

# **Eine Basis-Systemarchitektur zur Sensordatenfusion von Umfeldsensoren für Fahrerassistenzsysteme**

Dem Fachbereich Maschinenbau  
an der Technischen Universität Darmstadt  
zur  
Erlangung des Grades eines Doktor-Ingenieurs (Dr.-Ing.)  
eingereichte

D i s s e r t a t i o n

vorgelegt von

**Dipl.-Wirtsch.-Ing. Michael Darms**

aus Berlin-Wilmersdorf

Berichterstatter: Prof. Dr. rer. nat. H. Winner

Mitberichterstatter: Prof. Dr. B. Schiele

Tag der Einreichung: 27.04.2007

Tag der mündlichen Prüfung: 17.07.2007

Darmstadt 2007

D17

## Vorwort

Diese Arbeit entstand während und im Anschluss an meine Zeit als wissenschaftlicher Mitarbeiter am Fachgebiet Fahrzeugtechnik der Technischen Universität Darmstadt (FZD).

Danken möchte ich meinem Doktorvater Prof. Dr. rer. nat. Hermann Winner für die hervorragende Betreuung der Arbeit, die große Bereitschaft zu tiefgehenden fachlichen Diskussionen und für die mir gewährten Freiheiten in der Forschung und beim Anfertigen der Arbeit.

Herrn Prof. em. Dr.-Ing. Bert Breuer, dem ehemaligen Leiter des Fachgebiets, danke ich für seine kritischen Fragen und Hinweise in Doktorandenseminaren.

Herrn Prof. Dr. Bernt Schiele möchte ich für die Übernahme des Korreferats danken.

Die Arbeit baut auf dem interdisziplinären Projekt PRORETA auf, welches in Kooperation mit dem Institut für Automatisierungstechnik des Fachbereichs Elektrotechnik und dem Fachgebiet Arbeitswissenschaft des Fachbereichs Maschinenbaus durchgeführt und von der Continental Teves AG & Co. oHG gefördert wurde.

Für anregende Diskussionen möchte ich dem Projektleiter auf Seiten der Universität Prof. Dr.-Ing. Rolf Isermann sowie den Fachgebietsleitern Prof. em. Dr.-Ing. Kurt Landau und Prof. Dr.-Ing. Ralph Bruder danken. Für die erfolgreiche und äußerst angenehme Zusammenarbeit danke ich den PRORETA Team-Mitgliedern Eva Bender, Matthias Schorn und Ulrich Stählin.

Für die Forschungsunterstützung und die vertrauensvolle Zusammenarbeit möchte ich allen am Projekt beteiligten Mitarbeitern des Industriepartners, stellvertretend dem Projektleiter Dr. Peter Rieth, danken. Für ergiebige fachliche Diskussionen möchte ich mich insbesondere bei der A.D.C. GmbH, stellvertretend Jürgen Brugger und Stefan Heinrich, bedanken.

Für die Unterstützung beim Aufbau des Versuchsträgers und der Versuchseinrichtungen möchte ich den beteiligten mechanischen und elektrotechnischen Werkstätten der Universität, stellvertretend Robert Korndörfer und Harald Bathke von FZD, danken.

Allen Mitarbeitern des Fachgebiets, einschließlich des Sekretariats, möchte ich für die äußerst angenehme, konstruktive und von Selbständigkeit und Bereitschaft zur Verantwortungsübernahme geprägte Arbeitsatmosphäre danken. Ich wünsche mir, dass die entstandenen Freundschaften dauerhaft anhalten werden.

Besonderer Dank gilt zudem den von mir betreuten Studenten, die durch ihre Mitarbeit in Form von Studien- und Diplomarbeiten bzw. als wissenschaftliche Hilfskräfte einen großen Beitrag geleistet haben.

Schlussendlich hat meine liebe Frau Sabine Schmitt mit ihrer Geduld und Rücksicht sowie durch ihre stetige Unterstützung einen großen Anteil am Gelingen dieser Arbeit. Dafür bedanke ich mich von Herzen.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation zur Arbeit . . . . .	1
1.2	Begriffsdefinitionen zur Sensordatenfusion . . . . .	2
1.3	Architektur . . . . .	3
1.3.1	Begriffsdefinitionen . . . . .	3
1.3.2	Zweck einer Architektur . . . . .	4
1.3.3	Entwicklungsmethoden zum Entwurf einer Architektur . . . . .	5
1.4	Ziele der Arbeit . . . . .	6
1.5	Methodik und Aufbau der Arbeit . . . . .	7
<b>2</b>	<b>Architekturen zur Umfelderkennung</b>	<b>9</b>
2.1	Hauptkomponenten eines Fahrerassistenzsystems mit Umfelderkennung . . . . .	9
2.2	Architekturprinzipien und Architekturmuster . . . . .	13
2.2.1	Allgemeine Architekturprinzipien . . . . .	13
2.2.2	Architekturmuster zur Sensordatenfusion von Umfoldsensoren . . . . .	15
2.3	Grundannahmen für eine Architektur zur Sensordatenfusion für Fahrerassis- tenzsysteme . . . . .	22
<b>3</b>	<b>Datenfilterung für unterschiedliche Fahrerassistenzsysteme</b>	<b>24</b>
3.1	Ziele der Datenfilterung . . . . .	24
3.2	Auslegung eines Kalmanfilters mit festen Parametern für eine adaptive Fahr- geschwindigkeitsregelung . . . . .	25
3.3	Auslegung eines Kalmanfilters mit festen Parametern für ein System zur Un- fallvermeidung . . . . .	31
3.4	Auslegung eines Kalmanfilters mit festen Parametern für eine adaptive Fahr- geschwindigkeitsregelung und ein System zur Unfallvermeidung . . . . .	36
3.5	Auslegung eines adaptiven Kalmanfilters für eine adaptive Fahrgeschwindig- keitsregelung und ein System zur Unfallvermeidung . . . . .	38
3.6	Fazit zur Datenfilterung . . . . .	42
<b>4</b>	<b>Datenassoziation für unterschiedliche Fahrerassistenzsysteme</b>	<b>44</b>
4.1	Ziele der Datenassoziation . . . . .	44
4.2	Vorgehen bei der Datenassoziation . . . . .	45

4.3	Zusammenhang der Datenassoziation mit der Merkmalsextraktion und Objekthypothesenbildung . . . . .	48
4.4	Zusammenhang der Datenassoziation mit der Datenfilterung . . . . .	51
4.5	Fazit zur Datenassoziation . . . . .	59
<b>5</b>	<b>Basis-Systemarchitektur zur Sensordatenfusion für Fahrerassistenzsysteme</b>	<b>62</b>
5.1	Überblick zur Architektur . . . . .	62
5.2	Sensorebene . . . . .	65
5.3	Fusionsebene . . . . .	66
5.4	Robustheit bezüglich der Grundannahmen . . . . .	68
5.5	Zusammenfassung der Eigenschaften . . . . .	69
<b>6</b>	<b>Praktische Umsetzung der Basis-Systemarchitektur</b>	<b>71</b>
6.1	Versuchsträger und Umfoldsensoren . . . . .	71
6.2	Umsetzung der Fusionsebene . . . . .	73
6.2.1	L2-Darstellung . . . . .	73
6.2.2	Datenfilterung . . . . .	75
6.2.3	Klassifikation . . . . .	77
6.2.4	Objekthypothesenmanagement . . . . .	78
6.2.5	Anforderungsmanagement . . . . .	79
6.3	Umsetzung der Sensorebene für den Laserscanner . . . . .	80
6.3.1	L1-Darstellung . . . . .	80
6.3.2	Merkmalsextraktion . . . . .	80
6.3.3	Assoziation . . . . .	81
6.4	Umsetzung der Sensorebene für den Videosensor . . . . .	82
6.4.1	L1-Darstellung . . . . .	82
6.4.2	Merkmalsextraktion & Klassifikation . . . . .	82
6.4.3	Assoziation . . . . .	83
6.5	Verteilungssicht . . . . .	84
6.6	Laufzeitsicht . . . . .	85
6.7	Vorteile der Fusion . . . . .	86
6.8	Fazit zur Umsetzung . . . . .	89
<b>7</b>	<b>Fazit &amp; Ausblick</b>	<b>90</b>
<b>A</b>	<b>Filter</b>	<b>92</b>
A.1	Discrete White Noise Acceleration Modell (DWNA) . . . . .	92
A.2	Discrete Wiener Process Acceleration Modell (DWPA) . . . . .	92
A.3	Übertragungsfunktion des $\alpha\beta\gamma$ -Filters . . . . .	92
<b>B</b>	<b>Berechnungen zur adaptiven Fahrgeschwindigkeitsregelung</b>	<b>94</b>
B.1	Modellierung des Fahrzeugverhaltens . . . . .	94

---

B.2 Übertragungsfunktion . . . . .	94
<b>C Berechnungen zur Time To Collision</b>	<b>96</b>
<b>D Projekt Proreta</b>	<b>97</b>
D.1 Versuchsträger . . . . .	97
D.2 Projektteam . . . . .	98
<b>Literaturverzeichnis</b>	<b>99</b>

# Formelzeichen und Abkürzungen

## Abkürzungen

ACC	Adaptive Cruise Control (S. 25)
allg.	allgemein
Asc	Assoziation (Association)
CAN	Controller Area Network
DWNA	Discrete White Noise Acceleration (S. 40)
DWPA	Discrete Wiener Process Acceleration (S. 27)
FAS	Fahrerassistenzsystem (S. 6)
FZG	Fahrzeug
ggf.	gegebenenfalls
IMM	Interacting Multiple Model (S. 39)
JDL	Joint Directors of Laboratories (S. 2)
JPDA	Joint Probabilistic Data Association Filter (S. 52)
KF	Kalmanfilter
MH	Merkmalshypothese
OH	Objekthypothese
PC	Personal Computer
sensorspez.	sensorspezifisch
SH	Szenehypothese
u.a.	unter anderem
z.T.	zum Teil

## Lateinische Buchstaben

$A_{\text{ego}}$	z-Transformierte von $a_{\text{ego}}$ (S. 28)
$a_{\text{ego}}$	Beschleunigung des betrachteten Fahrzeugs [ $\text{m/s}^2$ ] (S. 26)
$A_{\text{fw}}$	z-Transformierte von $a_{\text{fw}}$ (S. 28)
$a_{\text{fw}}$	Beschleunigung des vorausfahrenden Fahrzeugs [ $\text{m/s}^2$ ] (S. 28)
$a_{\text{rel}}$	Relative Beschleunigung [ $\text{m/s}^2$ ] (S. 27)
$\hat{a}_{\text{rel}}$	Geschätzte relative Beschleunigung [ $\text{m/s}^2$ ] (S. 27)
$a_{\text{rel},x}$	Relativbeschleunigung in x-Richtung [ $\text{m/s}^2$ ] (S. 75)
$a_{\text{rel},y}$	Relativbeschleunigung in y-Richtung [ $\text{m/s}^2$ ] (S. 75)

$a_{\text{set}}$	Soll-Beschleunigung des betrachteten Fahrzeugs [m/s <sup>2</sup> ] (S. 26)
$d$	Abstand [m] (S. 87)
$d_0$	Abstand zum Zeitpunkt $t_0$ [m] (S. 33)
$d_{\text{rel}}$	Relativabstand [m] (S. 26)
$\hat{d}_{\text{rel}}$	Geschätzter Relativabstand [m] (S. 27)
$d_{\text{set}}$	Folgeabstand [m] (S. 26)
$e$	Integral der quadrierten Abweichung der Beschleunigung $a_{\text{ego}}$ [m <sup>2</sup> /s <sup>3</sup> ] (S. 30)
$f_a$	Wahrscheinlichkeitsdichte der Beschleunigung $a_{\text{fw}}$ (S. 34)
$F_i$	Filter auf Fusionsebene (S. 62)
$F(k)$	Zustandsübergangsmatrix (S. 75)
$f_t$	Wahrscheinlichkeitsdichte des Fehlers $\Delta \hat{t}_{\text{TTC}}$ (S. 34)
$f_{z\bar{z}}$	Gemeinsame Dichte von $z$ und $\bar{z}$ (S. 54)
$g$	Güte einer Objekthypothese (S. 78)
$g_{\text{FL}}$	Nicht-sensorspezifischer Anteil der Güte $g$ (S. 78)
$g_{\text{SL}}$	Sensorspezifischer Anteil der Güte $g$ (S. 78)
$g_{\text{SH}}$	Güte der Szenenhypothese (S. 79)
$H_{a_{\text{fw}}a_{\text{ego}}}(z)$	Übertragungsfunktion von $a_{\text{fw}}$ zu $a_{\text{ego}}$ (S. 94)
$H_{a_{\text{set}}a_{\text{ego}}}(z)$	Übertragungsfunktion von $a_{\text{set}}$ zu $a_{\text{ego}}$ (S. 94)
$H_{av}(z)$	Übertragungsfunktion von Beschleunigung zu Geschwindigkeit (S. 95)
$H_{ax}(z)$	Übertragungsfunktion von Beschleunigung zu Ort (S. 95)
$H_{d\hat{a}}(z)$	Übertragungsfunktion des $\alpha\beta\gamma$ -Filters von $d_{\text{rel}}$ nach $\hat{a}_{\text{rel}}$ (S. 93)
$H_{d\hat{d}}(z)$	Übertragungsfunktion des $\alpha\beta\gamma$ -Filters von $d_{\text{rel}}$ nach $\hat{d}_{\text{rel}}$ (S. 93)
$H_{d_{\text{rel}}a_{\text{ego}}}(z)$	Übertragungsfunktion von $d_{\text{rel}}$ nach $a_{\text{rel}}$ (S. 28)
$H_{d\hat{v}}(z)$	Übertragungsfunktion des $\alpha\beta\gamma$ -Filters von $d_{\text{rel}}$ nach $\hat{v}_{\text{rel}}$ (S. 93)
$h_{na}(n)$	Impulsantwort von $n$ zur $a_{\text{ego}}$ (S. 30)
$M$	Anzahl der Filterkerne im IMM Filter (S. 39)
$m_{x,y}$	Leistungsdichte zur Formschätzung [m <sup>2</sup> /s] (S. 76)
$\mathcal{N}(m, s)$	Normalverteilung mit Mittelwert $m$ und Standardabweichung $s$ (S. 27)
$n$	Zufälliger Abstandswert (Abstandsrauschen in der Entfernungsmessung) [m] (S. 26)
$n_d(k)$	Mittelwertfreier Zufallsprozess zur Modellierung der Dynamik (S. 75)
$n_v(k)$	Mittelwertfreier Zufallsprozess zur Schätzung der Form (S. 75)
$p$	Element der Zustandsübergangsmatrix (S. 40)
$Q$	Systemrauschen (S. 75)
$q_{x,y}$	Leistungsdichte zur Modellierung der Dynamik [m <sup>2</sup> /s <sup>5</sup> ] (S. 76)
$S$	Innovationskovarianzmatrix (S. 53)
$s$	Wurzel der mittleren quadratischen Abweichung von $\hat{v}_q$ und $v_q$ [m/s] (S. 87)
$T$	Zeitschritt [s] (S. 27)
$t_{\text{Fusion}}$	Zeitachse der Fusionsebene (S. 85)
$t_k$	Diskreter Zeitpunkt [s] (S. 75)
$t_{\text{Laser}}$	Zeitachse der Sensorebene des Laserscanners (S. 85)

$t_{\text{TTC}}$	Time To Collision [s] (S. 31)
$\hat{t}_{\text{TTC}}$	Geschätzte $t_{\text{TTC}}$ [s] (S. 32)
$t_{\text{Video}}$	Zeitachse der Sensorebene des Videosensors (S. 85)
$\mathcal{V}$	Volumen in dem ein Messwert mit einer vorgegebenen Wahrscheinlichkeit gefunden werden kann, auch Gate (S. 53)
$v$	Modellrauschen (Normalverteilte Zufallsvariable) (S. 27)
$v_{\text{ego}}$	Eigengeschwindigkeit [m/s] (S. 26)
$v_{\text{q}}$	Mit einem Correvit-Sensor ermittelte Quergeschwindigkeit [m/s] (S. 87)
$\hat{v}_{\text{q}}$	Geschätzte Quergeschwindigkeit [m/s] (S. 87)
$\hat{v}_{\text{q,Fusion}}$	Aus Laserscanner- und Videodaten geschätzte Quergeschwindigkeit [m/s] (S. 87)
$\hat{v}_{\text{q,Laser}}$	Aus den Laserscannerdaten geschätzte Quergeschwindigkeit [m/s] (S. 87)
$v_{\text{rel}}$	Relativgeschwindigkeit [m/s] (S. 26)
$\hat{v}_{\text{rel}}$	Geschätzte Relativgeschwindigkeit [m/s] (S. 27)
$v_{\text{rel,x}}$	Relativgeschwindigkeit in x-Richtung [m/s] (S. 75)
$v_{\text{rel,y}}$	Relativgeschwindigkeit in y-Richtung [m/s] (S. 75)
$x$	Zustandsvektor (S. 75)
$\tilde{x}$	Entfernung einer potentiellen Fahrzeugheckansicht [m] (S. 83)
$x_{\text{ego}}$	Absolute Position des betrachteten Fahrzeugs [m] (S. 26)
$x_{\text{fw}}$	Absolute Position des vorausfahrenden Fahrzeugs [m] (S. 26)
$\tilde{y}$	Ablage einer potentiellen Fahrzeugheckansicht [m] (S. 83)
$z$	Messvektor (S. 53)
$ZF_{t_{\text{TTC}}}$	Zielfunktion zur Minimierung der Fehler bei der $t_{\text{TTC}}$ -Berechnung [s <sup>2</sup> ] (S. 34)
$\bar{z}$	Prädizierter Messvektor (S. 53)

## Griechische Buchstaben

$\Delta t_{\text{max}}$	Obere Toleranz für $\Delta \hat{t}_{\text{TTC}}$ [s] (S. 35)
$\Delta t_{\text{min}}$	Untere Toleranz für $\Delta \hat{t}_{\text{TTC}}$ [s] (S. 35)
$\Delta \hat{t}_{\text{TTC}}$	Fehler der $t_{\text{TTC}}$ -Schätzung [s] (S. 32)
$\Delta v_{\text{q}}$	Schätzfehler der Quergeschwindigkeit [m/s] (S. 87)
$\Delta_n \hat{t}_{\text{TTC}}$	Fehler der $t_{\text{TTC}}$ -Schätzung aufgrund des Messrauschens [s] (S. 32)
$\Delta_T \hat{t}_{\text{TTC}}$	Fehler der $t_{\text{TTC}}$ -Schätzung aufgrund des Zeitverzugs [s] (S. 32)
$\epsilon$	Normalisierte, quadrierte Innovation [-] (S. 53)
$\gamma$	Quantil der $\chi^2$ -Verteilung [-] (S. 53)
$\lambda$	Manöverindex [-] (S. 27)
$\lambda_{\text{opt}}^{\text{asc}}$	Optimaler Manöverindex für die Datenassoziation [-] (S. 56)
$\lambda_{\text{max,dyn}}^{\text{acc}}$	Maximal notwendiger Manöverindex für den ACC-Regelkreis [-] (S. 31)
$\lambda_{\text{min,sta}}^{\text{acc}}$	Minimaler Manöverindex für Eigenstabilität des ACC-Regelkreises [-] (S. 29)
$\lambda_{\text{min,pla}}^{\text{acc}}$	Minimaler Manöverindex für Kolonnenstabilität des ACC-Regelkreises [-] (S. 28)
$\lambda_{\text{max,com}}^{\text{acc}}$	Minimaler Wert des Manöverindex für ein komfortables Folgeverhalten im stationären Folgebetrieb [-] (S. 30)

---

$\nu$	Abstand von Messung zu prädizierter Messung (S. 54)
$\Omega$	Normierte Frequenz [–] (S. 28)
$\Pi$	Zustandsübergangsmatrix des IMM Filters (S. 40)
$\sigma_a$	Standardabweichung der Beschleunigung $a_{\text{ego}}$ im stationären Folgebetrieb [m/s <sup>2</sup> ] (S. 30)
$\sigma_n$	Standardabweichung der Entfernungsmessung, Sensorrauschen [m] (S. 26)
$\sigma_v$	Standardabweichung des Modellrauschens (S. 27)
$\sigma_v^{\text{opt}}$	Optimaler Wert des Modellrauschens für die Datenassoziation (S. 61)
$\sigma_w$	Standardabweichung des Messrauschens [m] (S. 27)
$\sigma_{\bar{z}}$	Varianz der prädizierten Messung (S. 54)
$\tau_0$	Zeitlücke [s] (S. 26)
$\tau_d$	Zeitkonstante des ACC-Reglers [s] (S. 27)
$\tau_v$	Zeitkonstante des ACC-Reglers [s] (S. 27)
$\xi_i$	Eckpunkt eines Polygons (S. 74)

### Operatoren

$E\{\cdot\}$	Erwartungswert
$\mathcal{O}(\cdot)$	O-Notation zur Beschreibung des Aufwands eines Algorithmus
$\mathcal{P}(\cdot)$	Wahrscheinlichkeit
$\text{Var}\{\cdot\}$	Varianz

# 1 Einleitung

## 1.1 Motivation zur Arbeit

Die Funktion vieler Fahrerassistenzsysteme baut auf Informationen aus dem Umfeld des Fahrzeugs auf.<sup>1</sup> Ein bereits in anderen Disziplinen und in Forschungsprojekten im Bereich der Fahrerassistenzsysteme genutzter Ansatz Kosten zu sparen und die Leistungsfähigkeit der Systeme zu steigern, ist die Fusion der Information verschiedener Umfoldsensoren.<sup>2</sup> Eine im Designprozess für die praktische Umsetzung grundlegende und kostenbestimmende Frage ist hierbei, wie sich die Architektur einer solchen Sensordatenfusion hinsichtlich Software und Hardware gestaltet.<sup>3</sup> In aktuellen Forschungsprojekten im Bereich der Fahrerassistenzsysteme sind verschiedene Architekturen zur Sensordatenverarbeitung von Umfoldsensoren zu finden, die zum Teil unterschiedliche Vorteile, bzw. Grundannahmen für den Aufbau haben.<sup>4</sup> Auch wurde in der Automobilindustrie bisher keine einheitliche Architektur festgelegt, die einen verbindlichen Standard oder einen de-facto-Standard darstellt. Dies erschwert die unternehmensübergreifende Zusammenarbeit (Zulieferer, Original Equipment Manufacturer), die gezielte Entwicklung von auf eine Architektur angepassten Sensoren und Algorithmen, sowie die Migration zu neuen Assistenzfunktionen und Sensorengenerationen.<sup>5</sup>

Die vorliegende Arbeit greift diese Problematik auf. Es werden allgemeingültige Grundannahmen hinsichtlich der Datenfilterung und Datenassoziation erarbeitet, die beim Aufbau einer Architektur zur Sensordatenfusion von Umfoldsensoren für Fahrerassistenzsysteme beachtet werden müssen, bzw. zur Optimierung der Architektur herangezogen werden können. Schließlich wird eine Basis-Systemarchitektur aufbauend auf diesen Grundannahmen entworfen, welche eine Grundlage für die Gestaltung einer Serien-Systemarchitektur darstellen kann. In Abschnitt 1.4 werden die Ziele der Arbeit detailliert dargelegt. Hierzu werden in den fol-

---

<sup>1</sup>Einen breiten Überblick gibt z.B. Bishop, R.: *Vehicle Technology and Trends* (2005).

<sup>2</sup>Siehe z.B. Hall, D. L.: *Handbook of multisensor data fusion* (2001), Kapitel 1; Klaus, F.: *Techniken und Methoden der Multisensor-Datenfusion* (2004), S. 3-7.

<sup>3</sup>Siehe z.B. Hall, D. L.: *Handbook of multisensor data fusion* (2001), Kapitel 16; Klaus, F.: *Techniken und Methoden der Multisensor-Datenfusion* (2004), S. 51; Dietmayer, K.; Kirchner, A.; Kämpchen, N.: *Fusionsarchitekturen* (2005).

<sup>4</sup>Siehe Abschnitt 2, S. 9.

<sup>5</sup>Siehe auch die Ausführungen in Hall, D. L.: *Handbook of multisensor data fusion* (2001), Kapitel 2.3.

genden Abschnitten zunächst die in der Arbeit benötigten Begriffe definiert und notwendige Grundlagen zur Sensordatenfusion<sup>6</sup> und Architekturentwicklung<sup>7</sup> zusammengefasst.

