötig und so klein wie möglich sein, damit eine ausreichend schnelle Erkennung möglich ist.
3. **Speicherplatzbedarf:** Das Gedächtnisfenster sollte so groß wie nötig und so klein wie möglich sein, damit eine hohe Speichereffizienz und ein konstanter Speicherplatzbedarf gewährleistet werden kann.

Eine allgemeingültige Größe für das Analyse- und Gedächtnisfenster kann nicht angegeben werden, da sie beispielsweise je nach Sensor und Anwendung variieren kann. Im Falle der Geschwindigkeit für den adaptiven Kurvenwarner konnten mit einem Analysefenster der Größe 5 und Gedächtnisfenster der Größe 25 Ergebnisse erzielt werden. Die Messwerte in den Fenstern sind für den Hypothesentest notwendig und müssen zusätzlich zur geschätzten Verteilung der Kerndichteschätzung gespeichert werden. Der Speicherbedarf wird dadurch zwar erhöht, jedoch ist die Fenstergröße statisch und damit der Speicherbedarf weiterhin konstant.

Mit diesen Stichprobenfenstern können die Messwerte bestimmt werden, die durch den Zweistichproben-t-Test überprüft werden sollen. Wie die Erkennung genau funktioniert, wird im Folgenden erläutert.

### 5.1.5 Erkennen von sprunghaften Änderungen

Liegt das Analyse- und Gedächtnisfenster zur Bildung von Stichproben direkt hintereinander, können sprunghafte Änderungen sehr gut erkannt werden. Findet eine sprunghafte Änderung statt, liegen die Messwerte mit der neuen Verteilung innerhalb des Analysefensters. Damit kann der Hypothesentest die veränderte Verteilung der Messwerte im Analysefenster im Vergleich zu der des Gedächtnisfensters erkennen und damit auch die sprunghafte Änderung.

In Abbildung 5.4 ist diese Fensterkonfiguration über einer vereinfachten Messsequenz mit sprunghafter Änderung gezeigt. Das Analysefenster  $a1$  enthält im dargestellten Moment bereits Messwerte nach der Änderung. Zusätzlich eingezeichnet ist beispielhaft die Differenz des Mittelwerts beider Fenster ( $g1$  zu  $a1$ ) in Form eines Pfeils. Diese Differenz ist entscheidend für die Zuverlässigkeit der Änderungserkennung. Je größer diese Differenz ist, desto einfacher und zuverlässiger kann der Hypothesentest eine Änderung erkennen.

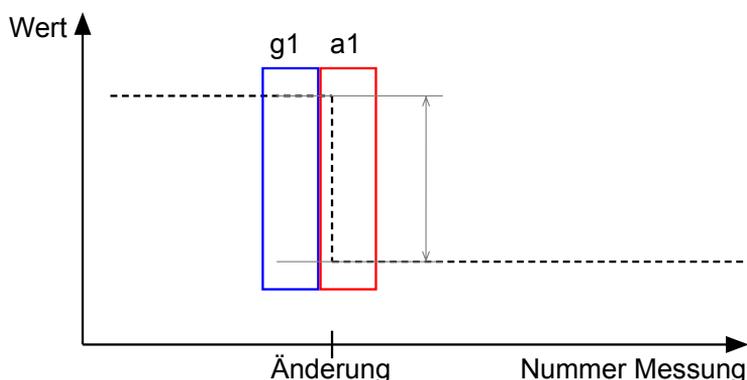


Abb. 5.4: Die Fensterkonfiguration zur Erkennung von sprunghaften Änderungen über einer Messwertreihe mit sprunghafter Änderung.

### 5.1.6 Erkennen von schleichenden Änderungen

Mit der bisher vorgestellten Fensterkonfiguration können schleichende Änderungen nur schlecht erkannt werden. Da sich die Änderung über viele Messwerte auswirkt, ist der Unterschied zwischen den Verteilungen im Analyse- und Gedächtnisfenster kleiner als bei einer sprunghaften Änderung.

## 5.1 Änderungserkennung zur Detektion von ungültigen Informationen

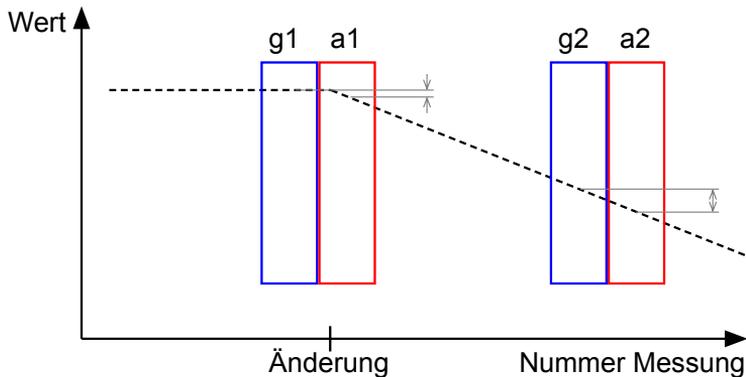


Abb. 5.5: Die Fensterkonfiguration zur Erkennung von sprunghaften Änderungen über einer Messwertreihe mit schleichender Änderung.

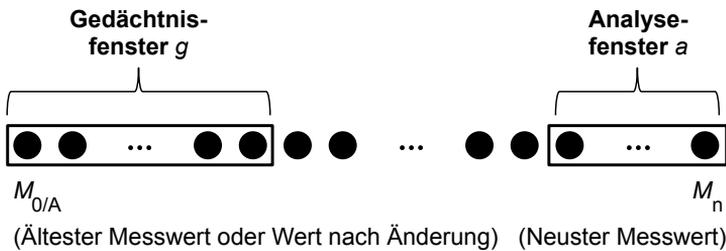


Abb. 5.6: Zwei Fenster bilden die Stichproben für die Erkennung von schleichenden Änderungen.

Dies ist beispielhaft in Abbildung 5.5 dargestellt. Die Differenz zwischen den beiden Fenstern  $g_1$  und  $a_1$  ist direkt nach dem Anfang der schleichenden Änderung relativ klein (vergleiche Differenz von  $g_1$  und  $a_1$  in Abbildung 5.4). Auch im weiteren Verlauf der Messsequenz nach der Änderung ist die Differenz zwischen den Fenstern, zum Beispiel bei  $g_2$  und  $a_2$ , nicht signifikant. Dadurch kann der Hypothesentest diese Änderung nicht oder nur sehr unzuverlässig erkennen.

Mit einer neuen Fensterkonfiguration zur Bildung der Stichproben, kann die Differenz zwischen den beiden Fenstern vergrößert werden: Wird das Gedächtnisfenster nicht mehr mitbewegt, liegt es statisch über den ältesten Messwerten (jene die zuerst gelernt wurden). Es wird nur bei einer erkannten Änderung umgesetzt. Die neue Fensterkonfiguration ist in Abbildung 5.6 dargestellt.

## 5 Erkennung von ungültigen Informationen im lernenden System

---

Bei einer Messsequenz  $M$  mit  $n$  Messwerten seit der letzten Änderung  $A$  oder seit Beginn 0, falls bisher keine Änderung stattgefunden hat:

$$M = [M_{0/A}, \dots, M_n] \quad (5.7)$$

gilt für das Analysefenster die gleiche Bildungsvorschrift wie bisher:

$$M_{Analyse} = [M_{n+1-a}, \dots, M_n] \quad (5.8)$$

und für das Gedächtnisfenster eine angepasste Bildungsvorschrift:

$$M_{Gedächtnis2} = [M_{0/A}, \dots, M_{0/A+g}] \quad (5.9)$$

Die Bildungsvorschrift für das Gedächtnisfenster ändert sich in der neuen Fensterkonfiguration nicht mehr mit der Anzahl der Messung  $n$ , sondern nur noch in Abhängigkeit der letzten Änderung  $A$ .

Abbildung 5.7 zeigt die Fensterkonfiguration des zweiten Hypothesentests über einer Messsequenz mit schleichender Änderung. Das statische Gedächtnisfenster  $gs$  liegt über den ältesten Messwerten, beziehungsweise den ältesten Messwerten seit der letzten Änderung. Die Differenzen zwischen dem Gedächtnisfenster und den Analysefenstern  $a1$  und  $a2$  sind jeweils als Pfeile eingezeichnet. Die Analysefenster  $a1$  und  $a2$  in Abbildung 5.7 sind identisch mit den Fenstern  $a1$  und  $a2$  aus der Abbildung 5.5, um eine Vergleichbarkeit der eingezeichneten Differenz zu ermöglichen. Die Fensterkonfiguration für schleichende Änderungen bringt für das Analysefenster  $a1$  keine Verbesserung. Doch im weiteren Verlauf der Änderung wächst die Differenz kontinuierlich an. Das Analysefenster  $a2$  zeigt mit der neuen Fensterkonfiguration für schleichende Änderungen eine signifikante Differenz.

Damit auch schleichende Änderungen zuverlässig erkannt werden können, wird eine zweite Instanz des Hypothesentests parallel zur ersten ausgeführt. Die Stichproben der zweiten Instanz werden mit der beschriebenen neuen Fensterkonfiguration erzeugt. Ziel der neuen Fensterkonfiguration ist die Vergrößerung der Differenz zwischen den beiden Fenstern während einer schleichenden Änderung. Durch eine vergrößerte Differenz kann der zweite Hypothesentest die Änderung zuverlässiger erkennen.

Mit dieser zweiten Fensterkonfiguration in Verbindung mit einer zweiten, parallel arbeitenden Instanz des Zweistichproben-t-Test ist auch eine Erkennung von schleichenden Änderungen möglich. Die Messungen des neuen Gedächtnisfensters müssen dafür zusätzlich gespeichert werden, das den Speicherbedarf leicht erhöht. Mit einer Gedächtnisfenstergröße von 35 Messungen konnten beim Anwendungsfall des adaptiven Kurvenwarners schleichende Änderungen erkannt werden.

## 5.1 Änderungserkennung zur Detektion von ungültigen Informationen

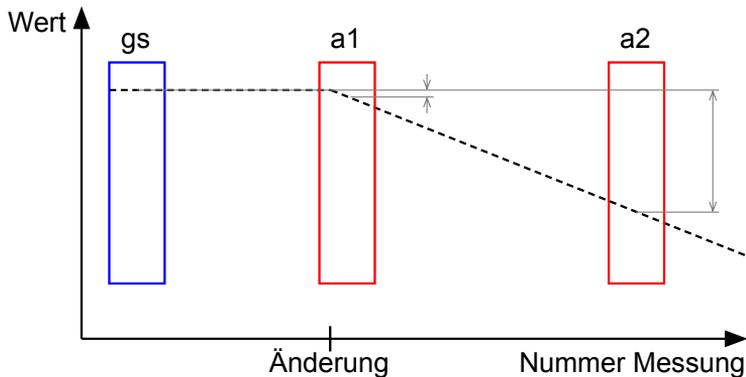


Abb. 5.7: Fensterkonfiguration der erweiterten Änderungserkennung über einer Messwertreihe mit schleichender Änderung.

### 5.1.7 Integration der Änderungserkennung in den Lernprozess

In diesem Abschnitt wird erläutert wie die Ergebnisse der Hypothesentests den Lernprozess beeinflussen. Eine Möglichkeit die ungültigen Daten zu korrigieren, ist deren Entfernung aus dem lernenden System. Dafür wird im Folgenden ein Algorithmus vorgestellt.

Der Zweistichproben-t-Test gibt mit seinen Hypothesen Hinweise auf eine Änderung in den Sensormesswerten. Das Entfernen von ungültig gewordenen Informationen wird im Lernprozess bei der Kerndichteschätzung gesteuert. Um das Entfernen im Lernprozess auf Basis der ausgegebenen Hypothesen auszulösen, ist ein Algorithmus notwendig, der in diesem Abschnitt erarbeitet wird. Da dieser Algorithmus auf dem Zweistichproben-t-Test basiert, wird er als **t-Test-Vergessen** bezeichnet.

In Abschnitt 4.1.1 wurde die Lernfunktion der Kerndichteschätzung und deren Verwendung in dem Lernprozess beschrieben. Die Gleichung 4.2 der Lernfunktion mit Kerndichteschätzung  $Lern_{KDE}$  besteht vereinfacht betrachtet aus zwei Komponenten:

1. Eine Komponente  $K_{alt}$  für die bisher geschätzte Verteilung bestehend aus zuvor erhobenen Messwerten.
2. Eine zweite Komponente  $K_{neu}$  für die Verteilung des neuen Messwerts, der hinzugefügt werden soll.

## 5 Erkennung von ungültigen Informationen im lernenden System

So lassen sich zwei Faktoren  $\alpha$  und  $\beta$  einführen, die eine Gewichtung der alten, ungültigen Messwerte  $K_{alt}$  und der neuen Messwerte  $K_{neu}$  ermöglichen:

$$Lern_{KDE} = \alpha \cdot K_{alt} + \beta \cdot K_{neu} \quad (5.10)$$

Diese Schätzung der Verteilung wird für jeden Rasterpunkt einer Situation berechnet, wie in Abschnitt 4.1 beschrieben. Folglich wird auch die Änderungserkennung für jeden Rasterpunkt getrennt durchgeführt.

Der *t-Test-Vergessen*-Algorithmus verwendet als Eingabewert die Hypothesen  $H_0$  (Verteilungen der Stichproben sind gleich) und  $H_1$  (Verteilungen der Stichproben unterscheiden sich) des Zweistichproben-t-Tests und gibt die beiden Gewichte  $\alpha$  und  $\beta$  für die Lernfunktion aus. Dieser Aufbau ist in Abbildung 5.8 zu sehen.



Abb. 5.8: Die Steuerung des Lernprozesses über das *t-Test-Vergessen*.

**Das t-Test-Vergessen:** Der *t-Test-Vergessen*-Algorithmus entscheidet anhand der  $H_0$  und  $H_1$  Hypothesen, ob eine Änderung vorliegt. Dabei werden die  $H_1$  Hypothesen der beiden Instanzen mit dem logischen Operator *ODER* zusammengeführt und verwendet. Wird vom Hypothesentest wiederholt eine unterschiedliche Verteilung zwischen den beiden Stichproben erkannt, ist also die Hypothese  $H_1$  über mehrere neue Messungen hintereinander wahr, dann wird dies vom Algorithmus als Änderung eingestuft. Sobald eine Änderung erkannt wird, werden die Faktoren zur Steuerung der Gewichtung der alten und neuen Messungen angepasst. Der Vergessensfaktor  $\gamma$  im Wertebereich  $[0, 1]$  soll zur Konfiguration der Faktoren  $\alpha$  und  $\beta$  verwendet werden und damit bestimmen, wie schnell die alten Messungen von den neuen Messungen nach der Änderung verdrängt werden. Der Faktor  $\alpha$  gewichtet die alten Messwerte und berechnet sich nach:

$$\alpha = \gamma \quad (5.11)$$

Der Faktor  $\beta$  gewichtet wiederum die neuen Messwerte und wird wie folgt berechnet:

$$\beta = 1 - \gamma \quad (5.12)$$

## 5.1 Änderungserkennung zur Detektion von ungültigen Informationen

Das bedeutet eine schnelle Abwertung der alten, ungültigen Messwerte bei einem kleinen  $\gamma$  Wert.

Das Verhalten des *t-Test-Vergessen*-Algorithmus ist in Abbildung 5.9 als Zustandsdiagramm dargestellt. Zustände *A* und *B* sind die *Analyse-zustände* des Algorithmus, bei denen auf eine Änderung gewartet wird. Zustand *C* ist der *Vergessenszustand*, bei dem die alten Messwerte vor der Änderung abgewertet und damit gelöscht werden. Was genau in welchen Zuständen passiert, wird im Folgenden beschrieben.

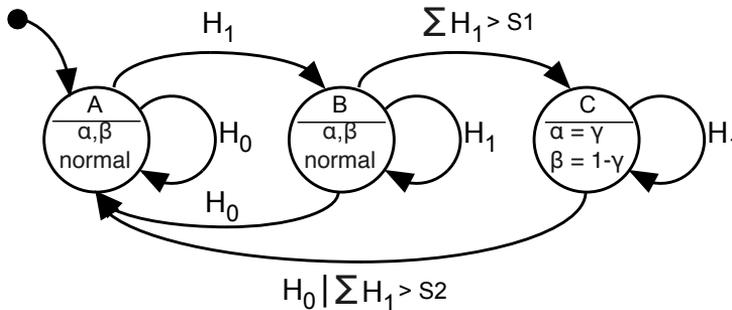


Abb. 5.9: Zustandsdiagramm vom *t-Test-Vergessen*-Algorithmus.

- **Zustand A:** Dies ist der Startzustand und beobachtet die Ergebnisse des Hypothesentests. Er wird verlassen sobald eine  $H_1$  Hypothese als Eingabewert vorliegt.
- **Zustand B:** Dieser Zustand beobachtet die folgenden Hypothesen, nachdem neue Messungen durchgeführt wurden, und fällt bei einer  $H_0$  Hypothese zurück in den Zustand A. Die Faktoren  $\alpha$  und  $\beta$  sind in den Zuständen A und B im Normalzustand, bei dem jede Messung die gleiche Gewichtung erhält. Konkret gilt für  $\alpha = \frac{n-1}{n}$  und  $\beta = \frac{1}{n}$ , wobei  $n$  die Anzahl der bisher in die geschätzte Verteilung eingeflossenen Messungen ist.
- **Zustand C:** Wird in Zustand B eine Änderung erkannt, die sich durch die Überschreitung der Anzahl an  $H_1$  Hypothesen über den Schwellwert  $S_1$  äußert, wird in den Zustand C gewechselt. Dort verändern sich die Faktoren  $\alpha$  und  $\beta$  der Lernfunktion, wie in den Gleichungen 5.11 und 5.12 beschrieben, wobei  $\gamma$  der vorgestellte Vergessensfaktor mit Wertebereich  $[0, 1]$  ist.

## 5 Erkennung von ungültigen Informationen im lernenden System

---

Wird die Änderung im Zustand  $C$  wieder erkannt ( $H_1$  ist wahr), erhöht sich mit jeder Fahrt  $f$  die Konfidenz über eine korrekt erkannte Änderung. Je höher diese Konfidenz ist, desto stärker werden die alten Messungen abgewertet, da die Gewichtungsfaktoren bei jeder Fahrt erneut mit den Termen  $K_{alt}$  und  $K_{neu}$  der Lernfunktion multipliziert wird (siehe Gleichung 5.10). Diese Steigerung findet bis zu einem Schwellwert  $S2$  statt. Werden mehr  $H_1$  Hypothesen als  $S2$  erkannt, wird wieder in den Zustand  $A$  gewechselt. Das Gedächtnisfenster der zweiten Instanz des Hypothesentests wird dabei versetzt und das  $t - Test - Vergessen$  kann die nächste sprunghafte oder schleichende Änderung erkennen.

Für den Anwendungsfall des adaptiven Kurvenwarners konnten Änderungen mit folgenden Schwellwerten innerhalb weniger Fahrten nach der Änderung erkannt werden:  $S1 = 3$  und  $S2$  wurde auf die Größe des Analysefensters gesetzt ( $S2 = 5$ ).

**Abwertung der ungültigen Informationen:** Der Verlauf des Anteils der alten und neuen Messwerte im lernenden System ist in Abbildung 5.10 beispielhaft für  $\gamma = 0.4$  in Prozent dargestellt. Dieser Verlauf ergibt sich, wenn der Hypothesentest durchgehend die Hypothese  $H_1$  ausgibt. Bei Messung 1 wurde keine Änderung erkannt und die Gewichtung ist im Normalzustand: Jeder Messwert wird gleichgewichtet. Ab Messung 2 wird eine Änderung erkannt und die Gewichtung der alten Messwerte nimmt durch  $\alpha$  ab. Im Umkehrschluss nimmt die Gewichtung der neuen Messwerte durch den Faktor  $\beta$  zu. Schon ab der dritten Messung nach der Änderung haben die alten Messwerte nur noch einen Anteil von 6.4% an der geschätzten Verteilung dieser Messgröße. Die neuen Messwerte nach der Änderung dominieren dann bereits mit 93.6%.

Neben der Anpassung der Gewichtung wird auch die Bandbreite der Kerndichtefunktion zurückgesetzt. Die neu geschätzte Verteilung der Kerndichteschätzung basiert durch die Änderung auf weniger Messwerten als davor. Da die Bandbreite nach der Anzahl der bereits eingeflossenen Messwerte  $n$  gewählt werden sollte [81, 47], wird sie bei einer Änderung zurückgesetzt und mit  $n = 1$  neu berechnet.

## 5.2 Evaluation der Änderungserkennung t-Test-Vergessen

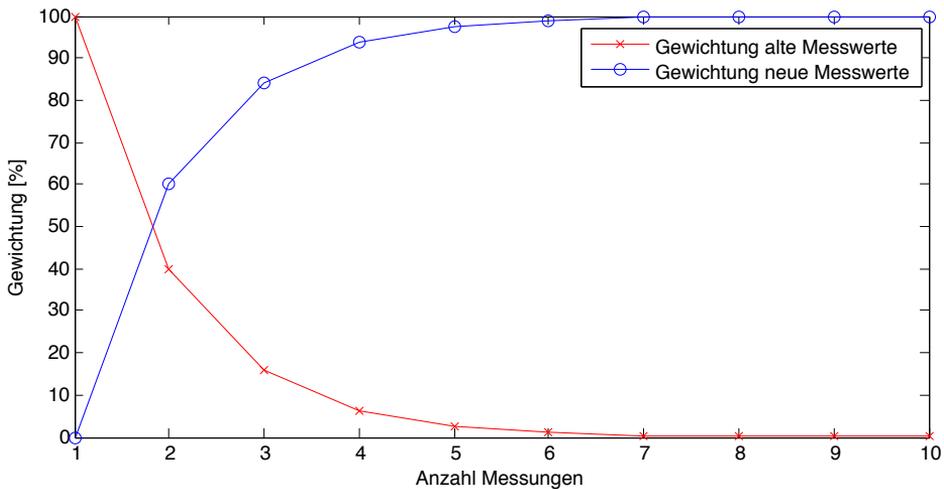


Abb. 5.10: Verlauf der Gewichtung der ungültigen und neuen Informationen nach Erkennung einer Änderung.

## 5.2 Evaluation der Änderungserkennung t-Test-Vergessen

In diesem Abschnitt wird ein Vergleich der Leistungsfähigkeit des *t-Test-Vergessen* mit bisherigen Verfahren durchgeführt. Dazu werden sowohl künstliche als auch realitätsnahe, simulierte Messwerte verwendet. Echte Messwerte aus dem Versuchsträger konnten nicht in ausreichender Menge für diese Evaluation erhoben werden. Mit diesen Messwerten soll die Leistungsfähigkeit bei sprunghaften und schleichenden Änderungen evaluiert werden.

### 5.2.1 Erstellung von Messsequenzen

Für den belastbaren Vergleich der Verfahren zur Änderungserkennung sind viele Messungen erforderlich. Die dafür verwendeten Messsequenzen werden auf verschiedene Arten erzeugt:

1. **Künstliche Messwerte:** Erzeugung von künstlichen Sensormesswerten durch geeignete Funktionen.
2. **Simulierte Messwerte:** Führen eines Fahrzeugs in einer realitätsnahen Fahrzeugsimulation und Speichern der dabei simulierten Sensormesswerte.

### Künstliche Messwerte

Über geeignete Funktionen können künstliche Messwerte erzeugt werden, die echten Messwerten ähneln. Der große Vorteil ist der vergleichsweise geringe Aufwand zum Erzeugen der Messungen.

Bei der Berechnung von Messwerten wird davon ausgegangen, dass sich eine sprungartige Änderung in Form eines verschobenen Mittelwerts auf die Messwerte auswirkt. Wird eine Kurve unter Normalbedingungen im Mittel mit  $v_{Mittel}$  befahren, wird sich der Mittelwert der Geschwindigkeit nach der Änderung um  $v_{Abweichung}$  verschieben. Da kein Fahrer in jeder Kurvenfahrt exakt die gleiche Geschwindigkeit fährt, entsteht eine Varianz der Werte, die berücksichtigt werden muss. Diese wird in Form eines normalverteilten Rauschens  $g()$  mit einer passenden Standardabweichung  $\sigma_g$  zum Mittelwert addiert.