## 1.2 Begriffsdefinitionen zur Sensordatenfusion

In der Arbeit wird für Sensordatenfusion die Definition nach Steinberg; Bowman; White verwendet:

„Data fusion is the process of combining data or information to estimate or predict entity states.“<sup>8,9</sup>

In der Definition wird der allgemeine Begriff der Entität verwendet, also ein abstraktes Objekt, dem Informationen zugeordnet werden können. Im weiteren Teil der Arbeit wird der Begriff Objekt direkt gebraucht, da dieser in den meisten Quellen im Bereich der Fahrerassistenzsysteme verwendet wird. Ein Objekt kann dabei ein beobachtetes Fahrzeug sein, allerdings wäre z.B. auch der freie Raum vor dem Fahrzeug als ein solches Objekt definierbar. Für beide Objekte ließe sich z.B. das Attribut Form schätzen.<sup>10</sup>

Im Joint Directors of Laboratories (*JDL*) Datenfusionsmodell<sup>11</sup> wird in eine so genannte Low-Level Verarbeitung und eine High-Level Verarbeitung unterschieden.<sup>12</sup> Im Rahmen der Low-Level Verarbeitung findet die Track-Schätzung<sup>13</sup> und die Objekt-Diskriminierung statt. Unter Track-Schätzung wird dabei die Schätzung der Zustände eines Objekts im regelungstechnischen Sinne verstanden (z.B. Position, Geschwindigkeit). Die Objekt-Diskriminierung wird wiederum in Detektion und Klassifizierung unterschieden.<sup>14</sup> Im Rahmen der Detektion wird erkannt, dass das Objekt vorhanden ist, die Klassifizierung ordnet das Objekt einer vordefinierten Klasse zu (z.B. Fahrzeug, Person). Zur High-Level Verarbeitung zählt unter anderem die Situationsanalyse<sup>15</sup> und die Steuerung des Fusionsprozesses.<sup>16</sup>

<sup>6</sup>Siehe z.B. Hall, D. L.: *Handbook of multisensor data fusion* (2001); Klein, L. A.: *Sensor and data fusion* (1999); Klaus, F.: *Techniken und Methoden der Multisensor-Datenfusion* (2004); Bar-Shalom, Y.; Li, X.-R.: *Multitarget-multisensor tracking* (1995); Brooks, R. R.; Iyengar, S. S.: *Multi sensor fusion* (1998); Waltz, E.; Llinas, J.: *Multisensor data fusion* (1990).

<sup>7</sup>Siehe z.B. Starke, G.: *Effektive Software-Architekturen* (2005); Vogel, O.: *Software-Architektur* (2005) oder Bass, L.; Clements, P.; Kazman, R.: *Software Architecture in Practice* (2003).

<sup>8</sup>Übersetzung: Datenfusion ist ein Prozess, bei dem Daten oder Informationen kombiniert werden, um Zustände einer Entität zu schätzen oder zu präzisieren.

<sup>9</sup>Steinberg, A.; Bowman, C.; White, F.: *Revisions to the JDL Data Fusion Model* (1998), zitiert in Hall, D. L.: *Handbook of multisensor data fusion* (2001), Kapitel 2.2.

<sup>10</sup>Vgl. auch die Diskussion in Dietmayer, K.; Kirchner, A.; Kämpchen, N.: *Fusionsarchitekturen* (2005), S. 67.

<sup>11</sup>Siehe z.B. Klein, L. A.: *Sensor and data fusion* (1999), S. 49.

<sup>12</sup>Diese müssen nicht zwangsweise aufeinander aufbauen, siehe Steinberg, A.; Bowman, C.: *Revisions to the JDL Data Fusion Model* (2001).

<sup>13</sup>Das „Tracking“.

<sup>14</sup>Vgl. Klein, L. A.: *Sensor and data fusion* (1999), S. 51.

<sup>15</sup>JDL Datenfusionsmodell: Situation Assessment.

<sup>16</sup>JDL Datenfusionsmodell: Process Refinement.

Im Rahmen der Arbeit liegt der Fokus auf der Low-Level Verarbeitung, also der Track-Schätzung und der Objekt-Diskriminierung. In Anlehnung an andere Arbeiten<sup>17</sup> und um Verwechslungen mit anderen Auffassungen für den Begriff Low-Level Fusion<sup>18</sup> zu vermeiden, wird im Folgenden der Begriff Objekt-Schätzung verwendet.<sup>19</sup>

## 1.3 Architektur

### 1.3.1 Begriffsdefinitionen

Die Architektur eines Software-Systems nimmt im Entwicklungsprozess der Software<sup>20</sup> eine zentrale Rolle ein.<sup>21</sup> Der Begriff der Architektur ist hierbei nicht eindeutig festgelegt.<sup>22</sup> Für die vorliegende Arbeit werden die Definition nach Bass; Clements; Kazman verwendet.<sup>23</sup> Definiert wird hier der Begriff Softwarearchitektur

„Die Softwarearchitektur eines Systems beschreibt dessen Software-Struktur respektive dessen Strukturen, dessen Softwarebausteine sowie deren sichtbare Eigenschaften und Beziehungen zueinander.“<sup>24</sup>

und die allgemeiner gefasste Systemarchitektur

„Die Systemarchitektur eines Systems beschreibt dessen Struktur respektive dessen Strukturen, dessen Bausteine (Software- und Hardware-Bausteine) sowie deren sichtbare Eigenschaften und Beziehungen zueinander.“<sup>25</sup>.

Bei der Beschreibung einer Architektur werden dabei verschiedene Sichten unterschieden.<sup>26</sup> Die wichtigsten Sichten einer Software-Architektur sind:<sup>27</sup>

<sup>17</sup>Siehe z.B. Becker, J.-C.: *Fusion der Daten der objekterkennenden Sensoren* (2002).

<sup>18</sup>Siehe z.B. Dietmayer, K.; Kirchner, A.; Kämpchen, N.: *Fusionsarchitekturen* (2005), S. 67.

<sup>19</sup>JDL Datenfusionsmodell: Object Assessment.

<sup>20</sup>Siehe hierzu z.B. Jacobson, I.; Booch, G.; Rumbaugh, J.: *The unified software development process* (1999) oder Beck, K.: *Extreme Programming* (2003); vgl. Vogel, O.: *Software-Architektur* (2005), S. 270. Zum in der Automobilindustrie verbreiteten V-Modell siehe z.B. KBSt: *V-Modell XT* (2006).

<sup>21</sup>Vgl. Starke, G.: *Effektive Software-Architekturen* (2005), S. 26.

<sup>22</sup>Siehe z.B. Vogel, O.: *Software-Architektur* (2005), S. 8-11 oder Starke, G.: *Effektive Software-Architekturen* (2005), S. 14-18. Etwa 50 verschiedene Definitionen sind auf den Seiten des Software Engineering Institute der Carnegie Mellon University zu finden: <http://www.sei.cmu.edu/architecture/definitions.html> (14.09.2006).

<sup>23</sup>Bass, L.; Clements, P.; Kazman, R.: *Software Architecture in Practice* (2003).

<sup>24</sup>Vogel, O.: *Software-Architektur* (2005), S. 46.

<sup>25</sup>Vogel, O.: *Software-Architektur* (2005), S. 48.

<sup>26</sup>Ähnlich wie bei Gebäude-Architekturen (hier z.B. Grundriss oder der Elektroplan), vgl. Starke, G.: *Effektive Software-Architekturen* (2005), S. 15.

<sup>27</sup>Vgl. Starke, G.: *Effektive Software-Architekturen* (2005), S. 86.

**Kontextsicht** Kontextsichten stellen den Zusammenhang eines Systems mit seiner Umgebung dar. Diese Sichten beinhalten meist wenig technische Informationen.

**Bausteinsicht** Bausteinsichten zeigen den internen (statischen) Aufbau eines Systems, dessen Subsysteme, Komponenten und deren Schnittstellen zueinander. Die Bausteinsicht stellt die wichtigste Sicht im Architekturentwurf dar.

**Laufzeitsicht** Laufzeitsichten erklären, welche Bausteine des Systems zur Laufzeit existieren und wie diese (dynamisch) zusammenwirken.

**Verteilungssicht** Verteilungssichten beschreiben die Hardwarekomponenten, auf welchen das System läuft und wie die Softwarebausteine auf diese verteilt werden. Hierzu zählen u.a. Rechner, Prozessoren und Netzwerktopologien.

Die einzelnen Sichten hängen voneinander ab und können nicht vollkommen getrennt voneinander entworfen werden.<sup>28</sup> Der Fokus dieser Arbeit liegt auf Baustein-, Laufzeit- und Verteilungssichten.

### 1.3.2 Zweck einer Architektur

Die Architektur trägt dazu bei, ein System verständlich zu machen und dessen Komplexität zu beherrschen. Für die an der Entwicklung des Systems beteiligten Personen werden hier die Struktur und das Zusammenwirken der einzelnen Teile dokumentiert.<sup>29</sup> Die Architektur erleichtert damit, auch durch die Definition geeigneter, eindeutiger Begriffe und Schnittstellen, die Kommunikation über das System.

Die Organisation hat einen Einfluss auf die Architektur und umgekehrt.<sup>30</sup> So können bei der Innenstrukturierung des Forschungs- und Entwicklungsbereiches eines Unternehmens Mitarbeiter nach wissenschaftlich-technischer Spezialisierung in Abteilungen zusammen gefasst werden.<sup>31</sup> Die Bildung der Abteilungen kann dabei an den Komponenten der Architektur orientiert sein, wie z.B. einer Tracking-Abteilung, die sich mit der Datenfilterung beschäftigt und einer Bildverarbeitungs-Abteilung, die Algorithmen zur Extraktion von Merkmalen aus einem Videobild entwickelt.

Die Architektur des Systems trägt zudem dazu bei, den Entwicklungsprozess zu strukturieren.<sup>32</sup> Dies gilt auch über die Unternehmensgrenzen hinaus, da die Architektur und der Grad der Kopplung<sup>33</sup> innerhalb des Systems beeinflussen, in wie weit Komponenten von verschiedenen Lieferanten hergestellt werden können.

<sup>28</sup>Vgl. Starke, G.: *Effektive Software-Architekturen* (2005), S. 85; Vogel, O.: *Software-Architektur* (2005), S. 81.

<sup>29</sup>Vgl. Starke, G.: *Effektive Software-Architekturen* (2005), S. 26.

<sup>30</sup>Vgl. Vogel, O.: *Software-Architektur* (2005), S. 247; Starke, G.: *Effektive Software-Architekturen* (2005), S. 24.

<sup>31</sup>Siehe z.B. Specht, G.; Beckmann, C.: *F&E-Management* (1996), S. 262-270.

<sup>32</sup>Vgl. Starke, G.: *Effektive Software-Architekturen* (2005), S. 29.

<sup>33</sup>Siehe z.B. Vogel, O.: *Software-Architektur* (2005), S. 114.

### 1.3.3 Entwicklungsmethoden zum Entwurf einer Architektur

Für die Entwicklung einer Architektur gibt es keine deterministischen Verfahren, die in jedem Fall zu einer guten Architektur führen.<sup>34</sup> Als sinnvoll hat sich ein iteratives und inkrementelles Vorgehen erwiesen.<sup>35</sup>

Jacobson; Booch; Rumbaugh schlagen hierbei die Orientierung an Anwendungsfällen<sup>36</sup> vor, die Auswahl der entsprechenden Anwendungsfälle für den Architekturentwurf hängt dabei von den Umständen des jeweiligen Projekts ab.<sup>37</sup>

Nach Vogel steht am Anfang der Architekturentwicklung ein so genanntes Skelettsystem.<sup>38</sup> Dieses ist ein lauffähiges System, welches in der Struktur bereits der Architektur des endgültigen Systems entspricht. Es enthält die tragenden Bausteine, wobei diese zunächst die Funktionalität zur Verfügung stellen, die zur Prüfung der relevanten Anwendungsfälle notwendig sind. In der weiteren Entwicklung werden die einzelnen Bausteine weiter verfeinert und der funktionale Umfang des Systems weiter erhöht.<sup>39</sup>

Ziel ist schließlich eine stabile Vorversion der endgültigen Architektur,<sup>40</sup> die den Rahmen bzw. die Schablone<sup>41</sup> für das System vorgibt, da spätere Änderungen in der Architektur in der Regel teuer und mit hohem Aufwand verbunden sind.<sup>42</sup>

Starke stellt dabei folgende Punkte zum effektiven Entwurf einer Architektur als grundlegende Aktivitäten dar:<sup>43</sup>

**Informationen sammeln** Bestehendes Wissen darüber, wie ähnliche Probleme gelöst wurden, wird gesammelt. Hierdurch kann zu Beispielen und Mustern gelangt werden, die es erlauben, nicht das gesamte System neu zu entwerfen.<sup>44</sup>

**Einflussfaktoren finden** Die für die Architektur relevanten Einflussfaktoren werden bestimmt. Hierzu zählen organisatorische Faktoren (z.B. Organisationsstruktur des Unternehmens, organisatorische Standards wie Vorgehensmodelle, etc.), technische Faktoren (z.B. verfügbare Hardware) und die Systemfaktoren (Anforderungen an das System).<sup>45</sup>

<sup>34</sup>Vgl. Starke, G.: *Effektive Software-Architekturen* (2005), S. 31.

<sup>35</sup>Siehe z.B. Starke, G.: *Effektive Software-Architekturen* (2005), S. 24; Vogel, O.: *Software-Architektur* (2005), S. 270 - 293; Jacobson, I.; Booch, G.; Rumbaugh, J.: *The unified software development process* (1999), S. 69 - 71.

<sup>36</sup>Use Cases, siehe auch Vogel, O.: *Software-Architektur* (2005), S. 278.

<sup>37</sup>Siehe hierzu auch die Ausführungen in Jacobson, I.; Booch, G.; Rumbaugh, J.: *The unified software development process* (1999), S. 330 - 333.

<sup>38</sup>Vgl. Vogel, O.: *Software-Architektur* (2005), S. 284. Siehe auch Jacobson, I.; Booch, G.; Rumbaugh, J.: *The unified software development process* (1999), S. 69.

<sup>39</sup>Vgl. Vogel, O.: *Software-Architektur* (2005), S. 285.

<sup>40</sup>Siehe auch die Ausführungen in Jacobson, I.; Booch, G.; Rumbaugh, J.: *The unified software development process* (1999), S. 69 - 71 zur so genannten Architecture Baseline.

<sup>41</sup>Vgl. Vogel, O.: *Software-Architektur* (2005), S. 285.

<sup>42</sup>Vgl. Jacobson, I.; Booch, G.; Rumbaugh, J.: *The unified software development process* (1999), S. 71.

<sup>43</sup>Vgl. Starke, G.: *Effektive Software-Architekturen* (2005).

<sup>44</sup>Vgl. Starke, G.: *Effektive Software-Architekturen* (2005), S. 33.

<sup>45</sup>Vgl. Starke, G.: *Effektive Software-Architekturen* (2005), S. 44 - 50.

**Heuristiken anwenden** Nutzen von bestehendem Erfahrungswissen aus den Bereichen der System- und Softwarearchitektur (Entwurfsmuster, Entwurfsprinzipien). Diese geben Hinweise auf dem Weg zu einer optimalen Lösung, garantieren diese aber nicht.<sup>46</sup>

**Systemidee entwickeln** Die Systemidee führt zusammen mit den Heuristiken bzw. Lösungsstrategien und den Anforderungen an die Architektur zum Architekturentwurf, welcher dann im Rahmen des iterativen Vorgehens weiter entwickelt wird.<sup>47</sup>

## 1.4 Ziele der Arbeit

Ziel der Arbeit ist die Aufstellung und Validierung von allgemeingültigen Grundannahmen hinsichtlich der Track-Schätzung, die in einer Systemarchitektur zur Sensordatenfusion von Umfeldsensoren für Fahrerassistenzsysteme (*FAS*) berücksichtigt werden müssen bzw. zur Optimierung des Aufbaus der Architektur herangezogen werden können. Weiteres Ziel ist die Beschreibung und Validierung einer auf diesen Annahmen aufbauenden Vorversion einer Systemarchitektur. Diese Vorversion wird im Folgenden als Basis-Systemarchitektur bezeichnet.<sup>48</sup>

Die Basis-Systemarchitektur kann Architekten einer noch zu entwickelnden Serien-Systemarchitektur als Grundlage dienen, auf der aufbauend sich schließlich das Gesamtsystem durch eine feinere Ausgestaltung der einzelnen Komponenten konzipieren und umsetzen lässt.

Zentrale Fragestellungen, die dabei im Rahmen der Arbeit beantwortet werden, sind:

- Wie können unterschiedliche Assistenz-Funktionen, welche die gleiche Umfeldsensorik verwenden, in einem Gesamtsystem gemeinsam betrieben werden?
  - Können für alle Assistenzfunktionen die Zustandsgrößen beobachteter Objekte mit dem gleichen Algorithmus geschätzt werden?
  - Kann für die Datenassoziation der gleiche Algorithmus zur Schätzung der Zustandsgrößen beobachteter Objekte eingesetzt werden wie für die Assistenzfunktionen?
  - Können für alle Assistenzfunktionen die gleichen Algorithmen zur Datenassoziation verwendet werden?

---

<sup>46</sup>Vgl. Starke, G.: *Effektive Software-Architekturen* (2005), S. 117-146; Vogel, O.: *Software-Architektur* (2005), S. 111-240.

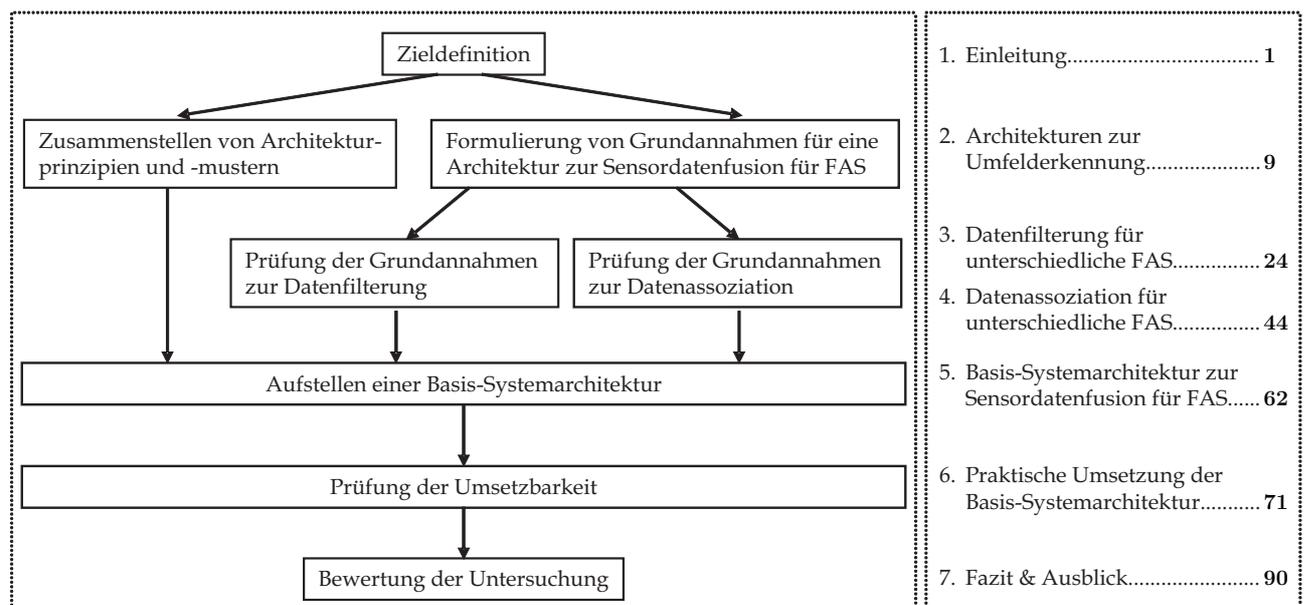
<sup>47</sup>Vgl. Starke, G.: *Effektive Software-Architekturen* (2005), S. 41.

<sup>48</sup>In Anlehnung an den Begriff „Architecture Baseline“ vgl. Jacobson, I.; Booch, G.; Rumbaugh, J.: *The unified software development process* (1999), S. 69. In Anlehnung an die Ausführungen in Abschnitt 1.3.3, S. 5 kann man diese auch als Rahmen-Systemarchitektur bezeichnen.

- Wie kann eine Architektur gestaltet werden, damit unterschiedliche Organisationseinheiten/Unternehmen in den Entwicklungsprozess des Gesamtsystems eingebunden werden können?
  - Wie ist die Kapselung der Bausteine in Modulen hierfür zu gestalten?
  - Wie können die Module auf verschiedene Hardware verteilt werden?
- Wie kann eine Architektur gestaltet werden, damit die Migration zu neuen Assistenzfunktionen mit ggf. weiterentwickelten Umfoldsensoren ohne Änderungen an der Architektur und bestehenden Algorithmen erfolgen kann?

## 1.5 Methodik und Aufbau der Arbeit

Abbildung 1.1 zeigt die Methodik und den Aufbau der Arbeit, welche sich an den in Abschnitt 1.3.3 beschriebenen Aktivitäten orientiert.



**Abbildung 1.1:** Methodik und Aufbau der Arbeit.

Um die in Abschnitt 1.4 beschriebenen Ziele der Arbeit zu erreichen, wurden auf Basis bestehender Literatur die Hauptkomponenten eines Fahrerassistenzsystems mit Umfelderkennung und die Hauptschritte des Prozesses der Sensordatenfusion identifiziert sowie deren Funktion beschrieben. Zudem wurden allgemeine Prinzipien zum Aufstellen einer Architektur auf Basis existierender Literatur aus dem Bereich des Softwareengineering recherchiert und allgemeine Muster für Architekturen zur Sensordatenfusion anhand bestehender Literatur zusammengestellt. Die Ergebnisse sind in Kapitel 2 zusammengefasst.

Mit den Ergebnissen der Recherche wurden für die Umfelddatenverarbeitung für Fahrerassistenzsysteme allgemeingültige Grundannahmen hinsichtlich der Datenfilterung und Da-

tenassoziation aufgestellt, die beim Aufstellen einer Architektur zur Sensordatenfusion für Fahrerassistenzsysteme berücksichtigt werden müssen bzw. zur Optimierung der Architektur herangezogen werden können. Diese Grundannahmen bilden mit Abschnitt 2.3 den Abschluss des Kapitels 2, welches damit die Grundlage für die weitere Arbeit darstellt.

Im nächsten Schritt wurde die Gültigkeit der Grundannahmen geprüft. Hierzu wurden sowohl Simulationen als auch existierende Literatur zur Umfelderkennung für Fahrerassistenzsysteme herangezogen. Die Ergebnisse sind in den Kapiteln 3 und 4 zu finden. Hierbei sind in den einzelnen Kapiteln die theoretischen Grundlagen zur jeweiligen Annahme nochmals dargelegt.

Auf Basis der bestätigten Grundannahmen und der allgemeinen Prinzipien zum Aufstellen einer Architektur wurde unter Zuhilfenahme der Ergebnisse der Architekturmusterrecherche eine Basis-Systemarchitektur zur Sensordatenfusion für Fahrerassistenzsysteme aufgestellt. Der Fokus liegt hierbei auf der Bausteinsicht. Die Architektur wird in Kapitel 5 beschrieben. Zudem wird diskutiert wie robust die Architektur hinsichtlich der Gültigkeit der Grundannahmen ist.

Zur Prüfung der praktischen Umsetzbarkeit der Basis-Systemarchitektur wurde diese im Projekt PRORETA auf einem Versuchsträger implementiert. In Kapitel 6 werden die algorithmische Umsetzung der Bausteine und Schnittstellen der Architektur erläutert sowie die realisierte Verteilungs- und Laufzeitsicht beschrieben. Anhand von praktischen Versuchen werden die Vorteile der Fusion für das Projekt dargelegt und die Umsetzbarkeit der Architektur dokumentiert.

Zum Abschluss werden in Kapitel 7 die Kernaussagen der Arbeit zusammengefasst und ein Fazit zur Arbeit gezogen. Zudem wird ein Ausblick auf weitere offene Fragen im Bereich der Architekturentwicklung für die Umfelderkennung für Fahrerassistenzsysteme gegeben.



(Energie) überlagert durch Störsignale (Rauschen) empfangen und in Rohsignale (z.B. Spannungen, Ströme) gewandelt. Die Rohsignale werden als physikalische Messgrößen interpretiert (z.B. Intensitäten, Frequenzen, etc.), welche schließlich die Rohdaten des Sensors bilden. Im Rahmen der Signalverarbeitung werden dabei (physikalische) Annahmen für die Interpretation getroffen (z.B. maximaler Empfangspegel, Impulsformen, etc.). Werden diese verletzt, entstehen so genannte Artefakte, welche eine systembedingte Schwäche darstellen.<sup>3</sup>

Im zweiten Schritt, im Folgenden als Wahrnehmen bezeichnet, werden aus den Rohdaten auf Basis von Annahmen bzw. Heuristiken Merkmale extrahiert (z.B. Kanten, Extremwerte), aus welchen Merkmals hypothesen zu einem angenommenen Objekt (Objekthypothesen) abgeleitet werden (z.B. Koordinaten der Unterkante eines Fahrzeugs aus einem Videobild<sup>4</sup>, Rückfront bzw. Seiten eines Fahrzeugs aus Entfernungsmessungen eines Laserscanners<sup>5</sup> oder auch Radgeschwindigkeit aus den Signalen eines Drehzahlsensors<sup>6</sup>). Aufgrund der Anwendung von Heuristiken kann es zu einer Fehlinterpretation kommen, wie z.B. bei der Segmentierung der Entfernungsmessungen eines Laserscanners.<sup>7</sup> Die Merkmals hypothesen gelten also nur unter der Annahme, dass die zugrunde liegende Heuristik/Hypothese bei der Extraktion des Merkmals korrekt war.

Die Signalverarbeitung und Merkmalsextraktion können mit Hilfe von bereits im System vorhandenen Informationen vorkonditioniert werden. Über prädizierte Zustände von beobachteten Objekten und die entsprechenden Varianzen kann der Wertebereich einer Messung vorhergesagt werden. Auf diese Weise können z.B. Verstärkungsfaktoren in der Hardware gesetzt werden (z.B. Verstärkung des Empfangssignals eines Laserscanners) oder Suchbereiche in den Rohdaten definiert werden, in denen Heuristiken angewandt werden (z.B. bei einer Videodatenverarbeitung). Auch die Objektklasse kann zur Vorkonditionierung oder zur Auswahl einer spezifischen Heuristik zur Merkmalsextraktion genutzt werden.

Das Verhalten der Signalverarbeitung und Merkmalsextraktion kann zudem von der Situationsanalyse<sup>8</sup> beeinflusst werden. So ist es z.B. möglich den Messbereich der Hardware an eine entsprechende Situation anzupassen (z.B. Schwenken der Optik eines Laserscanners für Kurvenfahrt bei der Adaptive Cruise Control<sup>9</sup>), oder die verfügbare Rechenzeit kann situationsangepasst auf ausgewählte Daten verwendet werden (z.B. Bestimmung der Breite nur der als potentielle Kollisionsobjekte identifizierten Objekthypothesen per Videodatenverarbeitung<sup>10</sup>).

<sup>3</sup>Analog zu systematischen Fehlern, mit dem Unterschied, dass nicht bekannt ist, ob die aktuelle Situation ein Artefakt hervorruft, weshalb der Fehler nicht kompensiert werden kann.

<sup>4</sup>Siehe z.B. Holt, V. v.: *Multisensorielle Fahrumgebungserfassung* (2005), S. 59 - 76.

<sup>5</sup>Siehe z.B. Sparbert, J.; Dietmayer, K.; Streller, D.: *Laser Range Images* (2001).

<sup>6</sup>Siehe z.B. Bauer, H.; Zabler, E.: *Sensoren im Kraftfahrzeug* (2001), S. 57.

<sup>7</sup>Siehe z.B. Streller, D.: *Multi-Hypothesen-Ansatz* (2006), S. 30; Kirchner, A.: *Sensordatenverarbeitung eines Laserscanners* (2000), S. 36. Siehe auch das Beispiel auf S. 49.

<sup>8</sup>Siehe S. 12.

<sup>9</sup>So realisiert im Laserscanner AIS100 von ContinentalTemic ADC.

<sup>10</sup>Siehe hierzu auch Abschnitt 6.2.5, S. 79.

## Datenassoziation

Die in der Signalverarbeitung und Merkmalsextraktion gewonnenen Merkmals Hypothesen werden während der Datenassoziation den bereits im System bekannten Objekthypothesen zugeordnet.<sup>11</sup> Für einen Sensor, der immer die Größe desselben Objekts bestimmt (z.B. Drehzahlsensor), ist dies eindeutig möglich (z.B. Zuordnung zum Ego-Fahrzeug<sup>12</sup>), für Sensoren, die mehrere Objekte vermessen (z.B. Umfeldsensoren), sind Algorithmen zur Zuordnung notwendig.<sup>13</sup> Messungen, die nicht zugeordnet werden können, werden genutzt, um neue Objekthypothesen zu generieren.<sup>14</sup>

## Objekthypothesen-Verwaltung

Die Anzahl der im System verwaltbaren Objekthypothesen ist aufgrund der beschränkten Rechenzeit begrenzt. Zudem wird zu Objekten, die den Erfassungsbereich der Sensoren verlassen haben, keine neue Information erzeugt, so dass die Informationen über das Objekt immer unzuverlässiger werden. Auch können sich bestehende Objekthypothesen als unwahrscheinlich erweisen oder neue Objekthypothesen auftreten (siehe Datenassoziation). Der Prozess des Verwaltens (Löschen, Beibehalten, Generieren) der Hypothesen wird in der Objekthypothesen-Verwaltung durchgeführt.

Die Objekthypothesen-Verwaltung wird von der Situationsanalyse beeinflusst, da hier entschieden wird, welche Objekte für die aktuelle Situation relevant sind und nicht gelöscht werden dürfen.

## Datenfilterung

Ein Filter bzw. Schätzer<sup>15</sup> wird eingesetzt, um auf Basis von Messungen Informationen zu verbessern oder neue Informationen zu gewinnen.<sup>16</sup>

Die Datenfilterung kann von den Ergebnissen der Klassifikation beeinflusst werden. Für eine über die Sensoren beobachtete Person kann z.B. ein anderes Bewegungsmodell zur Schätzung genutzt werden als für ein Fahrzeug.<sup>17</sup> Zudem können die Parameter der Schätzung an die Anforderungen des Reglers angepasst werden.

---

<sup>11</sup>Siehe auch Kapitel 4, S. 44. Hier wird die Datenassoziation ausführlich behandelt.

<sup>12</sup>Siehe hierzu aber auch die Anmerkung auf S. 44.

<sup>13</sup>„Multitarget Tracking“, siehe z.B. Bar-Shalom, Y.; Li, X.-R.: *Multitarget-multisensor tracking* (1995).

<sup>14</sup>Siehe unten.

<sup>15</sup>Die Begriffe werden im Folgenden synonym verwendet.

<sup>16</sup>Vgl. Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 2. Siehe auch Kapitel 3, S. 24.

<sup>17</sup>Siehe z.B. Cramer, H.: *Modelle zur multisensoriellen Erfassung des Fahrzeugumfeldes* (2006), oder Streller, D.; Fürstenberg, K.; Dietmayer, K.: *Vehicle and Object Models* (2002).

## Klassifikation

Während der Klassifikation werden Objekthypothesen aufgrund zugeordneter Eigenschaften einer vordefinierten Klasse zugewiesen.<sup>18</sup> Die Eigenschaften können aus den Rohdaten des Sensors<sup>19</sup> (z.B. Temperaturen, Formen, Muster) oder aus den Zustandsvariablen<sup>20</sup> der Objekthypothese (z.B. Geschwindigkeiten, Position in Relation zu anderen Objekten) stammen.<sup>21</sup> Die Zuweisung kann nicht immer eindeutig erfolgen, so dass es evtl. nur möglich ist, Wahrscheinlichkeiten anzugeben, zu welcher Klasse die Objekthypothese gehört.

Die Information über die Objektklasse kann im Rahmen der Datenfilterung und der Situationsanalyse genutzt werden. In der Situationsanalyse hilft die Information, Strategien zum Umgang mit der aktuellen Situation zu finden (z.B. Notbremsung auf eine als Fahrzeug klassifizierte Objekthypothese) oder das Verhalten eines Reglers einzustellen (z.B. Verhalten beim Überqueren einer Fahrbahnmarkierung).

## Situationsanalyse

Die Situationsanalyse bestimmt das Gesamtverhalten des Fahrerassistenzsystems. Beispielsweise steht hinter der adaptiven Fahrgeschwindigkeitsregelung (ACC) ein Zustandsautomat, der das Verhalten der Anwendung in verschiedenen Situationen festlegt.<sup>22</sup>

Die Situationsanalyse basiert auf den Ergebnissen der Klassifikation (z.B. Auswahl eines als Fahrzeug klassifizierten Objekts für die ACC-Regelung), den Ausgangssignalen der Komponenten Datenfilterung und Regler (z.B. Umschalten zwischen Abstands- und Abbremsregelung im ACC-Folgebetrieb<sup>23</sup>) sowie den Eingaben des Fahrers.

Auf Basis der Situationsanalyse kann die Signalverarbeitung und Merkmalsextraktion beeinflusst werden (siehe oben). Weiterhin können die Parameter der Zustandsschätzung an die Anforderungen des Reglers angepasst werden und der aktive Regler (Regler dessen Signale zur Regelung verwendet werden) kann ausgewählt werden.

## Regler

Im Allgemeinen wird die Funktion eines Fahrerassistenzsystems über verschiedene Regler bereitgestellt, die auf bestimmte Betriebspunkte optimiert sind.<sup>24</sup> Die Auswahl des aktiven

<sup>18</sup>Siehe z.B. Klein, L. A.: *Sensor and data fusion* (1999), S. 51.

<sup>19</sup>Siehe z.B. Fardi, B.; Scheunert, U.; Wanielik, G.: *Pedestrian Detection in Infrared Images* (2005).

<sup>20</sup>Siehe z.B. Streller, D.; Fürstenberg, K.; Dietmayer, K.: *Vehicle and Object Models* (2002).

<sup>21</sup>Einen allgemeinen Überblick gibt Klein, L. A.: *Sensor and data fusion* (1999), S. 52.

<sup>22</sup>Siehe z.B. Mayr, R.: *Regelungsstrategien für die automatische Fahrzeugführung* (2001), Witte, S.: *Simulationsuntersuchungen ACC* (1996) und Abschnitt 3.2.

<sup>23</sup>Vgl. Mayr, R.: *Regelungsstrategien für die automatische Fahrzeugführung* (2001), S. 75-76.

<sup>24</sup>Siehe z.B. Mayr, R.: *Regelungsstrategien für die automatische Fahrzeugführung* (2001), S. 75. Siehe auch Abschnitt 3.2.

Reglers basiert auf der Situationsanalyse. Die Regler wiederum nutzen die Ergebnisse der Datenfilterung als Eingangsgrößen.<sup>25</sup>

## 2.2 Architekturprinzipien und Architekturmuster

### 2.2.1 Allgemeine Architekturprinzipien

Zunächst werden für die Arbeit wichtige allgemeine Architekturprinzipien zusammengefasst.<sup>26</sup> Im Anschluss daran werden bewährte Architekturmuster für die Sensordatenfusion von Umfeldsensoren aufgezeigt. Auf die vorgestellten Prinzipien und Muster wird bei der Beschreibung der Basis-Systemarchitektur in Kapitel 5 Bezug genommen.<sup>27</sup>

#### Prinzip der losen Kopplung

Die Kopplung gibt an, wie stark einzelne Bausteine voneinander abhängen. Nach dem Prinzip der losen Kopplung soll die Kopplung zwischen den Bausteinen so gering wie möglich gehalten werden. Je geringer die Kopplung der Bausteine, desto unabhängiger sind die Bausteine voneinander. Dies erhöht die Änderbarkeit und Flexibilität des Gesamtsystems. Zudem macht es das System verständlicher, da die Anzahl der zusätzlichen Bausteine, die zum Verständnis des betrachteten, einzelnen Bausteins notwendig sind, minimiert wird. Eine hohe Kopplung hingegen führt zu einer hohen Komplexität und Starrheit des Entwurfs.<sup>28</sup>

#### Prinzip der hohen Kohäsion

Ein Baustein selbst besteht meist aus mehreren Teilen. Die Kohäsion gibt an, wie stark dessen Elemente inhaltlich zusammengehören. Nach dem Prinzip der hohen Kohäsion, sollen die Elemente innerhalb eines Bausteins möglichst eng miteinander verbunden sein. Wie die lose Kopplung führt dies zu einer besseren Änderbarkeit und Verständlichkeit der Bausteine. Üblicherweise führt eine hohe Kohäsion zu einer losen Kopplung.<sup>29</sup> Architekturen, die das Prinzip der hohen Kohäsion und das Prinzip der losen Kopplung umsetzen, verbessern die Verständlichkeit des Gesamtsystems.<sup>30</sup>

<sup>25</sup>Kapitel 3, S. 24, beschäftigt sich ausführlich mit diesem Thema.

<sup>26</sup>Für Details siehe Vogel, O.: *Software-Architektur* (2005), S. 112-136 und Starke, G.: *Effektive Software-Architekturen* (2005), S. 125-138.

<sup>27</sup>Siehe Kapitel 5, S. 62.

<sup>28</sup>Vgl. Starke, G.: *Effektive Software-Architekturen* (2005), S. 130; Vogel, O.: *Software-Architektur* (2005), S. 115-116.

<sup>29</sup>Vgl. Starke, G.: *Effektive Software-Architekturen* (2005), S. 131; Vogel, O.: *Software-Architektur* (2005), S. 117-118.

<sup>30</sup>Vgl. Vogel, O.: *Software-Architektur* (2005), S. 118.

## Prinzip des Entwurfs für Veränderung

Nach dem Prinzip des Entwurfs für Veränderungen sollen vorhersehbare Änderungen wie z.B. erwartbare Weiterentwicklungen mit in den Architekturentwurf einbezogen werden. Hochflexible Architekturen haben allerdings oftmals Nachteile gegenüber inflexiblen Architekturen (z.B. Ressourcen-Verbrauch), so dass bei unwahrscheinlichen Veränderungen die Einplanung sorgfältig abgewogen werden muss.<sup>31</sup>

## Separation-of-Concerns Prinzip

Das Separation-of-Concerns Prinzip besagt, dass verschiedene Aspekte eines Problems voneinander getrennt werden, so dass jedes Teilproblem eigenständig behandelt werden kann. Das Prinzip reduziert die Komplexität und erlaubt ein arbeitsteiliges Vorgehen.<sup>32</sup>

## Information Hiding Prinzip

Das Information Hiding Prinzip besagt, dass einem Klienten nur der notwendige Teilausschnitt der Information zugänglich gemacht wird, den er zur Erfüllung der Aufgabe benötigt.<sup>33</sup>

Anwendung findet dies in der Modularisierung eines Systems. Strukturen innerhalb des Systembausteins werden gekapselt und nach außen nur durch definierte Schnittstellen bekannt gegeben.<sup>34</sup>

Eine Schichten-Architektur realisiert üblicherweise das Information Hiding Prinzip. Jede Schicht sieht nur die direkt darunter liegende Schicht. Die nutzende Schicht sollte keine Implementierungsdetails der darunter liegenden Schicht sehen. Der Zugriff erfolgt über klar definierte Schnittstellen.<sup>35</sup>

## Offen-Geschlossen Prinzip

Zur Erfüllung des Offen-Geschlossen Prinzips sollen Bausteine offen für Änderungen, aber geschlossen für die Nutzung ihrer internen Details durch andere Systembausteine sein. Komponenten sollten so beschaffen sein, dass eine Erweiterung ohne Änderungen am bestehenden Quellcode möglich ist. Die Offenheit kann durch das Prinzip Entwurf für Veränderungen, die Geschlossenheit durch Information Hiding erreicht werden.<sup>36</sup>

<sup>31</sup>Vgl. Vogel, O.: *Software-Architektur* (2005), S. 119 - 121.

<sup>32</sup>Vgl. Vogel, O.: *Software-Architektur* (2005), S. 121 - 124.

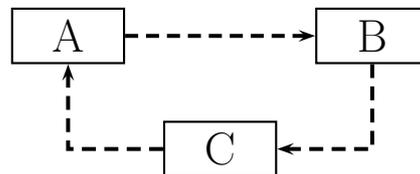
<sup>33</sup>Vgl. Vogel, O.: *Software-Architektur* (2005), S. 124.

<sup>34</sup>Vgl. Vogel, O.: *Software-Architektur* (2005), S. 124.

<sup>35</sup>Vgl. Vogel, O.: *Software-Architektur* (2005), S. 126.

<sup>36</sup>Vgl. Starke, G.: *Effektive Software-Architekturen* (2005), S. 131; Vogel, O.: *Software-Architektur* (2005), S. 133.

## Vermeidung von zyklischen Abhängigkeiten



**Abbildung 2.2:** Zyklische Abhängigkeit von Bausteinen.

(Quelle: Starke, G.: *Effektive Software-Architekturen* (2005), S. 136.)

Abbildung 2.2 zeigt ein Beispiel für eine zyklische Abhängigkeit. Diese erschweren die Wartbarkeit und Erweiterbarkeit/Änderbarkeit eines Systems. Bei einer Änderung an Modul A müssen im Beispiel aufgrund der Abhängigkeit auch Module B und C getestet werden. Zyklische Abhängigkeiten sollen daher vermieden werden.<sup>37</sup>

### 2.2.2 Architekturmuster zur Sensordatenfusion von Umfoldsensoren

Ein Blick über aktuelle Veröffentlichungen im Bereich der Sensordatenfusion für Fahrerassistenzsysteme zeigt, dass sich bisher keine allgemein anerkannte Architektur zur Sensordatenfusion von Umfoldsensoren für Fahrerassistenzsysteme herausgebildet hat.<sup>38</sup> Vielmehr existiert eine Vielzahl verschiedener Lösungsansätze, die sich z.T. in ihren Grundannahmen unterscheiden.

Aufgrund der Fülle an unterschiedlichen Fusionskonzepten kann an dieser Stelle keine vollständige Auflistung aller bestehenden Arbeiten erfolgen. Von daher werden im Folgenden etablierte, allgemeine Architekturmuster aus der Grundlagenliteratur zur Sensordatenfusion von Umfoldsensoren<sup>39</sup> aufgezeigt, auf welche sich die speziellen Architekturmuster aus dem Bereich der Fahrerassistenzsysteme zurückführen lassen. Im Text wird dabei auf die Vor- und Nachteile sowie die Eigenschaften der jeweiligen Konzepte eingegangen.

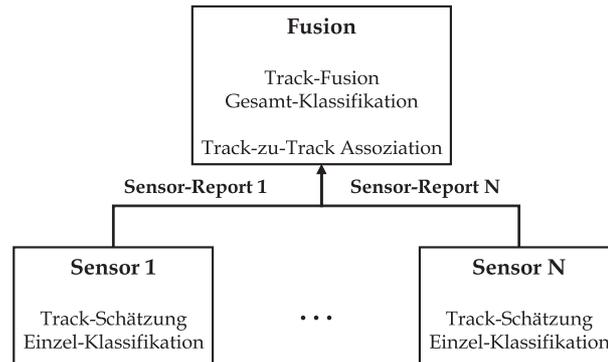
<sup>37</sup>Vgl. Starke, G.: *Effektive Software-Architekturen* (2005), S. 136.

<sup>38</sup>Siehe hierzu z.B. Dietmayer, K.; Kirchner, A.; Kämpchen, N.: *Fusionsarchitekturen* (2005); Naab, K.: *Sensorik- und Signalverarbeitungsarchitekturen* (2004); EU-Projekte ProFusion1 bzw. ProFusion2 N.N.: [www.prevent-ip.org](http://www.prevent-ip.org) (2007).

<sup>39</sup>Siehe z.B. Hall, D. L.: *Handbook of multisensor data fusion* (2001); Klein, L. A.: *Sensor and data fusion* (1999); Klaus, F.: *Techniken und Methoden der Multisensor-Datenfusion* (2004); Bar-Shalom, Y.; Li, X.-R.: *Multitarget-multisensor tracking* (1995); Bar-Shalom, Y.: *MMT Vol 3: Applications and advances* (2000); Blackman, S. S.; Popoli, R.: *Design and analysis of modern tracking systems* (1999); Brooks, R. R.; Iyengar, S. S.: *Multi sensor fusion* (1998); Waltz, E.; Llinas, J.: *Multisensor data fusion* (1990).

## Dezentral - Zentral - Hybrid

Die Einteilung in dezentrale, zentrale und hybride Fusion bezieht sich auf die Bausteinsicht. Sie basiert auf dem Grad der Datenverarbeitung in den Sensoren, den Ergebnissen der Datenverarbeitung in den Sensoren und der Stelle der Zusammenführung der Daten im Fusionsprozess<sup>40</sup> und wird meist im Zusammenhang mit der Track-Schätzung verwendet.<sup>41</sup>



**Abbildung 2.3:** *Dezentrale Architektur.*

Abbildung 2.3 zeigt eine **dezentrale Architektur**.<sup>42</sup> In den Sensoren wird individuell die Objekt-Diskriminierung und Track-Schätzung vollzogen. Die Ergebnisse werden in einem zentralen Baustein zusammengeführt, ggf. mit einer Rückführung von Ergebnissen der zentralen Fusion zu den Sensoren.<sup>43</sup> In diesem Falle kann jeder dezentrale Baustein zusätzlich die Funktionen des zentralen Bausteins übernehmen, so dass eine Redundanz erzielt wird.<sup>44</sup>

In Bezug auf die Objekt-Diskriminierung ist diese Form der Architektur optimal, insofern die Sensoren hinsichtlich dieser Operation orthogonal zueinander sind.<sup>45</sup> Dies ist z.B. der Fall wenn Sensorprinzipien auf Basis unterschiedlicher physikalischer Effekte genutzt werden, die keine Fehlerkennungen aufgrund der gleichen Phänomene auslösen.<sup>46</sup> Für die Zusammenführung werden dabei zwei Informationen benötigt, zum einen die Diskriminierungsentscheidung, zum anderen ein Maß für die Güte der Entscheidung.<sup>47, 48</sup>

Auch für die Track-Schätzung kann die Architektur optimal im Sinne der Minimierung der Schätzfehler bzgl. der wahren Zustände sein.<sup>49</sup> Dies allerdings nur unter relativ einschrän-

<sup>40</sup>Vgl. Klein, L. A.: *Sensor and data fusion* (1999), S. 68-69.

<sup>41</sup>Siehe z.B. Hall, D.; Llinas, J.: *An introduction to multisensor data fusion* (1997), S. 15.

<sup>42</sup>In der Literatur auch als sensor-level fusion, autonomous fusion, distributed fusion, post-individual sensor processing fusion bezeichnet; siehe Klein, L. A.: *Sensor and data fusion* (1999), S. 68.

<sup>43</sup>Siehe z.B. Bar-Shalom, Y.; Li, X.-R.: *Multitarget-multisensor tracking* (1995), S. 435.

<sup>44</sup>Siehe z.B. Bar-Shalom, Y.; Li, X.-R.: *Multitarget-multisensor tracking* (1995), S. 466.

<sup>45</sup>Auch als „orthogonale Sensoren“ bezeichnet.

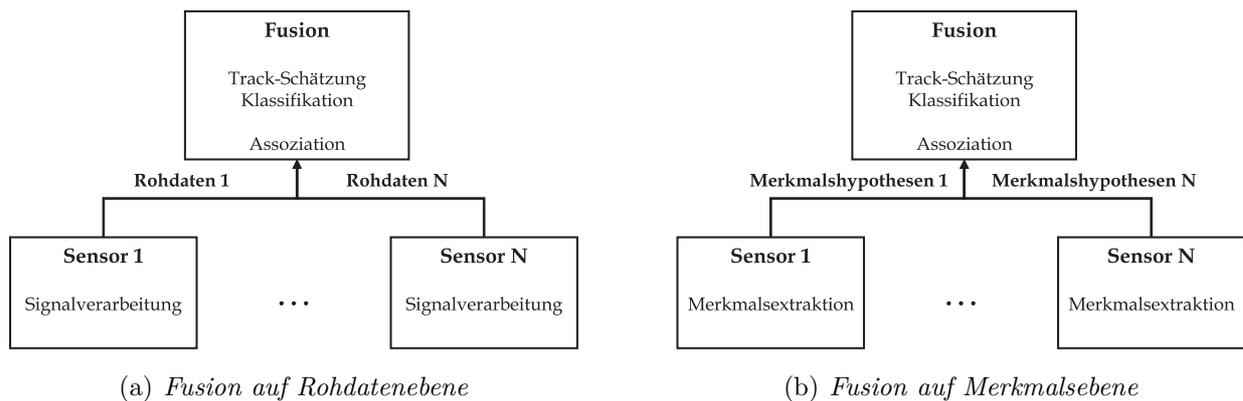
<sup>46</sup>Siehe Robinson, G.; Aboutalib, A.: *Trade-off analysis of multisensor fusion levels* (1990), zitiert in Klein, L. A.: *Sensor and data fusion* (1999), S. 69.

<sup>47</sup>Vgl. Klaus, F.: *Techniken und Methoden der Multisensor-Datenfusion* (2004), S. 71. Hier wird auch ein Überblick über Algorithmen zur Fusion angegeben.

<sup>48</sup>Siehe auch Abschnitt 2.2.2: Fusion auf Entscheidungsebene.

<sup>49</sup>Siehe z.B. Bar-Shalom, Y.; Li, X.-R.: *Multitarget-multisensor tracking* (1995), S. 468.

kenden Voraussetzungen. Zum einen müssen die Sensoren die gleichen Zustände schätzen, was wiederum voraussetzt, dass für jeden der Sensoren die Beobachtbarkeit<sup>50</sup> der Zustände gegeben ist.<sup>51</sup> Zum anderen muss bei Anwendung der Fusion per Kalmanfilter in den Sensoren das gleiche Systemmodell inkl. Systemrauschen verwendet und die Abhängigkeit der Fehler bei der Fusion berücksichtigt werden. Sind die Messzeitpunkte der Sensoren zudem unterschiedlich, ergeben sich wiederum nur näherungsweise optimale Lösungen hinsichtlich der erreichbaren Genauigkeit.<sup>52</sup>



**Abbildung 2.4:** Zentrale Architektur.

Abbildung 2.4 zeigt eine **zentrale Architektur**.<sup>53</sup> Die Daten in den Sensoren werden nur minimal vorverarbeitet (Merkmals- oder Rohdatenebene) und dann in einem zentralen Baustein zusammengeführt, ggf. mit einer Rückführung zu den Sensoren.<sup>54</sup>

Hinsichtlich der Objekt-Diskriminierung ist diese Form der Architektur der dezentralen Architektur überlegen, wenn die Sensoren nicht orthogonal zueinander sind. Sind die Sensoren orthogonal zueinander, unterscheiden sich die Ergebnisse nicht.<sup>55</sup>

Für die Track-Schätzung ist die zentrale Architektur optimal, ohne die einschränkenden Voraussetzungen bei der dezentralen Architektur. Zudem lassen sich auch Messungen, die nicht vom gleichen Zeitpunkt herrühren, optimal zusammenführen.<sup>56</sup>

Die Hauptnachteile der zentralen Architektur sind zum einen die geringere Flexibilität, da bei jeder Erweiterung die interne Struktur im zentralen Baustein geändert werden muss, zum

<sup>50</sup>Siehe hierzu z.B. Lunze, J.: *Regelungstechnik 2* (2006), S. 449; Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 28; Chen, Z.: *Local observability* (1991).