Messwerte für ein normales Durchfahren einer Kurve werden mit der Gleichung für  $w_n$  berechnet:

$$w_n = v_{Mittel} + \sigma_g \cdot g() \quad (5.13)$$

und Fahrten durch die Kurve nach einer Änderung mit der Gleichung für  $w_{a1}$ :

$$w_{a1} = v_{Mittel} + v_{Abweichung} + \sigma_g \cdot g() \quad (5.14)$$

Für schleichende Änderungen wird die Abweichung  $v_{Abweichung}$  über eine gegebene Anzahl an Messwerten gleichmäßig hinzugefügt. Die schleichende Änderung startet bei Messung  $n_{as}$  und endet bei Messung  $n_{ae}$  und kann mit  $w_{a2}(n)$  berechnet werden:

$$w_{a2}(n) = \begin{cases} v_{Mittel} + \sigma_g \cdot g() & , n < n_{as} \\ v_{Mittel} + \frac{n-n_{as}}{n_{ae}-n_{as}} \cdot v_{Abweichung} + \sigma_g \cdot g() & , n_{as} \leq n \leq n_{ae} \\ v_{Mittel} + v_{Abweichung} + \sigma_g \cdot g() & , n > n_{ae} \end{cases} \quad (5.15)$$

Mit diesen Gleichungen lassen sich beliebig viele verschiedene Messsequenzen zum Testen generieren, die den erhobenen echten Messwerten ähneln.

### Simulierte Messwerte

Die zweite Art Messwerte zu erheben, erfolgt über eine Simulation. Zum Generieren von realitätsnahen Messwerten werden in der Simulation Kurvenfahrten durchgeführt. Dabei wird das Fahrzeug manuell mit einem

## 5.2 Evaluation der Änderungserkennung t-Test-Vergessen



Abb. 5.11: Fahrzeug wird in der realitätsnahen Simulation über ein Lenkrad und Pedalen (nicht abgebildet) gesteuert.

Lenkrad, Gas- und Bremspedal gesteuert (siehe Abbildung 5.11). Damit wird eine möglichst natürliche Steuerung des Fahrzeugs gewährleistet. So entstehen keine unerwünschten, unnatürlichen Artefakte, wie sie beispielsweise durch eine Steuerung über eine Tastatur oder einen Spielecontroller entstehen würden.

Im Simulator werden die Fahrzeugsensoren nachgebildet, wodurch realitätsnahe Sensormesswerte für die Fahrten zur Verfügung stehen. Die simulierten Sensoren erheben in den gleichen festen Zyklen die Messwerte, wie die äquivalenten Sensoren im Versuchsträger. In der Tabelle 5.2 sind die simulierten Sensoren aufgelistet.

Als Grundlage für die Simulationsumgebung wird das Straßennetz aus dem OSM Kartenmaterial verwendet. Um Messequenzen für den adaptiven Kurvenwarner zu erzeugen, wird ein Kartenausschnitt von Berlin

## 5 Erkennung von ungültigen Informationen im lernenden System

Sensor	Einheit
Geschwindigkeit	$\frac{m}{s}$
Querbeschleunigung	$\frac{m}{s^2}$
Längsbeschleunigung	$\frac{m}{s^2}$
Hochbeschleunigung	$\frac{m}{s^2}$
Gierrate	$\frac{1}{s}$
Lenkradwinkel	°
Ausrichtung des Fahrzeugs	°
GPS Position	Längen- & Breitengrad
Gaspedalposition	%
Bremspedalposition	%

Tabelle 5.2: Die wichtigsten Sensordaten und deren Einheiten, die von der Simulation berechnet werden.

importiert. Die verwendete Kurve befindet sich auf der Straße *Alt-Lietzow* und ist in Abbildung 5.12 dargestellt.

Nach dem Import des OSM Kartenmaterials in die Simulation, ist dieses Straßennetz befahrbar. Ein Bildschirmfoto dieser Kurve im Normalzustand ist in Abbildung 5.13 zu sehen. Um eine Änderung zu simulieren, werden Hindernisse in diese Kurve eingefügt, wie in Abbildung 5.14 dargestellt. Die paarweise aufgestellten Hindernisse verengen die Fahrspur in der Kurve, ähnlich einer Spurverengung bei Baustellen. Darüber hinaus verändern die aufgestellten Hindernisse den Spurverlauf durch die Kurve (grüne Linie in Abbildung 5.14).

Für die Erhebung von Messwerten in den beiden beschriebenen Szenarien, *Normalzustand* und *mit Änderung*, wurden Regeln aufgestellt. Diese Regeln bestimmen, ob eine Messfahrt gültig ist und für den Vergleich verwendet werden darf:

1. Das Fahrzeug darf die Straße zu keinem Zeitpunkt der Fahrt verlassen.
2. Die aufgestellten Hindernisse dürfen nicht in Berührung mit dem Fahrzeug kommen.

In Abbildung 5.15 sind mehrere GPS Spuren von Fahrten im Simulator, durch die gewählte Kurve zu sehen. Die eingezeichneten Schilder stellen den erkannten Kurveneingang jeder einzelnen Durchfahrt dar. Für diese Kurvenfahrten sind jeweils alle Sensorinformationen aus Tabelle 5.2

## 5.2 Evaluation der Änderungserkennung t-Test-Vergessen



Abb. 5.12: Kurve der Straße *Alt-Lietzow* in Berlin die für die Messfahrten in der Simulation befahren wurde (Quelle: OSM Kartenmaterial).

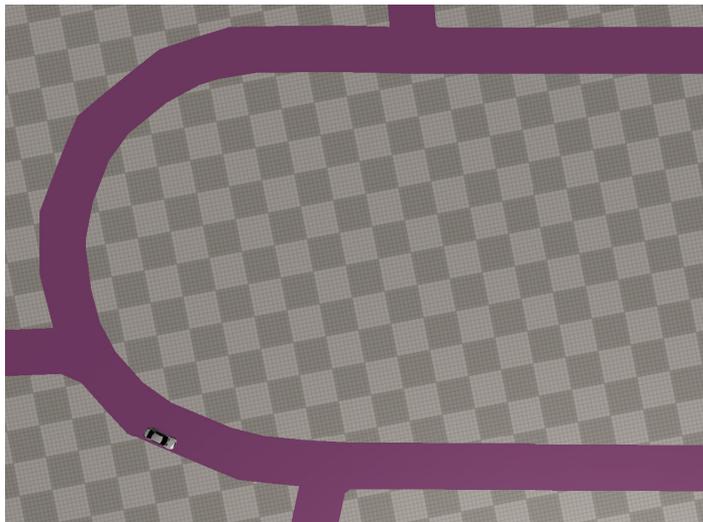


Abb. 5.13: Bildschirmfoto der Simulation mit ausgewählter Kurve (Straße *Alt-Lietzow* in Berlin) im Normalzustand.

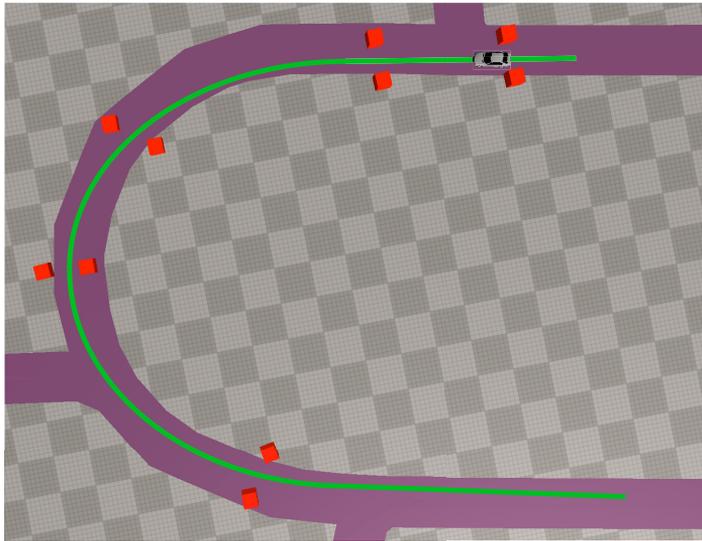


Abb. 5.14: Bildschirmfoto der Simulation mit ausgewählter Kurve (Straße *Alt-Lietzow* in Berlin) und einer Änderung durch Hindernisse auf der Strecke.

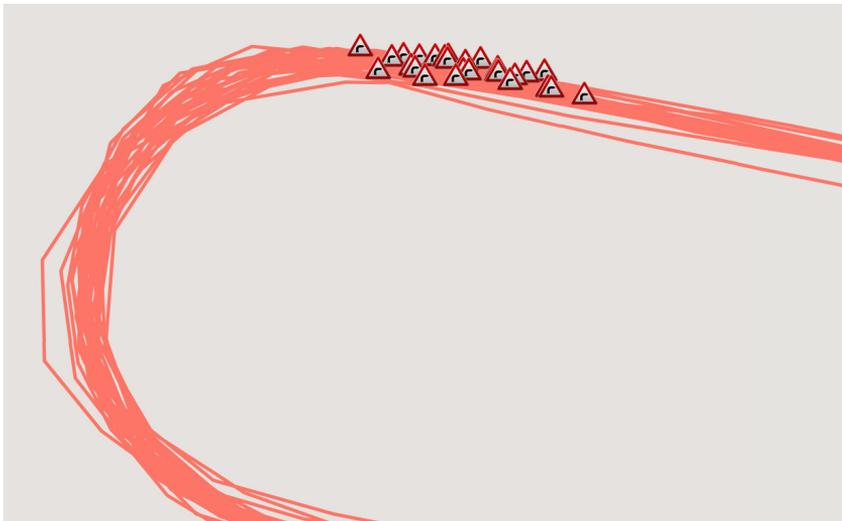


Abb. 5.15: GPS Spuren des Fahrzeugs durch eine Kurve mit Markierungen für erkannte Kurveneingänge.

verfügbar und können entweder direkt in das lernende System als CAN-Nachrichten eingespielt oder als Datensammlung gespeichert und mit anderen Programmen ausgewertet werden.

Mit diesen Möglichkeiten Messwerte zu erhalten, kann eine eingehende Prüfung der Leistungsfähigkeit des *t-Test-Vergessens* erfolgen.

### 5.2.2 t-Test-Vergessen bei sprungartigen Änderungen

In diesem Abschnitt wird das in der vorliegenden Arbeit neu entwickelte Verfahren zur Änderungserkennung, das *t-Test-Vergessen* (siehe Abschnitt 5.1), mit einfachen Verfahren ohne dedizierte Änderungserkennung und der Änderungserkennung aus [47] verglichen. Diese Verfahren sind:

- Gleichgewichtung aller Messwerte
- Mitteln einer limitierten Anzahl der neusten Messwerte [13]
- Änderungserkennung über die *Eskalationsstufe* [47]

Dazu wurden Messsequenzen mit je 100 Messungen künstlich erzeugt und mit den zu vergleichenden Verfahren verarbeitet. Für eine Messsequenz mit sprungartiger Änderung wird folgendes Schema verwendet: Die erste Hälfte (Messung 1 bis 50) besteht aus Messungen vor einer Änderung und die zweite Hälfte (Messung 51-100) besteht aus Messungen nach einer Änderung.

#### Vergleich der Änderungserkennungen mit einer Messsequenz

Eine beispielhafte Messsequenz aus simulierten Messwerten nach dem genannten Schema ist im ersten Plot der Abbildung 5.16 zu sehen. Der Änderungspunkt ist als vertikaler schwarzer Balken eingezeichnet. An diesem Punkt fällt die durchschnittliche Geschwindigkeit von circa  $50 \frac{km}{h}$  auf circa  $40 \frac{km}{h}$ . Im zweiten Plot sind die entsprechenden Ergebnisse des Zweistichproben-t-Test und der Änderungserkennung über die Eskalationsstufe aus [47] zu sehen.

- Die Eskalationsstufe (grün dargestellt) kann vier Stufen annehmen: Stufe 0 hat den Wert 0 auf der y-Achse, Stufe 1 den Wert  $\frac{1}{3}$ , Stufe 2 den Wert  $\frac{2}{3}$  und Stufe 3 den Wert 1 (Details dazu siehe Abschnitt 5.1.1). Je höher die Stufe ist, desto wahrscheinlicher ist laut diesem Verfahren eine Änderung erkannt worden.

## 5 Erkennung von ungültigen Informationen im lernenden System

- Die Ergebnisse des Hypothesentests des *t-Test-Vergessen* sind in blau (Wahrscheinlichkeit für  $H_0 = 1$ , Stichproben haben die gleiche Verteilung) und rot ( $H_1 = 1$ , Stichproben haben eine unterschiedliche Verteilungen) dargestellt.

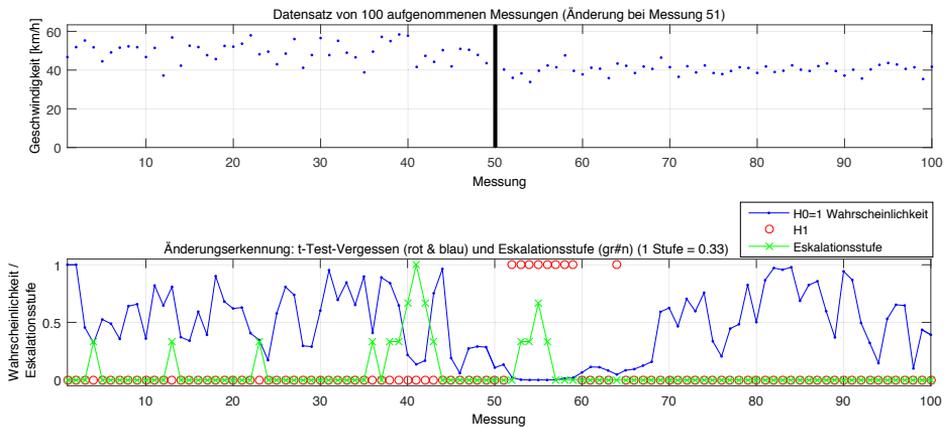


Abb. 5.16: Erster Plot: Messsequenz mit 100 simulierten Messwerten und sprunghafter Änderung bei Messung 51. Zweiter Plot: Ergebnisse der Änderungserkennung mit Eskalationsstufe aus [47] und dem *t-Test-Vergessen*.

An der Eskalationsstufe ist erkennbar, dass die Änderungserkennung aus [47] bereits bei Messung 40 eine Änderung vermutet. Dies ist ein Falschalarm, da an dieser Stelle in der Messsequenz keine Änderung stattgefunden hat. Die eigentliche Änderung bei Messung 51 wird von beiden Verfahren mit kurzer Verzögerung erkannt.

Erkennt ein Verfahren eine Änderung (egal ob Falschalarm oder korrekte Erkennung), werden die für ungültig gehaltenen Daten in den gelernten Informationen abgewertet, wodurch die geschätzten Verteilungen der Kerndichteschätzung beeinflusst werden.

Insgesamt werden für die folgenden Vergleiche vier Verteilungen berechnet (*t-Test-Vergessen*, Eskalationsstufe aus [47], Gleichgewichtung einer limitierten Anzahl von Messwerten aus [13], Gleichgewichtung aller Messwerte).

In Abbildung 5.17 sind die vier geschätzten Verteilungen mit der Kerndichteschätzung nach der Verarbeitung einer Anzahl an Messungen der Messsequenz aus Abbildung 5.16 zu sehen: Der erste Plot zeigt die Verteilungen nach der Verarbeitung der ersten 60 Messungen (9 Messungen

## 5.2 Evaluation der Änderungserkennung t-Test-Vergessen

nach der Änderung). Der zweite Plot zeigt die geschätzten Verteilungen nach der Verarbeitung der gesamten 100 Messungen. Es sind deutliche

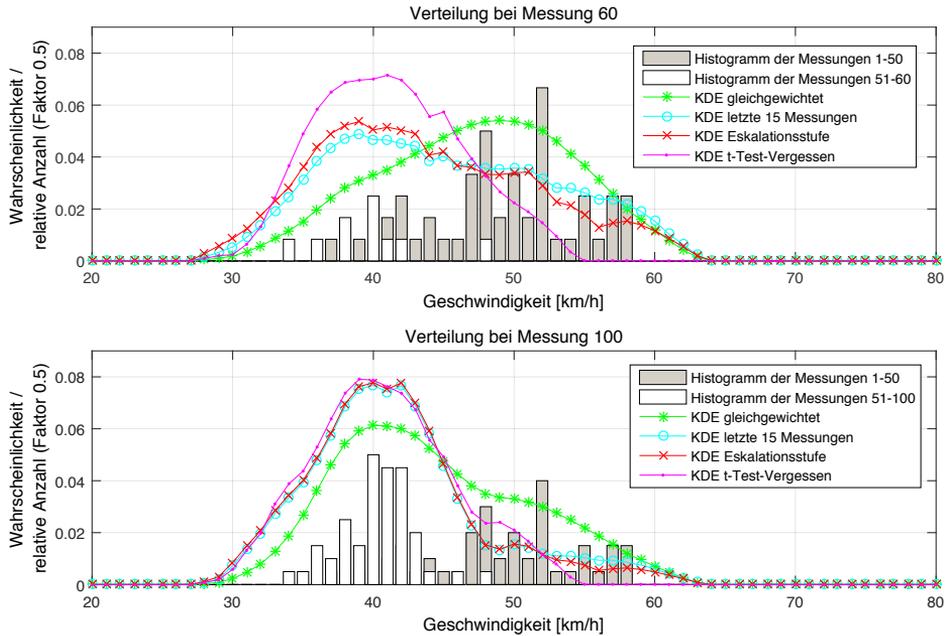


Abb. 5.17: Geschätzte Verteilungen mit den verschiedenen Verfahren zur Änderungserkennung nach Verarbeitung von Messung 60 und 100 der Messsequenz.

Unterschiede zwischen diesen vier Verteilungen zu sehen:

1. Im ersten Plot sind bereits 9 Messungen nach der Änderung verarbeitet worden. Hier sind die größten Unterschiede zwischen den Verfahren zu erkennen. Die neue mittlere Geschwindigkeit nach der Änderung liegt bei circa  $40 \frac{km}{h}$ , welcher vom *t-Test-Vergessen* am besten repräsentiert wird. Die Änderungserkennung über die *Eskalationsstufe*, die *Limitierung der Messwerte* aus [13] und das einfache *Gleichgewichten aller Messwerte*, berücksichtigen noch deutlich sichtbar Messungen, die vor der Änderung liegen. Im Hintergrund sind zwei Histogramme gezeigt, welche die relative Häufigkeit der gemessenen Geschwindigkeiten zeigen. Die Messungen vor der Änderung sind in grau und nach der Änderung in weiß dargestellt.

## 5 Erkennung von ungültigen Informationen im lernenden System

---

2. Im zweiten Plot wurden alle 100 Messungen der Messesequenz verarbeitet. Die geschätzten Verteilungen der Verfahren sind nach Verarbeitung der 100 Messungen sehr ähnlich. Nur die einfache Gleichgewichtung wird noch deutlich von den Messungen vor der Änderung beeinflusst. Auch in diesem Plot zeigen die beiden Histogramme im Hintergrund jeweils die Messungen vor und nach der Änderung als Referenz.

Dieser Vergleich basiert auf einer einzigen Messesequenz. Für einen belastbaren Vergleich muss die durchschnittliche Leistungsfähigkeit der Verfahren für jede Änderungsart (sprungartig und schleichend) verglichen werden. Dazu werden viele Messesequenzen benötigt.

### Vergleich der Änderungserkennungen mit tausend Messesequenzen

Um die durchschnittliche Leistungsfähigkeit der Verfahren bei einer sprungartigen Änderung zu vergleichen, werden 1000 Messesequenzen verwendet. Als Grundlage für die Messesequenzen (Schema: Messung 1-50 vor Änderung, Messung 51-100 nach Änderung) werden künstliche und simulierte Messwerte verwendet.

- Messesequenzen, die aus künstlichen Messwerten erzeugt werden, können direkt über die Gleichungen 5.13, 5.14 und 5.15 berechnet werden. Sie unterscheiden sich aufgrund ihrer zufällig erzeugten Rauschkomponente voneinander.
- Werden Messesequenzen aus simulierten Daten erzeugt, werden die Messungen aus einer Datensammlung von 251 Fahrten durch die *Alt-Lietzow*-Kurve in Berlin (siehe Abbildung 5.12) entnommen. Dabei wurden die Fahrten vorher per Hand in Messungen *vor* und *nach der Änderung* klassifiziert. Für eine Messesequenz aus simulierten Daten werden die Messungen 1 bis 50 aus der Klasse *vor Änderung* und die Messungen 51 bis 100 aus der Klasse *nach Änderung* zufällig gewählt. Dabei wird überprüft, dass jede erzeugte Messesequenz einzigartig ist.

Die so generierten Datensätze, bestehend aus 1000 einzigartigen Messesequenzen, werden mit den folgenden vier Verfahren verarbeitet:

- Das erste Verfahren besitzt keine Erkennung von Änderungen und gewichtet alle Messungen gleich.

## 5.2 Evaluation der Änderungserkennung t-Test-Vergessen

---

- Das zweite Verfahren limitiert die verwendeten Messungen auf die letzten Messungen, wie in [13] vorgeschlagen. In den folgenden Vergleichen werden dabei die letzten 15 Messungen berücksichtigt.
- Das dritte Verfahren ist die Änderungserkennung über die Eskalationsstufe aus [47].
- Das vierte Verfahren ist das in der vorliegenden Arbeit entwickelte *t-Test-Vergessen*.

Für den Vergleich wurden die 100 Messungen einer Messsequenz nacheinander vom lernenden System verarbeitet. Bei dieser Verarbeitung wird nach dem Hinzufügen der nächsten Messung die Änderungserkennung ausgeführt, die gegebenenfalls die als ungültig eingestuft Messungen entfernt. Für die nicht entfernten Messungen wird die Wahrscheinlichkeitsverteilung mit der Kerndichteschätzung berechnet. Aus dieser Verteilung kann die wahrscheinlichste Geschwindigkeit ausgelesen werden. So entsteht nach der Verarbeitung der Messsequenz aus 100 Messungen eine neue Sequenz bestehend aus der wahrscheinlichsten Geschwindigkeit nach der Verarbeitung jeder einzelnen Messung. Diese wahrscheinlichste Geschwindigkeit wird für alle 1000 Messsequenzen berechnet und gemittelt. Die über alle Messsequenzen gemittelte wahrscheinlichste Geschwindigkeit spiegelt die durchschnittliche Leistungsfähigkeit eines Verfahrens wider, eine Änderung zu erkennen. Dieser Prozess wird für alle vier Verfahren getrennt durchgeführt, um einen belastbaren Vergleich zu ermöglichen.

Im Folgenden werden drei nach diesem Prinzip durchgeführte Vergleiche vorgestellt:

1. In Abbildung 5.18 wurden 1000 Messsequenzen mit Funktionen generiert die vor der Änderung einen Mittelwert von  $60 \frac{km}{h}$  und eine Standardabweichung von  $10 \frac{km}{h}$  haben. Die Änderung wurde durch einen um  $-20 \frac{km}{h}$  verschobenen Mittelwert simuliert.

Die beste Erwartung wäre der optimale Verlauf (blaue Linie). Dieser kann nur erreicht werden, wenn der Änderungspunkt bekannt ist. Je näher ein Verlauf eines Verfahrens an den optimalen Verlauf herankommt, desto besser ist das Verfahren. Deutlich zu sehen, ist die schlechte Leistungsfähigkeit der einfachen Gleichgewichtung aller Messwerte, die auf Änderungen kaum reagiert. Schlechter als dieser Verlauf sollte ein Verfahren nicht sein, da es sonst keinen Mehrwert bieten würde. Die Limitierung der Messwerte zeigt bereits einen deutlich besseren Verlauf als die Gleichgewichtung. Das

## 5 Erkennung von ungültigen Informationen im lernenden System

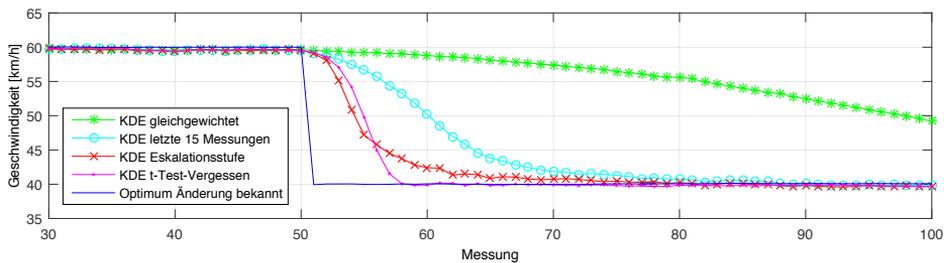


Abb. 5.18: Vergleich der verschiedenen Verfahren zur Änderungserkennung mit künstlichen erzeugten Messsequenzen.

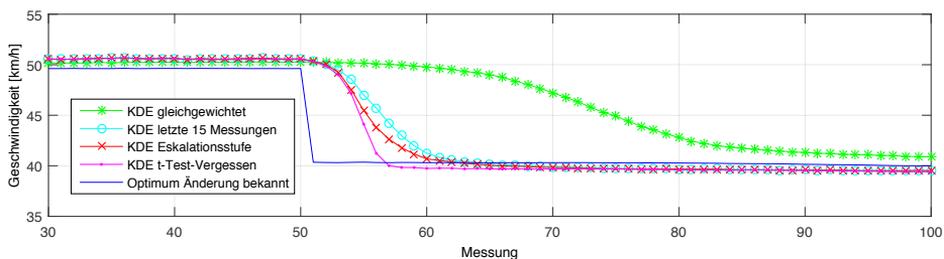


Abb. 5.19: Vergleich der verschiedenen Verfahren zur Änderungserkennung mit simulierten Messsequenzen.

Verfahren mit Eskalationsstufe reagiert in diesem Vergleich schneller als das *t-Test-Vergessen*, erreicht jedoch den neuen Wert nach der Änderung deutlich später.

- Die Abbildung 5.19 zeigt einen ähnlichen Vergleich wie die vorherige Abbildung, für diesen Vergleich wurden jedoch simulierte Messsequenzen verwendet. Der Mittelwert vor der Änderung ist circa  $50 \frac{km}{h}$  und sinkt nach der Änderung um circa  $10 \frac{km}{h}$  auf circa  $40 \frac{km}{h}$ .

In diesem Vergleich reagiert das *t-Test-Vergessen* schneller als die Eskalationsstufe auf die Änderung und nimmt erneut früher den neuen Wert nach der Änderung an. Das Verhalten der anderen Verfahren entspricht den Ergebnissen aus dem vorherigen Vergleich.

- In Abbildung 5.20 wurden ebenfalls simulierte Messsequenzen verwendet. Zusätzlich wurde die optimale Bandbreite (nach [81, 47]) der Kerndichteschätzung verdoppelt, um die Auswirkungen der Verteilungsschätzung auf die Änderungserkennung über die Eskalati-

## 5.2 Evaluation der Änderungserkennung t-Test-Vergessen

onsstufe von [47] zu zeigen.

Durch die veränderte Bandbreite im Lernprozess wird die Änderungserkennung von [47] negativ beeinflusst und verhält sich nahezu wie die Limitierung der Messwerte (vorgeschlagen in [13]). Das *t-Test-Vergessen* hingegen ist unabhängig vom Lernprozess und damit auch unabhängig von möglicherweise schlecht gewählten Konfigurationen der Kerndichteschätzung.

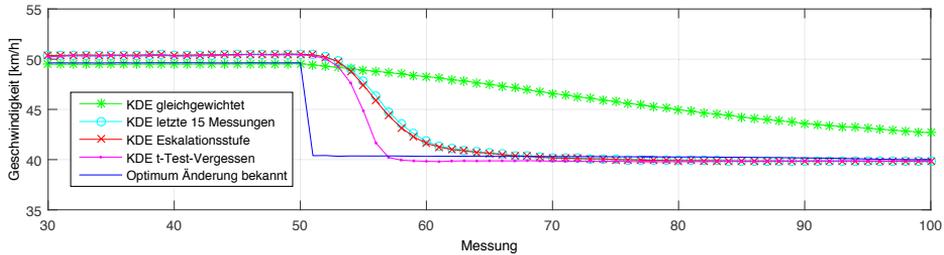


Abb. 5.20: Vergleich der verschiedenen Verfahren zur Änderungserkennung mit simulierten Messsequenzen bei verdoppelter, optimaler Bandbreite der Kerndichteschätzung.

In allen Vergleichen konnte sich das *t-Test-Vergessen* gegenüber den anderen Verfahren bei der Erkennung von sprunghaften Änderungen durchsetzen. Es konnte schneller auf Änderungen reagieren, schneller den neuen Wert nach der Änderung annehmen und zeigte eine konstante Leistungsfähigkeit. Auch bei absichtlich unvorteilhaft gewählter Konfiguration für die Kerndichteschätzung im Lernprozess bleibt die Leistungsfähigkeit des *t-Test-Vergessen* erhalten.

### 5.2.3 t-Test-Vergessen bei schleichenden Änderungen:

Wie in Abschnitt 5.1.6 beschrieben, wurde dem *t-Test-Vergessen* ein zweiter, parallel arbeitender Hypothesentest für die Erkennung von schleichenden Änderungen hinzugefügt. Dieser hat dafür eine andere Fensterkonfiguration zur Erzeugung der Stichproben. Für diesen Hypothesentest wird das Gedächtnisfenster nicht mit den Analysefenster über die neusten Messungen gelegt. Stattdessen bleibt das Gedächtnisfenster stationär über den ältesten Messungen bis eine Änderung erkannt wird und wird nur dann verschoben.

## 5 Erkennung von ungültigen Informationen im lernenden System

---

In diesem Abschnitt soll die Leistungsfähigkeit des *t-Test-Vergessen* mit zwei Hypothesentests zur Erkennung von schleichenden Änderungen verglichen werden. Als Vergleich wird die einfache Version des *t-Test-Vergessen* mit nur einem Hypothesentest verwendet, um zu zeigen, welche Verbesserungen mit der zweiten Instanz des Hypothesentests erzielt werden können. Im weiteren Verlauf dieses Abschnitts wird die Änderungserkennung mit zwei Hypothesentests *t-Test-Vergessen* genannt und die Änderungserkennung mit einem Hypothesentest *einfaches t-Test-Vergessen*.

Die Verfahren in der Literatur wurden nicht explizit für schleichende Änderungen entwickelt. Zudem zeigte das *t-Test-Vergessen* die beste Leistungsfähigkeit bei den bisherigen Vergleichen. Aus diesen Gründen werden die Verfahren aus der Literatur für die folgenden Vergleiche mit schleichenden Änderungen nicht mehr berücksichtigt. Stattdessen wird nur das *t-Test-Vergessen* mit dem *einfachen t-Test-Vergessen* verglichen.

Für die folgenden beiden Vergleiche wurden 1000 einzigartige Messsequenzen mit einer schleichenden Änderung über die Gleichung 5.15 erzeugt. In den folgenden Abbildungen sind die Mittelwerte der wahrscheinlichsten Geschwindigkeit über alle 1000 Sequenzen zu sehen.

1. Die erste schleichende Änderung verringert den Mittelwert um  $20 \frac{km}{h}$  von  $60 \frac{km}{h}$  auf  $40 \frac{km}{h}$ . Dabei wirkt sich die Änderung über 40 Messwerte gleichmäßig aus (von Messung 40 bis 80).

Die Abbildung 5.21 zeigt die durchschnittliche Leistungsfähigkeit beider Versionen des *t-Test-Vergessen* bei dieser schleichenden Änderung. Das *t-Test-Vergessen* kann dem optimalen Verlauf besser folgen als das *einfache t-Test-Vergessen*. Dabei wird die Differenz zum optimalen Verlauf im Mittel um 39.13% reduziert. Bei Messung 74 kann die Differenz sogar kurzzeitig um 78.60% verringert werden und ist damit deutlich näher am optimalen Verlauf. Beide Versionen folgen dem Plateau ab Messung 80 bereits nach 3 Messungen.

2. Die zweite schleichende Änderung verringert den Mittelwert ebenfalls um  $20 \frac{km}{h}$  (von  $60 \frac{km}{h}$  auf  $40 \frac{km}{h}$ ), wobei in diesem Vergleich der Gradient der schleichenden Änderung flacher ist und sich über 60 Messwerte gleichmäßig auswirkt.

Abbildung 5.22 zeigt den Vergleich der beiden *t-Test-Vergessen* Versionen. Nach einer Verzögerung von circa 7 Messungen nach dem Start der schleichenden Änderung reagiert der zweite Hypothesentest des *t-Test-Vergessen* und erkennt die Änderung. Erneut kann

## 5.2 Evaluation der Änderungserkennung *t*-Test-Vergessen

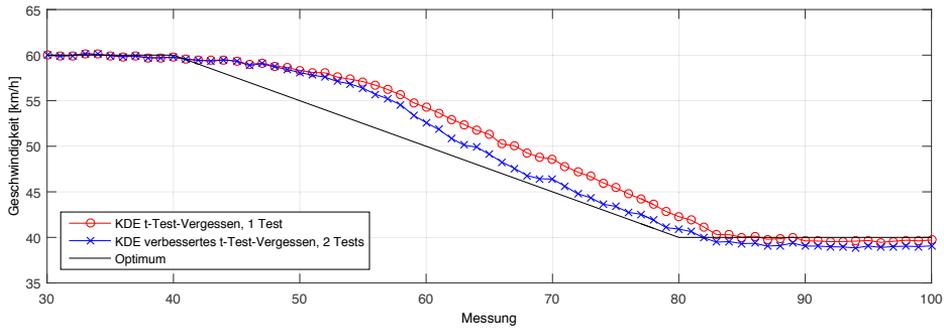


Abb. 5.21: Vergleich des *t*-Test-Vergessen mit einem und zwei Hypothesentests bei der Erkennung einer schleichenden Änderung ( $-20 \frac{km}{h}$  von Messung 40 bis 80).

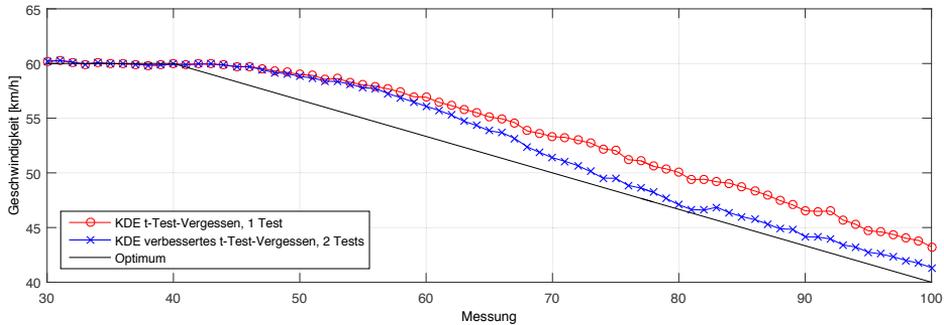


Abb. 5.22: Vergleich des *t*-Test-Vergessen mit einem und zwei Hypothesentests bei der Erkennung einer schleichenden Änderung ( $-20 \frac{km}{h}$  von Messung 40 bis 100).

das *t*-Test-Vergessen dem optimalen Verlauf besser folgen als das einfache *t*-Test-Vergessen, reduziert im Mittel die Differenz sogar um 43.72% und bei Messung 81 sogar um 89.51%.

Die Vergleiche zeigen die Fähigkeit des *t*-Test-Vergessens, mit zwei Hypothesentests eine schleichende Änderung deutlich besser zu erkennen als zuvor. Er ist in der Lage, abhängig von der Art der schleichenden Änderung, die Differenz zum optimalen Verlauf um bis zu  $\approx 90\%$  und im Mittel um mindestens  $\approx 40\%$  zu verbessern.

Mit dem in der vorliegenden Arbeit vorgestellten *t*-Test-Vergessen können durch zwei getrennte Hypothesentests sowohl sprungartige als auch

schleichende Änderungen schneller erkannt werden als mit bisher vorgestellten Verfahren. Das *t-Test-Vergessen* kann dabei dem optimalen Verlauf (bekannter Änderungspunkt) deutlich besser folgen als die anderen verglichenen Verfahren. Dabei ist die Leistungsfähigkeit des *t-Test-Vergessen* unabhängig vom Lernprozess.

### 5.3 Zusammenfassung

In diesem Kapitel wurden ungültig gewordene Informationen im lernenden System betrachtet.

Sind durch Änderungen gelernte Informationen ungültig geworden, müssen diese ausgeschlossen oder aus dem lernenden System gelöscht werden. Dafür ist eine Änderungserkennung notwendig. Es wurde das *t-Test-Vergessen* entwickelt, das in der Lage ist, verschiedene Arten von Änderungen (*sprungartig* und *schleichend*) mit Hilfe eines Hypothesentests, dem Zweistichproben-t-Test, zu erkennen.

Mit einer ausführlichen Evaluation konnte die höhere Leistungsfähigkeit des *t-Test-Vergessens* im Vergleich zu bisherigen Verfahren nachgewiesen werden. Das *t-Test-Vergessen* konnte Änderungen schneller als bisherige Verfahren erkennen und ist dabei vom Lernprozess unabhängig.

Durch diese verbesserte Änderungserkennung können ungültige Informationen im lernenden System schneller und zuverlässiger entfernt werden. Damit erhalten die FAS seltener ungültige Informationen als zuvor und es kommt seltener zu einer Einschränkung oder sogar einem vollständigen Ausfall des FAS durch ungültige Informationen.

In Abbildung 5.23 wurde das *t-Test-Vergessen* in das lernende System der vorliegenden Arbeit eingeordnet.

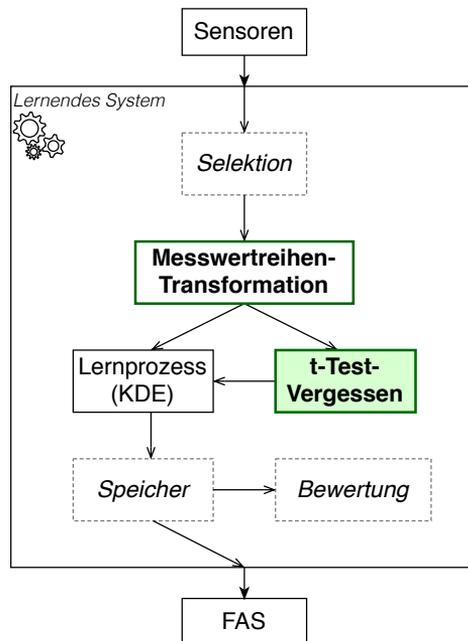


Abb. 5.23: Komponenten des lernenden Systems der vorliegenden Arbeit.  
Die Beiträge des 5. Kapitels sind grün hervorgehoben.



# Anwendung des lernenden Systems

---

In folgenden Kapitel wird der durch die *Messwertreihen-Transformation* und dem *t-Test-Vergessen* erweiterte Lernprozess in ein Gesamtsystem integriert, welches das lernende System bildet. Das lernende System wird im Versuchsträger angewendet und damit ein prototypisches, lernendes FAS realisiert, der adaptive Kurvenwarner.

Für das Gesamtsystem werden weitere Komponenten benötigt, die in Abschnitt 6.1 entwickelt werden. Ein persistenter Datenspeicher für die gelernten Informationen, eine Selektion relevanter Sensordaten und eine Priorisierung gelernter Informationen werden vorgestellt und evaluiert.

Anschließend wird mit dem lernenden System in Abschnitt 6.2 der adaptive Kurvenwarner im Versuchsträger realisiert und getestet.

## 6.1 Benötigte Komponenten für das Gesamtsystem

Das lernende System besteht nicht nur aus dem Lernprozess, sondern benötigt noch weitere Komponenten damit ein funktionsfähiges lernendes System für Fahrzeuge entsteht.

Um die gelernten Informationen für jede zukünftige Fahrt nutzen zu können, ist ein persistenter Datenspeicher notwendig. Dieser muss gelernte Informationen schnell bereitstellen, um die Funktion der FAS nicht einzuschränken. Dafür soll eine passende Datenbank und ein passendes Adressierungsverfahren gewählt werden.

Zum Bereitstellen der Messwertreihen für den Lernprozess ist es erforderlich, die relevanten Sensordaten zu selektieren und aufzuzeichnen. Es wird deswegen eine Selektion relevanter Daten am Beispiel des adaptiven Kurvenwarners vorgestellt.

Damit das lernende System einwandfrei funktionieren kann und neue Informationen gespeichert werden können, muss immer freier Speicher verfügbar sein. Um dies zu gewährleisten, soll eine Bewertung von gelernten Informationen zur gezielten Löschung der unwichtigsten Daten entwickelt werden.

### 6.1.1 Persistente Speicherung gelernter Informationen

In diesem Abschnitt wird die Datenadressierung und der Datenspeicher für das lernende System im Fahrzeug betrachtet.

Eine verzögerte Auslieferung der gelernten Informationen von wenigen Millisekunden an die FAS kann deren Funktion einschränken oder verhindern. Deswegen muss der Datenspeicher schnelle Zugriffszeiten vorweisen und die Informationen schnell genug zur Verfügung stellen. So müssen die Informationen über eine bekannte Kurve vor dem Warnzeitpunkt vorliegen. Zusätzlich muss dies mit den limitierten Ressourcen im Fahrzeug möglich sein.

In den vorgestellten Arbeiten zu lernenden Systemen im Fahrzeug aus Abschnitt 3.4 werden diese Anforderungen und Einschränkungen nicht ausreichend betrachtet. Die dort verwendeten Datenspeicher wurden nicht eingehend für die Verwendung im Fahrzeug überprüft. Es werden keine Empfehlungen für das zu verwendende Adressierungsverfahren von Daten ausgesprochen. Auch werden keine Empfehlungen für ein geeignetes Datenbankmodell (DBM) oder ein Datenbanksystem (DBS) gegeben.

Aus diesem Grund wurde für das lernende System der vorliegenden Arbeit eine Betrachtung von Adressierungsverfahren und bekannten DBM bezüglich der Eignung durchgeführt. In diesem Abschnitt werden die Ergebnisse dieser Betrachtung vorgestellt.

#### Adressierungsverfahren

Für die Verwaltung von positionsabhängigen Informationen in einer Datenbank wird häufig eine *direkte Adressierung über die Position* bestehend aus Längen- und Breitengrad verwendet, wie auch in [13] vorgeschlagen. Liegen die zu speichernden Informationen jedoch ausschließlich auf einem Straßennetz, wie in der vorliegenden Arbeit, kann auch eine *Adressierung über eine digitale Karte* durchgeführt werden. Dabei werden die Informationen mit dem Straßennetz der digitalen Karte verknüpft. Diese Adressierung nutzt die einzigartigen Identifikationsnummern von Kanten und Knoten des Kartengraphs als Schlüssel für die Informationen (siehe Abschnitt 2.1.5), die dort gespeichert werden sollen. Im Folgenden werden kurz die für die vorliegende Arbeit wichtigen Unterschiede dieser beiden Verfahren erläutert.

## 6.1 Benötigte Komponenten für das Gesamtsystem

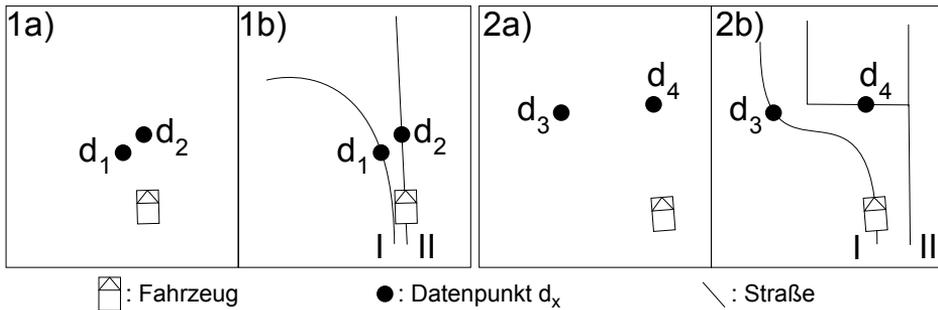


Abb. 6.1: Die Relevanz der Datenpunkte  $d_1$  bis  $d_4$  kann durch das Straßennetz eindeutig bestimmt werden.

**Relevanz:** Bei der Adressierung über eine digitale Karte werden die gelernten Informationen direkt an Straßen und Kreuzungen gespeichert. Da die Fahrzeugposition in der Karte immer bekannt ist, kann eindeutig bestimmt werden, welche gelernten Informationen für das Fahrzeug relevant sind, auch in einem dichten Straßennetz. Wird hingegen eine gelernte Information nur an eine Position gebunden, ist dies nicht automatisch gegeben. So existieren Situationen, in denen nur mit der Position nicht eindeutig bestimmt werden kann, ob eine Information relevant ist.

In Abbildung 6.1 ist dieser Unterschied dargestellt. Die beiden Beispiele 1a) und 2a) zeigen die Fahrzeugposition und jeweils zwei Positionen von gelernten Informationen  $d_1$  bis  $d_4$ . Sind die gelernten Informationen wie in 1a) und 2a) nur über die Position adressiert, ist nicht eindeutig, welche der Informationen für das Fahrzeug relevant sind. Wird hingegen die Adressierung über eine digitale Karte wie in 1b) und 2b) durchgeführt, kann eindeutig bestimmt werden, welche der gelernten Informationen für das Fahrzeug relevant sind: In Beispiel 1b) befindet sich das Fahrzeug auf Straße II und nur  $d_2$  ist relevant. In Beispiel 2b) ist  $d_3$  relevant, da das Fahrzeug auf Straße I fährt.

**Aufwand:** Beim Laden und Speichern von positionsabhängigen Informationen müssen bei beiden Verfahren die gleichen Schritte durchgeführt werden. Wie in Abbildung 6.2 dargestellt, sind die Schritte *Schreiben* in die Datenbank, *Suche* in der Datenbank und das *Mapmatching* notwendig, bevor die Daten den FAS bereitgestellt werden können.

Der Unterschied der beiden Verfahren in Bezug auf den Berechnungsaufwand ist der Zeitpunkt, bei dem die Schritte durchgeführt werden. Bei

## 6 Anwendung des lernenden Systems

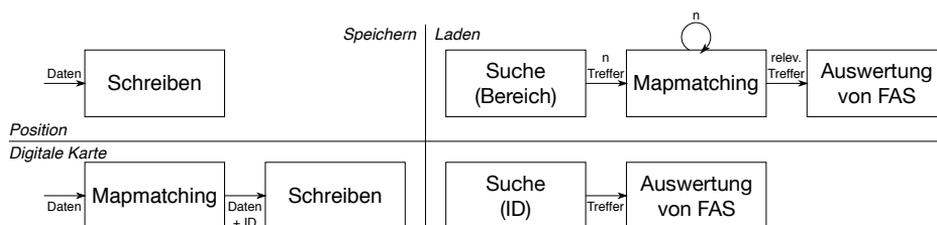


Abb. 6.2: Reihenfolge der Schritte beim Speichern und Laden von Informationen der beiden Adressierungsverfahren.

der Adressierung über die Position muss beim Laden erst eine Umkreissuche durchgeführt werden und alle gefundenen Informationen einzeln über das Mapmatching in den Kartengraph eingeordnet werden. Dies kann das Laden unter Umständen sehr stark verzögern, wenn im Suchbereich viele Treffer vorhanden sind. Diese müssen alle einzeln über das Mapmatching in das Kartenmaterial eingeordnet werden. Soll für das Mapmatching noch die Fahrzeugbewegung berücksichtigt werden, müsste diese zusätzlich zu den Informationen gespeichert werden.

Bei der Adressierung über eine digitale Karte wird das Mapmatching gleich beim Speichern der Daten ausgeführt. Dabei ist die aktuelle Fahrzeugbewegung für das Mapmatching vorhanden und muss nicht zusätzlich gespeichert werden. Da der Mapmatching-Schritt damit beim Laden entfällt, können die Daten schneller den FAS bereitgestellt werden als mit der Adressierung über die Position.

**Vorladen:** Wird vom Fahrer eine Route vorgegeben, können alle gelernten Informationen entlang dieser Route vorgeladen werden. Dadurch muss der Ladevorgang des Datenspeichers nicht unmittelbar vor Erreichen einer Situation durchgeführt werden und die Zugriffszeiten auf diese Informationen werden reduziert. Auch das Abweichen des Fahrers von dieser Route durch unerwartete Abbiegevorgänge kann berücksichtigt werden. Es kann ein *dynamischer elektronischer Horizont* berechnet werden, der alle Straßen enthält, auf die abgelenkt werden kann. Dabei kann eine Tiefe dieses Horizonts angegeben werden, die angibt wie viele Kanten des Kartengraphs von der aktuellen Position vorgeladen werden sollen.

In Abbildung 6.3 ist beispielhaft der dynamische elektronische Horizont für die Tiefe 1 und 2 über dem Straßennetz eingezeichnet. Rund und schwarz dargestellt sind die Endknoten der Kanten bis zur Tiefe 1. Qua-

## 6.1 Benötigte Komponenten für das Gesamtsystem

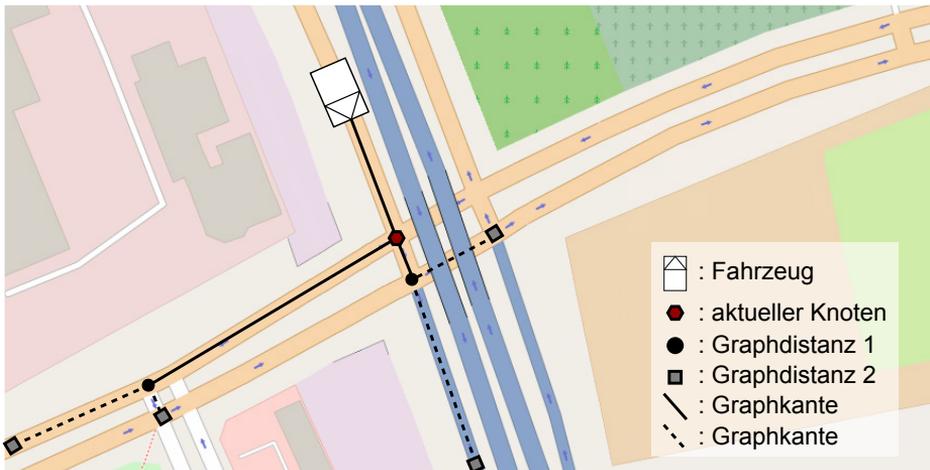


Abb. 6.3: Beispiel für die vorgeladenen Kanten des dynamischen elektronischen Horizonts in zwei verschiedenen Knoten-Tiefen.

dratisch und grau dargestellt sind die Endknoten der Kanten bis zur Tiefe 2.