<sup>51</sup>Mit einer Monokamera lässt z.B. die Entfernung nicht schätzen, ohne einschränkende Modellannahmen (ebene Fahrbahn) zu treffen; siehe z.B. Thomanek, F.: *Visuelle Erkennung und Zustandsschätzung von Straßenfahrzeugen* (1996).

<sup>52</sup>Siehe Bar-Shalom, Y.; Li, X.-R.: *Multitarget-multisensor tracking* (1995), S. 447.

<sup>53</sup>In der Literatur auch als central-level fusion, centralized fusion, pre-individual sensor processing fusion bezeichnet; siehe Klein, L. A.: *Sensor and data fusion* (1999), S. 68.

<sup>54</sup>Siehe Klein, L. A.: *Sensor and data fusion* (1999), S. 72.

<sup>55</sup>Siehe Robinson, G.; Aboutalib, A.: *Trade-off analysis of multisensor fusion levels* (1990), zitiert in Klein, L. A.: *Sensor and data fusion* (1999), S. 72.

<sup>56</sup>Siehe Bar-Shalom, Y.; Li, X.-R.: *Multitarget-multisensor tracking* (1995), S. 468.

anderen das hohe Datenvolumen, welches auf den Schnittstellen zwischen Sensorbausteinen und Fusionsbaustein anfällt.<sup>57</sup>

Bei der **hybriden Architektur** werden der zentrale und der dezentrale Ansatz kombiniert. Der zentrale Fusionsbaustein kann neben den minimal vorverarbeiteten Daten (Rohdaten) auch bereits in den Sensoren vorverarbeitete Daten (Tracks) bekommen. Diese können wiederum zusätzlich Eingang für einen dezentralen Fusionsbaustein im gleichen System sein. Die Ergebnisse dieses dezentralen Bausteins können in den Fusionsalgorithmus des zentralen Fusionsbausteins mit einfließen.<sup>58</sup>

Bar-Shalom; Li geben als Beispiel für den Einsatz einer hybriden Architektur ein Szenario an, welches in verschiedene Erfassungsbereiche aufgeteilt ist, die jeweils von einer Multisensorplattform erfasst werden. Innerhalb einer Plattform kommt eine zentrale Architektur zum Einsatz, über die Bereiche hinweg wird das Gesamtbild mittels einer dezentralen Architektur ermittelt.<sup>59</sup>

### **Rohdaten-Ebene - Merkmals-Ebene - Entscheidungs-Ebene**

Die Einteilung in Fusion auf Rohdaten-<sup>60</sup>, Merkmals und Entscheidungsebene bezieht sich auf die Auflösung der in den Fusionsalgorithmus eingebrachten Daten und den Grad der Vorverarbeitung der Sensordaten.<sup>61</sup> Sie bezieht sich also auf die Laufzeitsicht und wird üblicherweise im Zusammenhang mit Algorithmen zur Objekt-Diskriminierung verwendet.<sup>62</sup>

Bei der **Fusion auf Rohdatenebene** werden minimal vorverarbeitete und in der Auflösung der beteiligten Sensoren vorliegende Daten (beispielsweise Pixel in der Bildverarbeitung) in einer zentralen Architektur fusioniert. Auf diese Weise können z.B. Informationen aus verschiedenen Spektren (Infrarot, sichtbares Licht) vor einer anschließenden Bildverarbeitung zusammengeführt werden.<sup>63</sup> Vorteil des Ansatzes ist die Verfügbarkeit der vollständigen Information aus den Sensoren, auf die der Fusionsalgorithmus abgestimmt werden kann. Hauptnachteile sind das hohe Datenaufkommen zwischen den Sensoren und dem zentralen Baustein, sowie die erschwerte Änderbarkeit und Erweiterbarkeit der optimierten Algorithmen im zentralen Baustein.

Bei der **Fusion auf Merkmalsebene** werden zunächst die Merkmale extrahiert und dann die Fusion durchgeführt. In einer zentralen Architektur wird so die Kommunikationsbandbreite zwischen Sensorbausteinen und zentralem Baustein auf Kosten eines Informationsverlustes reduziert.

<sup>57</sup>Siehe Klein, L. A.: *Sensor and data fusion* (1999), S. 73.

<sup>58</sup>Vgl. Klein, L. A.: *Sensor and data fusion* (1999), S. 73.

<sup>59</sup>Für Details siehe Bar-Shalom, Y.; Li, X.-R.: *Multitarget-multisensor tracking* (1995), S. 438.

<sup>60</sup>Auch Pixelebene.

<sup>61</sup>Vgl. Klein, L. A.: *Sensor and data fusion* (1999), S. 79.

<sup>62</sup>Siehe z.B. Hall, D.; Llinas, J.: *An introduction to multisensor data fusion* (1997), S. 16.

<sup>63</sup>Vgl. Klein, L. A.: *Sensor and data fusion* (1999), S. 74.

Die **Fusion auf Entscheidungsebene** entspricht der dezentralen Architektur. Im Gegensatz zur Fusion auf Merkmalsebene wird hier bereits in den Sensoren die Objekt-Diskriminierung durchgeführt, also eine Entscheidung getroffen. Die Ergebnisse werden dann in einem zentralen Baustein zusammen mit den Informationen aus der Track-Schätzung kombiniert.<sup>64</sup> Diese muss dabei nicht nach dem Prinzip einer dezentralen Architektur aufgebaut sein.

Tabelle 2.1 fasst die Architekturprinzipien *Dezentral - Zentral - Hybrid* und *Pixel-Ebene - Merkmals-Ebene - Entscheidungsebene* und deren Abhängigkeiten zusammen.

**Tabelle 2.1:** *Fusionsarchitekturen (Bausteinsicht).*

(In Anlehnung an: Hall, D. L.; McMullen, S. A.: *Mathematical techniques in multisensor data fusion* (2004), S. 360-361, siehe auch Klein, L. A.: *Sensor and data fusion* (1999), S. 73.)

Typ	Beschreibung	Fusionsebene	Bemerkung
Zentral	Fusion von Rohdaten	Rohdaten	<ul style="list-style-type: none"> <li>• Minimaler Informationsverlust.</li> <li>• Benötigt im Vergleich höchste Bandbreite zur Kommunikation zwischen Sensorbausteinen und zentralem Baustein.</li> <li>• Optimal bei orthogonalen und nichtorthogonalen Sensoren.</li> </ul>
	Fusion von Merkmalen	Merkmal	<ul style="list-style-type: none"> <li>• Benötigt geringere Kommunikationsbandbreite als bei Fusion auf Rohdatenebene.</li> <li>• Informationsverlust aufgrund der Merkmalsextraktion.</li> <li>• Bei nichtorthogonalen Sensoren können die Vorteile der Fusion auf Rohdatenebene nicht genutzt werden.</li> </ul>
Dezentral	Fusion von Zustandsvariablen und Diskriminierungsentscheidungen	Entscheidungsebene	<ul style="list-style-type: none"> <li>• Informationsverlust aufgrund der Merkmalsextraktion.</li> <li>• Optimale Objekt-Diskriminierung bei orthogonalen Sensoren.</li> <li>• Optimale Track-Schätzung nur unter restriktiven Bedingungen.</li> <li>• Abhängigkeiten der in den Sensorbausteinen ermittelten Ergebnisse muss bei Fusion berücksichtigt werden.</li> <li>• Redundanz kann erzielt werden, indem in mehreren dezentralen Bausteinen die Fusion berechnet wird.</li> </ul>
Hybrid	Kombination von Zentral und Dezentral	Kombination aller Ebenen möglich	<ul style="list-style-type: none"> <li>• Kombiniert die Eigenschaften der zentralen und dezentralen Architektur.</li> <li>• Im Vergleich höhere Komplexität der Architektur.</li> </ul>

<sup>64</sup>Vgl. Klein, L. A.: *Sensor and data fusion* (1999), S. 76.

## Synchronisiert - Unsynchronisiert

In der Laufzeitsicht kann eine Unterscheidung in synchronisierte und unsynchronisierte Sensoren getroffen werden. Die Unterscheidung bezieht sich auf den zeitlichen Ablauf, in dem die Daten in den Sensoren aufgezeichnet werden.<sup>65</sup>

Bei **synchronisierten Sensoren** ist die Datenakquisition zeitlich aufeinander abgestimmt. Ein Spezialfall sind synchrone Sensoren, bei denen die Datenaufnahme gleichzeitig stattfindet. Bei **unsynchronisierten Sensoren** findet die Datenaufnahme in einem sensorindividuellen und nicht auf die übrigen Sensoren abgestimmten Takt, der nicht notwendigerweise konstant sein muss, statt.<sup>66</sup>

Nachteil der Synchronisierung ist der zusätzliche Aufwand in Hard- und ggf. Software, Vorteil das bereits im Designprozess bekannte zeitliche Verhalten des Systems.<sup>67</sup>

## Neue Daten - Datenkonstellation - Externes Ereignis

Hinsichtlich des Ereignisses, das eintreten muss, damit die Fusion der Daten durchgeführt wird, kann aus Laufzeitsicht unterschieden werden in: bei Vorliegen neuer Daten, bei Vorliegen einer bestimmten Datenkonstellation und aufgrund eines externen Ereignisses.

Wird jeweils bei Vorliegen **neuer Daten** fusioniert, geht keine Information verloren. Je nachdem, ob mit synchronisierten oder nicht synchronisierten Sensoren gearbeitet wird, müssen im Fusionsprozess Lösungen für die Verarbeitung der Daten gefunden werden, die nicht in der zeitlichen Reihenfolge der Messungen am Fusionsbaustein eintreffen.<sup>68</sup> In einer dezentralen Struktur können die aktuellsten fusionierten Daten zu den Sensoren zurückgeführt werden, so dass in den Bausteinen jeweils die aktuellste Schätzung, z.B. zur Vorkonditionierung von Algorithmen, vorliegt.<sup>69</sup>

Wird beim Auftreten **bestimmter Datenkonstellationen** fusioniert, z.B. immer dann, wenn die Daten von bestimmten Sensoren vorliegen, müssen Ressourcen für eine Zwischenspeicherung der Daten vorgehalten werden. Zudem stehen die fusionierten Daten nicht zum frühestmöglichen Zeitpunkt zur Verfügung. Werden unsynchronisierte Sensoren verwendet

<sup>65</sup>Siehe z.B. Bar-Shalom, Y.; Li, X.-R.: *Multitarget-multisensor tracking* (1995), S.87. Für den Automotive Bereich geben die folgenden Artikel einen Überblick: Narbe, B. et al.: *Datennetzkonzepte - Teil 1* (2003a); Narbe, B. et al.: *Datennetzkonzepte - Teil 2* (2003b); Mauthener, M. et al.: *Out-of-Sequence Measurements* (2006).

<sup>66</sup>Die Bezeichnung asynchron wird in diesem Fall nicht gewählt, da asynchrone Sensoren synchronisiert sind, z.B. abgestimmt abwechselnde Messwertaufnahme.

<sup>67</sup>Auch bei einem unsynchronisierten Sensornetzwerk lassen sich unter bestimmten Voraussetzungen zeitlich beschränkte Vorhersagen über das zukünftige Verhalten machen, siehe z.B. Kämpchen, N.; Dietmayer, K.: *Data synchronization strategies for multi-sensor fusion* (2003).

<sup>68</sup>Siehe z.B. Bar-Shalom, Y.: *Update with out-of-sequence measurements in tracking* (2002); Stüker, D.: *Heterogene Sensordatenfusion* (2004), S. 23.

<sup>69</sup>Siehe auch Abschnitt 2.1, S. 9.

muss entschieden werden, in welcher Form die Daten in den Fusionsprozess eingebracht werden.<sup>70</sup>

Werden die Ergebnisse eines zentralen Fusionsbausteins nicht wieder zu den Sensoren zurückgeführt, so kann die Fusion zu beliebigen Zeitpunkten durch ein **externes Ereignis** ausgelöst werden. Dies ermöglicht die Anpassung der Datenrate an den weiterverarbeitenden Prozess und damit eine Anpassung der Ressourcen, stellt aber im Allgemeinen hinsichtlich der Genauigkeit der Track-Schätzung keine optimale Lösung dar.<sup>71</sup>

### **Originaldaten - Gefilterte Daten - Prädizierte Daten**

Hinsichtlich der Art der Daten, die in den Fusionsprozess einfließen, kann in Originaldaten, gefilterte Daten und prädizierte Daten unterschieden werden (Laufzeitsicht).

Bei **Originaldaten** gehen die zeitlich ungefilterten Daten in den Fusionsprozess ein. Hiermit ist eine optimale Track-Schätzung möglich.

Werden bereits **gefilterte Daten** verwendet (z.B. in einer dezentralen Architektur) kann unter restriktiven Bedingungen eine optimale Track-Schätzung erfolgen.<sup>72</sup> Werden die gefilterten Daten allerdings wie ungefilterte Daten behandelt und einem weiteren Filter zur Schätzung übergeben, entsteht eine Filterkette. Dies führt im Allgemeinen zu längeren Signallaufzeiten. Zudem sind die Messfehler, welche aufgrund der bereits stattgefundenen Filterung eigentlich Schätzfehler sind, nun korreliert. Für ein optimales Ergebnis der Schätzung muss dies in der Modellierung beachtet werden.

Die Verwendung von **prädizierten Daten** (z.B. auf Basis von Modellen) ist ebenfalls möglich. Häufig wird dieses Verfahren verwendet, um zum Vorliegen einer bestimmten Datenkonstellation gesammelte Messdaten auf einen Zeitpunkt zu beziehen und diese zunächst zu so genannten Super-Messungen<sup>73</sup> zusammenzufassen. Bar-Shalom; Li vertreten hierzu die Ansicht, dass dieses Verfahren für unsynchronisierte Sensoren nicht zum optimalen Ergebnis hinsichtlich des erreichbaren Schätzfehlers führt.<sup>74</sup>

### **Parallel - Sequentiell**

Eine weitere in der Literatur zu findende Einteilung betrifft das Fusionsverfahren an sich. Unterschieden wird hier in **parallele Fusion**, bei der die vorliegenden Messungen in einem Schritt fusioniert werden und in **sequentielle Fusion**, bei der die Messungen in mehreren

---

<sup>70</sup>Siehe unten.

<sup>71</sup>Siehe z.B. Bar-Shalom, Y.; Li, X.-R.: *Multitarget-multisensor tracking* (1995), S. 434.

<sup>72</sup>Siehe oben.

<sup>73</sup>Supermeasurements.

<sup>74</sup>„Ad-hoc techniques to 'move' the measurement times to a common time are possible but questionable“, Bar-Shalom, Y.; Li, X.-R.: *Multitarget-multisensor tracking* (1995), S. 436.

aufeinander folgenden Schritten zusammengeführt werden (Laufzeitsicht). Beide Verfahren sind unter der Voraussetzung linearer Systeme und synchronisierter Sensoren äquivalent.<sup>75</sup>

Dietmayer; Kirchner; Kämpchen sprechen bei vorliegen synchronisierter Sensoren und paralleler Fusion auch von expliziter Fusion, bei unsynchronisierten Sensoren und sequentieller Fusion auch von impliziter Fusion.<sup>76</sup>

## 2.3 Grundannahmen für eine Architektur zur Sensordatenfusion für Fahrerassistenzsysteme

Wie in Abschnitt 1.3.3 beschrieben, gibt es kein Verfahren zum Entwurf einer Architektur, welches Optimalität garantiert. Zentrum der Architekturentwicklung in dieser Arbeit bilden daher die folgenden, für Fahrerassistenzsysteme allgemeingültigen Grundannahmen. Die als *stark* bezeichneten Grundannahmen müssen dabei in jedem Architekturentwurf berücksichtigt werden, die übrigen können zur Optimierung des Aufbaus der Architektur herangezogen werden.<sup>77</sup> In der Formulierung der Annahmen werden dabei im Folgenden Algorithmen als unterschiedlich verstanden, wenn sie sich entweder im Programmablaufplan oder im Parametersatz unterscheiden.

**Starke Annahme für funktionsbezogene Mehrfachfilter** Für bestimmte Ungenauigkeiten in Messungen (Sensorrauschen) existieren mindestens zwei verschiedene Assistenzfunktionen, die sich hinsichtlich der Anforderungen an Dynamik und Genauigkeit der Datenfilterung so signifikant unterscheiden, dass kein Algorithmus zur Datenfilterung gefunden werden kann, der einen gemeinsamen Betrieb beider Funktionen mit demselben Algorithmus ermöglicht.

**Schwache Annahme für funktionsbezogene Mehrfachfilter** Für bestimmte Ungenauigkeiten in Messungen (Sensorrauschen) existieren mindestens zwei verschiedene Assistenzfunktionen, die sich hinsichtlich der Anforderungen an Dynamik und Genauigkeit der Datenfilterung so signifikant unterscheiden, dass, wenn nur ein Algorithmus zur Datenfilterung verwendet wird, die Funktionen zwar beide betrieben werden können, allerdings nicht beide gemeinsam optimal.

**Starke Annahme für getrennte Filter zur Datenassoziation** Für bestimmte Ungenauigkeiten in Messungen (Sensorrauschen) existiert mindestens eine Assistenzfunktion, bei der sich die Anforderungen an Dynamik und Genauigkeit für die Datenfilterung bzgl. der Funktionalität und bzgl. der Datenassoziation so signifikant unterscheiden, dass kein Algorithmus zur Datenfilterung gefunden werden kann, der die Erfüllung beider Aufgaben mit demselben Algorithmus ermöglicht.

<sup>75</sup>Siehe z.B. Bar-Shalom, Y.; Li, X.-R.: *Multitarget-multisensor tracking* (1995), S. 88-89.

<sup>76</sup>Siehe Dietmayer, K.; Kirchner, A.; Kämpchen, N.: *Fusionsarchitekturen* (2005).

<sup>77</sup>Die Grundannahmen „überdauern“ damit die Designfreiheiten (und Fehler) des Architekturentwurfs.

**Schwache Annahme für getrennte Filter zur Datenassoziation** Für bestimmte Ungenauigkeiten in Messungen (Sensorrauschen) existiert mindestens eine Assistenzfunktion, bei der sich die Anforderungen an Dynamik und Genauigkeit für die Datenfilterung bzgl. der Funktion der Anwendung und bzgl. der Datenassoziation so signifikant unterscheiden, dass, wenn nur ein Algorithmus zur Datenfilterung verwendet wird, zwar beide Aufgaben gelöst werden können, allerdings nicht gemeinsam optimal.

**Annahme für das Ausreichen eines Einzelfilters für die Datenassoziation** Für alle Ungenauigkeiten in Messungen und für jede Kombination von Assistenzfunktionen kann ein Algorithmus zur Datenfilterung gefunden werden, der die Anforderungen an Dynamik und Genauigkeit für die Datenassoziation in so weit erfüllt, dass der Betrieb aller Funktionen für den Nutzer nicht merkbar schlechter ermöglicht wird, als wenn jeder Funktion ein eigenes Filter für die Datenassoziation zugeordnet werden würde.

**Annahme zur Abhängigkeit der Datenassoziation von den Rohdaten** Es existieren Sensoren, bei welchen die Datenassoziation unter Einbeziehung von Rohdaten signifikant weniger fehlerhafte Zuordnungen mit sich bringt, als eine Datenassoziation ohne die Verwendung von Rohdaten.

Die Gültigkeit der Annahmen wird in den folgenden Kapiteln 3 und 4 dargelegt. Im Anschluss wird in Kapitel 5 eine Basis-Systemarchitektur vorgestellt, die auf diesen Grundannahmen aufbaut.

## 3 Datenfilterung für unterschiedliche Fahrerassistenzsysteme

### 3.1 Ziele der Datenfilterung

Ein Filter bzw. Schätzer<sup>1</sup> wird eingesetzt, um auf Basis von Messungen Informationen zu verbessern oder um neue Informationen zu gewinnen.<sup>2</sup> Beispiele sind:

- Trennung von Signal und Störung.<sup>3</sup>
- Rekonstruktion von Zustandsgrößen, die nicht direkt gemessen werden.<sup>4</sup>

Die Gestaltung und Anpassung der Parameter des Filters findet dabei nach für den jeweiligen Fall festzulegenden Optimierungskriterien statt.<sup>5</sup>

Soll z.B. die ungestörte Größe möglichst exakt geschätzt werden, kann das Optimierungskriterium ein Abstandsmaß zwischen geschätzter Größe (im Folgenden mit einem Dach ( $\hat{\cdot}$ ) gekennzeichnet) und wahrer Größe sein, wie z.B. bei der Herleitung des (linearen) Kalmanfilters, das den mittleren quadratischen Fehler der geschätzten Zustandsvariablen eines dynamischen Systems minimiert.<sup>6</sup>

Ist der Schätzer Teil eines Regelkreises,<sup>7</sup> beeinflusst er das dynamische Verhalten des Gesamtsystems.<sup>8</sup> Die Parameter des Schätzers müssen in diesem Fall an die Anforderungen des Regelkreises angepasst werden. Hierbei muss ein Kompromiss zwischen der Dynamik des Filters und dem erzielbaren Schätzfehler eingegangen werden.<sup>9</sup> Wird als Regler ein Zustandsregler eingesetzt, sichert das Separationstheorem<sup>10</sup> zumindest die Stabilität des Gesamtsystems, insofern der Schätzer stabil ist. Regler und Beobachter können daher getrennt voneinander

<sup>1</sup>Die Begriffe werden im Folgenden synonym verwendet.

<sup>2</sup>Vgl. Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 2; Hänsler, E.: *Statistische Signale* (1997), S. 189.

<sup>3</sup>Siehe z.B. Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 2.

<sup>4</sup>Siehe z.B. Föllinger, O.: *Regelungstechnik* (1990), S. 501.

<sup>5</sup>Siehe z.B. Hänsler, E.: *Statistische Signale* (1997), S. 190-198.

<sup>6</sup>Siehe z.B. Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 200-210.

<sup>7</sup>Je nach Art der auftretenden Störung wird zur Rekonstruktion von Zustandsgrößen ein Beobachter anstatt eines Schätzers bzw. Filters verwendet. Siehe hierzu die Ausführungen in Föllinger, O.: *Regelungstechnik* (1990), S. 513.

<sup>8</sup>Siehe z.B. Lunze, J.: *Regelungstechnik 2* (2006), S. 339 bzw. Föllinger, O.: *Regelungstechnik* (1990), S. 517.

<sup>9</sup>Siehe z.B. Lunze, J.: *Regelungstechnik 2* (2006), S. 336-338.

<sup>10</sup>Siehe z.B. Lunze, J.: *Regelungstechnik 2* (2006), S. 335 bzw. Föllinger, O.: *Regelungstechnik* (1990), S. 514.

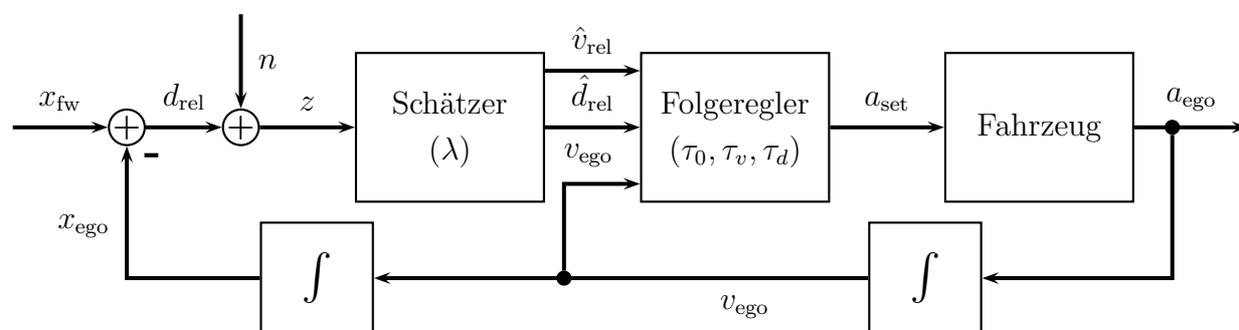
entworfen werden, wobei die Eigenwerte des Schätzers in Abhängigkeit der Eigenwerte des Reglers gewählt werden sollen.<sup>11</sup>

Zur Reduzierung der Komplexität des Gesamtsystems ist es wünschenswert, nur ein gemeinsames Filter für die Schätzung der Zustandsgrößen von Umfeldobjekten für alle Anwendungen in der Architektur vorzusehen. In den folgenden Abschnitten wird anhand der Auslegung eines Schätzers für eine adaptive Fahrgeschwindigkeitsregelung und für ein System zur Unfallvermeidung gezeigt, dass diese Forderung nur eingeschränkt haltbar ist.

## 3.2 Auslegung eines Kalmanfilters mit festen Parametern für eine adaptive Fahrgeschwindigkeitsregelung

Eine adaptive Fahrgeschwindigkeitsregelung<sup>12</sup> (auch Adaptive Cruise Control, ACC) stellt eine Erweiterung der konventionellen Fahrgeschwindigkeitsregelung dar.

„An enhancement to conventional cruise control systems [...], which allows the subject vehicle to follow a forward vehicle at an appropriate distance by controlling the engine and/or power train and potentially the brake.“<sup>13</sup>



**Abbildung 3.1:** Vereinfachter Aufbau eines ACC Regelkreises für die Folgeregelung. In Klammern sind die Parameter der jeweiligen Elemente angegeben.

Abbildung 3.1 zeigt ein vereinfachtes Signalfussdiagramm eines ACC Regelkreises für den Betriebszustand Folgeregelung<sup>14</sup>. Die Eigengeschwindigkeit  $v_{\text{ego}}$  wird einem vorausfahrenden Fahrzeug (Führungsfahrzeug) angepasst und ein über die Zeitlücke  $\tau_0$  definierter Folgeabstand  $d_{\text{set}} = \tau_0 v_{\text{ego}}$  eingeregelt.

<sup>11</sup>Siehe z.B. Föllinger, O.: *Regelungstechnik* (1990), S. 513 bzw. das Entwurfsverfahren in Lunze, J.: *Regelungstechnik 2* (2006), S. 337.

<sup>12</sup>Siehe ISO-Norm: *Adaptive Cruise Control systems* (2002a) oder z.B. Bauer, H.: *Adaptive Fahrgeschwindigkeitsregelung ACC* (2002).

<sup>13</sup>ISO-Norm: *Adaptive Cruise Control systems* (2002a), S. 1.

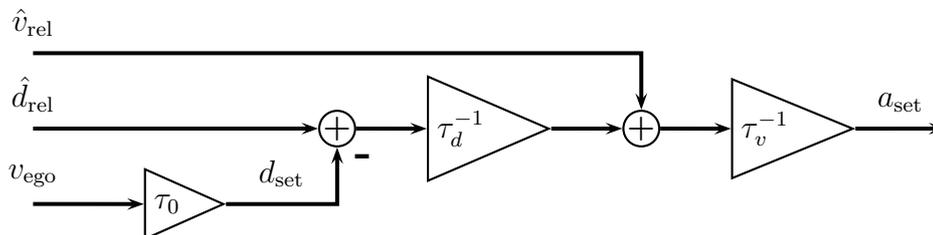
<sup>14</sup>Für ein vollständiges Zustandsdiagramm zur ACC-Funktionsbeschreibung siehe z.B. Witte, S.: *Simulationsuntersuchungen ACC* (1996), S. 14.

$x_{\text{fw}}$  gibt im Modell die absolute Position des Führungsfahrzeugs an,  $x_{\text{ego}}$  die absolute Position des geregelten Fahrzeugs.  $a_{\text{set}}$  stellt die Sollbeschleunigung und  $a_{\text{ego}}$  die Istbeschleunigung des geregelten Fahrzeugs dar.  $\tau_v$  und  $\tau_d$  sind Parameter des Reglers, die weiter unten erläutert werden.

Der Relativabstand  $d_{\text{rel}}$  und die Relativgeschwindigkeit  $v_{\text{rel}}$  werden aus der Messgröße  $z$  über einen Schätzer bestimmt. Die Messgröße  $z$  ergibt sich dabei aus der Addition des Relativabstands  $d_{\text{rel}}$  mit einem als normalverteiltem weißen Rauschen  $n \sim \mathcal{N}(0, \sigma_n)$  angenommenen Messfehler. Mit der Größe  $\sigma_n$  wird damit die Messgenauigkeit des verwendeten Sensors modelliert, im Folgenden wird zur Vereinfachung direkt vom Sensorrauschen  $\sigma_n$  gesprochen.

Bei der Auslegung des Folgereglers muss u.a. ein Kompromiss zwischen Regelgeschwindigkeit, Komfort und Kolonnenstabilität eingegangen werden.<sup>15</sup> Eine Lösung bieten nichtlineare Regelalgorithmen<sup>16</sup> oder die Nutzung mehrerer unterschiedlicher Regler, die für verschiedene Situationen optimiert wurden und zwischen denen situationsabhängig umgeschaltet wird.<sup>17</sup>

Im Folgenden wird die stationäre Folgeregelung betrachtet, d.h. Einschwingvorgänge von Regler und Schätzer sind bereits abgeklungen und es treten nur kleine Abweichungen in Abstand und Relativgeschwindigkeit auf. Der Folgeregler lässt sich in diesem Fall durch einen linearisierten Ansatz gemäß Abbildung 3.2 beschreiben.<sup>18</sup>



**Abbildung 3.2:** Linearisierter ACC-Folgeregler für die stationäre Folgeregelung.

Die Parameter  $\tau_d$  und  $\tau_v$  beeinflussen das Verhalten des Reglers bzgl. Abweichungen vom Sollwert des Relativabstands  $d_{\text{set}}$  und einer Relativgeschwindigkeit  $v_{\text{rel}} \neq 0$ . In der Praxis<sup>19</sup> hat es sich dabei als sinnvoll erwiesen, dass eine Relativgeschwindigkeit von  $|v_{\text{rel}}| \approx 1$  m/s die gleiche Sollbeschleunigung  $a_{\text{set}}$  bewirkt, wie eine Abstandsabweichung  $|d_{\text{rel}} - d_{\text{set}}| \approx 5$  m . . . 10 m. Zudem soll eine Abstandsabweichung von 20 m, was in etwa mit der Entfernung eines vor dem geregelten Fahrzeug einsicherenden Fahrzeugs übereinstimmt, zu einer Verzögerung von ca. 1 m/s<sup>2</sup> führen. Dies entspricht annähernd dem Gaswegnehmen bzw. einer leichten Brem-

<sup>15</sup>Siehe z.B. Winner, H.: *Adaptive Cruise Control* (1999), S. 30.25 und Witte, S.: *Simulationsuntersuchungen ACC* (1996), S. 122-133.

<sup>16</sup>Siehe z.B. Winner, H.: *Adaptive Cruise Control* (1999), S. 30.25.

<sup>17</sup>Siehe z.B. Mayr, R.: *Regelungsstrategien für die automatische Fahrzeugführung* (2001), S. 75 oder Mayr, R.; Bauer, O.: *Safety issues in intelligent cruise control* (1999).

<sup>18</sup>Siehe Witte, S.: *Simulationsuntersuchungen ACC* (1996), S. 120 und S. A-18.

<sup>19</sup>Siehe Winner, H.: *Adaptive Cruise Control* (1999), S. 30.26.

sung und wird mit  $\tau_v \tau_d = 20 \text{ s}^2$  erfüllt. Für den stationären Folgebetrieb wird im Weiteren deshalb  $\tau_v = 4 \text{ s}$  und  $\tau_d = 5 \text{ s}$  gesetzt, um auch die erst genannten Bedingungen zu erfüllen.<sup>20</sup> Ist nichts anderes erwähnt, gilt für die Zeitlücke  $\tau_0 = 1,5 \text{ s}$ .

Als Schätzer wird ein Kalmanfilter<sup>21</sup> eingesetzt. Da dieses auch für ein System zur Unfallvermeidung eingesetzt werden soll und hierfür die Relativbeschleunigung  $a_{\text{rel}}$  benötigt wird<sup>22</sup>, werden die Zustände Distanz ( $\hat{d}_{\text{rel}}$ ), relative Geschwindigkeit ( $\hat{v}_{\text{rel}}$ ) und relative Beschleunigung ( $\hat{a}_{\text{rel}}$ ) geschätzt. Für die Relativbewegung wird ein Modell zugrunde gelegt, nach dem sich  $a_{\text{rel}}$  in jedem Zeitschritt  $T$  um  $v \sim \mathcal{N}(0, \sigma_v)$  ändert, innerhalb des Zeitschritts aber konstant bleibt. Dieses wird auch als Discrete Wiener Process Acceleration (*DWPA*) Modell bezeichnet.<sup>23</sup>

Im eingeschwungenen Zustand geht dieses Filter in das lineare und zeitinvariante  $\alpha\beta\gamma$ -Filter über, dessen Verhalten nur noch vom dimensionslosen Manöverindex

$$\lambda = \frac{\sigma_v T^2}{\sigma_w} \quad (3.1)$$

abhängt.<sup>24</sup> Die Größe  $\sigma_v$  gibt dabei die Standardabweichung des Modellrauschens an und modelliert das Beschleunigungsinkrement über der Abtastperiode  $T$ . Die Größe  $\sigma_w$  ist die modellierte Standardabweichung des Messrauschens der Entfernungsmessung.<sup>25</sup> Zur Vereinfachung wird im Folgenden auch direkt vom Modellrauschen  $\sigma_v$  und dem Messrauschen  $\sigma_w$  gesprochen.

Je höher der Manöverindex  $\lambda$  gewählt wird, desto schneller folgen die geschätzten Größen den realen, das Filter wird dynamischer. Im Gegenzug sinkt die Genauigkeit der Schätzung, da das Sensorrauschen weniger stark herausgefiltert wird.

Für die weitere Berechnung wird das System nach Abbildung 3.1 als zeitdiskretes System mit Abtastperiode  $T = 90 \text{ ms}$  angenommen.<sup>26</sup> Die Umsetzung der Sollbeschleunigung  $a_{\text{set}}$  durch das Fahrzeug wird als  $PT_1$  Glied mit der Zeitkonstante  $T_1 = 0,15 \text{ s}$  modelliert<sup>27</sup>, welches durch eine bilineare  $z$ -Transformation in den zeitdiskreten Raum abgebildet wird.<sup>28</sup>

Zunächst wird die Kolonnenstabilität betrachtet. Kolonnenstabilität liegt dann vor, wenn Anregungen in Form von Geschwindigkeitsänderungen des vorausfahrenden Fahrzeugs über die

<sup>20</sup>Siehe auch Witte, S.: *Simulationsuntersuchungen ACC* (1996), S. A-20.

<sup>21</sup>Siehe z.B. Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 200ff.

<sup>22</sup>Siehe Abschnitt 3.3.

<sup>23</sup>Siehe Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 274. Die Gleichungen sind in Abschnitt A.2 auf S. 92 zu finden.

<sup>24</sup>Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 289.

<sup>25</sup>Das modellierte Messrauschen  $\sigma_w$  wird hier absichtlich nicht mit  $\sigma_n$  bezeichnet, da  $\sigma_w$  ein Modellierungsparameter ist und nicht gleich  $\sigma_n$  gewählt werden muss.

<sup>26</sup>Dies entspricht z.B. der Abtastperiode des im Projekt PRORETA verwendeten Laserscanners, siehe Abschnitt 6.1, S. 71.

<sup>27</sup>Siehe Witte, S.: *Simulationsuntersuchungen ACC* (1996), S. A-20.

<sup>28</sup>Siehe Anhang B.1, S. 94.

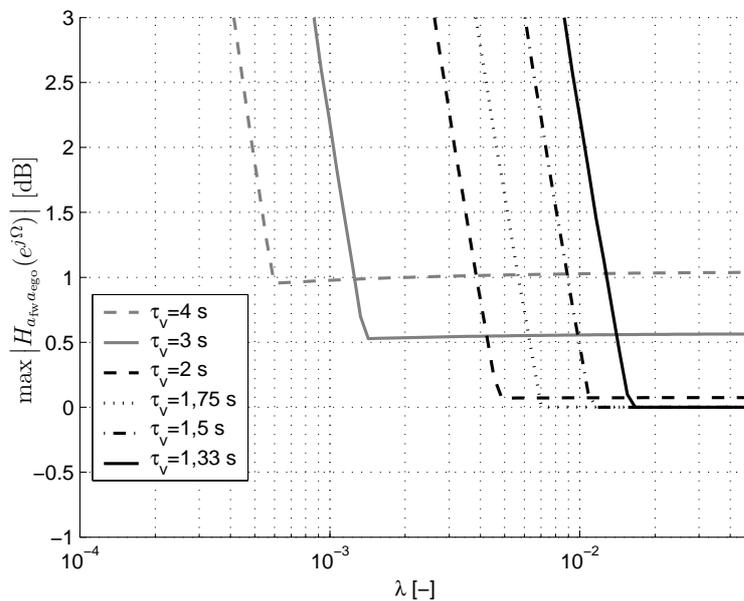
Kolonnenposition abgeschwächt werden.<sup>29</sup> Dies ist erfüllt, wenn für die Übertragungsfunktion der Beschleunigung von vorausfahrendem Fahrzeug zu betrachtetem Fahrzeug gilt:<sup>30</sup>

$$|H_{a_{fw}a_{ego}}(e^{j\Omega})| \leq 1, \text{ mit } 0 \leq \Omega < 2\pi \text{ und } H_{a_{fw}a_{ego}}(z) = \frac{A_{fw}(z)}{A_{ego}(z)}. \quad (3.2)$$

Hierbei gilt im Modell

$$H_{a_{fw}a_{ego}}(z) = \frac{H_{ax}(z)H_{d_{rel}a_{ego}}(z)}{1 + H_{ax}(z)H_{d_{rel}a_{ego}}(z)} = \frac{L(z)}{1 + L(z)}, \quad (3.3)$$

wobei  $H_{d_{rel}a_{ego}}(z)$  die Übertragungsfunktion von  $d_{rel}$  nach  $a_{ego}$  des am Addierer aufgebrochenen Kreises darstellt und  $H_{ax}(z)$  die Übertragungsfunktion von Beschleunigung zu Ort, sowohl für das vorausfahrende Fahrzeug als auch für das betrachtete Fahrzeug.<sup>31</sup>



**Abbildung 3.3:** Einfluss des Manöverindex  $\lambda$  auf die Varianz der Fahrzeugbeschleunigung  $a_{ego}$  und den Zeitverzug des Kalmanfilters (Zeitlücke  $\tau_0 = 1,5$  s).

Abbildung 3.3 zeigt den Maximalwert der Übertragungsfunktion in Abhängigkeit von  $\lambda$  für verschiedene Werte von  $\tau_v$ . Mit den oben gesetzten Parametern  $\tau_v = 4$  s und  $\tau_d = 5$  s erfüllt selbst die Übertragungsfunktion ohne Schätzer die Bedingung (3.2) nicht. Durch eine Verringerung von  $\tau_v$  kann dies erreicht werden. Dies bedeutet, dass der Regler „härter“ ausgelegt ist, zulasten des Komforts.<sup>32</sup> In der praktischen Realisierung wird zwischen den beiden Varianten mithilfe von Kennlinien umgeschaltet.<sup>33</sup> Unterhalb von  $\tau_v \approx 1,72$  s wird der Regler ohne Schätzer kolonnenstabil. Für diesen Wert kann ein Manöverindex  $\lambda_{min,pla}^{acc} \approx 7,0 \cdot 10^{-3}$  gefunden werden, so dass Kolonnenstabilität erreicht wird.

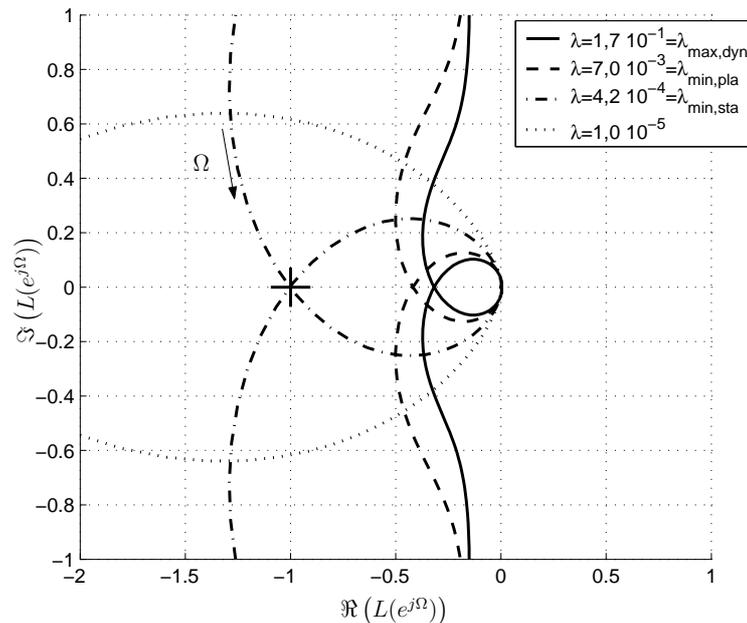
<sup>29</sup>Siehe Wallentowitz, H.: *Sicherheitsanalyse zum automatischen Fahren* (2001), S. 92 bzw. Witte, S.: *Simulationsuntersuchungen ACC* (1996), S. 90ff.

<sup>30</sup>Witte, S.: *Simulationsuntersuchungen ACC* (1996), S. 91.

<sup>31</sup>Die Herleitung dieser Beziehung sowie die zahlenmäßige Berechnung der Übertragungsfunktionen ist im Anhang B.2 auf S. 94 zu finden.

<sup>32</sup>Siehe auch Wallentowitz, H.: *Sicherheitsanalyse zum automatischen Fahren* (2001), S. 93.

<sup>33</sup>Siehe z.B. Wallentowitz, H.: *Sicherheitsanalyse zum automatischen Fahren* (2001), S. 93.



**Abbildung 3.4:** Ortskurven  $L(e^{j\Omega}) = H_{ax}(e^{j\Omega})H_{d_{rel}a_{ego}}(e^{j\Omega})$  für  $\tau_v = 1,75$  s und verschiedene Werte des Manöverindex  $\lambda$ .

Neben der Kolonnenstabilität muss das System aus Regler und Schätzer selbst stabil sein. Bezogen auf den Kolonnenverkehr wird dies auch als Einzelstabilität bezeichnet.<sup>34</sup> Zur Betrachtung kann wiederum Gleichung (3.3) herangezogen werden. Damit die Einzelstabilität erfüllt ist, muss  $H_{a_{fw}a_{ego}}(z)$  stabil sein.  $H_{a_{fw}a_{ego}}(z)$  hat die Form der Übertragungsfunktion eines Standardregelkreises, so dass das Nyquistkriterium für zeitdiskrete Übertragungsfunktionen<sup>35</sup> zur Stabilitätsbetrachtung verwendet werden kann. Die Ortskurve von  $L(e^{j\Omega}) = H_{ax}(e^{j\Omega})H_{d_{rel}a_{ego}}(e^{j\Omega})$  für  $0 \leq \Omega < 2\pi$  darf danach den kritischen Punkt  $-1$  in der  $z$ -Ebene nicht umschließen.

Abbildung 3.4 zeigt die Ortskurven für  $\tau_v = 1,75$  s und verschiedene Werte des Manöverindex  $\lambda$ . Für Werte  $\lambda < \lambda_{min,sta}^{acc} = 4,2 \cdot 10^{-4}$  wird das System instabil, was auf die Trägheit des Schätzers zurückzuführen ist. Der Wert liegt unterhalb des Wertes für Kolonnenstabilität  $\lambda_{min,pla}^{acc}$ , so dass bei Auslegung bzgl. Kolonnenstabilität die Einzelstabilität gegeben ist.<sup>36</sup>

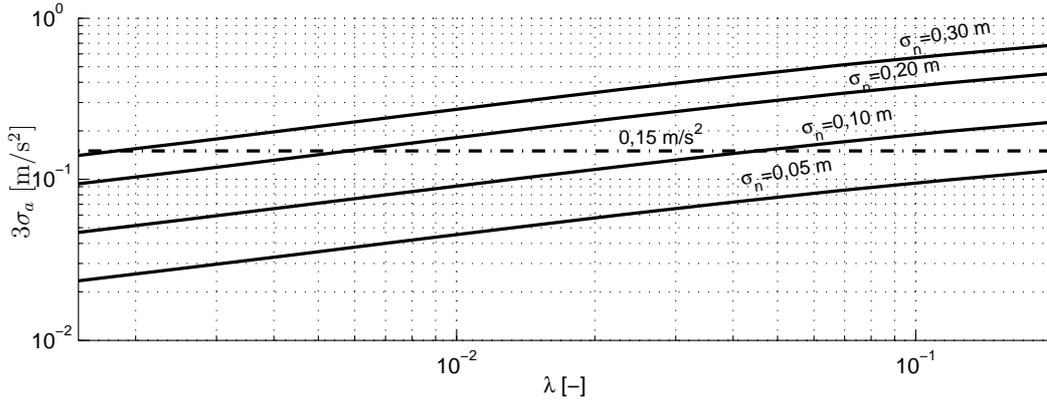
Der Mensch nimmt Beschleunigungen ab einer Amplitude von  $0,15 \text{ m/s}^2$  wahr.<sup>37</sup> Der Einfluss der Störung  $n$  auf die Fahrzeugbeschleunigung  $a_{ego}$  muss daher im stationären Folgebetrieb unterhalb dieses Wertes bleiben, damit das System vom Fahrer nicht als unkomfortabel

<sup>34</sup>Siehe z.B. Witte, S.: *Simulationsuntersuchungen ACC* (1996), S. 91.

<sup>35</sup>Siehe z.B. Lunze, J.: *Regelungstechnik 2* (2006), S. 497.

<sup>36</sup>Zu erkennen ist hieran auch, dass das System für  $\lambda$  in der Nähe von  $\lambda_{min,sta}^{acc}$  dazu neigt, abklingend zu schwingen. Dieser Effekt ist um so ausgeprägter, je kleiner  $\tau_v$  gewählt wird und bei einer sprungförmigen Änderung von  $a_{fw}$  und  $\lambda = \lambda_{min,pla}^{acc}$  schon deutlich im Verlauf von  $a_{ego}$  zu erkennen.

<sup>37</sup>Siehe Meyer-Gramcko, F.: *Verkehrsunfall und Fahrzeugtechnik, Nr. 3, 1990* (1990), zitiert in Huang, B.: *Regelkonzepte zur Fahrzeugführung* (2004), S. 26.



**Abbildung 3.5:** Einfluss des Manöverindex  $\lambda$  auf die Varianz der Fahrzeugbeschleunigung  $a_{\text{ego}}$  (Zeitlücke  $\tau_0 = 1,5$  s).

empfohlen wird:

$$|a_{\text{ego}}| < 0,15 \text{ m/s}^2. \quad (3.4)$$

Die Fahrzeugbeschleunigung  $a_{\text{ego}}$  ist nach dem Modell gemäß Abbildung 3.1 im stationären Folgebetrieb ein stationärer, normalverteilter Zufallsprozess mit Mittelwert Null. Die Varianz  $\sigma_a^2$  des Prozesses berechnet sich daher mit der Impulsantwort von Störung zur Fahrzeugbeschleunigung  $h_{na}(n)$  und dem Sensorrauschen  $\sigma_n$  zu<sup>38</sup>

$$\sigma_a^2 = \sigma_n^2 \sum_{n=0}^{\infty} h_{na}(n)^2. \quad (3.5)$$

Abbildung 3.5 zeigt den  $3\sigma$ -Wert der Fahrzeugbeschleunigung  $a_{\text{ego}}$  in Abhängigkeit des Manöverindex  $\lambda$  für  $\tau_0 = 1,5$  s. Der Betrag der Amplituden von 99,7% der Musterfunktionen von  $a_{\text{ego}}$  liegt unterhalb dieser Grenze. Wird ein Wert  $\lambda = \lambda_{\text{max,com}}^{\text{acc}}$  gewählt, so dass der  $3\sigma$ -Wert gleich  $0,15 \text{ m/s}^2$  ist, wird Bedingung (3.4) somit in 99,7% aller Fälle erfüllt.