Eine allgemeine Aussage über die empfohlene Tiefe kann nicht getroffen werden, da diese Entscheidung von mehreren Faktoren abhängt. Befindet sich das Fahrzeug im Bereich von Autobahnen und Landstraßen, kann eine Tiefe von 1 bis 2 bereits viele hundert Meter bis Kilometer weit reichen, im Stadtbereich mit dichtem Straßennetz hingegen möglicherweise keine hundert Meter weit. Wegen der limitierten Ressourcen im Fahrzeug könnte eine dynamische Tiefe Sinn machen: Sie könnte solange iterativ erhöht werden, bis der für das Vorladen bereitgestellte Zwischenspeicher vollständig genutzt ist.

Die *Adressierung über eine digitale Karte* hat für die vorliegende Arbeit entscheidende Vorteile:

- Die Relevanz von gelernten Informationen für das Fahrzeug ist durch die Zuordnung in das Straßennetz eindeutig.
- Der Aufwand des Mapmatching entsteht beim Speichern der Informationen, wodurch das Laden schneller ausgeführt wird. Da die gelernten Informationen möglichst schnell zur Verfügung gestellt werden sollen, wird ein schnelles Laden bevorzugt.

- Mit Hilfe des Straßennetzes können zusätzlich die Informationen von einer bekannten Route und möglichen Abbiegevorgängen vorgeladen werden. Dieses Vorladen von gelernten Informationen reduziert weiter die Zugriffszeiten.

Aus den genannten Gründen wird in der vorliegenden Arbeit die *Adressierung über eine digitale Karte* verwendet.

### Datenbankmodell und Datenbanksystem

Das lernende System muss in der Lage sein gelernte Informationen persistent in einem Datenspeicher abzulegen. Dafür sollen die gelernten Informationen mit einem eigenen Datenformat in einem DBS gespeichert werden. Für eine einfache Austauschbarkeit wird das DBS modular in das lernende System integriert. So kann beispielsweise bei einer Entwicklung des Steuergeräts ein alternativer Datenspeicher verwendet werden.

Es existieren verschiedene DBM, die unterschiedliche Eigenschaften haben. Auch deren konkrete Implementierung im Form eines DBS können Unterschiede aufweisen. Für eine Wahl eines passenden Datenspeichers für die vorliegende Arbeit werden die folgenden Anforderungen an diesen formuliert:

1. **Passgenauigkeit:** Das DBM soll von der angebotenen Funktionalität her zur Adressierung über eine digitale Karte passen.
2. **Zugriffszeit:** Ein wichtiges Kriterium für den Datenspeicher sind die Zugriffszeiten. Diese sollten beim DBM und DBS so niedrig wie möglich sein, um die Funktion von FAS nicht einzuschränken.

Diese Anforderungen werden von dem Key-Value-DBM und konkret vom DBS *Oracle Berkeley DB Java Edition* erfüllt. Die *Oracle Berkeley DB Java Edition* ist frei erhältlich. Das Key-Value-DBM passt zur Adressierung über eine digitale Karte, da Informationen über Schlüssel adressiert werden. Die einzigartige Identifikationsnummer einer Straße im Kartenmaterial kann als Schlüssel verwendet werden und die Informationen entsprechen den gelernten Sensordaten auf dieser Straße. Durch die Organisation und Verwaltung der Daten anhand des Kartenmaterials in der Anwendungsschicht werden keine umfangreichen und komplexen Funktionen vom DBS benötigt; das Key-Value-DBM deckt die benötigte Funktionalität vollständig ab.

## 6.1 Benötigte Komponenten für das Gesamtsystem

Die niedrigen Zugriffszeiten gelten als eine der bedeutendsten Eigenschaften des Key-Value-DBM. Zur Evaluation der Zugriffszeiten wird ein Geschwindigkeitstest mit Daten aus dem lernenden System durchgeführt. In einem Schreib- und Lesetest werden die Geschwindigkeiten von der *Berkeley DB Java Edition* (Version 5.0.103) mit dem weit verbreiteten, objektrelationalen DBS *PostgreSQL* (Version 9.0.4) verglichen.

**Konfiguration der Geschwindigkeitstests:** Für den Test wurden zum einen identische *sequenzielle* Schreib- und Lesezugriffe mit den beiden DBS durchgeführt. Zum anderen wurden identische *zufällige* Schreib- und Lesezugriffe durchgeführt. Die Schreib- und Lesezugriffe wurden getrennt durchgeführt. Um Caching zu vermeiden, wurden mehrere Durchläufe mit verschiedenen Datenmengen ausgeführt. In Tabelle 6.1 sind die durchgeführten Tests zur Übersicht aufgelistet.

Testnummer	Zugriffsart	Objekte	Reihenfolge
1	Schreiben	10000	sequenziell
2	Schreiben	100000	sequenziell
3	Schreiben	200000	sequenziell
4	Schreiben	100000	zufällig
5	Schreiben	200000	zufällig
6	Lesen	10000	sequenziell
7	Lesen	100000	sequenziell
8	Lesen	200000	sequenziell
9	Lesen	100000	zufällig
10	Lesen	200000	zufällig

Tabelle 6.1: Konfiguration der durchgeführten Geschwindigkeitstests mit dem Key-Value-DBS.

Die Testobjekte waren zufällige Byteströme mit einer Größe von 18.2 kByte, was der durchschnittlichen Größe eines gelernten Datensatzes einer Kurve mit den grundlegenden Sensordaten entspricht. Diese grundlegenden Daten enthalten die Fahrzeuggeschwindigkeit, Fahrzeugdynamik, Positionsdaten, Fahrereingaben und weitere Metadaten (wie beispielsweise die Identifikationsnummer der Straße aus der digitalen Karte).

Jeder Datensatz wird über einen Schlüssel gespeichert, der als Adresse innerhalb des DBS fungiert. Für die sequenziellen Tests wurde der verwendete Schlüsselwert nach jeder Operation um 1 inkrementiert. Für die zu-

## 6 Anwendung des lernenden Systems

fälligen Tests wurde eine zufällige Reihenfolge der verwendeten Schlüssel im Vorfeld berechnet. Damit beim zufälligen Schreiben keine Schlüssel-Kollisionen in dem DBS entstehen, wurde dabei kein Schlüssel doppelt verwendet.

Die Geschwindigkeit wurde in  $\frac{\text{Objekte}}{\text{Sekunde}}$  ermittelt, indem die Zeit zwischen dem Start des Speicher- beziehungsweise Ladevorgangs und dessen Fertigstellung gemessen wurde. Anschließend wurde mit dieser Dauer und der Objektanzahl des Tests die Rate der Objekte pro Sekunde berechnet.

**Ergebnisse der Geschwindigkeitstests:** Die Ergebnisse des Tests sind in den Abbildungen 6.4 für das Speichern und 6.5 für das Laden dargestellt.

Beim Test der Speichergeschwindigkeit (Abbildung 6.4) ist kaum ein Unterschied zwischen dem sequenziellen und zufälligen Speichern der beiden Datenbanken (DB) vorhanden.

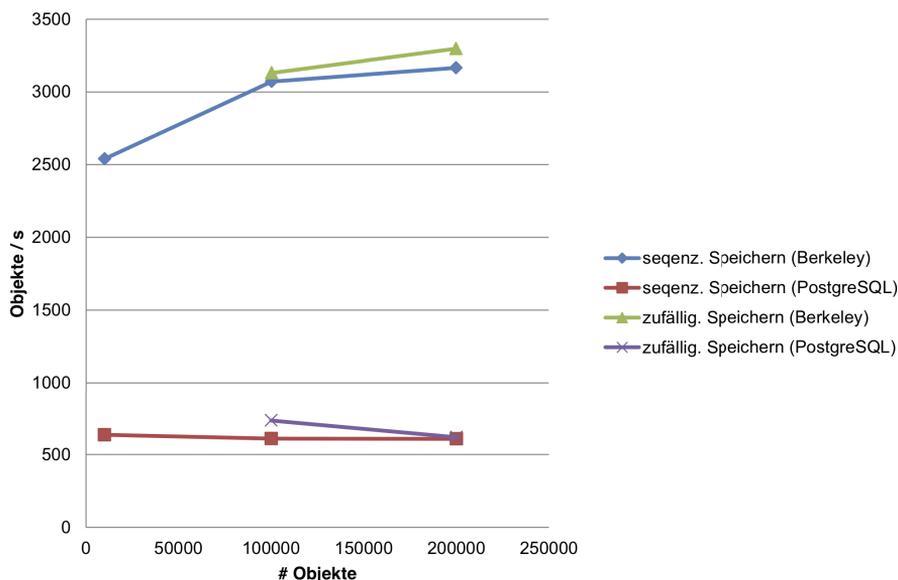


Abb. 6.4: Geschwindigkeitsvergleich der Datenbanken Oracle Berkeley DB Java Edition 5.0.103 und PostgreSQL 9.0.4 beim Speichern mit zufälligen und sequenziellen Schlüssel.

## 6.1 Benötigte Komponenten für das Gesamtsystem

---

Dies war zu erwarten, da sämtliche Schreibvorgänge bei beiden Varianten ohne Kollision durchgeführt wurden und der Wert des Schlüssels beim Ablegen der Daten in die jeweilige DB keinen Unterschied macht. Die Werte für das sequenzielle Speichern (insbesondere bei der Berkeley DB) mit 10000 Objekten wird als Ausreißer behandelt. Grund dafür ist die kurze Ausführungsdauer des Tests, bei der das Caching des DBS und andere Effekte des Betriebssystems *Microsoft Windows 7* einen vergleichsweise großen Einfluss haben. Für weitere Vergleiche werden deswegen nur die Ergebnisse der Tests mit 100000 und 200000 Objekten verwendet.

Die Geschwindigkeit der Speicherzugriffe unterscheidet sich in diesem Test deutlich zwischen der PostgreSQL und Berkeley DB. Aufgrund der kleinen Unterschiede zwischen den Tests mit 100000 und 200000 Objekten werden die Ergebnisse für weitere Vergleiche gemittelt. Damit ist die *Berkeley DB Java Edition* im Mittel mit  $\approx 3166 \frac{\text{Objekte}}{s}$  circa fünf mal so schnell wie die *PostgreSQL* mit  $\approx 609 \frac{\text{Objekte}}{s}$ .

Im Gegensatz zum Speichern, spielt beim Laden die Anzahl der gespeicherten Objekte in der DB eine Rolle. Auch ist die Art und Weise, wie die Daten aus der DB geladen werden (sequenziell oder zufällig) von Bedeutung. Die Anzahl der Objekte in der DB beeinflusst maßgeblich die Suchdauer eines speziellen Schlüssels. Zu erwarten ist, dass sich beim Laden von Daten mit zufälligem Schlüssel die Dauer der Suche proportional zur Anzahl der Objekte in der DB verhält.

In Abbildung 6.5 sind die Ergebnisse des Tests der Ladegeschwindigkeit gezeigt. Das erwartete Verhalten der Ladegeschwindigkeiten wird durch die Messergebnisse bestätigt: Je mehr Objekte sich in der DB befanden, desto länger dauerte die Suche nach zufälligen Schlüsseln. Sequenzielles Abfragen der Schlüssel hingegen zeigte keine sichtbare Veränderung der Geschwindigkeit bei steigender Anzahl an Objekten in der DB. Die Berkeley DB ist gegenüber der PostgreSQL DB in jedem Aspekt des Tests schneller, jedoch nicht so deutlich wie beim Speichern. Wird das sequenzielle und zufällige Laden zusammengefasst, ist die *Berkeley DB Java Edition* im Mittel mit  $\approx 3431 \frac{\text{Objekte}}{s}$  circa anderthalb mal so schnell wie die *PostgreSQL* mit  $\approx 2308 \frac{\text{Objekte}}{s}$ .

Wichtig zu erwähnen ist, dass dieser Geschwindigkeitstest der DBS nicht generalisiert werden darf. In diesem Test wurde lediglich das für die vorliegende Arbeit relevante Szenario in Form von Adressierung, Datenstruktur, -inhalt und -menge getestet. Jedes DBM hat Stärken und Schwächen, und in diesem speziellen Fall ist das Key-Value DBS in Bezug auf die Zugriffsgeschwindigkeiten dem objektrelationalen DBS überlegen.

## 6 Anwendung des lernenden Systems

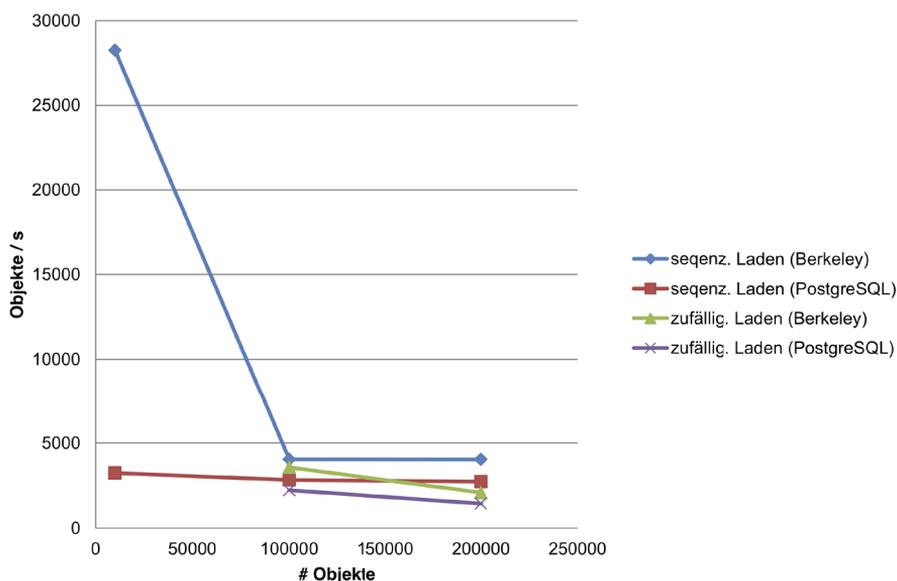


Abb. 6.5: Geschwindigkeitsvergleich der Datenbanken Oracle Berkeley DB Java Edition 5.0.103 und PostgreSQL 9.0.4 beim Laden mit zufälligen und sequenziellen Schlüsseln.

Aus den genannten Gründen wird das Key-Value DBS *Oracle Berkeley DB Java Edition 5.0.103* in der vorliegenden Arbeit verwendet in Verbindung mit der Adressierung über eine digitale Karte. Im folgenden Text beziehen sich die *Datenbank* relevanten Begriffe und Erwähnungen ausschließlich auf dieses DBS.

### 6.1.2 Selektion relevanter Informationen

In dem für diese Arbeit verwendeten Versuchsträger können alleine über die vorhandenen sieben CAN-Busse insgesamt  $437.5 \frac{kByte}{s}$  Daten übertragen werden. Durch eine einstündige Fahrt würde damit ein Datenvolumen von  $3.15 GByte$  entstehen. Bei der üblichen Lebensdauer eines Fahrzeugs von 12 Jahren [39] ist es derzeit nicht möglich, alle erzeugten Sensordaten im Fahrzeug lokal zu speichern. Es ist deswegen unumgänglich den Umfang der Sensordaten zu reduzieren, bevor sie in das lernende System überführt werden. Durch eine Selektion der für die FAS relevanten Sensordaten, ist eine deutliche Reduzierung der Datenmenge möglich.

## 6.1 Benötigte Komponenten für das Gesamtsystem

---

Es können zwei Arten der Selektion durchgeführt werden, um die Relevanz von Sensordaten zu bestimmen:

1. **Selektion benötigter Sensordaten:** Welche Sensordaten werden von einem FAS benötigt?
2. **Selektion relevanter Situationen:** In welchen Situationen werden diese Sensordaten vom FAS benötigt?

**Selektion benötigter Sensordaten:** Gewisse Sensordaten werden von vielen FAS, wie auch vom adaptiven Kurvenwarner, verwendet und bilden die Grundlage für deren Funktion. Dazu gehören die Sensormesswerte der Dynamik und GPS Position des Fahrzeugs sowie die Fahrereingaben. Zu Fahrzeugdynamik zählen Sensormesswerte wie die Geschwindigkeit, Gierate und Beschleunigungen in Längs- und Querrichtung. Die Fahrereingaben beschränken sich auf den Lenkradwinkel und die Pedalstellung von Gas und Bremse. Diese Sensordaten werden in der vorliegenden Arbeit als *grundlegende Sensordaten* bezeichnet.

Die grundlegenden Sensordaten haben eine Datenrate von circa  $16 \frac{kBit}{s}$  und generieren pro Stunde Fahrt circa  $7.2 MByte$  Daten. Im Vergleich zum möglichen Datendurchsatz der CAN-Busse im Versuchsträger haben diese Sensordaten nur einen Anteil von 3.7%. Dieser Wert stellt einen ungefähren Wert einer unteren Grenze dar, zu der, je nach FAS, noch weitere Sensordaten hinzukommen können.

Alleine mit dieser einfachen Maßnahme der Selektion der benötigten Sensordaten konnten in diesem Fall 96.3% der möglichen Daten vor der Verarbeitung durch das lernende System herausgefiltert werden. Ein allgemeingültiger Wert für diese Einsparung lässt sich nicht angeben, da dieser von den benötigten Sensordaten der FAS abhängig ist.

**Selektion relevanter Situationen:** Die von den FAS benötigten Sensordaten sind jedoch nicht durchgehend von Bedeutung, sondern je nach Assistenzsystem nur in relevanten Situationen. Für den adaptiven Kurvenwarner sind Messgrößen beispielsweise nur in Kurven relevant. Auch für die Funktion der adaptiven Automatikschaltung sind die Messgrößen nur bei Straßenabschnitten mit Steigungen notwendig. Für diese Art der Selektion muss eine Erkennung für die entsprechende Situation existieren, damit nur die notwendigen Sensordaten aufgezeichnet werden.

**Relevante Situationen für den adaptiven Kurvenwarner:** Relevante Situationen für den adaptiven Kurvenwarner sind Fahrten durch Kurven. Es wird also für die Selektion der relevanten Situation in diesem Fall eine Kurvenerkennung benötigt. Im Folgenden wird diese Erkennung anhand von Sensoren im Detail erläutert.

Für die Kurvenerkennung müssen drei Bedingungen erfüllt sein:

1. **Mindestgeschwindigkeit:** Ziel dieser Bedingung ist das Herausfiltern von Abbiege-, Park- und Rangiervorgängen, welche hauptsächlich bei niedrigen Geschwindigkeiten durchgeführt werden. Bei Tests zeigte eine Mindestgeschwindigkeit  $v_{min}$  von  $32.5 \frac{km}{h}$  sehr gute Ergebnisse:

$$v_{min} = \begin{cases} wahr & , v \geq 32.5 \frac{km}{h} \\ falsch & , \text{sonst} \end{cases} \quad (6.1)$$

Um einen häufigen Wechsel zwischen *wahr* und *falsch* bei Geschwindigkeiten um die Mindestgeschwindigkeit herum zu vermeiden, wird eine Hysterese von  $2.5 \frac{km}{h}$  verwendet.

2. **Querbeschleunigung:** Die eigentliche Erkennung einer Kurve wird über die Querbeschleunigung des Fahrzeugs realisiert. Auf gerader Strecke ist die Querbeschleunigung nahe Null und nimmt in Kurven größere Werte an. Je höher die Kurvengeschwindigkeit ist, desto höher ist auch die Querbeschleunigung [69]. Damit nicht kleinere Biegungen der Straße fälschlicherweise als Kurve erkannt werden, wird auch für die Querbeschleunigung ein Mindestwert  $a_{min}$  vorausgesetzt. Mit einem Wert von  $1.5 \frac{m}{s}$  konnten in Tests sehr gute Ergebnisse erzielt werden.

$$a_{min} = \begin{cases} wahr & , |a| \geq 1.5 \frac{m}{s} \\ falsch & , \text{sonst} \end{cases} \quad (6.2)$$

Wie auch bei der Geschwindigkeit wird eine Hysterese verwendet. Diese wird bei der Querbeschleunigung auf  $0.5 \frac{m}{s}$  gesetzt.

3. Zum Herausfiltern von kurzen Lenkmanövern wie einem Spurwechsel, Abbiegevorgang oder einer Kurskorrektur, wird eine Zeitdauer vorgegeben, in der die anderen Bedingungen mindestens erfüllt sein müssen. Erst wenn die Bedingungen länger als  $t_{min} = 3s$  erfüllt sind, wird eine Kurve angenommen.

## 6.1 Benötigte Komponenten für das Gesamtsystem

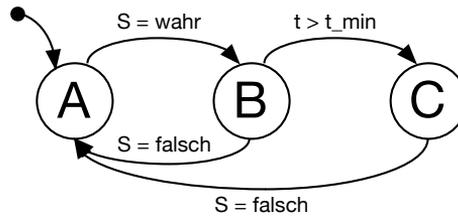


Abb. 6.6: Statechart der Kurvenerkennung für den adaptiven Kurvenwarner.

In Abbildung 6.6 ist die Kurvenerkennung als Statechart mit

$$S = v_{min} \wedge a_{min} \quad (6.3)$$

dargestellt. Wurden die definierten Schwellwerte überschritten, wird  $S$  wahr und ein möglicher Kurveneingang wurde erkannt (Zustand  $B$ ). Werden die Schwellwerte früher als  $t_{min}$  wieder unterschritten, handelt es sich um keine relevante Kurve. Ist  $S$  länger als  $t_{min}$  wahr, wurde eine Kurve erkannt (Zustand  $C$ ). Werden Schwellwerte dann unterschritten wird  $S$  falsch, der Kurvenausgang wurde erreicht und das Fahrzeug befindet sich wieder auf einer geraden Strecke (Zustand  $A$ ).

Dieser Erkennungsalgorithmus wurde mit Sensormesswerten aus einem Simulator (siehe Abschnitt 5.2) und echten Sensormesswerten aus Messfahrten mit dem Versuchsträger getestet. Dabei konnten die Kurven erfolgreich erkannt werden. In Abbildung 5.15 sind die GPS Spuren und die dabei erkannten, sehr unterschiedlichen Kurveneingänge von 27 Fahrten durch eine Kurve auf der Straße *Alt-Lietzow* in Berlin zu sehen. Die erkannten Kurveneingänge sind auf einem circa 25 Meter langen Straßensegment am Anfang der Kurve verteilt. Die sehr unterschiedlichen Positionen der dabei erkannten Kurveneingänge entstehen primär durch die Variation der Geschwindigkeit und des Einlenkverhaltens (sanft gleichmäßig oder kurz abrupt). Aus diesem Grund beginnen und enden die dabei aufgezeichneten Messreihen an unterschiedlichen Positionen. Diese Ergebnisse zeigen die Notwendigkeit der vorgestellten dynamischen Anpassung des gemeinsamen Rasters bei der Zusammenführung von Messwertreihen (siehe auch Abschnitt 4.1.2 und 4.2).

**Weitere Bedingungen für die Selektion relevanter Situationen:** Der umgebende Verkehr kann große Auswirkungen auf die zu lernenden Messgrößen haben und diese damit verfälschen. Ist ein nahes vorausfahrendes

Fahrzeug vorhanden, kann davon ausgegangen werden, dass der Fahrer seine eigene Geschwindigkeit nicht frei wählt. In dieser Situation folgt der Fahrer wahrscheinlich dem vorausfahrenden Fahrzeug und passt seine Geschwindigkeit entsprechend an. Der adaptive Kurvenwarner soll jedoch die Geschwindigkeit lernen, die der Fahrer frei wählt. Um eine solche Situation zu erkennen, kann beispielsweise ein verbauter Radarsensor verwendet werden. Dieser liefert Informationen über ein vorausfahrendes Fahrzeug als sogenanntes Radarziel, unter anderem wird die Distanz des Ziels gemessen. Ist kein Radarziel vorhanden oder ein Ziel weit genug entfernt, soll die Geschwindigkeit gelernt werden. Ist jedoch ein nahes Ziel (beispielsweise innerhalb 300m vor dem Fahrzeug) vorhanden, wird das Lernen ausgesetzt oder mit einer geringeren Gewichtung gelernt.