Je kleiner das Sensorrauschen  $\sigma_n$ , desto höher ist  $\lambda_{\text{max,com}}^{\text{acc}}$ . Für große Werte von  $\lambda$  wird der Einfluss des Schätzers auf das dynamische Verhalten des Regelkreises allerdings immer geringer. Abbildung 3.6 zeigt das Integral  $e$  der quadrierten Abweichung<sup>39</sup> zwischen der Beschleunigung  $a_{\text{ego}}$  des Regelkreises mit Schätzer und einem idealen Regelkreis ohne Schätzer<sup>40</sup> für verschiedene Werte von  $\lambda$ :

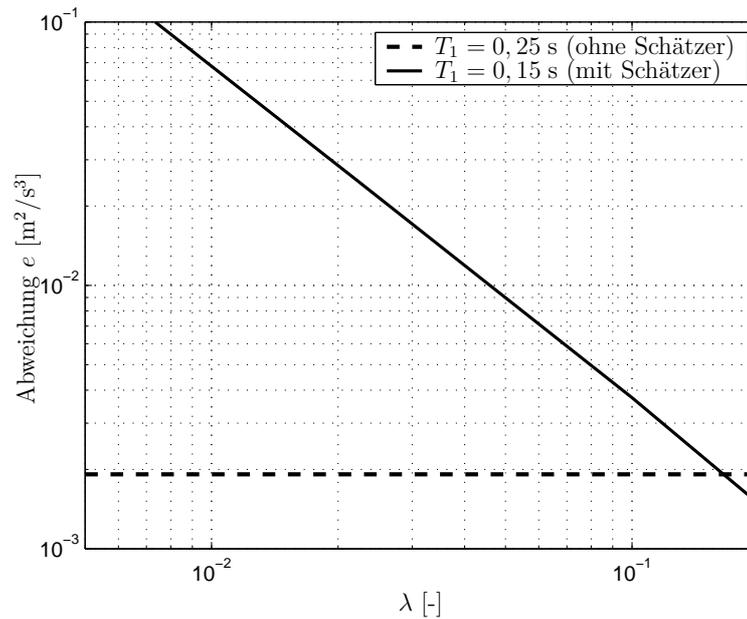
$$e = \int_0^{\infty} (a_{\text{ego}}(t)|_{\text{ohne Schätzer}} - a_{\text{ego}}(t)|_{\text{mit Schätzer}})^2 dt. \quad (3.6)$$

Zugrunde gelegt ist dabei jeweils eine sprungförmige Änderung der Beschleunigung  $a_{\text{fw}}$  von Null auf  $1 \text{ m/s}^2$ . Das betrachtete Fahrzeug ist dabei als  $PT_1$ -Glied mit  $T_1 = 0,15$  s modelliert.

<sup>38</sup>Siehe z.B. Hänsler, E.: *Statistische Signale* (1997), S. 127ff.

<sup>39</sup>Vgl. auch die Bewertung des Regelfehlers in Horn, M.; Dourdoumas, N.: *Regelungstechnik* (2004), S. 181.

<sup>40</sup>Im Modell werden hier die Eingangsgrößen des Reglers nicht über einen Schätzer bestimmt, sondern die wahren Größen direkt verwendet.



**Abbildung 3.6:** Integral der quadrierten Abweichung  $e$  von  $a_{\text{ego}}$  bei sprungförmiger Änderung der Größe  $a_{\text{fW}}$  von Null auf  $1 \text{ m/s}^2$  aufgrund des Beobachters ( $\tau_0 = 1,5 \text{ s}$ ,  $\tau_v = 1,5 \text{ s}$ ,  $\tau_d = 5 \text{ s}$ ).

Zudem gilt  $\tau_v = 1,5 \text{ s}$ , also ein Regler der dynamischer, als für Kolonnenstabilität notwendig ist. Zum Vergleich ist  $e$  zu einem Regelkreis mit gleichem Regler, aber ohne Schätzer und mit  $T_1 = 0,25 \text{ s}$  eingetragen. Dies entspricht einer um ca. 100 ms langsameren Reaktion des betrachteten Fahrzeugs auf eine Beschleunigungsanforderung und liegt in der Größenordnung der Abtastperiode  $T$ . Ab einem Wert von  $\lambda_{\text{max,dyn}}^{\text{acc}} \approx 0,17$  ist der Wert  $e$  des Regelkreises mit  $T_1 = 0,15 \text{ s}$  und Schätzer geringer als der des Regelkreises mit  $T_1 = 0,25 \text{ s}$  ohne Schätzer, so dass es nicht notwendig ist,  $\lambda$  weiter zu erhöhen.

Die Größen  $\lambda_{\text{min,sta}}^{\text{acc}}$ ,  $\lambda_{\text{min,pla}}^{\text{acc}}$ ,  $\lambda_{\text{max,com}}^{\text{acc}}$  und  $\lambda_{\text{max,dyn}}^{\text{acc}}$  legen die Randbedingungen fest, innerhalb derer ein Wert für den Manöverindex gewählt werden kann. In Abbildung 3.12 auf S. 37 wird dies nochmals visualisiert.

### 3.3 Auslegung eines Kalmanfilters mit festen Parametern für ein System zur Unfallvermeidung

Zur Auslegung des Kalmanfilters wird ein Sicherheitssystem betrachtet, welches Aktionen wie Warnungen<sup>41</sup>, Maßnahmen zur Unfallfolgenminderung oder Unfallvermeidung<sup>42</sup> auf Basis der Auswertung der Zeit bis zu einer potentiellen Kollision mit einem vorausfahrenden Fahrzeug  $t_{TTC}$  auslöst. Unter Einbeziehung des Relativabstands  $d_{\text{rel}}$ , der Relativgeschwindigkeit  $v_{\text{rel}}$

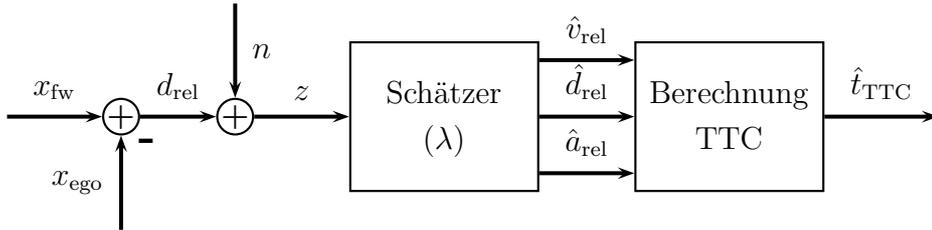
<sup>41</sup>Siehe z.B. ISO-Norm: *Forward vehicle collision warning systems* (2002b).

<sup>42</sup>Siehe hierzu z.B. Kodaka, K. et al.: *Rear-End Collision Velocity Reduction System* (2003); Knothe, F. et al.: *Mercedes Benz CL-Klasse* (2006), S. 801-810.

und der Relativbeschleunigung  $a_{\text{rel}}$  gilt für diese<sup>43</sup>

$$t_{\text{TTC}} = \frac{-v_{\text{rel}} - \sqrt{v_{\text{rel}}^2 - 2a_{\text{rel}}d_{\text{rel}}}}{a_{\text{rel}}}. \quad (3.7)$$

Die Größen  $d_{\text{rel}}$ ,  $v_{\text{rel}}$  und  $a_{\text{rel}}$  werden wie im vorausgehenden Abschnitt über ein Kalmanfilter aus Abstandswerten bestimmt, welchen eine additive Störung  $n$  überlagert ist (siehe Abbildung 3.7).



**Abbildung 3.7:**  $t_{\text{TTC}}$ -Berechnung über einen Schätzer. In Klammern die Parameter der jeweiligen Elemente.

Die Größe  $\hat{t}_{\text{TTC}}$  ist damit eine Zufallsvariable, ebenso wie der Schätzfehler  $\Delta\hat{t}_{\text{TTC}}$

$$\Delta\hat{t}_{\text{TTC}} = t_{\text{TTC}} - \hat{t}_{\text{TTC}}. \quad (3.8)$$

Dieser setzt sich aus zwei Komponenten zusammen:

$$\Delta\hat{t}_{\text{TTC}} = \Delta_n\hat{t}_{\text{TTC}} + \Delta_T\hat{t}_{\text{TTC}}. \quad (3.9)$$

Zum einen wird durch das Kalmanfilter ein systematischer Fehler aufgrund des Zeitverzugs des Filters hervorgerufen ( $\Delta_T\hat{t}_{\text{TTC}}$ ), zum anderen ergibt sich ein zufälliger Fehler ( $\Delta_n\hat{t}_{\text{TTC}}$ ) aufgrund der Störung  $n$ . Die Varianz dieses Fehlers lässt sich dabei wie folgt abschätzen<sup>44</sup>:

$$\begin{aligned} \text{Var}\{\Delta_n\hat{t}_{\text{TTC}}\} \approx & \left(\frac{\partial t_{\text{TTC}}}{\partial d_{\text{rel}}}\right)^2 \sigma_{\hat{d}_{\text{rel}}}^2 + \left(\frac{\partial t_{\text{TTC}}}{\partial v_{\text{rel}}}\right)^2 \sigma_{\hat{v}_{\text{rel}}}^2 + \left(\frac{\partial t_{\text{TTC}}}{\partial a_{\text{rel}}}\right)^2 \sigma_{\hat{a}_{\text{rel}}}^2 + \\ & + 2 \left( \frac{\partial t_{\text{TTC}}}{\partial d_{\text{rel}}} \frac{\partial t_{\text{TTC}}}{\partial v_{\text{rel}}} \sigma_{\hat{d}_{\text{rel}}\hat{v}_{\text{rel}}} + \frac{\partial t_{\text{TTC}}}{\partial d_{\text{rel}}} \frac{\partial t_{\text{TTC}}}{\partial a_{\text{rel}}} \sigma_{\hat{d}_{\text{rel}}\hat{a}_{\text{rel}}} + \frac{\partial t_{\text{TTC}}}{\partial v_{\text{rel}}} \frac{\partial t_{\text{TTC}}}{\partial a_{\text{rel}}} \sigma_{\hat{v}_{\text{rel}}\hat{a}_{\text{rel}}} \right) \end{aligned} \quad (3.10)$$

Die partiellen Ableitungen<sup>45</sup> müssen hierbei am ohne die Störung  $n$  aufgrund des Zeitverzugs des Kalmanfilters gegebenen Arbeitspunkt für (3.7) berechnet werden. Die Varianzen und Kovarianzen der Schätzungen  $\hat{d}_{\text{rel}}$ ,  $\hat{v}_{\text{rel}}$  und  $\hat{a}_{\text{rel}}$  lassen sich dabei analog (3.5) berechnen:<sup>46</sup>

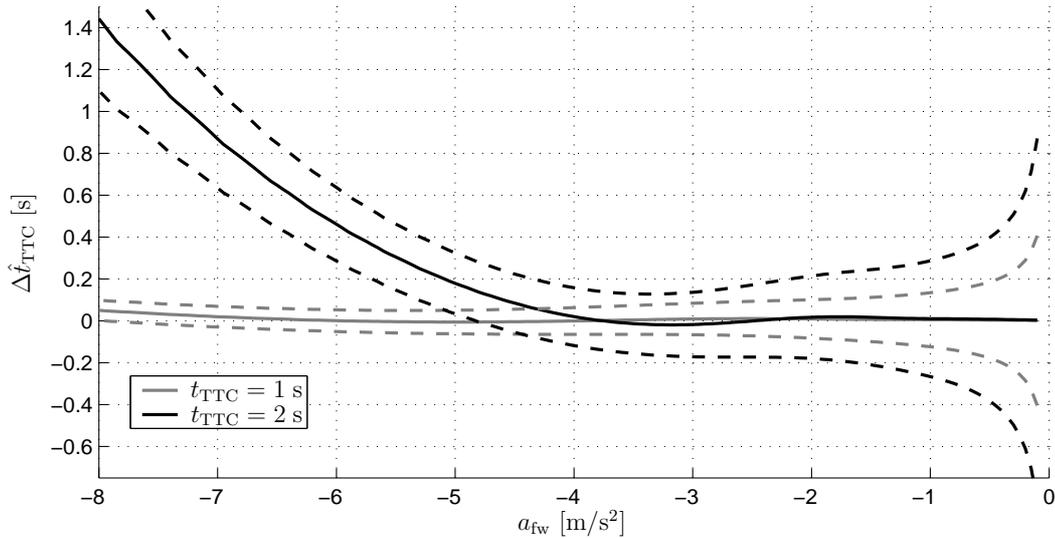
$$\sigma_{ij} = \sigma_n^2 \sum_{n=0}^{\infty} h_{ni}(n)h_{nj}(n) \quad \text{mit } i, j \in \{\hat{d}_{\text{rel}}, \hat{v}_{\text{rel}}, \hat{a}_{\text{rel}}\}. \quad (3.11)$$

<sup>43</sup>Im betrachteten Arbeitspunkt gilt  $d_{\text{rel}} > 0$ ,  $v_{\text{rel}} < 0$  und  $a_{\text{rel}} < 0$ .

<sup>44</sup>Allgemeines Fortpflanzungsgesetz statistischer Messfehler, siehe z.B. Gränicher, H.: *Messung beendet - was nun?* (1996), S. 8-4.

<sup>45</sup>Die Gleichungen der partiellen Ableitungen finden sich in Anhang C auf Seite 96.

<sup>46</sup>Siehe z.B. Hänsler, E.: *Statistische Signale* (1997), S. 148.



**Abbildung 3.8:** Mittlerer Fehler und  $3\sigma$ -Bereiche (gestrichelt) für  $t_{\text{TTC}}$  bei  $\sigma_n = 0,2 \text{ m}$ ,  $\lambda = \lambda_{\text{max,com}}^{\text{acc}} = 5,9 \cdot 10^{-3}$  und  $d_0 = 33,33 \text{ m}$  bzw.  $\tau_0 = 1,5 \text{ s}$ .

Simulationsuntersuchungen zeigen, dass  $\Delta \hat{t}_{\text{TTC}}$  annähernd normalverteilt ist. Daher wird für die weiteren Berechnungen angenommen, dass gilt:

$$\Delta \hat{t}_{\text{TTC}} \sim \mathcal{N}(\Delta_T \hat{t}_{\text{TTC}}, \text{Var}\{\Delta_n \hat{t}_{\text{TTC}}\}). \quad (3.12)$$

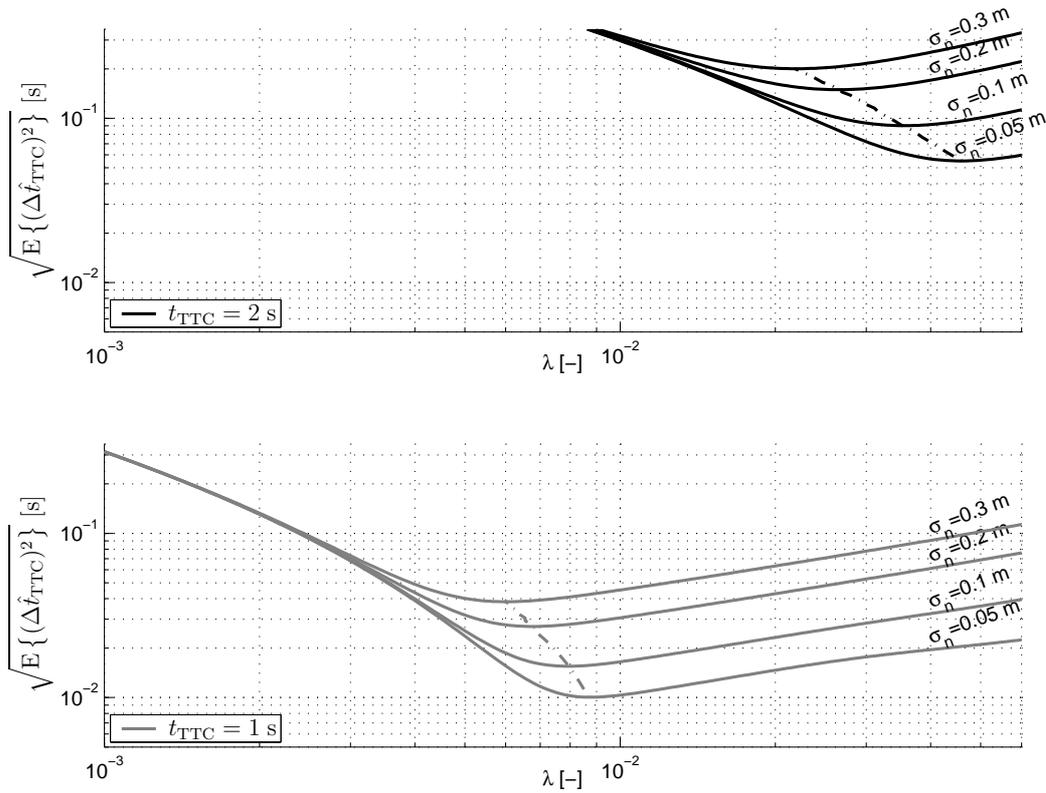
Für die folgenden Betrachtungen gelten die gleichen Voraussetzungen wie in Abschnitt 3.2 und die Abtastperiode des Systems wird wieder auf  $T = 90 \text{ ms}$  gesetzt. Betrachtet wird eine Folgefahrt, bei der zunächst beide Fahrzeuge mit einer Geschwindigkeit von  $80 \text{ km/h}$  fahren und einen konstanten Abstand  $d_0$  gemäß einer Zeitlücke von  $1,5 \text{ s}$  einhalten.<sup>47</sup> Zum Zeitpunkt  $t_0$  verzögert das vorausfahrende Fahrzeug mit der Beschleunigung  $a_{\text{fw}}$ , das betrachtete Fahrzeug hält die Geschwindigkeit.<sup>48</sup>

Abbildung 3.8 zeigt den mittleren Fehler und die  $3\sigma$ -Bereiche für  $t_{\text{TTC}} = 2 \text{ s}$  und  $t_{\text{TTC}} = 1 \text{ s}$ , berechnet nach (3.12) in Abhängigkeit von  $a_{\text{fw}}$  für  $\sigma_n = 0,2 \text{ m}$  und dem für diese Bedingungen optimalen Manöverindex bzgl. Komfort für ACC  $\lambda = \lambda_{\text{max,com}}^{\text{acc}} = 5,9 \cdot 10^{-3}$ . Für Beschleunigungen  $a_{\text{rel}} > -1 \text{ m/s}^2$  steigt der Fehler für beide TTC-Zeiten aufgrund der hohen Empfindlichkeit der TTC-Berechnung in diesem Bereich stark an. Für  $t_{\text{TTC}} = 2 \text{ s}$  und Beschleunigungen  $a_{\text{rel}} < -4 \text{ m/s}^2$  macht sich der systematische Fehler deutlich bemerkbar. Bei Beschleunigungen um  $-8 \text{ m/s}^2$  liegt der maximale Fehler bei über  $1,5 \text{ s}$ .

Durch eine Variation des Manöverindex  $\lambda$  und damit der Dynamik des Schätzers lassen sich  $\Delta_T \hat{t}_{\text{TTC}}$  und  $\Delta_n \hat{t}_{\text{TTC}}$  beeinflussen. Eine Vergrößerung von  $\lambda$  führt dabei zu einem kleineren  $\Delta_T \hat{t}_{\text{TTC}}$  und einem größeren  $\Delta_n \hat{t}_{\text{TTC}}$ . Um einen optimalen Wert für  $\lambda$  zu finden, wird als

<sup>47</sup>Entspricht  $33 \text{ m}$ . Eine typische Situation auf einer Landstraße.

<sup>48</sup>Anmerkung: Bei einer Beschleunigung von  $a_{\text{fw}} = -8 \text{ m/s}^2$  dauert es in diesem Fall  $2,4 \text{ s}$  bis zur Kollision.



**Abbildung 3.9:** Wurzel des mittleren quadratischen Fehlers  $\Delta\hat{t}_{\text{TTC}}$  in Abhängigkeit des Manöverindex  $\lambda$  für  $t_{\text{TTC}} = 2$  s (oben) und  $t_{\text{TTC}} = 1$  s (unten) bei  $d_0 = 33,33$  m bzw.  $\tau_0 = 1,5$  s. Strichpunktiert: Ortskurve der Minima.

Zielfunktion  $ZF_{t_{\text{TTC}}}$  der mittlere quadratische Fehler von  $\hat{t}_{\text{TTC}}$  minimiert:<sup>49</sup>

$$ZF_{t_{\text{TTC}}} = E \{ (\Delta\hat{t}_{\text{TTC}})^2 \} = \int \int (\Delta\hat{t}_{\text{TTC}})^2 f_a(a) f_t(t, a) dt da. \quad (3.13)$$

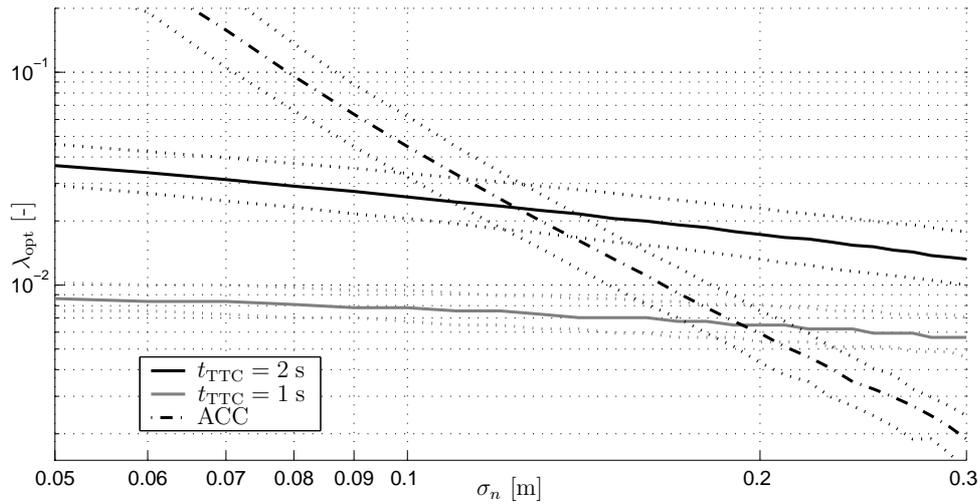
Dabei ist  $f_t(t, a)$  die Wahrscheinlichkeitsdichte des Fehlers  $\Delta\hat{t}_{\text{TTC}}$  nach (3.12) in Abhängigkeit der Beschleunigung  $a_{\text{fw}}$ .  $f_a(a)$  ist die Wahrscheinlichkeitsdichte möglicher Beschleunigungen  $a_{\text{fw}}$ . Diese wird im Folgenden als Gleichverteilung modelliert, so dass alle Beschleunigungen im Einsatzbereich des Systems  $[a_{\text{min}}; a_{\text{max}}]$  gleich gewichtet werden.<sup>50</sup> Hierbei wird  $a_{\text{min}} = -8$  m/s<sup>2</sup> bzw.  $a_{\text{max}} = -1$  m/s<sup>2</sup> gesetzt, da für betragsmäßig kleinere Beschleunigungen der Fehler  $\Delta\hat{t}_{\text{TTC}}$  sehr stark ansteigt und größere Verzögerungen im realen Straßenverkehr selten vorkommen.<sup>51</sup>

Abbildung 3.9 (oben) zeigt die Werte der Zielfunktion in Abhängigkeit des Manöverindex  $\lambda$  und des Messrauschens  $n$  für  $t_{\text{TTC}} = 2$  s, Abbildung 3.9 (unten) für  $t_{\text{TTC}} = 1$  s. Die

<sup>49</sup>Es lassen sich hier auch andere Zielfunktionen definieren. So kann z.B. die Wahrscheinlichkeit minimiert werden, dass der Fehler außerhalb bestimmter Grenzen liegt, wobei dies zu ähnlichen Ergebnissen führt. Allgemeine Betrachtungen zur Wahl der Zielfunktion finden sich z.B. in Hänslers, E.: *Statistische Signale* (1997), S. 189ff.

<sup>50</sup>Hier ließen sich natürlich andere Dichten wählen. In ISO-Norm: *Forward vehicle collision warning systems* (2002b), S. 15 wird eine empirisch ermittelte Verteilung angegeben.

<sup>51</sup>Siehe z.B. ISO-Norm: *Forward vehicle collision warning systems* (2002b), S. 15.

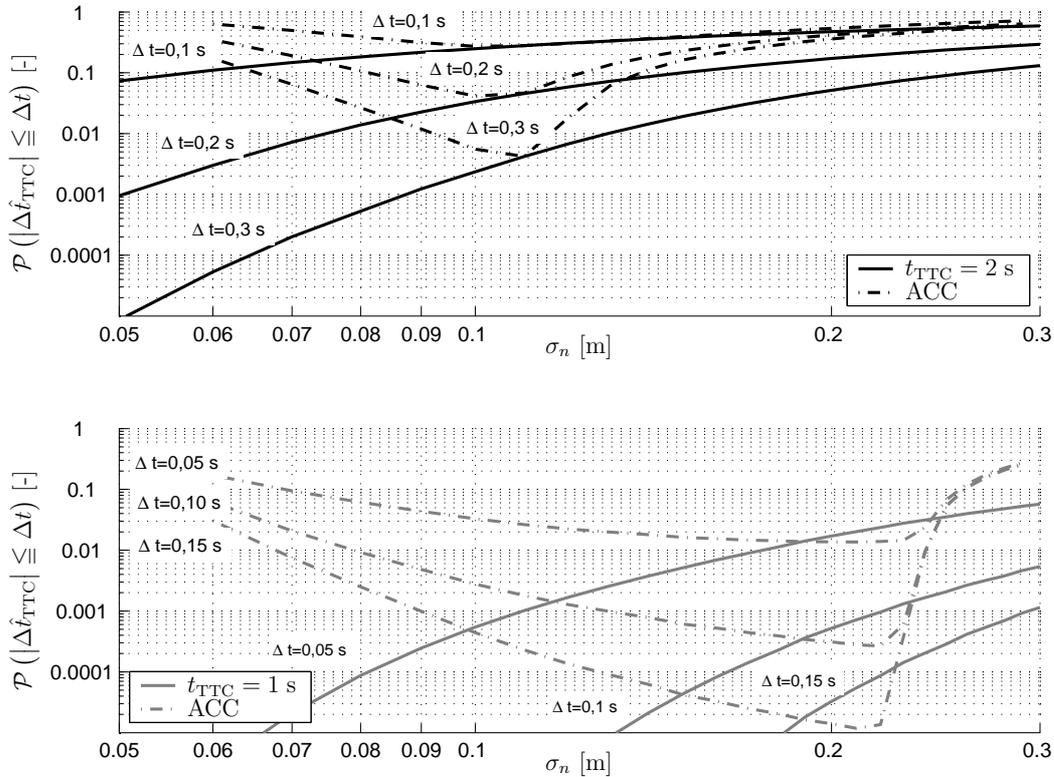


**Abbildung 3.10:** Optimale Werte für den Manöverindex  $\lambda$  für  $t_{\text{TTC}} = 2 \text{ s}$  und  $t_{\text{TTC}} = 1 \text{ s}$  für  $d_0 = 33,33 \text{ m}$  bzw.  $\tau_0 = 1,5 \text{ s}$  sowie für die stationäre Folgeregelung. Gepunktet: Toleranzbereiche bei einer Abweichung von  $\pm 10\%$  von der  $t_{\text{TTC}}$ -Zielfunktion bzw. von der Beschleunigungsschwelle  $|a_{\text{ego}}| < 0,15 \text{ m/s}^2$ .