Diese zusätzlichen Bedingungen für die Kurvenerkennung kann wie folgt ausgedrückt werden:

$$radar_{Ziel} = \begin{cases} wahr & , \text{kein Radarziel vorhanden} \\ falsch & , \text{sonst} \end{cases} \quad (6.4)$$

$$radar_{Distanz} = \begin{cases} wahr & , s \geq 300m \\ falsch & , \text{sonst} \end{cases} \quad (6.5)$$

Diese Bedingungen können mit der allgemeinen Erkennung von Kurven (Gleichung 6.3) verundet werden:

$$S^* = S \wedge radar_{Ziel} \wedge radar_{Distanz} \quad (6.6)$$

So wird auch ein vorausfahrendes Fahrzeug über den Radarsensor beim Lernen von Kurven berücksichtigt.

Sind gewisse Messgrößen für FAS nur während einer freien Fahrt oder nur während dem Folgen eines Fahrzeugs relevant, kann der Lernprozess mit entsprechenden Informationen über den umgebenden Verkehr gesteuert werden. Diese Informationen können mit Hilfe von umgebungserfassenden Sensoren wie einem Radarsensor, einer Stereokamera oder einem LiDAR erfasst und an das lernende System gegeben werden.

**Speicherplatzbedarf:** Im folgenden wird eine Abschätzung des Speicherplatzbedarfs des lernenden Systems am Beispiel der Geschwindigkeit durchgeführt.

Für die Geschwindigkeit wird eine Diskretisierung von  $1 \frac{km}{h}$ -Schritten gewählt und ein Arbeitsbereich von  $1 - 200 \frac{km}{h}$  angenommen. Damit müssen

je Position für die Verteilung der Kerndichtschätzung 200 Werte gespeichert werden. Diese Werte repräsentieren die relative Wahrscheinlichkeit einer bestimmten Geschwindigkeit an einer Position. Sie werden mit  $16\text{Bit}$  aufgelöst und haben damit eine ausreichende Genauigkeit von 0.000015. Darüber hinaus wird für die Kerndichteschätzung noch die Anzahl der erhobenen Messwerte benötigt, welche ebenfalls mit einer  $16\text{Bit}$  großen Variable gespeichert wird. Insgesamt sind für die geschätzte Wahrscheinlichkeitsverteilung an einer Position 201 Werte mit einer Größe von  $16\text{Bit}$  notwendig.

Für die Änderungserkennung müssen die Messwerte der Stichprobenfenster gespeichert werden. Für das Analysefenster wurden 5 Messwerte verwendet, das Gedächtnisfenster für sprunghafte Änderungen umfasst 25 Messwerte und das für schleichende Änderungen 35 Messwerte. Mit einer Auflösung von  $16\text{Bit}$  kann eine ausreichende Genauigkeit von  $\approx 0.003 \frac{\text{km}}{\text{h}}$  erreicht werden. Darüber hinaus muss der Zustand der Änderungserkennung gespeichert werden, wie beispielsweise die berechneten Hypothesen der letzten Durchläufe. Der Zustand einer Hypothesentest-Instanz lässt sich in einer  $8\text{Bit}$  Variable festhalten. Die Zustände beider Instanzen benötigen daher  $16\text{Bit}$ . Es müssen demnach 66 Werte mit je  $16\text{Bit}$  für die Änderungserkennung gespeichert werden.

Insgesamt sind nach dieser Rechnung  $267 \cdot 16\text{Bit} = 4272\text{Bit} = 534\text{Byte}$  Speicher notwendig, um für eine Position die Geschwindigkeit zu lernen. Dieser Speicherbedarf ist dabei konstant und unabhängig von der Anzahl der Messwerte. Wird wie vorgeschlagen eine Auflösung von  $1\text{m}$  für das lernende System gewählt, fällt pro Kilometer Straße eine Datenmenge von  $0.534\text{MByte}$ , für das gesamte überörtliche Straßennetz Deutschlands mit  $230782\text{km}$  [84] eine Datenmenge von  $123.2\text{GByte}$  und für das gesamte Straßennetz Deutschlands mit  $645000\text{km}$  [14] eine Datenmenge von  $344.4\text{GByte}$  an.

Weitere zu lernende Messgrößen erhöhen diese Datenmenge bei gleichen Auflösungen linear, jedoch können mit einem handelsüblichen Datenträger ( $1 - 6\text{TByte}$ ) mehrere Messgrößen für das gesamte deutsche Straßennetz gelernt werden.

Mit der Trennung von Daten, die eine Relevanz für die FAS haben, und jene, die nicht benötigt werden, kann ein großer Teil der zur Verfügung stehenden Daten aussortiert werden. Durch die vorgestellte Selektion kann der Ressourcenbedarf des lernenden Systems deutlich reduziert werden. Die Realisierung des lernenden Systems mit den limitierten Ressourcen

im Fahrzeug wird somit erleichtert.

Mit einer Erkennung von vorausfahrenden Fahrzeugen kann das Lernen von Kurvengeschwindigkeiten auf Situationen beschränkt werden in denen der Fahrer seine Geschwindigkeit frei wählt. So wird ein möglicher Einfluss vom umgebenden Verkehr auf die gelernten Informationen vermieden.

### 6.1.3 Priorisierung zum Löschen von unwichtigen Informationen

Mit einer Bewertung könnten gelernte Informationen priorisiert werden und gezielt unwichtige Informationen gelöscht werden, wenn der Speicher voll ist. In diesem Abschnitt wird dafür ein konfigurierbares und leicht erweiterbares Verfahren zur Bewertung gelernter Informationen vorgestellt.

#### Metriken zur Bewertung

Eine Metrik bewertet Eingabeparameter mit Hilfe einer Funktion. Diese zu bewertenden Eingabeparameter werden von den gelernten Informationen abgeleitet. Aus mehreren Metriken kann eine Priorität berechnet werden, die einen Vergleich von gelernten Informationen ermöglicht.

Im Folgenden werden vier Vorschläge für Metriken gemacht, die für lernende Systeme von Bedeutung sind. Dabei bewerten die Metriken Datensätze, welche eine Situation repräsentieren, wie beispielsweise eine Kurve.

- **Umfang:** Der Umfang eines Datensatzes bezeichnet die Anzahl an Messwerten, die bereits für diesen Datensatz erhoben wurden. Je mehr Messwerte dies sind, desto besser kann die Kerndichteschätzung deren Wahrscheinlichkeitsverteilung schätzen. Deswegen soll diese Metrik bei einem größer werdenden Umfang einen höheren Wert und damit eine höhere Priorität ausgeben. Als Eingabeparameter wird für diese Metrik die Anzahl der Messwerte verwendet.
- **Letzte Verwendung:** Die letzte Verwendung eines Datensatzes kann einen Rückschluss auf die Häufigkeit von dessen Verwendung geben. Je kürzer die Zeitspanne seit der letzten Verwendung ist, desto wahrscheinlicher wird der Datensatz benötigt. Wurde der Datensatz für einen langen Zeitraum nicht benötigt, ist dieser wahrscheinlich wenig relevant. Von dieser Metrik wird eine hohe Priorität bei kleinen Zeitspannen erwartet und im Verlauf zu größeren Zeitspannen muss sich die Priorität verringern. Der Eingabeparameter für diese Metrik ist die Zeitspanne seit der letzten Verwendung.

- **Alter:** Analog zum Alter von digitalen Karten kann auch beim Alter der Daten von Situationen angenommen werden, dass ältere Daten eher von der Realität abweichen [101]. Je älter der Datensatz ist, desto wahrscheinlicher hat beispielsweise eine Änderung in der Zwischenzeit stattgefunden und der Datensatz ist dadurch ungültig. Falls der Datensatz jünger ist, wird es sich wahrscheinlicher um einen gültigen Datensatz handeln. So erhalten jüngere Datensätze in dieser Metrik eine höhere Priorität. Das Alter eines Datensatzes ergibt sich aus dem Mittelwert des Alters der letzten Messwerte. Als Eingabeparameter für diese Metrik wird dieser Mittelwert verwendet.

Neu erfasste Daten müssen bei der Berechnung der Priorität gesondert behandelt werden:

- **Schutz neuer Daten:** Damit neue, dem lernenden System bisher unbekannt, Situationen nicht direkt wieder gelöscht werden, müssen sie geschützt werden. Neue Datensätze werden deswegen mit einem Schutz innerhalb eines Zeitfensters versehen. Dieser Schutz lässt sich mit einer Metrik realisieren: Dazu muss die Funktion bei Datensätzen unterhalb einer Zeitschwelle eine hohe Priorität und nach dieser Schwelle sehr schnell eine niedrige Priorität ausgeben. Eingabeparameter ist die Zeitspanne zwischen der Erhebung des ersten Messwerts und der aktuellen Zeit.

Damit der Speicher nicht vollständig mit neuen Datensätzen gefüllt wird, kann ein maximales Verhältnis von neuen, geschützten Datensätzen zu älteren, ungeschützten Datensätzen festgelegt werden. Wird dieses Verhältnis überschritten, wird der Schutz kurzzeitig ausgesetzt und neue Datensätze mit der niedrigsten Priorität gelöscht. Damit wird das Entfernen von länger bestehenden Daten durch unverhältnismäßig viele neue Datensätze verhindert.

Es werden die folgenden vier Metriken  $M$  zur Berechnung einer Priorität festgelegt:

- $M_1$ : Je größer der **Umfang** eines Datensatzes, desto höher ist die Priorität.
- $M_2$ : Je jünger die **letzte Verwendung** eines Datensatzes ist, desto höher ist die Priorität.

- $M_3$ : Je geringer das **Alter** des Datensatzes, desto höher ist die Priorität.
- $M_4$ : Neu erstellte Datensätze erhalten einen **Schutz** in Form einer hohen Priorität während einer festgelegten Zeitspanne.

Im nächsten Abschnitt wird erläutert, wie sich die Priorität durch die vier Metriken berechnet.

### Berechnung der Priorität

Die Berechnung der Priorität soll leicht erweiterbar und konfigurierbar sein. Damit die Gewichtung der einzelnen Metriken  $M_1$  bis  $M_4$  vereinfacht wird, sollen die Ausgabewerte der Metriken in einem Wertebereich von  $[0, 1]$  liegen. So können die einzelnen Metriken einfach über einen Faktor  $s_i$  gewichtet werden und die gesamte Priorität  $P$  liegt nach folgender Gleichung ebenfalls im Intervall  $[0, 1]$ :

$$P = \frac{\sum_{i=1}^n s_i \cdot M_i}{\sum_{i=1}^n s_i} \quad (6.7)$$

So können neue Metriken einfach hinzugefügt und die Gewichtung von bestehenden Metriken angepasst werden. Die Schnittstelle (Wertebereich der Priorität) im lernenden System muss bei solchen Änderungen nicht angepasst werden. Die Berechnung der vier vorgestellten Metriken wird im Folgenden erläutert.

**Metrik des Umfangs:** Die erste Metrik (Umfang des Datensatzes) lässt sich durch folgende Funktion berechnen:

$$M_1(n_{\text{Werte}}) = 1 - \frac{1}{0.1 \cdot n_{\text{Werte}} + 1} \quad (6.8)$$

Der verwendete Eingabeparameter der Funktion entspricht dabei der Anzahl sämtlicher Messwerte  $n_{\text{Werte}}$  im Datensatz. Der Verlauf der Funktion genügt den eingangs gestellten Anforderungen: Je weniger Messwerte in einem Datensatz vorhanden sind, desto kleiner ist die Priorität und vice versa. Bei  $n = 0$  Messwerten ist die Priorität ebenfalls 0, für  $n \rightarrow \infty$  konvergiert sie gegen 1. Der Verlauf von  $M_1$  ist in Abbildung 6.7 zu sehen. Auf der x-Achse ist der Wert des verwendeten Eingabeparameters  $E$  ( $n_{\text{Werte}}$ ) und auf der y-Achse der Funktionswert der Metrik  $M_1$  aufgetragen.

## 6.1 Benötigte Komponenten für das Gesamtsystem

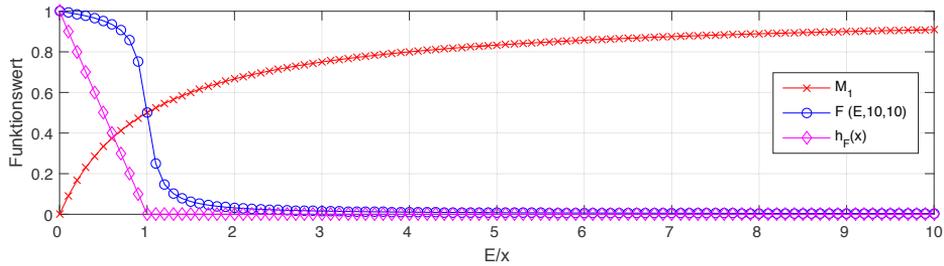


Abb. 6.7: Verlauf der Funktionen  $M_1$ ,  $F(E, 10, 10)$  und  $h_F(x)$  mit  $E$  beziehungsweise  $x$  im Intervall  $[0, 10]$ .

**Metrik der letzten Verwendung, des Alters und des Schutzes:** Bei den Metriken  $M_2$ ,  $M_3$  und  $M_4$  ist der Verlauf grundsätzlich ähnlich: Sie haben alle bei einem kleinen Eingabewert eine hohe Priorität, bei größer werdenden Werten soll die Priorität kleiner werden und sich 0 asymptotisch annähern. Zusätzlich soll die Priorität an einem einstellbaren Wechselpunkt zu einer geringen Priorität umschlagen können (beispielsweise für die Schutz-Metrik). Die Geschwindigkeit des Umschlagens soll dabei festgelegt werden können.

Der Arkuskotangens hat diesen beschriebenen Verlauf. Mit passend gesetzten Faktoren in der Funktion kann ein Wechselpunkt und die Steigung der Funktion in diesem Punkt festgelegt werden. So lassen sich die Metriken (letzte Verwendung, Alter, Schutz neuer Daten) aus einer auf dem Arkuskotangens basierenden Funktion  $F$  ableiten:

$$F(E, c_1, c_2) = \frac{\operatorname{arccot}(E \cdot c_1 - c_2)}{\pi} + h_F\left(E \cdot \frac{c_1}{c_2}\right) \cdot \left(1 - \left(\frac{\operatorname{arccot}(-c_2)}{\pi}\right)\right) \quad (6.9)$$

Für jede Metrik mit ihrem jeweiligen Eingabeparameter  $E$  kann eine passende Konfiguration der Funktion  $F$  über die Faktoren  $c_1$  und  $c_2$  vorgenommen werden, über die der Wechselpunkt und die Steigung in diesem Punkt eingestellt werden kann.

## 6 Anwendung des lernenden Systems

---

Die Hilfsfunktion  $h_F(x)$  entspricht einer an der x-Achse gespiegelten Sägezahnfunktion mit einer Periode von 1, wobei  $h_F(0) = 1$  und  $h_F(1) = 0$  ist. Die Hilfsfunktion wird im zweiten Term von  $F$  benötigt und sorgt dafür, dass unabhängig von den gewählten Werten für  $c_1$  und  $c_2$ , immer  $F(0) = 1$  gilt.

$$h_F(x) = \begin{cases} 0 & , x < 0 \\ 1 - x & , 0 \leq x \leq 1 \\ 0 & , x > 1 \end{cases} \quad (6.10)$$

Die Funktion  $F$  (beispielhaft mit  $c_1 = c_2 = 10$ ) und deren Hilfsfunktion  $h_F$  sind in Abbildung 6.7 zu sehen. Sie hat einen Wertebereich von  $[0, 1]$ : Bei  $E = 0$  ist  $F = 1$  und konvergiert für  $E \rightarrow \infty$  gegen 0. Mit dem Faktor  $c_1$  lässt sich der Wechsellpunkt (Durchgang durch den Wert 0.5) konfigurieren und mit dem Faktor  $c_2$  der Grad der negativen Steigung an diesem Punkt. Es werden nur positive Werte für  $E$  betrachtet, da negative (Zeit-)Werte nicht vorkommen können. Wird der Parameter  $c_1$  für die Steigung im Wechsellpunkt gewählt, kann der gewünschte Wechsellpunkt  $E_{Wechsel}$  anschließend mit dem Parameter  $c_2$  über die Gleichung 6.11 eingestellt werden:

$$c_2 = c_1 \cdot E_{Wechsel} \quad (6.11)$$

Im Folgenden werden die konkreten Funktionen der vorgeschlagenen Metriken  $M_2$ ,  $M_3$  und  $M_4$  vorgestellt:

- Bei der Metrik der *letzten Verwendung* eines Datensatzes  $M_2$  ist ein langsames Abfallen (verglichen mit der Steigung des normalen Arkuskotangens) der Priorität mit zunehmender Zeitdifferenz erwünscht. Aus diesem Grund muss der Parameter  $c_1$  klein gewählt werden, beispielsweise  $c_1 = 0.05$ , um eine flache und negative Steigung zu erhalten. Für die Metrik der *letzten Verwendung* wäre beispielsweise der Wechsellpunkt nach 30 Tagen sinnvoll. Wird die Zeitdifferenz zur letzten Verwendung  $\Delta V$ , also der Eingabeparameter  $E_{\Delta V}$ , in Tagen angegeben, ergibt sich mit Gleichung 6.11 für  $c_2 = 1.5$ . Die Metrik  $M_2$  kann so über folgende Gleichung berechnet werden:

$$M_2(E_{\Delta V}) = F(E_{\Delta V}, 0.05, 1.5) \quad (6.12)$$

Der Verlauf von Metrik  $M_2$  ist in Abbildung 6.8 zu sehen.

## 6.1 Benötigte Komponenten für das Gesamtsystem

- Für die Metrik über das *Alter* eines Datensatzes  $M_3$  wird ein größerer Zeitraum für den Abfall der Priorität gewählt. Deshalb wird der negative Gradient der Funktion mit  $c_1 = 0.01$  noch flacher als bei  $M_2$  gewählt. Der Wechsellpunkt wird bei einem durchschnittlichen Alter des Datensatzes von 180 Tagen eingestellt. Daraus ergibt sich mit Hilfe der Gleichung 6.11 für  $c_2 = 1.8$ . Diese Metrik kann wie folgt berechnet werden:

$$M_3(E_{Alter}) = F(E_{Alter}, 0.01, 1.8) \quad (6.13)$$

Der Verlauf von Metrik  $M_3$  ist ebenfalls in Abbildung 6.8 zu sehen.

- Die Metrik für den *Schutz neuer Daten*  $M_4$  soll den Schutz  $E_{Schutz}$  für 7 Tage aufrechterhalten (Wechsellpunkt). Dazu muss die Metrik bis dahin einen Wert nahe 1 und danach schnell einen Funktionswert nahe 0 annehmen. Es wird also ein großer negativer Gradient für diese Metrik benötigt, der sich mit einem vergleichsweise großen Wert für den Parameter  $c_1 = 140$  realisieren lässt. So ergibt sich für den Wechsellpunkt nach 7 Tagen mit Gleichung 6.11 für  $c_2 = 20$ . Die Schutz-Metrik  $M_4$  berechnet sich wie folgt:

$$M_4(E_{Schutz}) = F(E_{Schutz}, 20, 140) \quad (6.14)$$

Auch der Verlauf von Metrik  $M_4$  ist in Abbildung 6.8 zu sehen.

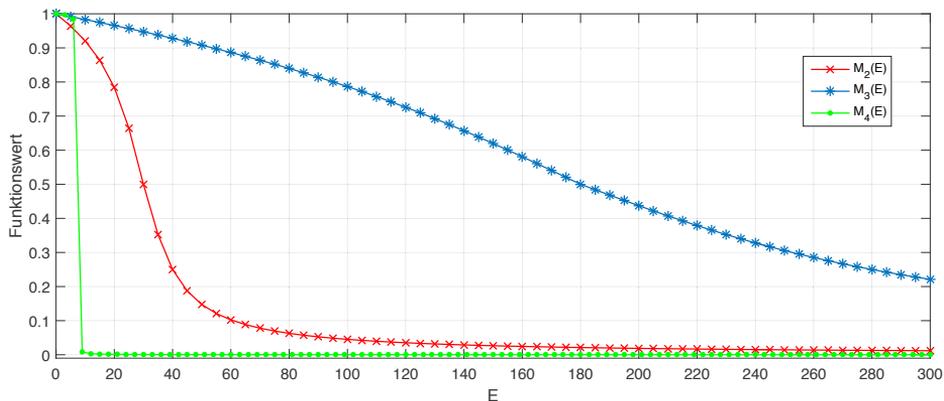


Abb. 6.8: Verlauf der Metriken  $M_2$ ,  $M_3$  und  $M_4$  mit  $E$  im Intervall  $[0, 300]$ .

### Bewertung von gelernten Informationen

Anhand von simulierten Daten wird gezeigt, dass die vorgeschlagenen Metriken für die Anwendung im Fahrzeug geeignet sind. Dazu werden 10000 Messungen von 500 Kurven, die je einen Datensatz bilden, über einen Zeitraum von zwei Jahren (730 Tage) simuliert. Die 500 Kurven, beziehungsweise Datensätze, besitzen eine gleichmäßig verteilte Wahrscheinlichkeit, die besagt, wie oft das Fahrzeug in den 730 Tagen an dieser Kurve vorbeifährt. Anschließend werden 10000 Messungen simuliert und den Datensätzen auf Basis dieser Wahrscheinlichkeit zugeordnet. Dabei wird auch der Lernprozess simuliert und die relevanten Eigenschaften für die Metriken berechnet. In Abbildung 6.9 sind diese Eigenschaften der 500 Datensätze als Histogramm dargestellt:

- Der erste Plot in Abbildung 6.9 zeigt, wie viele Datensätze welche **Anzahl an Messungen haben**. Dieser Umfang der Messungen wird für  $M_1$  benötigt.

Es existieren demnach viele Datensätze, die weniger als ~40 Messungen haben und wenige Datensätze die mehr als ~40 Messungen haben.

- Der zweite Plot in Abbildung 6.9 zeigt, wie viele Datensätze vor wie vielen Tagen **das letzte Mal verwendet wurden**. Die letzte Verwendung wird für  $M_2$  benötigt.

Die meisten Datensätze wurden in den letzten 100 Tagen verwendet und nur bei wenigen liegt die letzte Verwendung weiter als 100 Tage zurück.

- Der dritte Plot in Abbildung 6.9 zeigt, wie viele Datensätze welches **Alter** (Durchschnitt der letzten 7 Messungen) haben. Das durchschnittliche Alter wird für  $M_3$  benötigt.

Die meisten Datensätze haben ein Alter um 100 Tage. Die Anzahl dieser nimmt dann bis zum maximalen Alter von 730 Tagen ab.

- Der vierte Plot in Abbildung 6.9 zeigt, bei wie vielen Datensätzen dessen **erste Messung ein bestimmtes Alter** hat. Das Alter der ersten Messung wird für  $M_4$  benötigt.

Das Alter der ersten Messung jedes Datensatzes ist gleichmäßig über alle 730 Tage verteilt.