Zielfunktion nimmt jeweils ein eindeutiges Minimum an. Abbildung 3.10 zeigt schließlich die optimalen Werte für den Manöverindex  $\lambda$  in Abhängigkeit des Messrauschens  $n$ , ebenfalls für  $t_{\text{TTC}} = 2 \text{ s}$  und  $t_{\text{TTC}} = 1 \text{ s}$  mit Toleranzbändern, die sich bei einer Abweichung von  $\pm 10\%$  von der Zielfunktion ergeben. Für  $t_{\text{TTC}} = 1 \text{ s}$  liegen die Werte für den Manöverindex  $\lambda$  unterhalb denen für  $t_{\text{TTC}} = 2 \text{ s}$ , da sich zu diesem Zeitpunkt der systematische Fehler  $\Delta_T \hat{t}_{\text{TTC}}$  nicht mehr so stark auswirkt und durch das trägere Filter der zufällige Fehler  $\Delta_n \hat{t}_{\text{TTC}}$  verringert werden kann.

Der Anteil der Fehler  $\Delta \hat{t}_{\text{TTC}}$ , welcher bei gegebenem  $\lambda$  außerhalb vorgegebener Toleranzen  $\Delta t_{\text{min}}$  bzw.  $\Delta t_{\text{max}}$  liegt, lässt sich dabei wie folgt berechnen:

$$\mathcal{P}(\Delta t_{\text{min}} \leq \Delta \hat{t}_{\text{TTC}} \leq \Delta t_{\text{max}}) = \int_{a_{\text{min}}}^{a_{\text{max}}} \int_{-\infty}^{\Delta t_{\text{min}}} f_a(a) f_t(t, a) dt da + \int_{a_{\text{min}}}^{a_{\text{max}}} \int_{\Delta t_{\text{max}}}^{+\infty} f_a(a) f_t(t, a) dt da. \quad (3.14)$$

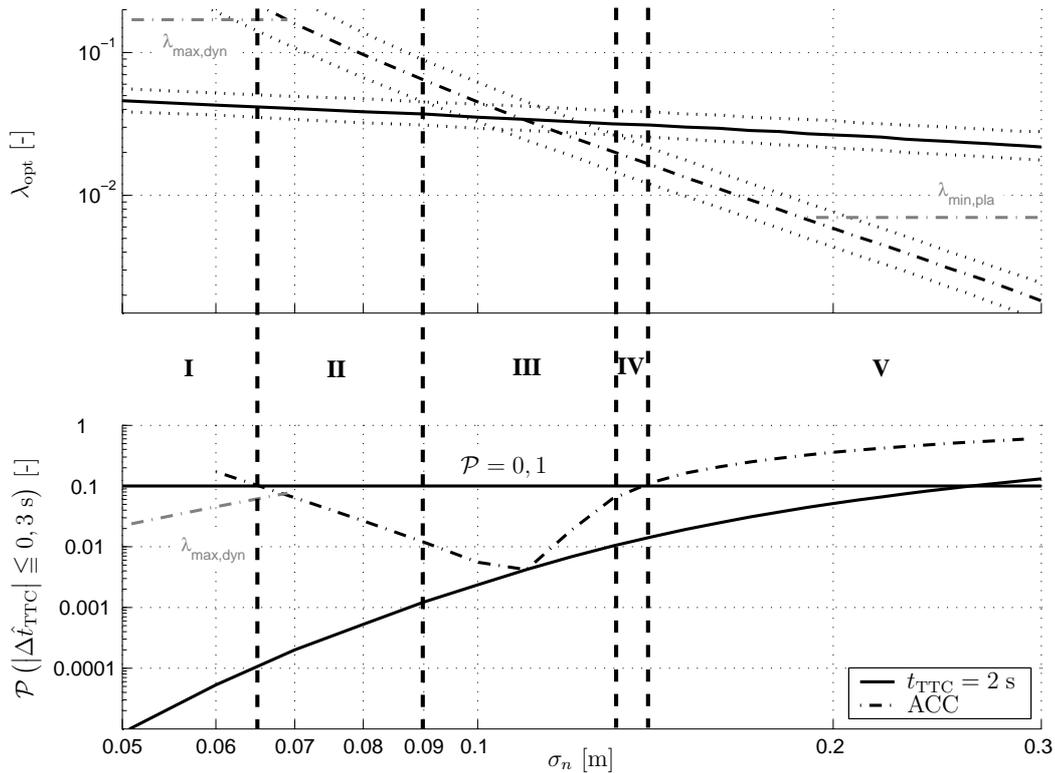


**Abbildung 3.11:** Anteil der Fehler  $\mathcal{P}(|\Delta \hat{t}_{TTC}| \leq \Delta t)$  bei Wahl des optimalen Manöverindex  $\lambda$  für  $t_{TTC} = 2 \text{ s}$  (oben) und  $t_{TTC} = 1 \text{ s}$  (unten) für  $d_0 = 33,33 \text{ m}$  bzw.  $\tau_0 = 1,5 \text{ s}$ . Strichpunktirt: Fehleranteil bei Wahl des optimalen Manöverindex für ACC.

Abbildung 3.11 (oben) zeigt den Anteil der Fehler für  $\Delta t_{min} = \Delta t_{max} = \Delta t$  bei optimalem Manöverindex  $\lambda_{opt}^{ttc}$  nach (3.13) für unterschiedliche Werte von  $\Delta t$  und  $t_{TTC} = 2 \text{ s}$ , Abbildung 3.11 (unten) für  $t_{TTC} = 1 \text{ s}$ . Zum Vergleich sind die Anteile der Fehler bei Wahl des optimalen Manöverindex für die stationäre ACC-Folgeregelung eingetragen. In einigen Bereichen ist der Fehler bei Wahl des optimalen Manöverindex für die stationäre ACC-Folgeregelung niedriger als bei Wahl des optimalen Manöverindex bezgl. der  $t_{TTC}$ -Schätzung. Der Wert der Zielfunktion  $ZF_{t_{TTC}}$  ist in diesen Bereichen dennoch schlechter.

### 3.4 Auslegung eines Kalmanfilters mit festen Parametern für eine adaptive Fahrgeschwindigkeitsregelung und ein System zur Unfallvermeidung

Abbildung 3.10 zeigt, dass bei Verwendung eines Kalmanfilters mit festen Parametern die Wahl des Manöverindex  $\lambda$  von der Genauigkeit des verwendeten Sensors (bzw. dem Sensorrauschen  $\sigma_n$ ) abhängt und dabei für verschiedene Assistenzsysteme unterschiedlich ausfallen kann. Im Folgenden wird gezeigt, welche Effekte der Betrieb eines  $t_{TTC}$ -basierten Systems



**Abbildung 3.12:** Auslegungsbereiche beim Betrieb eines ACC-Systems und eines TTC-basierten Systems mit  $t_{\text{TTC}} = 2 \text{ s}$  unter Verwendung eines konstanten Kalmanfilters. Oben: optimaler Manöverindex  $\lambda_{\text{opt}}$  für das jeweilige System mit 10% Toleranzbereich. Unten: TTC-Fehleranteil bei Betrieb mit optimalem Manöverindex.

und eines ACC-Systems mit einem gemeinsamen Kalmanfilter mit festen Parametern mit sich bringt.

Hierzu wird ein  $t_{\text{TTC}}$ -basiertes System betrachtet, welches bei  $t_{\text{TTC}} = 2 \text{ s}$  eine Aktion auslösen soll, wobei gefordert wird, dass für höchstens 10% der Fehler<sup>52</sup>  $|\Delta t_{\text{TTC}}| \leq 0,3 \text{ s}$  gilt<sup>53</sup>. Zudem wird ein ACC-System betrachtet, wobei gefordert wird, dass im stationären Folgebetrieb die maximale Amplitude der Beschleunigung  $a_{\text{ego}}$  kleiner als  $0,15 \text{ m/s}^2$  ist (bezogen auf einen  $3\sigma$ -Bereich).

Es ergeben sich die folgenden fünf Bereiche in Abhängigkeit der Messgenauigkeit  $\sigma_n$  des verwendeten Sensors für die Auslegung des gemeinsamen Kalmanfilters (siehe Abbildung 3.12):

#### **Bereich I** Filter nie im optimalen Arbeitspunkt für ACC - TTC optimal möglich

Um die Forderung  $\mathcal{P}(|\Delta t_{\text{TTC}}| \leq 0,3 \text{ s}) < 0,1$  einzuhalten, muss das Filter träger als für den ACC-Betrieb möglich ausgelegt werden. Hierbei kann bis auf den Wert  $\lambda_{\text{opt}}^{\text{ttc}}$  zurück

<sup>52</sup>Dieser Wert ist zunächst willkürlich. Die folgende Argumentation gilt aber auch für niedrigere Werte, allerdings verschieben sich dann die im Folgenden definierten Bereiche.

<sup>53</sup>Dies ist die Größenordnung der minimalen Reaktionszeit eines Menschen, siehe z.B. Bender, E.; Landau, K.: *PRORETA: Fahrerverhalten* (2006).

gegangen werden, so dass das  $t_{\text{TTC}}$ -basierte System im optimalen Bereich betrieben werden kann. Wird für ACC  $\lambda \leq \lambda_{\text{max,dyn}}^{\text{acc}}$  gewählt, kann im gesamten Bereich die Genauigkeit für die TTC Berechnung eingehalten werden.

**Bereich II** Filter entweder nicht im optimalen Arbeitspunkt für ACC oder für TTC

Die Forderung  $\mathcal{P}(|\Delta t_{\text{TTC}}| \leq 0,3 \text{ s}) < 0,1$  wird immer eingehalten. Es kann ein Wert  $\lambda_{\text{opt}}^{\text{ttc}} \leq \lambda \leq \lambda_{\text{max,com}}^{\text{acc}}$  gewählt werden, eines der Systeme wird dabei immer außerhalb des Optimums betrieben.

**Bereich III** Optimale Auslegung für ACC und TTC möglich

Die Forderung  $\mathcal{P}(|\Delta t_{\text{TTC}}| \leq 0,3 \text{ s}) < 0,1$  wird immer eingehalten. Im Rahmen einer Toleranz von 10% Abweichung von der jeweiligen Forderung können beide Systeme optimal betrieben werden.

**Bereich IV** Filter nie im optimalen Arbeitspunkt für TTC - ACC optimal möglich

Die Forderung  $\mathcal{P}(|\Delta t_{\text{TTC}}| \leq 0,3 \text{ s}) < 0,1$  wird immer eingehalten. Um die Forderung  $\max |a_{\text{ego}}| \leq 0,15 \text{ m/s}^2$  einhalten zu können, muss das Filter träger als optimal für das  $t_{\text{TTC}}$ -basierte System ausgelegt werden.

**Bereich V** Betrieb von ACC und TTC nicht gleichzeitig möglich

Um die Forderung  $\max |a_{\text{ego}}| \leq 0,15 \text{ m/s}^2$  einhalten zu können, muss die Bedingung  $\mathcal{P}(|\Delta t_{\text{TTC}}| \leq 0,3 \text{ s}) < 0,1$  gebrochen werden und umgekehrt. Es lässt sich daher kein Wert  $\lambda$  finden, der einen gleichzeitigen Betrieb beider Systeme unter den geforderten Randbedingungen erlaubt. Im letzten Teil des Bereichs müsste der Wert für  $\lambda$  sogar unterhalb von  $\lambda_{\text{min,pla}}^{\text{acc}}$  gewählt werden, um einen komfortablen Folgebetrieb für ACC zu gewährleisten.

### 3.5 Auslegung eines adaptiven Kalmanfilters für eine adaptive Fahrgeschwindigkeitsregelung und ein System zur Unfallvermeidung

Ein Filter mit festen Parametern hat eine feste Bandbreite. Ist diese zu gering, führt das zu einem hohen Zeitverzug, ist sie zu hoch, treten zu hohe Schätzfehler auf. Aufgrund der Unschärferelation der Informationstechnik<sup>54</sup>  $\Delta\omega\Delta t \leq \text{const.}$ , lässt sich dieses Dilemma mit einem konstanten Filter nicht umgehen.

Abhilfe bieten adaptive Filter, bei welchen die Filtereigenschaften mit Hilfe eines Adaptionsverfahrens angepasst werden.<sup>55</sup> Im Bereich der Zustandsschätzung per Kalmanfilter hat sich

<sup>54</sup>Siehe z.B. Hilberg, W.: *Grundlagen elektronischer Schaltungen* (1992), S. 41.

<sup>55</sup>Allgemeine Überlegungen zu adaptiven Filtern finden sich z.B. bei Hänsler, E.: *Statistische Signale* (1997), S. 336ff.

der so genannte Interacting Multiple Model Ansatz (*IMM*) als leistungsfähiges Verfahren erwiesen.<sup>56</sup>

Im Gegensatz zu klassischen Verfahren, bei denen die Parameter des Kalmanfilters adaptiert werden<sup>57</sup>, besteht der IMM Filter aus  $M \geq 2$  Kalmanfiltern mit unterschiedlichen Modellen bzw. Modellparametern. Im Algorithmus werden die geschätzten Zustandsgrößen der einzelnen Filter in jedem Zeitschritt miteinander kombiniert (Interaction/Mixing). Basis für die Kombination ist die Likelihood<sup>58</sup> für das jeweilige Modell, aus welcher sich eine Wahrscheinlichkeit für das Zutreffen des jeweiligen Modells errechnet (Mode Probability) und die ggf. zeitabhängigen<sup>59</sup> Übergangswahrscheinlichkeiten zwischen den Modellen. Dabei wird angenommen, dass das Verhalten des beobachteten Objekts immer mit einem der Modelle beschrieben werden kann und die Sprünge zwischen den Modellen mithilfe einer (homogenen) Markovkette<sup>60</sup> modelliert werden können.<sup>61</sup>

Zur Optimierung des Verhaltens des IMM-Filters steht eine hohe Anzahl von Freiheitsgraden zur Verfügung (Übergangswahrscheinlichkeiten, Messrauschen, Modellrauschen, Mess- und Systemmodelle der  $M$  Kalmanfilter). Zudem lassen sich die gleichen Eigenschaften des Filters oftmals mit verschiedenen Parametern beeinflussen.<sup>62</sup> Die Optimierung der Parameter kann dabei auf Basis von Richtlinien und Praxiserfahrung erfolgen,<sup>63</sup> oder es können numerische Optimierungsverfahren eingesetzt werden.<sup>64</sup>

Zur weiteren Untersuchung wird ein Sensor mit Sensorrauschen  $\sigma_n = 0,2$  m betrachtet, einem Wert aus Bereich V in Abbildung 3.12, bei dem der gleichzeitige Betrieb beider Anwendungen mit einem Filter mit konstanten Parametern nicht möglich ist und das Kalmanfilter für den ACC-Betrieb unterhalb der Grenze für Kolonnenstabilität betrieben werden müsste. Die Anforderungen an ein adaptives Filter sind dabei für beide Anwendungen unterschiedlich:

**ACC** Für das ACC-System wird im stationären Folgebetrieb eine Bandbreite benötigt, die Bedingung (3.4) erfüllt. Tritt eine Beschleunigung  $a_{\text{fw}}$  auf, ist eine Bandbreite erforderlich, die sowohl Eigen- und Kolonnenstabilität als auch einen schwingungsfreien Verlauf von  $a_{\text{ego}}$  erlaubt und die Störungen  $\sigma_a$  auf einem vertretbaren Niveau hält. Für den hier

<sup>56</sup>Siehe z.B. Niehsen, W.; Müller, M.: *IMM-Tracking-Filter für Fahrerassistenzsysteme* (2003); Kaempchen, N. et al.: *IMM object tracking* (2004); Weiss, K.; Kaempchen, N.; Kirchner, A.: *Multiple-model tracking* (2004). Allgemeine Betrachtungen zur Leistungsfähigkeit des Ansatzes finden sich z.B. bei Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 476.

<sup>57</sup>Siehe z.B. Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 424-440.

<sup>58</sup>Auch Mutmaßlichkeit, wird aber in der Regel nicht übersetzt.

<sup>59</sup>Siehe z.B. Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 460.

<sup>60</sup>Zu Markovketten siehe z.B. Isaacson, D. L.; Madsen, R. W.; Quenues, W.: *Markov chains* (1976).

<sup>61</sup>Vgl. Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 453-457.

<sup>62</sup>Vgl. die Ausführungen in Besada, J. et al.: *Design of IMM filter for radar tracking* (2005).

<sup>63</sup>„The mode set, the transition mode probability matrices, process noise standard deviations and initial estimates/variances are estimator design parameters, which should be selected based on the expected scenario and the designers’s experience“ Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 472.

<sup>64</sup>Siehe z.B. Besada, J. et al.: *Design of IMM filter for radar tracking* (2005).

betrachteten Fall  $\sigma_n = 0,2$  m bedeutet dies eine niedrigere Bandbreite im Folgebetrieb als bei  $a_{fw} \neq 0$ .

**TTC** Für das  $t_{TTC}$ -basierte System ist bei konstanter Beschleunigung  $a_{fw}$  eine Bandbreite nötig, die den zufälligen Fehler  $\Delta_n \hat{t}_{TTC}$  gering hält. Da dies zu einem hohen Zeitverzug führt, was wiederum  $\Delta_T \hat{t}_{TTC}$  ansteigen lässt, wird bei Änderungen von  $a_{fw}$  eine höhere Bandbreite benötigt.

Die unterschiedlichen Anforderungen führen schließlich zu unterschiedlichen adaptiven Filtern.

Ein mögliches IMM-Filter, welches die Anforderungen der ACC-Anwendungen erfüllt, besteht aus  $M = 2$  Modellen. Das erste ist das bereits in den vorherigen Abschnitten<sup>65</sup> verwendete DWPA Modell<sup>66</sup>, das zweite ein Discrete White Noise Acceleration (DWNA) Modell<sup>67</sup>, mit welchem nur die Größen  $d_{rel}$  und  $v_{rel}$  geschätzt werden. Hierbei wird angenommen, dass sich die Geschwindigkeit  $v_{rel}$  in jedem Zeitschritt  $T$  gemäß einer konstanten Beschleunigung  $v \sim \mathcal{N}(0, \sigma_v)$  ändert. Bei entsprechender Parametrierung des IMM-Filters wird das DWNA Modell im stationären Folgebetrieb ( $a_{fw} \approx 0$ ) stärker gewichtet und bestimmt damit die Eigenschaften des Filters, in dynamischen Situationen ( $a_{fw} \neq 0$ ) hingegen das DWPA Modell, welches dann eine dynamischere Schätzung der Zustandsgrößen erlaubt.

Die folgenden Parameter sind dabei für den ACC-Regelkreis sinnvoll:<sup>68</sup> Für beide Modelle gilt  $\sigma_w = \sigma_n = 0,2$  m. Für das DWPA Modell ( $a_{fw} \neq 0$ ) gilt  $\sigma_v = 0,6$  m/s<sup>2</sup>, für das DWNA Modell ( $a_{fw} \approx 0$ )  $\sigma_v = 0,05$  m/s<sup>2</sup>. Für die Zustandsübergangsmatrix zwischen den Modellen wird gesetzt:

$$\Pi = \begin{pmatrix} p_{11} = 0,9 & p_{12} = 1 - p_{11} \\ p_{21} = 1 - p_{22} & p_{22} = 0,9 \end{pmatrix}. \quad (3.15)$$

Ein mögliches IMM-Filter für die  $t_{TTC}$ -basierte Anwendung besteht ebenfalls aus  $M = 2$  Modellen. Vorteilhaft ist hier allerdings der Einsatz von zwei DWPA Modellen mit unterschiedlichem Modellrauschen  $\sigma_v$ . Bei entsprechender Parametrierung des IMM-Filters ist bei  $a_{fw} \approx \text{const.}$  das träge Modell wahrscheinlicher, so dass eine präzisere Schätzung der Zustandsgrößen ermöglicht wird. Bei Beschleunigungssprüngen dagegen das dynamischere Modell, was ein schnelleres Konvergieren des Filters bewirkt.

Die folgenden Parameter sind dabei für eine  $t_{TTC}$ -basierte Anwendung sinnvoll.<sup>69</sup> Für beide Modelle gilt  $\sigma_w = \sigma_n = 0,2$  m. Für das erste DWPA Modell gilt  $\sigma_v = 0,65$  m/s<sup>2</sup> (entspricht  $\lambda_{opt}^{ttc}$  für dieses Messrauschen), für das zweite ( $a_{fw} \approx \text{const.}$ )  $\sigma_v = 0,1$  m/s<sup>2</sup>. Der Manöverindex

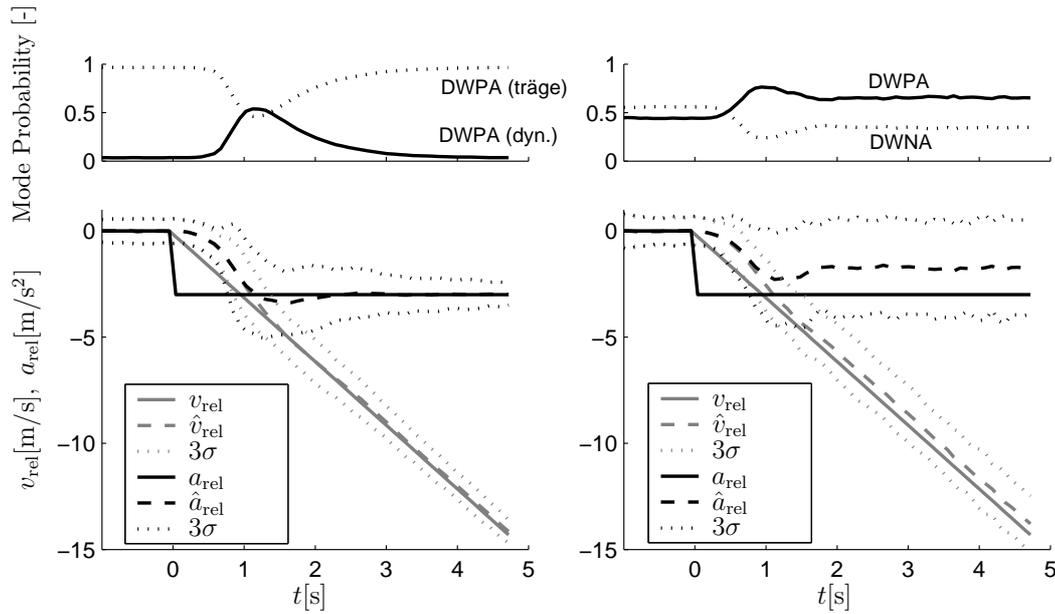
<sup>65</sup>Siehe S. 27.

<sup>66</sup>Siehe Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 274.

<sup>67</sup>Siehe Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 273. Die Gleichungen sind in Abschnitt A.1 auf S. 92 aufgeführt.

<sup>68</sup>Dies sind nicht notwendigerweise die optimalen Parameter für den ACC-Regelkreis, sie erfüllen aber die geforderten Randbedingungen sehr gut, vgl. Tabelle 3.1.

<sup>69</sup>Dies sind nicht notwendigerweise die optimalen Parameter für die  $t_{TTC}$ -basierte Anwendung, sie erfüllen aber die geforderten Randbedingungen sehr gut, vgl. Tabelle 3.1.