## 6.1 Benötigte Komponenten für das Gesamtsystem

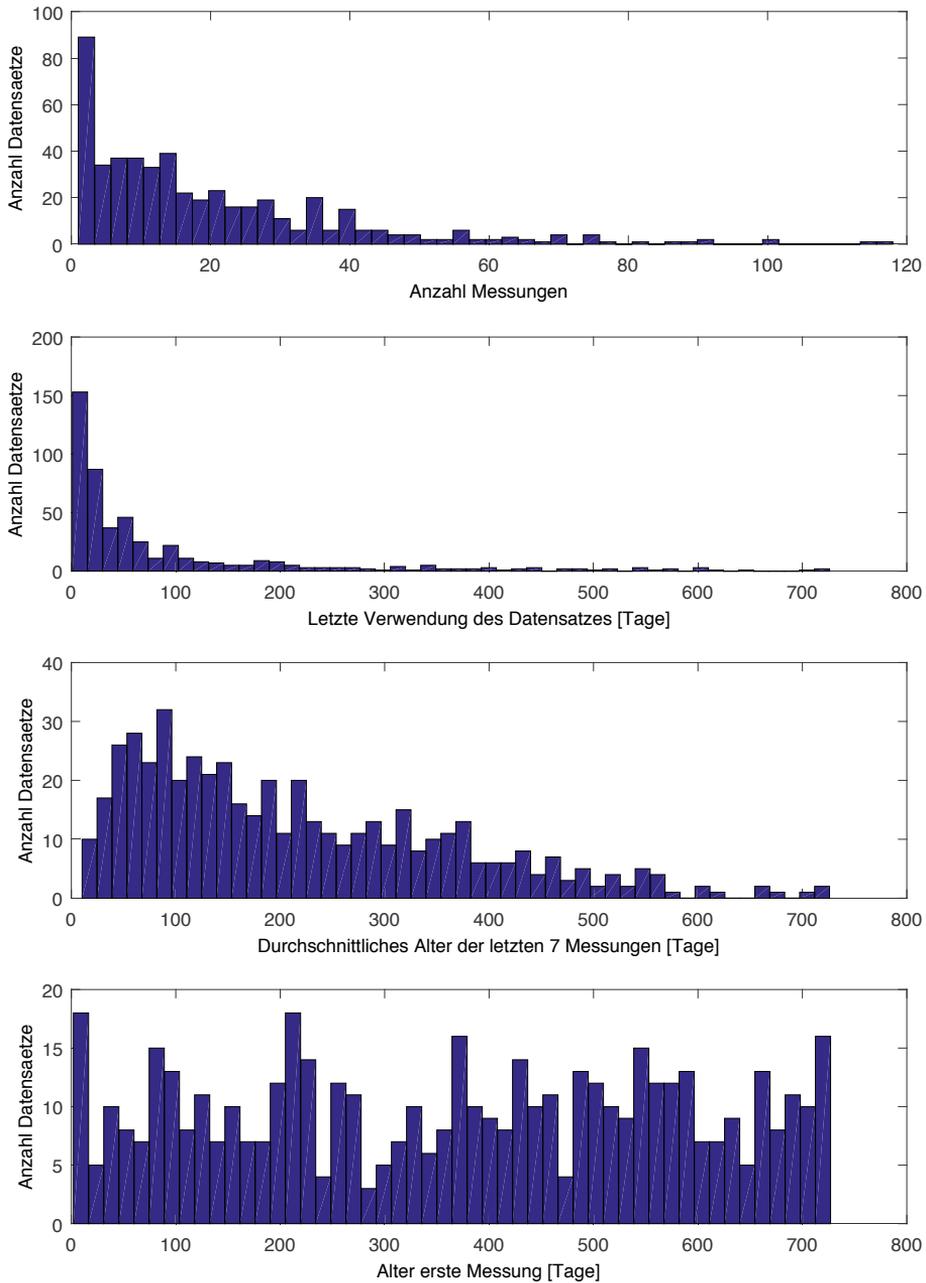


Abb. 6.9: Häufigkeitsverteilung der vier Eingabeparameter der 500 simulierten Datensätze als Histogramm.

Basierend auf diesen simulierten Datensätzen wurden die Metriken und die Priorität berechnet. Die Metriken  $M_1$ ,  $M_2$ ,  $M_3$ ,  $M_4$  und die Priorität  $P$  bilden alle auf das Intervall  $[0, 1]$  ab. Das Ziel der Metriken und vorrangig der Priorität ist, die Eingabeparameter so zu bewerten, dass eine gleichmäßige Verteilung in diesem Wertebereich entsteht. Die Ergebnisse sind in Abbildung 6.10 als Histogramm dargestellt:

- Der erste Plot in Abbildung 6.10 zeigt die Verteilung der *Umfang*-Metrik  $M_1$ . Diese Metrik konnte die Datensätze gleichmäßig verteilen. Die Lücken im Bereich 0 bis circa 0.5 entstehen aufgrund des ganzzahligen Eingabeparameters der Metrik (Anzahl der Messungen). Da sich die Funktion von  $M_1$  nur asymptotisch an den Wert 1 annähert, konnte bei der simulierten Anzahl an Messungen ein Wert von 1 noch nicht erreicht werden.
- Der zweite Plot in Abbildung 6.10 zeigt die Verteilung der *letzte Verwendung*-Metrik  $M_2$ . Auch diese Metrik kann die Datensätze, bis auf kleine Häufungen im Bereich um 0 und 1, gleichmäßig auf den Wertebereich abbilden.
- Der dritte Plot in Abbildung 6.10 zeigt die Verteilung der *Alter*-Metrik  $M_3$ . Bis auf kleine Häufungen in den Bereichen 0.1 und 0.9 ist auch diese Metrik in der Lage den Eingabeparameter gleichmäßig auf den Wertebereich abzubilden.
- Der vierte Plot in Abbildung 6.10 zeigt die Verteilung der *Schutz*-Metrik  $M_4$ . Der Schutz neuer Datensätze ist erwartungsgemäß nur für sehr wenige Datensätze aktiv, deren erste Messung jünger als 7 Tage ist. Alle anderen Messungen erhalten einen sehr niedrigen Wert und werden nicht mehr geschützt. Bei dieser Metrik soll aufgrund ihrer Sonderfunktion keine gleichmäßige Verteilung auf den Wertebereich erfolgen.
- Der fünfte Plot in Abbildung 6.10 zeigt die Verteilung der Priorität  $P$ . Sie kombiniert alle vier vorgestellten Metriken mit individuellen Gewichten (siehe Gleichung 6.7). Da  $M_4$  ein vorübergehender Schutz ist, würde sich die Priorität  $P$  außerhalb der Schutzzeit aus  $M_1$ ,  $M_2$  und  $M_3$  zusammensetzen (jeweils eine Gewichtung von  $\frac{1}{3}$ ).  $M_2$  und  $M_3$  basieren beide auf dem Alter des Datensatzes,  $M_1$  hingegen auf dem Umfang des Datensatzes. Damit diese beiden Eigenschaften

## 6.1 Benötigte Komponenten für das Gesamtsystem

---

(Umfang und Alter) gleichgewichtet (jeweils eine Gewichtung von  $\frac{1}{2}$ ) werden, wird für  $M_1$  eine Gewichtung von 2 gewählt:

$$P = \frac{2 \cdot M_1 + 1 \cdot M_2 + 1 \cdot M_3 + 1 \cdot M_4}{5} \quad (6.15)$$

Die Priorität ist gleichmäßig über den Wertebereich verteilt, mit der Ausnahme des oberen Bereichs (circa 0.8 bis 1) aufgrund von  $M_4$ , die in den meisten Fällen 0 ist.

Die Berechnung der Priorität ist nicht zeitkritisch und kann mit freien Ressourcen während der Fahrt oder nach einer Fahrt berechnet werden. Dadurch sind die Kosten der Berechnung nicht kritisch und müssen keinen Grenzwert einhalten.

Mit Hilfe der Priorität  $P$ , die aus den vorgeschlagenen Metriken  $M_1$ ,  $M_2$ ,  $M_3$  und  $M_4$  berechnet wird, kann eine Bewertung der Informationen im lernenden System durchgeführt werden. Damit können die unwichtigsten Informationen gezielt gelöscht werden, falls beispielsweise der Speicherplatz im Fahrzeug voll ist. Dadurch kann eine uneingeschränkte Funktion des lernenden Systems in Bezug auf freien Speicherplatz gewährleistet werden, ohne unwissentlich wichtige Daten zu löschen. So können im Fall des adaptiven Kurvenwarners neue Kurven immer gelernt werden.

Sobald jedoch Daten aus dem lernenden System gelöscht werden, wird das Potential des lernenden Systems eingeschränkt. Dies sollte deswegen verhindert werden. Die Berechnung einer Priorität ermöglicht es dem lernenden System einen Zustand herzustellen in dem weiterhin gelernt werden kann. Tritt ein solcher Fall ein, kann beispielsweise der Fahrer darüber informiert werden. So hat er die Möglichkeit den Speicherplatz zu vergrößern oder manuell gelernte Informationen zu löschen (beispielsweise für bestimmte Regionen), um die uneingeschränkte Funktion wiederherzustellen.

## 6 Anwendung des lernenden Systems

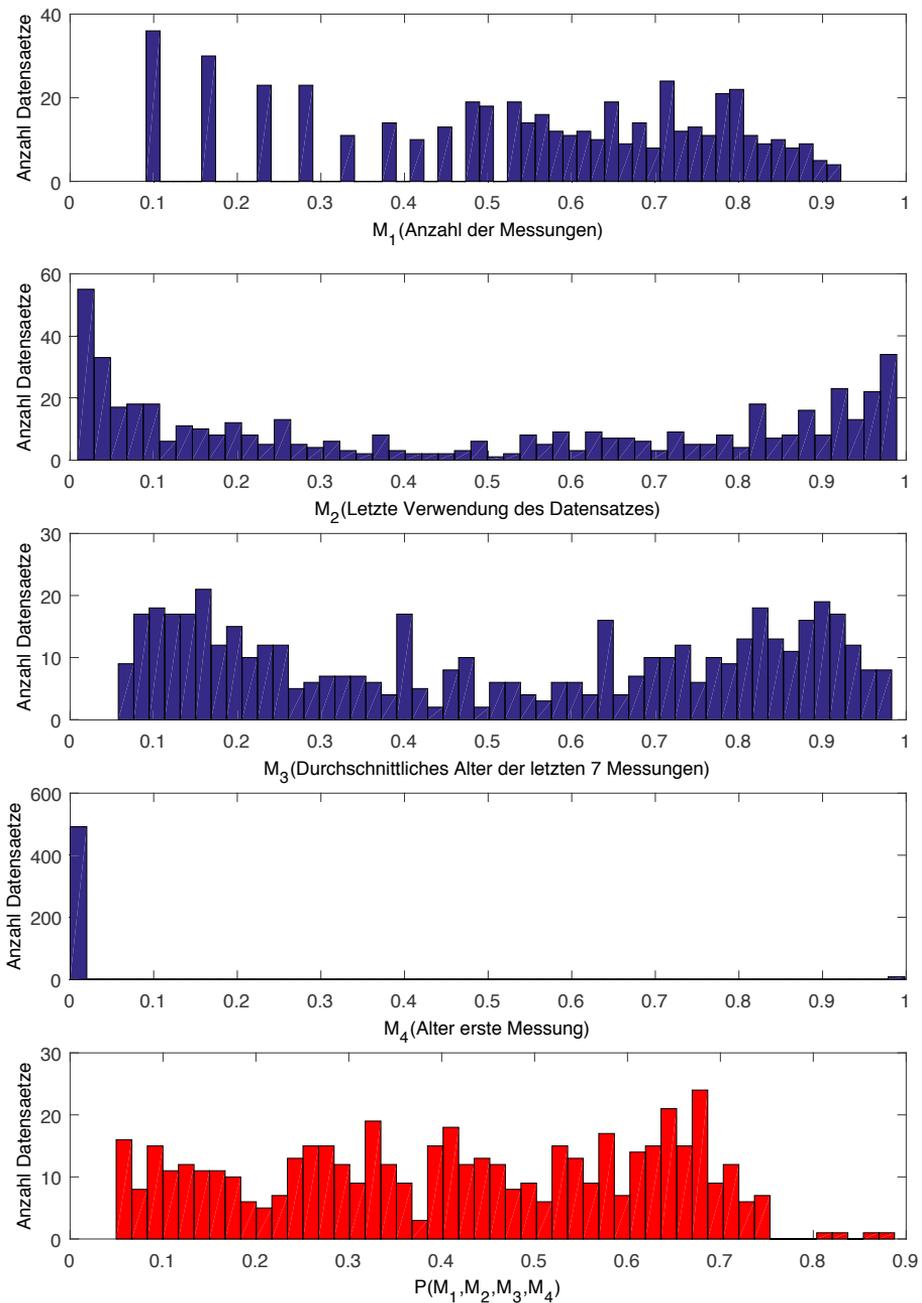


Abb. 6.10: Histogramm der Metriken  $M_1$ ,  $M_2$ ,  $M_3$ ,  $M_4$  und  $P$  der 500 simulierten Datensätze.

## 6.1 Benötigte Komponenten für das Gesamtsystem

Die drei Komponenten *Key-Value DBS*, *Selektion* und *Berechnung der Priorität* sollen mit dem erweiterten Lernprozess mit der *Messwertreihen-Transformation* und dem *t-Test-Vergessen* in ein Gesamtsystem integriert werden. Eine Übersicht dieser Komponenten und deren Einordnung in das lernende System ist in Abbildung 6.11 dargestellt.

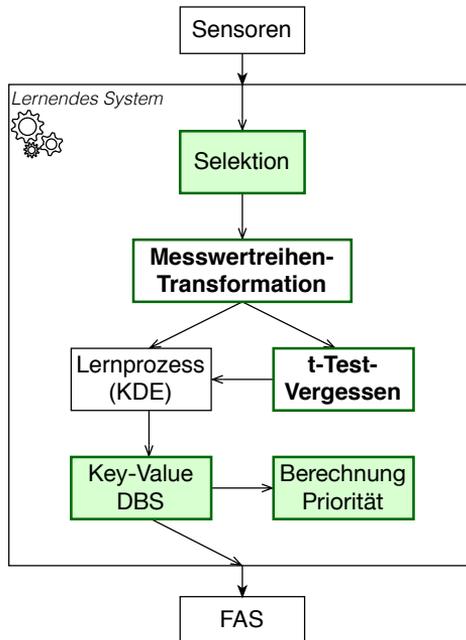


Abb. 6.11: Komponenten des lernenden Systems der vorliegenden Arbeit.  
Die Beiträge des 6. Kapitels sind grün hervorgehoben.

### 6.2 Der adaptive Kurvenwarner im Versuchsträger

Der adaptive Kurvenwarner basierend auf dem in der vorliegenden Arbeit entwickelten lernenden System wurde im Versuchsträger getestet.

Dafür stand ein Versuchsträger der *Daimler AG* zur Verfügung, ein *Mercedes-Benz* des Modells *S-Klasse* (Baujahr 2006). Für die Tests wurden die Beiträge der vorliegenden Arbeit implementiert und ein Framework für ein lernendes System erschaffen. Der adaptive Kurvenwarner wurde als FAS in diesem Framework realisiert. Das gesamte System wurde auf einem Laptop eingerichtet, wodurch der Laptop im Versuchsaufbau das Steuergerät ersetzt. Die Sensordaten konnten von den Fahrzeugbussen über mehrere USB-CAN-Adapter von PEAK [67] am Laptop ausgelesen werden. Zur Simulation des Steuergeräts und zum Auslesen der Sensordaten wurde ein Laptop des Modells *Lenovo ThinkPad T410s* [43] mit Intel Core i5-540M Prozessor, 8GB RAM und SSD verwendet. Als Akteur beziehungsweise HMI wurde ein Tablet verwendet, das über WLAN mit dem Laptop verbunden wurde. Das Tablet stellt die Fahrzeugposition auf der digitalen Karte und die ausgelösten Warnungen dar. Dieser Versuchsaufbau ist in Abbildung 6.12 schematisch dargestellt.

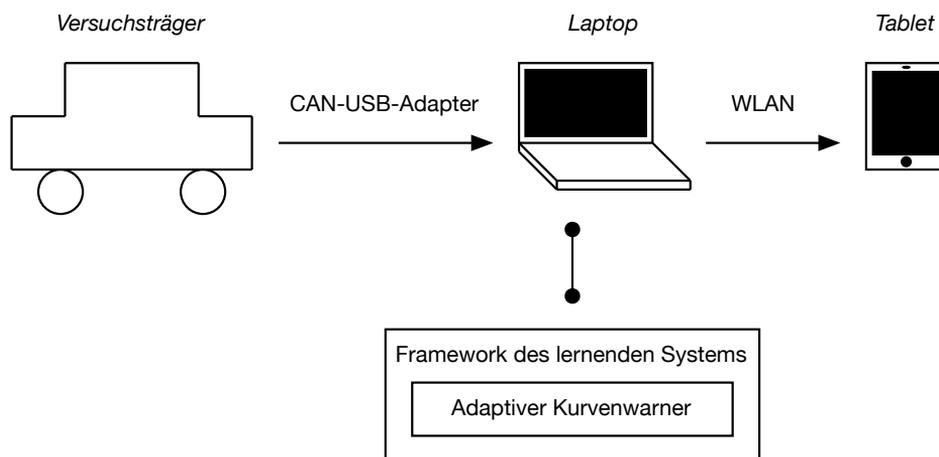


Abb. 6.12: Schematische Darstellung des Versuchsaufbaus.

Abbildung 6.13 zeigt den Innenraum des Versuchsträgers mit Laptop und dem in der Mittelkonsole angebrachten Tablet für das HMI. Das HMI des adaptiven Kurvenwarners verwendet alle drei Arten der HMI-Aktoren: Visuell, akustisch und haptisch (siehe 2.1.2). Visuell wird dem Fahrer auf

## 6.2 Der adaptive Kurvenwarner im Versuchsträger

einem Display die Fahrzeugposition auf der digitalen Karte präsentiert und bei einer Kurvenwarnung die gelernte Geschwindigkeit inklusive der Kurvenrichtung angezeigt. Die Kurvenwarnung wird akustisch durch Warntöne unterstützt. Beim Prototyp wurde für die Anzeige des HMI ein Tablet benutzt, das über eine Vibrationsfunktion verfügt. Dieser haptische Aktor wurde parallel zu den Warntönen verwendet. Für eine Umsetzung in einem Serienfahrzeug kann stattdessen ein Vibrationsmotor beispielsweise im Lenkrad oder im Sitz verwendet werden.



Abb. 6.13: Bild vom Versuchsträger mit HMI-Tablet (angebracht an die Mittelkonsole) und Laptop mit CAN-Anbindung auf dem das lernende System und der adaptive Kurvenwarner ausgeführt wird.

Die durchgeführten Testfahrten verliefen erfolgreich. Insgesamt wurde der adaptive Kurvenwarner in über 50 Fahrten getestet. Das lernende System konnte Kurven erkennen und im Datenspeicher mit der entsprechenden Identifikationsnummer der Straße aus der digitalen Karte ablegen. Jede weitere Fahrt durch eine erkannte Kurve wurde korrekt zugeordnet und die gelernten Informationen entsprechend durch den Lernprozess aktualisiert. Wenn auf eine bereits erkannte Kurve zugefahren wird, informiert das Tablet den Fahrer über die Position und Richtung der kommenden Kurve. Wird mit erhöhter Geschwindigkeit auf die Kurve zugefahren, löst der adaptive Kurvenwarner zuverlässig eine präzise Warnung über das Ta-

blet aus. Im folgenden werden die verschiedenen Hinweise und Warnungen des adaptiven Kurvenwarners beispielhaft an einer Fahrt durch die *Birkenallee*-Kurve vorgestellt (HMI Design von Fraunhofer Fokus):

- Hinweis 1: Beim Anfahren der bereits gelernten Kurve wurde eine Markierung in Form eines Schildes angezeigt. Das Schild informiert über die Position und Richtung der Kurve, die bereits erkannt und gelernt wurde (siehe Abbildung 6.14).
- Hinweis 2: Beim Annähern an die bekannte Kurve wird in der rechten oberen Ecke des Displays ein größeres Hinweisschild mit der Kurvenrichtung angezeigt (siehe Abbildung 6.15). Gleichzeitig wird der angezeigte Kartenausschnitt auf die Kurve fixiert.
- Warnung 1: Bei dieser Fahrt wurde die Geschwindigkeit über das gelernte Niveau erhöht. Dadurch würde das Fahrzeug zu schnell in die Kurve einfahren. Am rechten Rand des Displays wird eine Warnung ausgegeben, welche die gelernte Geschwindigkeit (abgerundet auf  $5 \frac{km}{h}$  Schritte) und die Kurvenrichtung enthält (siehe Abbildung 6.16). Dabei wird ein einfacher Warnton ausgegeben und die Vibration des Tablets aktiviert.
- Warnung 2: Reagiert der Fahrer nicht auf diese Warnung und verzögert nicht, wird die zweite Warnstufe aktiv. Die Geschwindigkeitswarnung mit Kurvenrichtung wird in diesem Fall als Vollbildwarnung dargestellt (siehe Abbildung 6.17). Zusätzlich wird ein Warnton zweifach ausgegeben und ebenfalls die Vibration aktiviert.

Die Tests des lernenden Systems am Beispiel des adaptiven Kurvenwarners im Versuchsträger demonstrierten die Tauglichkeit der Beiträge der vorliegenden Arbeit unter realen Bedingungen.

## 6.2 Der adaptive Kurvenwarner im Versuchsträger

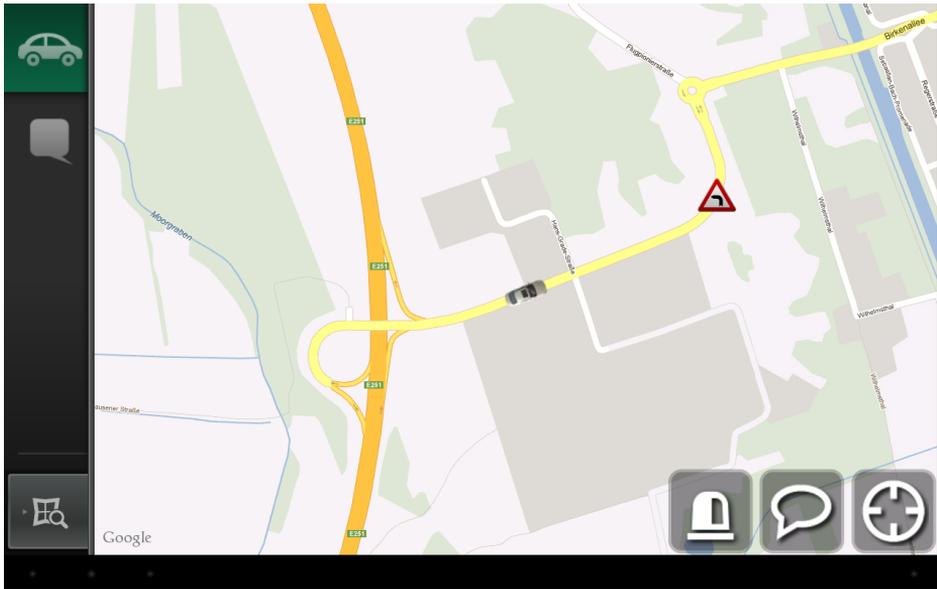


Abb. 6.14: HMI des adaptiven Kurvenwarners bei der Anfahrt an eine gelernte Kurve, die markiert wurde.

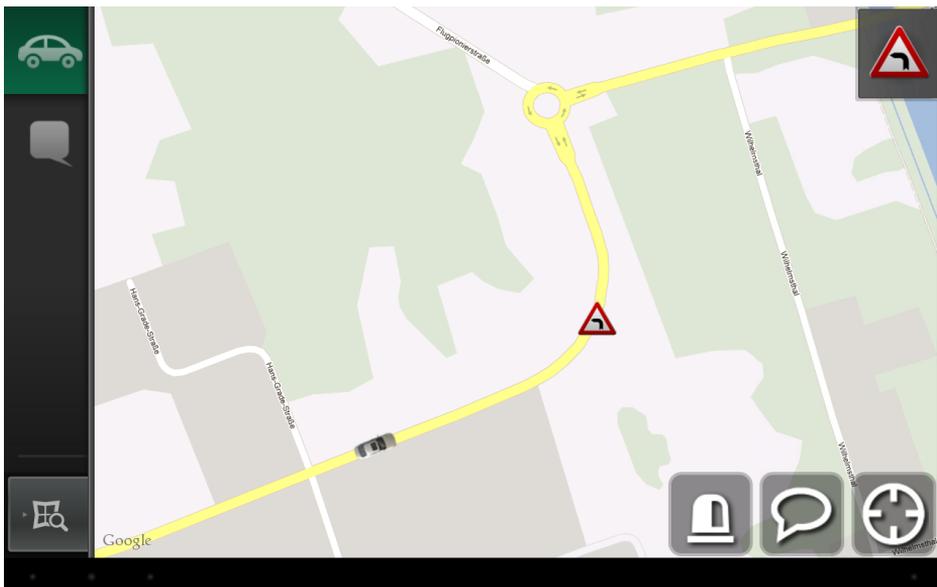


Abb. 6.15: HMI des adaptiven Kurvenwarners bei der Annäherung an eine gelernte Kurve mit Infosymbol in der oberen rechten Ecke.

## 6 Anwendung des lernenden Systems

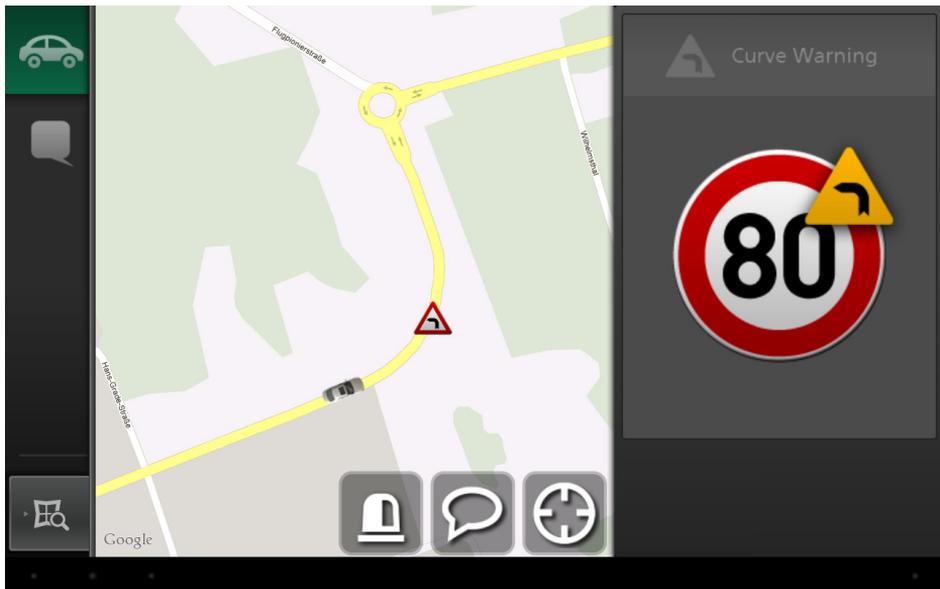


Abb. 6.16: HMI des adaptiven Kurvenwarners. Ersten Stufe der Warnung: Gelernte Geschwindigkeit und Kurvenrichtung am rechten Rand.

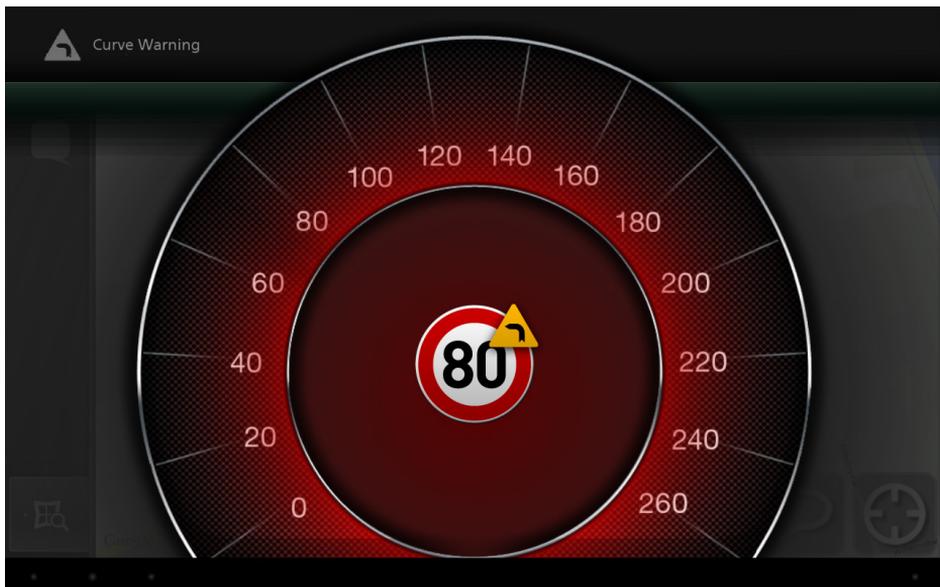


Abb. 6.17: HMI des adaptiven Kurvenwarners. Zweiten Stufe der Warnung: Gelernte Geschwindigkeit und Kurvenrichtung als Vollbild.

## 6.3 Zusammenfassung

In diesem Kapitel wurde gezeigt, wie die Realisierung eines lernenden Systems im Fahrzeug und des adaptiven Kurvenwarners, durch die Beiträge der vorliegenden Arbeit ermöglicht wurde. Der adaptive Kurvenwarner konnte in einem Versuchsträger mit realen Sensordaten in Echtzeit erfolgreich getestet werden.

Um ein lauffähiges Gesamtsystem zu erschaffen, wurde der erweiterte Lernprozess mit der *Messwertreihen-Transformation* und dem *t-Test-Vergessen* um drei weitere Komponenten ergänzt.

Es wurde ein Datenspeicher entwickelt, der die gelernten Informationen persistent speichern kann und einen schnellen Zugriff auf diese Daten bietet. Dafür wurde die *Oracle Berkeley DB*, ein Key-Value Datenbankmodell, in Kombination mit einer Adressierung über die Straßenidentifikationsnummern der digitalen Karte verwendet, die ein intelligentes Vorladen von Informationen ermöglicht.

Eine Selektion von relevanten Sensordaten konnte die Menge der zu verarbeitenden Daten deutlich reduzieren, indem nur benötigte Sensordaten in relevanten Situationen aufgezeichnet und gelernt werden. Dazu wurde eine Erkennung von Kurven für den adaptiven Kurvenwarner vorgestellt. Darüber hinaus wurde gezeigt, wie der Einfluss vom Straßenverkehr auf die gelernten Informationen ausgeschlossen werden kann.

Die Berechnung einer Priorität zur Bewertung von gelernten Informationen wurde mit 10000 simulierten Messungen in 500 Kurven erfolgreich überprüft. Mit vier vorgestellten Metriken, konnten die gelernten Informationen gleichmäßig auf den Wertebereich der Priorität verteilt werden. So können jederzeit die Informationen mit der niedrigsten Priorität bei unzureichend freiem Speicherplatz gelöscht werden. Damit kann stets freier Speicherplatz zum Lernen gewährleistet werden, ohne wichtige Daten zu entfernen.

Das lernende System mit dem adaptiven Kurvenwarner wurde implementiert, im Versuchsträger inklusive eines HMIs integriert und erfolgreich während der Fahrt im Fahrzeug getestet. Die präzise ausgelösten Kurvenwarnungen bei gelernten Kurven ermöglichten eine rechtzeitige Reduzierung der Geschwindigkeit auf eine gelernte normale Geschwindigkeit.

Die prototypische Umsetzung im Versuchsträger und die dabei entstandenen Ergebnisse zeigen, dass ein lernendes System mit den Beiträgen der vorliegenden Arbeit ein neues Spektrum an Verbesserungen bestehender und die Umsetzung neuer FAS ermöglicht.



# Zusammenfassung

---

Ziel dieser Arbeit war die Verbesserung bestehender und das Entwickeln neuer FAS mit Hilfe eines lernenden Systems im Fahrzeug. Als Leitbeispiel wurde dafür ein adaptiver Kurvenwarner verwendet, der mit den Beiträgen dieser Arbeit umgesetzt werden konnte und erfolgreich in einem Versuchsträger ausgeführt wurde.

## 7.1 Beiträge

Im Folgenden werden die konkretisierten Ziele dieser Arbeit aus Abschnitt 3.5 aufgegriffen, um die Beiträge der vorliegenden Arbeit zusammenzufassen.

### Lernen von heterogenen Sensormesswertreihen

**Ziel:** Der Lernprozess soll erweitert werden, damit heterogene Sensormesswertreihen mit der Kerndichteschätzung gelernt werden können.

FAS können mit den detaillierten Informationen durch das Lernen von Messwertreihen deutlich präziser agieren, wie es am adaptiven Kurvenwarner gezeigt wurde. Damit die Kerndichteschätzung für die Zusammenfassung von Messwertreihen verwendet werden kann, wurde in der vorliegenden Arbeit eine Messwertreihen-Transformation erarbeitet. Mit dieser Transformation werden die zeitabhängigen, heterogenen Messwertreihen der Sensoren auf ein gemeinsames, positionsabhängiges Raster abgebildet und damit homogenisiert. Für die FAS liegen damit für jeden Rasterpunkt die Wahrscheinlichkeitsverteilungen der Sensormesswerte vor und es kann beispielsweise das wahrscheinlichste Geschwindigkeitsprofil durch eine Kurve abgeleitet werden. Mit einem solchen Profil kann die Funktion von FAS erweitert werden. Der adaptive Kurvenwarner kann so deutlich präzisere Warnungen ausgeben, als mit bisher vorgestellten lernenden Systemen im Fahrzeug.

### Erkennung von ungültigen Informationen im lernenden System

**Ziel:** Es soll eine unabhängige und schnellere Änderungserkennung zur Detektion ungültiger Informationen im lernenden System erarbeitet werden.

In der vorliegenden Arbeit wurde eine neue, schnellere und unabhängige Änderungserkennung basierend auf einem Hypothesentest entwickelt. Mit dieser können Informationen aus dem lernenden System entfernt werden, die durch eine Änderung ungültig geworden sind. Der verwendete Zweistichproben-*t*-Test ermöglicht es dem *t-Test-Vergessen* durch passend gewählte Stichproben verschiedene Änderungstypen zu erkennen, dazu zählen *sprungartige* und *schleichende* Änderungen. Dabei werden die Sensormesswerte direkt für die Änderungserkennung verwendet, wodurch das *t-Test-Vergessen* unabhängig vom verwendeten Lernprozess ist. Dadurch ist die Leistungsfähigkeit der Änderungserkennung konstant und unabhängig von der Qualität des Lernprozesses. Mit dem *t-Test-Vergessen* kann die einwandfreie Funktion von FAS, wie dem adaptiven Kurvenwarner, schneller wiederhergestellt werden.

### Anwendung des lernenden Systems

**Ziel:** Das lernende System soll als lauffähiges Gesamtsystem im Versuchsträger integriert und der adaptive Kurvenwarner realisiert werden. Dazu soll der für das Lernen von Messwertreihen erweiterte Lernprozess, die neu entwickelte Änderungserkennung, der Datenspeicher, die Selektion relevanter Daten und die Priorisierung von Informationen implementiert und als ein lernendes System zusammengefügt werden.

Das lernende System wurde als lauffähiges Gesamtsystem im Versuchsträger integriert, um den adaptiven Kurvenwarner mit echten Sensordaten im Fahrzeug zu testen.

Dazu wurden drei periphere Komponenten entwickelt, implementiert und getestet.

1. Es wurde gezeigt, dass für einen schnellen, persistenten Datenspeicher die Adressierung über eine digitale Karte in Verbindung mit einem Key-Value-Datenbankmodell eine geeignete Wahl ist, um die gelernten Informationen schnell den FAS bereitzustellen.

## 7.2 Weiteres Beispiel für die Anwendung des lernenden Systems

---

2. Zur Verringerung des Verarbeitungsaufwands im lernenden System wurde eine Selektion benötigter Daten in relevanten Situationen vorgestellt. Als Beispiel wurde eine Kurvenerkennung für den adaptiven Kurvenwarner realisiert.
3. Damit neue Informationen gelernt werden können, müssen bei vollem Speicher Daten aus dem lernenden System entfernt werden. Es wurde ein leicht erweiterbares und konfigurierbares Verfahren zur Bewertung von gelernten Informationen entwickelt. Dazu wurde ein Algorithmus für eine Priorität basierend auf Metriken entwickelt, die verschiedene Eigenschaften der gelernten Informationen bewerten wie Alter, Messwertumfang und deren Nutzung. So können gezielt die unwichtigsten Daten aus dem lernenden System entfernt werden.

Diese drei Komponenten wurden in ein Gesamtsystem integriert und im Fahrzeug getestet. Ebenfalls wurde die *Messwertreihen-Transformation* und das *t-Test-Vergessen* mit echten Sensordaten aus dem Versuchsträger getestet und ausgewertet. Mit Hilfe dieser Beiträge konnte ein lernendes System im Fahrzeug realisiert werden. Damit ist es möglich die Funktion von FAS zu verbessern, wie es am Beispiel des adaptiven Kurvenwarner gezeigt wurde. Dieser konnte unbekannte Kurven erkennen, deren Geschwindigkeitsprofile lernen, den Fahrer rechtzeitig vor einer überhöhten Geschwindigkeit warnen und sich Änderungen anpassen. Es können auch die anderen skizzierten Verbesserungen bestehender FAS mit diesem lernenden System umgesetzt werden, wie beispielsweise die adaptive Automatikschaltung.

Ein lernendes System mit den Beiträgen dieser Arbeit eröffnet vielen FAS, die in modernen Fahrzeugen verbaut sind, neues Potential für eine verbesserte Funktion.

## 7.2 Weiteres Beispiel für die Anwendung des lernenden Systems

Eine weitere Anwendung des lernenden Systems wäre die Umsetzung der in Abschnitt 3.1 vorgestellten *adaptiven Automatikschaltung*.

Damit die Automatikschaltung vor Steigungen nicht einen möglicherweise falschen Gang einlegt, können die Positionen und der Grad der Stei-

gungen gelernt werden [13]. Dies sind die relevanten Situationen, in denen die adaptive Automatikschaltung aktiv ist und versucht einen Gang zu wählen, mit dem die dort typische Beschleunigung des Fahrers umgesetzt werden kann. Dafür muss das lernende System während einer Steigung die gemessene oder die angeforderte Fahrzeuglängsbeschleunigung lernen.

Mit dieser Information kann vor und bei Steigungen ein passender Gang eingelegt oder gehalten werden und damit unnötige Schaltvorgänge und Verschleiß reduzieren. Mit der bekannten Motorleistung über den Drehzahlverlauf, der Getriebeübersetzung und dem Fahrzeuggewicht kann die mögliche Längsbeschleunigung des Fahrzeugs berechnet werden. Unterschreitet diese Längsbeschleunigung die vom Fahrer angeforderte Längsbeschleunigung an einer Position, muss ein niedrigerer Gang gewählt werden.

### 7.3 Ausblick

Die vorliegende Arbeit verbessert den Lernprozess für ein lernendes System im Fahrzeug. Die Beiträge der vorliegenden Arbeit wurden unter Annahmen und Einschränkungen (siehe Abschnitt 3.2) entwickelt. Zukünftige Forschungen könnten das lernende System weiter verbessern, indem folgende Aspekte näher untersucht werden:

**Spurbasierte Zusammenführung von Messwertreihen:** Bisher wird die Zusammenführung von Messwertreihen unabhängig von der Spur durchgeführt, in der sich das Fahrzeug befindet. Dadurch werden die Messwertreihen aller Spuren einer Fahrtrichtung als eine Spur betrachtet und zusammengeführt. Wird das Fahrzeug eindeutig einer Spur zugeordnet, kann das lernende System Informationen für jede Spur getrennt lernen.

**Klassifikation von Modi nach Änderungen:** Änderungen in Sensorsignalen können aus verschiedenen Gründen entstehen. Die gefahrene Geschwindigkeit in einer Kurve kann sich durch Schlaglöcher oder Baustellen verändern oder aber durch bestimmte Witterungen wie Nebel, Nässe, Schnee oder Eis. Das Verwerfen der gelernten Informationen einer trockenen Fahrbahn, wenn Schnee liegt, ist nicht sinnvoll. Stattdessen wäre es sinnvoll verschiedene wiederkehrende Modi zu lernen, zwischen denen in solchen Fällen umgeschaltet wird. Diese Modi können als Trainingsdaten für Klassifikationsverfahren wie Support Vector Machines oder neuronale

Netze verwendet werden, mit denen versucht werden kann, die Messungen nach einer Änderung einem bekannten Modus zuzuordnen.

**Lernen komplexer Messwerte:** Es können skalare Messwertreihen gelernt werden, komplexe Messwerte wie (Stereo-)Kamerabilder oder 3D-Punktwolken vom LiDAR Sensor hingegen nicht. Der Lernprozess könnte erweitert werden, so dass multidimensionale Messwerte und Messwertreihen gelernt werden. Mit einer solchen Erweiterung könnten auch komplexere umgebungserfassende Sensoren für die Verbesserung von FAS verwendet werden und die verfügbaren Informationen im lernenden System deutlich vergrößern.

**Verwendung von Kommunikation:** Wird das Fahrzeug nicht als geschlossenes System betrachtet und können externe Quellen für Informationen einbezogen werden, wäre es möglich, allgemeingültige Informationen zwischen lernenden Systemen auszutauschen. Die dadurch deutlich höhere Anzahl von Messwerten könnte nicht nur das Lernen beschleunigen und verbessern, auch die Änderungserkennung würde davon stark profitieren. So wäre es beispielsweise möglich, eine Änderung in der Fahrzeugumgebung mit einem Schwarm von Fahrzeugen in einer kürzeren Zeitspanne zu erkennen und diese zu allen lernenden Systemen zu propagieren. Zudem würden lernende Systeme die Möglichkeit bekommen, Informationen über bisher nicht befahrene Strecken zu erhalten und schon bei der ersten Befahrung einen Mehrwert bieten.



# Abbildungsverzeichnis

2.1	Übersicht über die grundlegenden Komponenten von Fahrerassistenzsystemen und deren Zusammenspiel. . . . .	7
2.2	Beispielhafte Darstellung eines Kartengraphs mit einzigartigen Identifikationsnummern. (OSM Kartenmaterial) . . . .	16
3.1	Integration eines lernenden Systems in die bestehende IT-Infrastruktur von heutigen Fahrzeugen. . . . .	29
3.2	Detailgrad gelernter Informationen wirkt sich auf die Warnung des adaptiven Kurvenwarners aus. . . . .	35
3.3	Gemeinsame Komponenten der lernenden Systeme aus vorgestellten Arbeiten. . . . .	45
3.4	Komponenten des lernenden Systems die in der vorliegenden Arbeit entwickelt werden sollen. . . . .	48
4.1	Zwei unterschiedliche Wahrscheinlichkeitsverteilungen $A$ und $B$ der Geschwindigkeit $v$ die mit der Kerndichteschätzung $\hat{f}$ geschätzt wurden. (Grafik aus [47]) . . . . .	51
4.2	Beispiele der Heterogenität von Messwertreihen: Verschiebung und variierende Dichte. . . . .	53
4.3	Zwei Fahrten durch eine Kurve mit unterschiedlichen Kurveneingängen ( $S1, S2$ ) und Kurvenausgängen ( $E1, E2$ ) aufgrund der Arbeitsweise der Kurvenerkennung. . . . .	54
4.4	Messwertreihen mit beliebigen Messpositionen 1) können über eine Zwischenwertberechnung 2) auf ein Raster 3) abgebildet werden. . . . .	56
4.5	Das gemeinsame Raster wird dynamisch erweitert, wenn neue Messwertreihen das bisherige überschreiten. . . . .	57
4.6	Interpolationsverfahren im Vergleich für reale Geschwindigkeitsmesswerte: Linear und kubische Splines. . . . .	58
4.7	Selektive Interpolation zur Abbildung auf das Raster in Abhängigkeit zur Dichte der Messwertreihe. . . . .	59
4.8	Messwertreihen-Transformation einer zeitabhängigen Messwertreihe auf ein positionsabhängige Raster. . . . .	61
4.9	Darstellung der gemessenen und der interpolierten Werte einer Sensormesswertreihe. . . . .	62
4.10	Messwertreihen wurden über die Messwertreihen-Transformation auf das Raster abgebildet und können zusammengeführt werden. . . . .	62

4.11	Kurve auf der Birkenallee in Oranienburg nahe der B96 (Quelle: OSM Kartenmaterial). . . . .	64
4.12	GPS Spuren, die bei 5 Fahrten durch die <i>Birkenallee</i> Kurve aufgenommen wurden. . . . .	65
4.13	Ausschnitt der Rasterpunkte nach der Messwertreihen-Transformation der Fahrten durch die Kurve der <i>Birkenallee</i> . . .	66
4.14	Verteilung der Geschwindigkeit an einem Rasterpunkt der Kurve. . . . .	67
4.15	Wahrscheinlichstes Geschwindigkeitsprofil durch die <i>Birkenallee</i> Kurve abgebildet über ein $1m$ Raster. . . . .	68
4.16	Distanz der Warnungen jedes Rasterpunkts in Relation zum Kurveneingang der gelernten Geschwindigkeiten. . . . .	69
4.17	Wahrscheinlichstes Geschwindigkeitsprofil einer künstlichen Kurvenfahrt abgebildet über ein $1m$ Raster. . . . .	69
4.18	Distanz der Warnungen jedes Rasterpunkts in Relation zum Kurveneingang der künstlich erzeugten Kurvenfahrt. . . . .	70
4.19	Komponenten des lernenden Systems der vorliegenden Arbeit. Beiträge des 4. Kapitels sind grün hervorgehoben. . . .	72
5.1	Einteilung der Wahrscheinlichkeitsverteilung in fünf Bereiche über Flächeninhalte. (Grafik aus [47]) . . . . .	76
5.2	Eine Messsequenz mit einer Änderung nach Messung 50. Gezeigt ist 1) der optimale Verlauf und die Mittelwertverläufe von 2) Änderungserkennung mit KS-Test und 3) Änderungserkennung mit t-Test. . . . .	80
5.3	Zwei Fenster bilden die Stichproben für den Zweistichproben-Hypothesentest. . . . .	82
5.4	Die Fensterkonfiguration zur Erkennung von sprungartigen Änderungen über einer Messwertreihe mit sprungartiger Änderung. . . . .	84
5.5	Die Fensterkonfiguration zur Erkennung von sprungartigen Änderungen über einer Messwertreihe mit schleichender Änderung. . . . .	85
5.6	Zwei Fenster bilden die Stichproben für die Erkennung von schleichenden Änderungen. . . . .	85
5.7	Fensterkonfiguration der erweiterten Änderungserkennung über einer Messwertreihe mit schleichender Änderung. . . .	87
5.8	Die Steuerung des Lernprozesses über das <i>t-Test-Vergessen</i> . . . .	88
5.9	Zustandsdiagramm vom <i>t-Test-Vergessen</i> -Algorithmus. . . .	89

5.10	Verlauf der Gewichtung der ungültigen und neuen Informationen nach Erkennung einer Änderung. . . . .	91
5.11	Fahrzeug wird in der realitätsnahen Simulation über ein Lenkrad und Pedalen (nicht abgebildet) gesteuert. . . . .	93
5.12	Kurve der Straße <i>Alt-Lietzow</i> in Berlin die für die Messfahrten in der Simulation befahren wurde (Quelle: OSM Kartenmaterial). . . . .	95
5.13	Bildschirmfoto der Simulation mit ausgewählter Kurve (Straße <i>Alt-Lietzow</i> in Berlin) im Normalzustand. . . . .	95
5.14	Bildschirmfoto der Simulation mit ausgewählter Kurve (Straße <i>Alt-Lietzow</i> in Berlin) und einer Änderung durch Hindernisse auf der Strecke. . . . .	96
5.15	GPS Spuren des Fahrzeugs durch eine Kurve mit Markierungen für erkannte Kurveneingänge. . . . .	96
5.16	Erster Plot: Messsequenz mit 100 simulierten Messwerten und sprungartiger Änderung bei Messung 51. Zweiter Plot: Ergebnisse der Änderungserkennung mit Eskalationsstufe aus [47] und dem <i>t-Test-Vergessen</i> . . . . .	98
5.17	Geschätzte Verteilungen mit den verschiedenen Verfahren zur Änderungserkennung nach Verarbeitung von Messung 60 und 100 der Messsequenz. . . . .	99
5.18	Vergleich der verschiedenen Verfahren zur Änderungserkennung mit künstlichen erzeugten Messsequenzen. . . . .	102
5.19	Vergleich der verschiedenen Verfahren zur Änderungserkennung mit simulierten Messsequenzen. . . . .	102
5.20	Vergleich der verschiedenen Verfahren zur Änderungserkennung mit simulierten Messsequenzen bei verdoppelter, optimaler Bandbreite der Kerndichteschätzung. . . . .	103
5.21	Vergleich des <i>t-Test-Vergessen</i> mit einem und zwei Hypothesentests bei der Erkennung einer schleichenden Änderung ( $-20\frac{km}{h}$ von Messung 40 bis 80). . . . .	105
5.22	Vergleich des <i>t-Test-Vergessen</i> mit einem und zwei Hypothesentests bei der Erkennung einer schleichenden Änderung ( $-20\frac{km}{h}$ von Messung 40 bis 100). . . . .	105
5.23	Komponenten des lernenden Systems der vorliegenden Arbeit. Die Beiträge des 5. Kapitels sind grün hervorgehoben.	107
6.1	Die Relevanz der Datenpunkte $d_1$ bis $d_4$ kann durch das Straßennetz eindeutig bestimmt werden. . . . .	111

6.2	Reihenfolge der Schritte beim Speichern und Laden von Informationen der beiden Adressierungsverfahren. . . . .	112
6.3	Beispiel für die vorgeladenen Kanten des dynamischen elektronischen Horizonts in zwei verschiedenen Knoten-Tiefen. . . . .	113
6.4	Geschwindigkeitsvergleich der Datenbanken Oracle Berkeley DB Java Edition 5.0.103 und PostgreSQL 9.0.4 beim Speichern mit zufälligen und sequenziellen Schlüsseln. . . . .	116
6.5	Geschwindigkeitsvergleich der Datenbanken Oracle Berkeley DB Java Edition 5.0.103 und PostgreSQL 9.0.4 beim Laden mit zufälligen und sequenziellen Schlüsseln. . . . .	118
6.6	Statechart der Kurvenerkennung für den adaptiven Kurvenwarner. . . . .	121
6.7	Verlauf der Funktionen $M_1$ , $F(E, 10, 10)$ und $h_F(x)$ mit $E$ beziehungsweise $x$ im Intervall $[0, 10]$ . . . . .	127
6.8	Verlauf der Metriken $M_2$ , $M_3$ und $M_4$ mit $E$ im Intervall $[0, 300]$ . . . . .	129
6.9	Häufigkeitsverteilung der vier Eingabeparameter der 500 simulierten Datensätze als Histogramm. . . . .	131
6.10	Histogramm der Metriken $M_1$ , $M_2$ , $M_3$ , $M_4$ und $P$ der 500 simulierten Datensätze. . . . .	134
6.11	Komponenten des lernenden Systems der vorliegenden Arbeit. Die Beiträge des 6. Kapitels sind grün hervorgehoben. . . . .	135
6.12	Schematische Darstellung des Versuchsaufbaus. . . . .	136
6.13	Bild vom Versuchsträger mit HMI-Tablet (angebracht an die Mittelkonsole) und Laptop mit CAN-Anbindung auf dem das lernende System und der adaptive Kurvenwarner ausgeführt wird. . . . .	137
6.14	HMI des adaptiven Kurvenwarners bei der Anfahrt an eine gelernte Kurve, die markiert wurde. . . . .	139
6.15	HMI des adaptiven Kurvenwarners bei der Annäherung an eine gelernte Kurve mit Infosymbol in der oberen rechten Ecke. . . . .	139
6.16	HMI des adaptiven Kurvenwarners. Ersten Stufe der Warnung: Gelernte Geschwindigkeit und Kurvenrichtung am rechten Rand. . . . .	140
6.17	HMI des adaptiven Kurvenwarners. Zweiten Stufe der Warnung: Gelernte Geschwindigkeit und Kurvenrichtung als Vollbild. . . . .	140

# Abkürzungsverzeichnis

<b>ABC</b>	Active Body Control
<b>ABS</b>	Antiblockiersystem
<b>ACC</b>	Adaptive Cruise Control
<b>ASR</b>	Antriebsschlupfregelung
<b>BAS</b>	Bremsassistent
<b>CAN</b>	Controller Area Network
<b>DBM</b>	Datenbankmodell
<b>DBS</b>	Datenbanksystem
<b>DGPS</b>	Differential Global Positioning System
<b>ESP</b>	Elektronisches Stabilitätsprogramm
<b>FAS</b>	Fahrerassistenzsysteme
<b>GPS</b>	Global Positioning System
<b>HMI</b>	Human Machine Interface
<b>LiDAR</b>	Light Detection And Ranging
<b>LIN</b>	Local Interconnect Network
<b>MOST</b>	Media Oriented Systems Transport
<b>OSM</b>	OpenStreetMaps
<b>RADAR</b>	Radio Detection and Ranging



# Literaturverzeichnis

---

- [1] Ackermann, F. *Abstandsregelung mit Radar*. In: Spektrum der Wissenschaft, pp. 24–34, 1980.
- [2] ADAC. *Test Spurwechsel-Assistenten*. [https://www.adac.de/\\_mmm/pdf/Spurwechsel-Assistenten\\_86KB\\_23685.pdf](https://www.adac.de/_mmm/pdf/Spurwechsel-Assistenten_86KB_23685.pdf), 2008.
- [3] ADAC. *Grundlagen Assistenzsysteme*. <https://www.adac.de/infotestrat/technik-und-zubehoer/fahrerassistenzsysteme/grundlagen/default.aspx>, 2017.
- [4] Albrecht, F. *Fahrerassistenzsysteme und rechtliche Rahmenbedingungen*. In: Verkehrsdienst 2006, pp. 143–150, 2006.
- [5] Bär, T.; Kohlhaas, R.; Zöllner, J. M.; Scholl, K.-U. *Anticipatory driving assistance for energy efficient driving*. In: IEEE Forum on Integrated and Sustainable Transportation System (FISTS), pp. 1–6, 2011.
- [6] Basseville, M.; Nikiforov, I. V. et al. *Detection of abrupt changes: theory and application*, Vol. 104. Prentice Hall Englewood Cliffs, 1993.
- [7] Bíl, M.; Andrášik, R.; Janoška, Z. *Identification of hazardous road locations of traffic accidents by means of kernel density estimation and cluster significance evaluation*. Accident Analysis & Prevention, Vol. 55, pp. 265–273, 2013.
- [8] Bonett, D. G.; Price, R. M. *Statistical inference for a linear function of medians: confidence intervals, hypothesis testing, and sample size requirements*. Psychological methods, Vol. 7(3), pp. 370, 2002.
- [9] Brown, I. D. *A review of the 'looked but failed to see' accident causation factor*. In: Behavioural research in road safety: Eleventh Seminar, 2002.

- [10] Bundesgesetzblatt Teil II. *Gesetz zu den Übereinkommen vom 08. November 1968 über den Straßenverkehr und über Straßenverkehrszeichen, zu den europäischen Zusatzübereinkommen vom 1. Mai 1971 zu diesen Übereinkommen sowie zum Protokoll vom 1. März 1973 über Straßenmarkierungen.* Bonn, 1977. Bundesanzeiger Verlag GmbH. URL: [www.bgbl.de/Xaver/start.xav](http://www.bgbl.de/Xaver/start.xav).
- [11] Büning, H.; Trenkler, G. *Nichtparametrische statistische Methoden.* Walter de Gruyter, 1994.
- [12] Burckhardt, M. *Reaktionszeiten bei Notbremsvorgängen.* Verlag TÜV Rheinland, 1985.
- [13] Carlsson, A. *A system for the provision and management of route characteristic information to facilitate predictive driving strategies,* Vol. 40. Expert Verlag, 2008.
- [14] CIA. *The World Factbook - Roadways Country Comparison.* <https://www.cia.gov/library/publications/the-world-factbook/rankorder/2085rank.html?countryName=Germany&countryCode=gm&regionCode=eur>, 2010.
- [15] Conchillo, Á.; Recarte, M. Á.; Nunes, L.; Ruiz, T. *Comparing speed estimations from a moving vehicle in different traffic scenarios: absence versus presence of traffic flow.* The Spanish journal of psychology, Vol. 9(01), pp. 32–37, 2006.
- [16] Daniel, W. *Kolmogorov–Smirnov one-sample test.* In: Applied Non-parametric Statistics, pp. 319–330, 1990.
- [17] Deutschle, S. *Wer fährt?–Der Fahrer oder das System.* SVR Straßenverkehrsrecht, Vol. 6, pp. 201–206, 2005.
- [18] Donner, E.; Winkle, T.; Walz, R.; Schwarz, J. *RESPONSE 3–Code of Practice für die Entwicklung, Validierung und Markteinführung von Fahrerassistenzsystemen.* In: VDA. Technischer Kongress, Vol. 28, 29 p, 2007.
- [19] Dormann, C. F. *Parametrische Statistik: Verteilungen, maximum likelihood und GLM in R.* Springer-Verlag, 2013.
- [20] EUREKA. *PROMETHEUS - Programme for a european traffic system with highest efficiency and unprecedented safety.* <http://www.eurekanetwork.org/project/id/45>, 1995.

- [21] Fraedrich, E.; Lenz, B. *Autonomes Fahren-Mobilität und Auto in der Welt von morgen. Ausblick zur Akzeptanz des autonomen Fahrens im Projekt „Villa Ladenburg“ der Daimler und Benz Stiftung.* Technikfolgenabschätzung-Theorie und Praxis, Vol. 23(1), pp. 46–53, 2014.
- [22] Fraunhofer-Institut für eingebettete Systeme und Kommunikationstechnik ESK. *Zukünftige Softwarearchitekturen für Fahrzeuge.* [http://www.esk.fraunhofer.de/content/dam/esk/dokumente/PDB\\_adaptives\\_Bordnetz\\_dt\\_web.pdf](http://www.esk.fraunhofer.de/content/dam/esk/dokumente/PDB_adaptives_Bordnetz_dt_web.pdf), 2012.
- [23] Fuller, R. *Towards a general theory of driver behaviour.* Accident Analysis & Prevention, Vol. 37(3), pp. 461–472, 2005.
- [24] Ghosh, A. K.; Chaudhuri, P.; Sengupta, D. *Classification Using Kernel Density Estimates.* Technometrics, Vol. 48(1), pp. 120–132, 2006.
- [25] GIGA APPLE. *Bosch zeigt iPhone-Navi mit Kurvenwarner.* <http://www.giga.de/apps/app-store/news/bosch-zeigt-iphone-navi-mit-kurvenwarner/>, 2010.
- [26] Glaser, S.; Nouveliere, L.; Lusetti, B. *Speed limitation based on an advanced curve warning system.* In: 2007 IEEE Intelligent Vehicles Symposium, pp. 686–691, 2007.
- [27] Golem Media GmbH. *Wiener Übereinkommen erlaubt vollautomatisierte Autos.* <http://www.golem.de/news/autobahnpilot-wiener-uebereinkommen-erlaubt-vollautomatisierte-autos-1603-119922.html>, 2016.
- [28] Hallmark, S. L.; Hawkins, N.; Smadi, O. *Speed reduction impact of dynamic speed feedback signs on high crash curves.* In: 15th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 1115–1120, 2012.
- [29] Hashimoto, S.; Yoshiki, S.; Saeki, R.; Mimura, Y.; Ando, R.; Nanba, S. *Development and application of traffic accident density estimation models using kernel density estimation.* Journal of traffic and transportation engineering (English edition), Vol. 3(3), pp. 262–270, 2016.

- [30] Hazewinkel, M. *Encyclopedia of Mathematics*. Springer, Berlin, Germany, 2001.
- [31] Heinrichs, R. *Slowly Impacting Change Points In Measurements: An Improved Hypothesis Test Triggered Forgetting for Automotive Learning Systems*. In: Proceedings of the 4th Commercial Vehicle Technology Symposium (CVT 2016), pp. 355–364, 2016.
- [32] Heinrichs, R.; Fritzsche, M.; Radusch, I. *Improved automotive self learning system using hypothesis test triggered forgetting to adapt to change points*. In: IEEE Intelligent Vehicles Symposium Proceedings, pp. 176–182, 2014.
- [33] Hoppe, M.; Zobel, R.; Schlag, B. *Identifikation von Einflussgrößen auf Verkehrsunfälle als Grundlage für die Beurteilung von Fahrerassistenzsystemen am Beispiel von Kreuzungsunfällen*. In: Tagungsband zur 4. VDI-Tagung Fahrer im 21. Jahrhundert, 2007.
- [34] Hwang, S.-M.; Lee, H.-J.; Hong, K.-S.; Kang, B.-S.; Hwang, G.-Y. *New development of combined permanent-magnet type microspeakers used for cellular phones*. IEEE Transactions on Magnetics, Vol. 41(5), pp. 2000–2003, 2005.
- [35] Isermann, R. *Fahrdynamik-Regelung: Modellbildung, Fahrerassistenzsysteme, Mechatronik*. Springer-Verlag, 2007.
- [36] Jaensch, M. *Modulorientiertes Produktlinien Engineering für den modellbasierten Elektrik/Elektronik-Architekturentwurf*. Dissertation, Karlsruher Institut für Technologie (KIT), 2012.
- [37] Keßler, M.; Mangin, B. *Nutzerorientierte Auslegung von teilautomatisierten Einparkassistenzsystemen*. In: Tagungsband zur 4. VDI-Tagung Fahrer im 21. Jahrhundert, 2007.
- [38] Kopf, M. *Was nützt es dem Fahrer, wenn Fahrerinformations-und-assistenzsysteme etwas über ihn wissen?* In: Fahrerassistenzsysteme mit maschineller Wahrnehmung, pp. 117–139. Springer, 2005.
- [39] Kraftfahrt-Bundesamt. *Fahrzeugzulassungen (FZ) - Besitzumschreibungen und Außerbetriebsetzungen von Kraftfahrzeugen und Kraftfahrzeuganhängern nach Fahrzeugalter*. Vol. 16, 2013.

- [40] Kraiss, K.-F. *Benutzergerechte Automatisierung–Grundlagen und Realisierungskonzepte*. at-Automatisierungstechnik, Vol. 46(10), pp. 457–467, 1998.
- [41] Krebs, H.; Damianoff, N. *Geschwindigkeitsverhalten bei Gefahrenzeichen, in Kurven und an Stellen mit Geschwindigkeitsbeschränkung*. Forschung Straßenbau und Straßenverkehrstechnik. Bundesminister für Verkehr, Abt. Straßenbau, 1983.
- [42] Kulmala, R.; Mäurerer, H. *Final Report and Recommendations of the Implementation Road Map Working Group*, 2005. eSafety Forum, Brussels.
- [43] Lenovo. *Detailed specifications - ThinkPad T410s, T410si*. <https://support.lenovo.com/de/en/documents/migr-75308>, 2016.
- [44] Li, K.; Tan, H.-S.; Misener, J. A.; Hedrick, J. K. *Digital map as a virtual sensor–dynamic road curve reconstruction for a curve speed assistant*. Vehicle System Dynamics, Vol. 46(12), pp. 1141–1158, 2008.
- [45] Liu, S.; Yamada, M.; Collier, N.; Sugiyama, M. *Change-point detection in time-series data by relative density-ratio estimation*. Neural Networks, Vol. 43, pp. 72–83, 2013.
- [46] Lowbridge, P. *Low-cost mm-wave radar for cruise control*. III-Vs Review, Vol. 8(5), pp. 44–48, 1995.
- [47] Mauk, T. *Selbstlernende, zuverlässigkeitsorientierte Prädiktion energetisch relevanter Größen im Kraftfahrzeug*. Expert Verlag, 2011.
- [48] Mercedes-Benz. *BAS PLUS mit Kreuzungs-Assistent*. [http://m.mercedes-benz.de/de\\_DE/bas\\_plus\\_cross\\_traffic\\_assist/detail.html](http://m.mercedes-benz.de/de_DE/bas_plus_cross_traffic_assist/detail.html), 2013.
- [49] Mercedes-Benz. *PRE-SAFE® Sound: Rosa Rauschen, bevor es Krach macht*. <https://www.mercedes-benz.com/de/mercedes-benz/next/vernetzung/pre-safe-sound-rosa-rauschen-bevor-es-krach-macht/>, 2015.
- [50] Milošević, S.; Milić, J. *Speed perception in road curves*. Journal of Safety Research, Vol. 21(1), pp. 19–23, 1990.

- [51] Morosi, S.; Del Re, E.; Martinelli, A. *P2P cooperative GPS positioning with fine/coarse time assistance*. In: International Conference on Localization and GNSS (ICL-GNSS), pp. 1–6, 2013.
- [52] Murphy, K. P. *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [53] National Highway Traffic Safety Administration, United States Code of Federal Regulations. *Electronic Stability Control Systems*. Federal Motor Vehicle Safety Standard 126, Vol. 49, pp. 457–467, 2007.
- [54] Nedden, M. *Nichtparametrische Kerndichteschätzung*, pp. 115–132. Vieweg+Teubner Verlag, Wiesbaden, 2012.
- [55] Netzwelt. *Navigations-App von Bosch: Kurvenwarner sorgt für mehr Sicherheit*. <http://www.netzwelt.de/news/85026-navigations-app-bosch-kurvenwarner-sorgt-mehr-sicherheit.html>, 2010.
- [56] Norton, H. N. *Handbook of transducers*. Prentice Hall, 1989.
- [57] Nürnberger, G. *Approximation by spline functions*, Vol. 1. Springer Berlin, 1989.
- [58] Ogle, J.; Guensler, R.; Bachman, W.; Koutsak, M.; Wolf, J. *Accuracy of global positioning system for determining driver performance parameters*. Transportation Research Record: Journal of the Transportation Research Board, Vol. 1818(1), pp. 12–24, 2002.
- [59] OpenStreetMap. *Core elements in the OpenStreetMap data model - Way*. <http://wiki.openstreetmap.org/wiki/Way>, 2009.
- [60] OpenStreetMap. *Core elements in the OpenStreetMap data model - Node*. <http://wiki.openstreetmap.org/wiki/Node>, 2016.
- [61] OpenStreetMap. *OpenStreetMap's conceptual data model of the physical world - Elements*. <http://wiki.openstreetmap.org/wiki/Elements>, 2016.
- [62] Otto, H.-U. et al. *The ActMap Approach-Specifications of Incremental Map Updates for Advanced In-Vehicle Applications*. In: Tagungsband „ITS in Europe Congress“ in Hannover (D), 2005.

- [63] Otto, H.-U.; Flament, M.; Landwehr, M. *FeedMAP – Incremental Updating Meets Map-Based Probes*. In: World Congress on Intelligent Transport Systems, London, 2006.
- [64] Parker, R. *Permanent Magnets in Audio Devices*. IRE Transactions on Component Parts, Vol. 5(1), pp. 32–37, 1958.
- [65] Parkinson, B. W.; Enge, P. K. *Differential GPS*. Global Positioning System: Theory and applications., Vol. 2, pp. 3–50, 1996.
- [66] Paulus, C. L.; Stout, R. K. *Automatic steering and control mechanism*, 1943. US Patent 2,317,400.
- [67] PEAK-System Technik GmbH. *PCAN-USB, CAN-Interface für USB*. <http://www.peak-system.com/PCAN-USB.199.0.html>, 2016.
- [68] Polunchenko, A. S.; Tartakovsky, A. G. *State-of-the-art in sequential change-point detection*. Methodology and computing in applied probability, Vol. 14(3), pp. 649–684, 2012.
- [69] Reymond, G.; Kemeny, A.; Droulez, J.; Berthoz, A. *Role of lateral acceleration in curve driving: Driver model and experiments on a real vehicle and a driving simulator*. Human Factors: The Journal of the Human Factors and Ergonomics Society, Vol. 43(3), pp. 483–495, 2001.
- [70] Rössler, B.; Obojski, M.; Stanzel, M.; Ehmanns, D.; Kosch, T.; Hopstock, M.; Shooter, C.; Heenan, A.; Woolley, P.; Benson, C. et al. *D40.4 requirements for intersection safety applications*. INTERSAFE Deliverable, Vol. 258, 2005.
- [71] Salas, G. *Highway Coding for Route Destination and Position Coding*. In: Highway Research Board, Publ, 1968.
- [72] Salmon, P. M.; Regan, M. A.; Johnston, I. *Human error and road transport: phase one: a framework for an error tolerant road transport system*. Number 256. 2005.
- [73] Schmidtke, H.; Bernotat, R. *Ergonomie*. Hanser München, 1993.
- [74] Schönhage, A. *Approximationstheorie*. Walter de Gruyter, 1971.

- [75] Schraut, M. P. *Umgebungserfassung auf Basis lernender digitaler Karten zur vorausschauenden Konditionierung von Fahrerassistenzsystemen*. Dissertation, Technische Universität München, Universitätsbibliothek, 2000.
- [76] Schweizerische Bundeskanzlei. *AS 2016 1019, Änderungen am Übereinkommen vom 8. November 1968 über den Strassenverkehr*. <https://www.admin.ch/opc/de/official-compilation/2016/1019.pdf>, 2016.
- [77] Schweizerische Bundeskanzlei. *Übereinkommen vom 8. November 1968 über den Strassenverkehr*. <https://www.admin.ch/opc/de/classified-compilation/19680244/index.html>, 2016.
- [78] Seashore, J. E.; Miller, W. J. *Portable automobile diagnostic tool*, 1999. US Patent 5,916,286.
- [79] Siebert, G. *The impact of expected satellite availability on global positioning system (GPS) accuracy*. In: Fourth International Conference on Satellite Systems for Mobile Communications and Navigation, pp. 178–180, 1988.
- [80] Siegel, S.; Rennert, U.; Schüle, W. *Nichtparametrische statistische methoden*. Fachbuchhandlung für Psychologie Frankfurt, 1976.
- [81] Silverman, B. W. *Density estimation for statistics and data analysis*, Vol. 26. CRC press, 1986.
- [82] Statistisches Bundesamt. *Verkehrsunfälle 2010*. Vol. 49(0), pp. 1–323, 2011.
- [83] STR-Automatik GmbH. *Funktion eines Automatikgetriebes*. <https://www.str-automatik.de/automatikgetriebe/funktion/>, 2017.
- [84] Straßenweb.de. *Wie viele Straßen gibt es in Deutschland?* <https://www.strassenweb.de/deutschland/strassenstatistik.html>, 2017.
- [85] Streubel, T.; Hoffmann, K. H. *Fahrverhaltenanalyse an Kreuzungen auf Basis von Fahrdaten*. In: GMM-Fachbericht Automotive meets Electronics, 2014.

- [86] Tartakovsky, A.; Nikiforov, I.; Basseville, M. *Sequential analysis: Hypothesis testing and changepoint detection*. CRC Press, 2014.
- [87] Thadewald, T. *Uni-und bivariate Dichteschätzung: optimale Bandbreitenbestimmung und Anwendungen*. Dissertation, Freie Universität Berlin, 1998.
- [88] Thomas, B.; Löwenau, J.; Durekovic, S.; Otto, H.-U. *The ActMap-FeedMap framework a basis for in-vehicle ADAS application improvement*. In: Proceedings of the 2008 IEEE Intelligent Vehicles Symposium (IV'08), Netherlands, 2008.
- [89] Thrun, S.; Pratt, L. *Learning to learn*. Springer Science & Business Media, 2012.
- [90] Timpe, K.-P.; Jürgensohn, T.; Kolrep, H. *Mensch-Maschine-Systemtechnik. Konzepte, Modellierung, Gestaltung*. Evaluation, Vol. 2, 2002.
- [91] Toyota. *History of Safety Technology*. [http://www.toyota.com.cn/technology/download/toyotas\\_history\\_of\\_safety\\_technology.pdf](http://www.toyota.com.cn/technology/download/toyotas_history_of_safety_technology.pdf), 2009.
- [92] Toyota. *Toyota Adds to Pre-crash Safety Technologies*. <http://www.toyota.co.jp/en/news/09/0226.html>, 2009.
- [93] Uduwaragoda, E. R. I. A. C. M.; Perera, A. S.; Dias, S. A. D. *Generating lane level road data from vehicle trajectories using Kernel Density Estimation*. In: 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), pp. 384–391, 2013.
- [94] Uhler, W.; Knoll, P. M. *Surround Sensors-Enablers for Predictive Safety Systems*. Technical report, SAE Technical Paper, 2006.
- [95] Venkatraman, K.; Amutha, B.; Karthick, K.; Sankar, S. *A hybrid method for improving GPS accuracy for land vehicle navigation system*. In: International Conference on Emerging Trends in Robotics and Communication Technologies (INTERACT), pp. 74–79, 2010.
- [96] Vollrath, M.; Briest, S.; Schießl, C.; Drewes, J.; Becker, U. *Ableitung von Anforderungen an Fahrerassistenzsysteme aus Sicht der Verkehrssicherheit*. Wirtschaftsverlag NW, Verlag für neue Wissenschaft GmbH, 2006.

- [97] Wang, E.; Zhao, W.; Cai, M. *Research on improving accuracy of GPS positioning based on particle filter*. In: 8th IEEE Conference on Industrial Electronics and Applications (ICIEA), pp. 1167–1171, 2013.
- [98] Watanabe, T.; Kishimoto, N.; Hayafune, K.; Yamada, K.; Maede, N. *Development of an intelligent cruise control system*. In: Steps Forward. Intelligent Transport Systems World Congress, 1995.
- [99] Weiße, J. *Beitrag zur Entwicklung eines optimierten Bremsassistenten*. Ergonomia-Verlag, 2003.
- [100] Williams, B. *Intelligent transport systems standards*. Artech House, 2008.
- [101] Winner, H.; Hakuli, S.; Wolf, G. *Handbuch Fahrerassistenzsysteme*. Vieweg+ Teubner, Wiesbaden, Vol. 5, 2009.
- [102] World Health Organization. *Global status report on road safety 2013: supporting a decade of action*. 2013.
- [103] Xie, Z.; Yan, J. *Detecting traffic accident clusters with network kernel density estimation and local spatial statistics: an integrated approach*. Journal of transport geography, Vol. 31, pp. 64–71, 2013.
- [104] Zhou, J.; Wang, K.; Tang, S.; Wang, F.-Y. *Trajectory learning and analysis based on kernel density estimation*. In: 12th International IEEE Conference on Intelligent Transportation Systems (ITSC'09), pp. 1–6, 2009.
- [105] Zimmermann, W.; Schmidgall, R. *Bussysteme in der fahrzeugtechnik: protokolle, standards und softwarearchitektur*. Springer-Verlag, 2014.