**Abbildung 3.13:** Schätzung der relativen Geschwindigkeit  $\hat{v}_{\text{rel}}$  und relativen Beschleunigung  $\hat{a}_{\text{rel}}$  mit einem IMM-Filter, der auf die Anforderungen einer  $t_{\text{TTC}}$ -basierten Anwendung (links) bzw. auf einen ACC-Regler (rechts) abgestimmt ist. Oben: Mittlere „Mode Probability“ (Wahrscheinlichkeit für das jeweilige Modell).

dieses Modells ergibt sich zu  $\lambda = 2,7 \cdot 10^{-3} < \lambda_{\text{min,pla}}^{\text{acc}}$ . Für die Zustandsübergangsmatrix zwischen beiden Modellen gilt:

$$\Pi = \begin{pmatrix} p_{11} = 0,9 & p_{12} = 1 - p_{11} \\ p_{21} = 1 - p_{22} & p_{22} = 0,995 \end{pmatrix}. \quad (3.16)$$

Abbildung 3.13 zeigt eine Monte-Carlo-Simulation<sup>70</sup> der Schätzungen  $\hat{v}_{\text{rel}}$  und  $\hat{a}_{\text{rel}}$  mit den verschiedenen IMM-Filtern bei einer sprungförmigen Änderung der Größe  $a_{\text{fw}}$  von Null auf  $-3 \text{ m/s}^2$ . Deutlich zu erkennen ist das unterschiedliche Verhalten im Bereich  $a_{\text{rel}} \neq 0$ . Dies führt bei der  $t_{\text{TTC}}$ -basierten Anwendung zu einer präziseren Schätzung, beim ACC-System zu einem besseren Folgeverhalten.

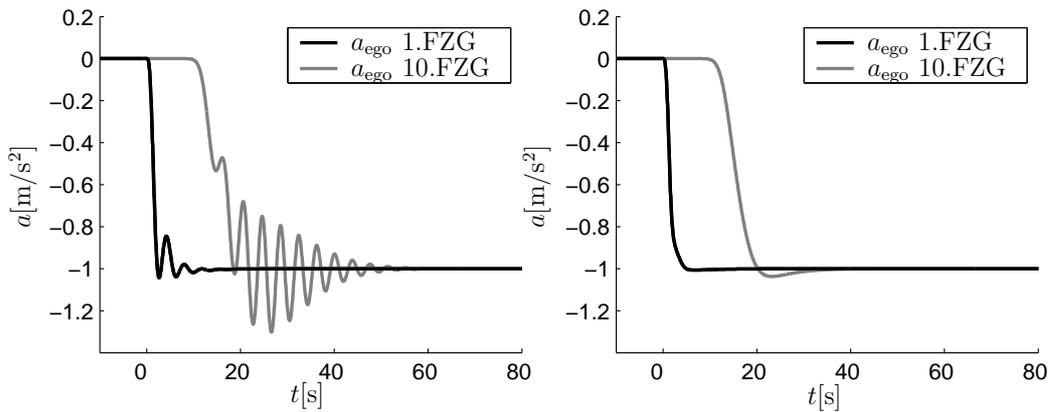
Abbildung 3.14 zeigt den Verlauf von  $a_{\text{ego}}$  für die verschiedenen IMM-Filter mit<sup>71</sup>  $\tau_v = 1,75 \text{ s}$ . Da das für die  $t_{\text{TTC}}$ -basierte Anwendung abgestimmte Filter kurz nach dem Beschleunigungssprung wieder in den trägen Modus fällt (der Manöverindex  $\lambda$  dieses Teilfilters liegt unterhalb der Grenze für Kolonnenstabilität und in der Nähe der Stabilitätsgrenze bzgl. der Einzelstabilität<sup>72</sup>), neigt das System zu Schwingungen. Zu erkennen ist, wie sich die Schwingung beim  $t_{\text{TTC}}$ -optimierten Filter entlang der Kolonnenposition verstärkt, das für ACC optimierte Filter hingegen nur ein kleines Überschwingen bewirkt.

Abbildung 3.15 zeigt schließlich die Fehler der  $t_{\text{TTC}}$ -Schätzung in Abhängigkeit des verwendeten Filters. Zu sehen ist die Leistungssteigerung durch das IMM-Filter im Bereich niedriger

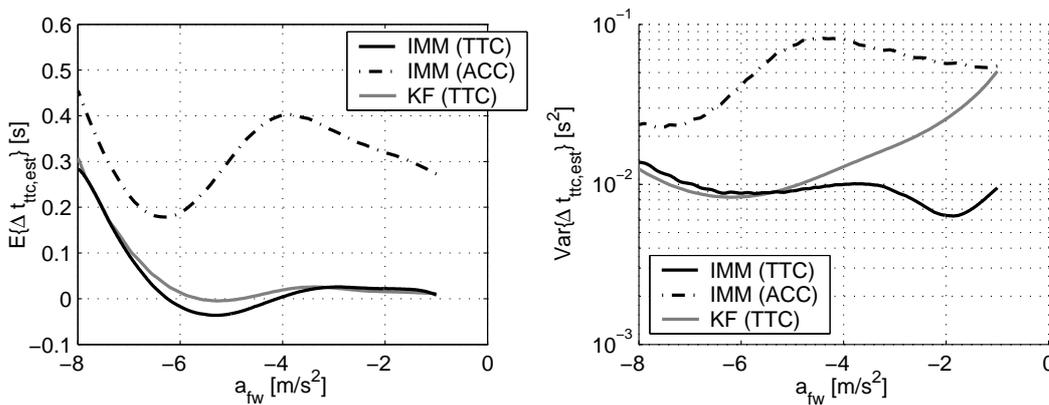
<sup>70</sup>Zur Methode der Monte-Carlo Simulation siehe z.B. Domschke, W.; Drexler, A.: *Operations-Research* (1998), S. 214 oder Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 168.

<sup>71</sup>Minimaler Wert für Kolonnenstabilität des Reglers ohne Schätzer, siehe S. 29.

<sup>72</sup>Siehe Abbildung 3.4, S. 29.



**Abbildung 3.14:** Verhalten der Fahrzeugbeschleunigung  $a_{ego}$  bei einem Regelkreis mit IMM-Filter, der auf die Anforderungen einer  $t_{TTC}$ -basierten Anwendung (links) bzw. auf einen ACC-Regler (rechts) abgestimmt ist ( $\tau_v = 1,75$  s).



**Abbildung 3.15:** Mittelwert (links) und Varianz (rechts) des Fehlers der TTC-Schätzung  $\Delta \hat{t}_{TTC}$  mit einem auf eine  $t_{TTC}$ -basierte Anwendung abgestimmten IMM-Filter und einem auf einen ACC-Regler abgestimmten IMM-Filter. Zum Vergleich sind die Werte für das optimale konstante Kalmanfilter für die TTC-Berechnung eingezeichnet.

Beschleunigungen im Vergleich zum optimalen konstanten Filter für dieses Messrauschen. Das auf dem ACC-Regelkreis abgestimmte IMM-Filter zeigt signifikant größere Fehler, sowohl in der Varianz als auch beim Mittelwert.

### 3.6 Fazit zur Datenfilterung

Die Ausführungen in Abschnitt 3.4 zeigen, dass in Abhängigkeit des Messrauschens Bereiche existieren, in denen eine  $t_{TTC}$ -basierte Anwendung und ein ACC-System nicht mit einem gemeinsamen konstanten Kalmanfilter betrieben werden können. Tabelle 3.1 fasst die Merkmale der Filter bei einem Sensorrauschen  $\sigma_n = 0,2$  m nochmals zusammen. Die Vorteile der adaptiven Filter gegenüber den konstanten Filtern sind deutlich erkennbar. Da die Anforderungen an die verschiedenen adaptiven Filter aber gegensätzlich sind - insbesondere benötigt

**Tabelle 3.1:** Gegenüberstellung der Merkmale der untersuchten Filter bei  $\sigma_n = 0,2$  m.

	$3\sigma_a/\text{m/s}^2$	$e/\text{m}^2/\text{s}^3$ †	$\mathcal{P}( \Delta t_{\text{TTC}}  < 0,3 \text{ s})$	$\mathcal{P}( \Delta t_{\text{TTC}}  < 0,2 \text{ s})$	$\mathcal{P}( \Delta t_{\text{TTC}}  < 0,1 \text{ s})$	$\sqrt{ZF_{t_{\text{TTC}}}}/\text{s}$
IMM (ACC)	< 0,15	0,04	$\approx 0,50$	$\approx 0,70$	$\approx 0,85$	$\approx 0,40$
IMM (TTC)	< 0,15	0,14 <sup>‡</sup>	$\approx 0,01$	$\approx 0,08$	$\approx 0,47$	$\approx 0,11$
KF (ACC)	0,15	0,13 <sup>‡</sup>	0,35	0,41	0,53	0,52
KF (TTC)	0,25	0,02	0,05	0,17	0,47	0,15

†: Abweichung  $e$  nach (3.6) zu einem Regelkreis ohne Schätzer bei einem Beschleunigungssprung  $a_{\text{fw}} = 0 \text{ m/s}^2 \rightarrow a_{\text{fw}} = 1 \text{ m/s}^2$  (Regler:  $\tau_v = 1,5 \text{ s}$ ; Fahrzeug:  $T_1 = 0,15 \text{ s}$ ).

‡: Nicht kolonnenstabil.

das Filter für ACC eine höhere Dynamik für  $a_{\text{fw}} \neq 0$  als das Filter für die  $t_{\text{TTC}}$ -basierte Anwendung - lässt sich auch kein gemeinsames adaptives Filter finden, welches beide Vorteile vereint.

Die Gültigkeit sowohl der starken als auch der schwachen Annahme für funktionsbezogene Mehrfachfilter konnte damit gezeigt werden. Zum Nachweis wurden Anwendungen herangezogen, die bereits in Serie sind oder bald in Serie gehen werden sowie Sensorgenauigkeiten betrachtet, mit welchen die jeweiligen Anwendungen realisierbar sind. Soll die System-Architektur für Sensoren unterschiedlicher Messgenauigkeit geeignet sein, muss also die Abhängigkeit des Filteralgorithmus von den Eigenschaften der Anwendung berücksichtigt werden.

Bemerkenswert ist, dass die schwache Annahme für funktionsbezogene Mehrfachfilter im gezeigten Fall für nahezu alle Werte des Sensorrauschens  $\sigma_n$  zutrifft. Selbst bei geringen Messungenauigkeiten des Sensors bringt es hier Vorteile, für verschiedene Anwendungen unterschiedliche Filteralgorithmen in der Architektur vorzusehen.

## 4 Datenassoziation für unterschiedliche Fahrerassistenzsysteme

### 4.1 Ziele der Datenassoziation

Im Rahmen der Datenassoziation werden die während der Signalverarbeitung und Merkmalsextraktion gewonnenen Merkmals hypothesen den bereits bekannten Objekthypothesen zugeordnet.<sup>1</sup> Misst ein Sensor genau eine Größe (z.B. Raddrehzahlsensor) und ist das dazugehörige Objekt eindeutig definierbar (z.B. Rad vorne rechts<sup>2</sup>), so ist eine eineindeutige Zuordnung möglich. Für Umfeldsensoren, die eine zunächst unbekannte Anzahl an Objekten vermessen, stellt die Datenassoziation eine Aufgabe dar, welche die Qualität der Schätzung maßgeblich beeinflusst.<sup>3</sup> Die folgenden Ausführungen beschäftigen sich mit dem Zuordnungsproblem bei Umfeldsensoren.<sup>4</sup>

Singh; Bailey unterscheiden dabei zwei Fehlerarten bei der Datenassoziation:<sup>5</sup>

**Typ I Fehler** Eine Merkmals hypothese wird keiner Objekthypothese zugeordnet, obwohl dies korrekt wäre. Dies führt zu einem Informationsverlust für die Objekt-Schätzung. Zudem können aufgrund der nicht assoziierten Merkmals hypothesen weitere Objekthypothesen zum realen Objekt erstellt werden.

<sup>1</sup>Verallgemeinerte Theorien der Sensordatenfusion kommen auch ohne den Schritt der expliziten Datenassoziation aus, siehe z.B. Mori, S.: *Random sets in data fusion* (2000); Steinberg, A.; Bowman, C.: *Revisions to the JDL Data Fusion Model* (2001).

<sup>2</sup>Auch hier kann z.B. ein Kabel vertauscht sein!

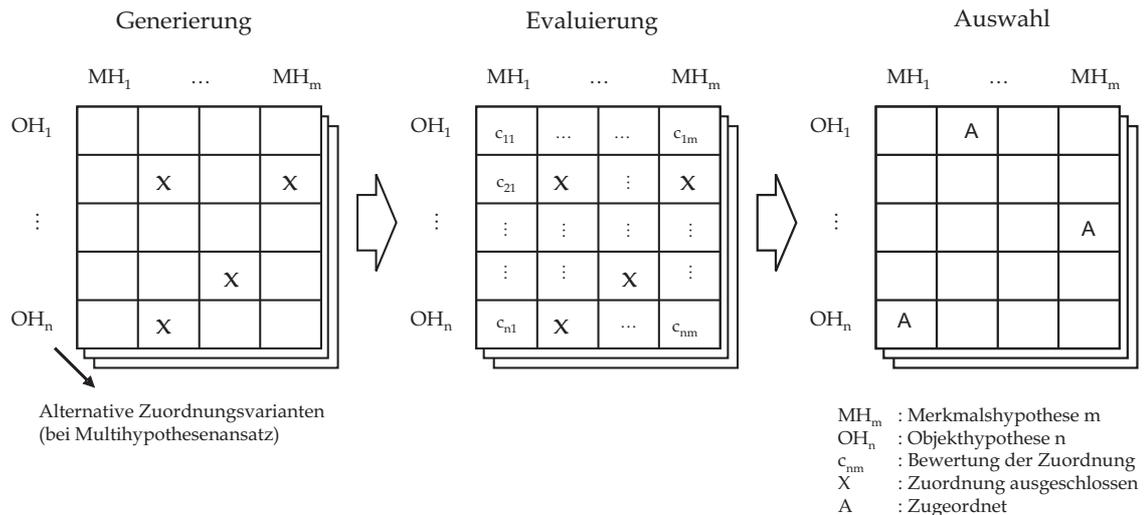
<sup>3</sup>Siehe hierzu z.B. die Ausführungen in Bar-Shalom, Y.; Li, X.-R.: *Multitarget-multisensor tracking* (1995), S.98-101; Holt, V. v.: *Multisensorielle Fahrumgebungserfassung* (2005), S.40-41; Stüker, D.: *Heterogene Sensordatenfusion* (2004), S.46-47.

<sup>4</sup>Ggf. ist eine Verallgemeinerung möglich, die auch z.B. Fahrzustandssensoren mit einschließt. Die Assoziation kann in diesem Falle die Plausibilität der Sensorsignale im Vergleich zu einer Prädiktion aus einem Fahrzeugmodell ermitteln, in welches auch Informationen aus den Umfeldsensoren mit einfließen können. Mit Hilfe von Multihypothesen-Ansätzen lassen sich ggf. auch nicht eindeutige Situationen mit einbeziehen (Straßenneigung, Steilkurven), welche sich wiederum durch die zusätzliche Informationen aus Umfeldsensoren auflösen lassen (z.B. Erkennung des Horizonts per Videosensorik). Zum Thema Fehlererkennung siehe z.B. Schulz, H.-G.: *Modellgestützte Fehlerdiagnose* (2006). Zur Erkennung des Fahrzustands aus Umfeldsensoren siehe z.B. Fuerstenberg, K.; Dittmer, M.: *Ego-motion estimation* (2003).

<sup>5</sup>Vgl. Singh, R.-N.; Bailey, W.: *Fuzzy logic applications* (1997); siehe auch Stüker, D.: *Heterogene Sensordatenfusion* (2004), S.46-47.

**Typ II Fehler** Eine Merkmalshypothese wird der falschen Objekthypothese zugeordnet. Hierdurch werden falsche Informationen in den Prozess der Objekt-Schätzung eingebracht, womit die Güte der Schätzung abnimmt.

## 4.2 Vorgehen bei der Datenassoziation



**Abbildung 4.1:** Aufteilung des Prozesses Datenassoziation.

Hall; Llinas teilen den Prozess der Datenassoziation in die folgenden drei Schritte ein<sup>6</sup> (siehe auch Abbildung 4.1):

**Generierung von Zuordnungshypothesen** Prinzipiell mögliche Zuordnungen von Merkmals-hypothesen zu Objekthypothesen werden gefunden. Ergebnis sind eine bzw. mehrere Matrizen mit prinzipiell möglichen Zuordnungen (Zuordnungsmatrizen).

**Evaluierung der Zuordnungshypothesen** Die gefundenen Zuordnungshypothesen werden mit dem Ziel einer quantitativen Bewertung bzw. einer Rangfolge evaluiert. Ergebnis sind quantitative Werte (z.B. Kosten) in der Zuordnungsmatrix bzw. den -matrizen.

**Auswahl der Zuordnungshypothesen** Aus den bewerteten Zuordnungsmöglichkeiten wird eine Auswahl getroffen, auf der die weitere Datenverarbeitung und damit insbesondere die Datenfilterung aufbaut.

Die drei Verarbeitungsschritte müssen nicht getrennt implementiert werden, vielmehr können sie voneinander abhängen. Allerdings wird empfohlen, die Schritte im Entwicklungsprozess voneinander zu entkoppeln: „Our recommended approach is to decouple these process designs so that they can be confidently engineered in isolation.“<sup>7</sup>

<sup>6</sup>Hall, D.; Llinas, J.: *An introduction to multisensor data fusion* (1997), S. 79.

<sup>7</sup>Hall, D. L.; McMullen, S. A.: *Mathematical techniques in multisensor data fusion* (2004), S. 113.

Bei der Gestaltung der Algorithmen stellen sich u.a. die folgenden Fragen:<sup>8</sup>

- Wird einer Objekthypothese genau eine Merkmalshypothese zugeordnet (eineindeutige Zuordnung: 1:1) oder nicht (mehrfache Zuordnung: 1:n, m:1, m:n)?
- Werden für die Zuordnung nur die Daten des aktuellen Mess-Zyklus verwendet oder die Daten mehrerer Zyklen?
- Wird eine einzelne Zuordnungsmatrix (eineindeutig oder auch nicht) oder werden mehrere alternative Zuordnungsmatrizen erzeugt (Multihypothesenansatz bzgl. der Zuordnung)?

Bei der Beantwortung der Fragen spielt die Qualität und Leistungsfähigkeit der zur Verfügung stehenden Ressourcen (wie z.B. Rechenkapazität, Leistungsfähigkeit der Sensoren) eine Rolle. Je nach Randbedingungen sind verschiedene Lösungen möglich.<sup>9</sup>

Die Hypothesengenerierung selbst lässt sich wiederum in zwei Teilschritte untergliedern: Aufstellung der Zuordnungshypothesen und Auswahl der prinzipiell möglichen Hypothesen. Zur Aufstellung der Zuordnungshypothesen können verschiedene Methoden herangezogen werden. Dies sind z.B.:<sup>10</sup>

**Physikalische Modelle** Sichtbereiche und Verdeckungen des verwendeten Sensors können berechnet werden. Objekthypothesen, die signifikant außerhalb des Sichtbereichs liegen, werden nicht in die Hypothesengenerierung mit einbezogen.

**Szenenwissen** Das Verhalten und der potentielle Aufenthaltsort von Objekten auf Basis des Wissens über die beobachtete Szene kann genutzt werden, wie z.B. Bereiche zum Auffinden von Straßenmarkierungen oder Verkehrsschildern.

**Probabilistische Modelle** Die erwartete Anzahl an Fehldetektionen kann in den Prozess mit einbezogen werden.

**Ad-hoc Methoden** Ein Beispiel hierfür ist das Aufstellen aller möglichen Zuordnungsmöglichkeiten. Hierzu braucht kein weiteres Vorwissen bekannt sein.

Zur Auswahl der prinzipiell möglichen Hypothesen sind z.B. die folgenden Methoden anwendbar:<sup>11</sup>

**Mustererkennungsalgorithmen** Unter Verwendung der Rohsignale und Rohdaten lassen sich Zuordnungen ausschließen (z.B. über Korrelationstechniken).

**Gating Techniken** Über z.B. physikalische Modelle kann ein Bereich, in dem sich Objekthypothesen bzw. davon abgeleitete Merkmalsypothesen zum aktuellen Messzeitpunkt mit einer bestimmten Wahrscheinlichkeit befinden, berechnet werden (Prädiktion). Aus

<sup>8</sup>Siehe Hall, D. L.; McMullen, S. A.: *Mathematical techniques in multisensor data fusion* (2004), S. 81.

<sup>9</sup>Siehe z.B. die Ausführungen in Hall, D. L.: *Handbook of multisensor data fusion* (2001), S. 345-383.

<sup>10</sup>Siehe Hall, D. L.; McMullen, S. A.: *Mathematical techniques in multisensor data fusion* (2004), S. 92-93. Hier werden auch weitere Methoden aufgeführt.

<sup>11</sup>Siehe Hall, D. L.; McMullen, S. A.: *Mathematical techniques in multisensor data fusion* (2004), S. 94-95. Hier werden auch weitere Methoden aufgeführt.

dem aktuellen Messzyklus stammende Merkmalshypothesen, die außerhalb eines solchen Bereiches liegen, werden nicht zur entsprechenden Objekthypothese assoziiert.

Zur Hypothesenevaluierung können u.a. probabilistische Modelle basierend auf der Bayes-Theorie, possibilistische Modelle basierend auf der Dempster-Shafer-Theorie, neuronale Netze oder auch Ad-hoc Techniken, wie z.B. eine ungewichtete Abstandsberechnung zwischen einer Prädiktion der Merkmale und den Merkmalen selbst, herangezogen werden.<sup>12</sup>

Zur Hypothesenauswahl existiert schließlich eine Vielzahl mathematischer Algorithmen.<sup>13</sup> Mit zunehmender Dimension und insbesondere dann, wenn Daten aus mehreren Zyklen in die Auswahl mit einbezogen werden, ist die Lösung aufwändig: „In problems of relatively large size, the basic assignment problem formulation leads to an NP-hard problem for which reasonable compute-time solutions may not be available.“<sup>14</sup>

Von der Komplexität beherrschbar sind Einfach-Hypothesen-Ansätze, die bei der Hypothesenauswahl nur die Daten des aktuellen Zyklus berücksichtigen.<sup>15</sup> Eine häufig vorzufindende Problemstellung ist dabei die Zuordnung von  $n$  Objekthypothesen zu  $m$  Merkmalshypothesen mit  $m \geq n$ , wobei einer Objekthypothese genau eine Merkmalshypothese zugeordnet wird.

Hierzu existieren exakte Verfahren, welche die Summe der Kosten in der Zuordnungsmatrix minimieren. Ein Beispiel ist der Munkres-Algorithmus, welcher einen Aufwand von  $\mathcal{O}(n^2m)$  hat.<sup>16</sup> Daneben existieren weniger aufwändige Algorithmen, die nur eine Näherungslösung liefern. Ein Beispiel ist das iterative Nächster-Nachbar Verfahren, welches sukzessive die Zuordnungen mit den geringsten Kosten bzw. der höchsten Wahrscheinlichkeit wählt und einen Aufwand von  $\mathcal{O}(m^2 \log_2 m)$  hat.<sup>17</sup>

Im Bereich der Fahrzeugtechnik setzen sich verschiedene Arbeiten mit dem Thema Datenassoziation auseinander.<sup>18</sup> Becker, Kirchner und Stüker kommen dabei zu dem Ergebnis, dass bei Laserscannern Ansätze, die bei der Hypothesenauswahl nur Näherungslösungen liefern, ausreichen. So schreibt hierzu z.B. Stüker:

„Obwohl theoretisch suboptimal liefert bereits der iterative Nächster-Nachbar-Algorithmus [...] zumeist die optimale Lösung. Die höhere Rechenkomplexität für

---

<sup>12</sup>Siehe Hall, D. L.; McMullen, S. A.: *Mathematical techniques in multisensor data fusion* (2004), S. 106-107.

Hier werden auch weitere Methoden aufgeführt.

<sup>13</sup>Siehe z.B. Hall, D. L.; McMullen, S. A.: *Mathematical techniques in multisensor data fusion* (2004), S. 119-124.

<sup>14</sup>Hall, D. L.; McMullen, S. A.: *Mathematical techniques in multisensor data fusion* (2004), S. 118.

<sup>15</sup>Stüker, D.: *Heterogene Sensordatenfusion* (2004), S. 50-58 gibt einen Überblick über verschiedene Zuordnungsverfahren.

<sup>16</sup>Siehe z.B. Becker, J.-C.: *Fusion der Daten der objekterkennenden Sensoren* (2002), S. 28.

<sup>17</sup>Siehe z.B. Becker, J.-C.: *Fusion der Daten der objekterkennenden Sensoren* (2002), S. 40.

<sup>18</sup>Siehe z.B. Becker, J.-C.: *Fusion der Daten der objekterkennenden Sensoren* (2002); Stüker, D.: *Heterogene Sensordatenfusion* (2004); Holt, V. v.: *Multisensorielle Fahrumgebungserfassung* (2005); Streller, D.: *Multi-Hypothesen-Ansatz* (2006).

optimale Verfahren, Alle-Nachbarn-Algorithmen und Multi-Hypothesen-Tracking ist daher unangemessen.<sup>19</sup>

Anders ist die Situation bei der Assoziation von aus Videobildern extrahierten Kanten, die ein Fahrzeug beschreiben. Hier schreibt Holt:

„Im Bereich der Bildbearbeitung kommt der Zuordnungsproblematik - im Vergleich zu anderen umfelderfassenden Sensoren - aufgrund der Vielzahl und Vielfalt der Merkmale eine nochmals gesteigerte Bedeutung zu.“<sup>20</sup>

Im Rahmen seiner Arbeit entwickelt er ein auf einem Hopfield-Netz<sup>21</sup> basierendes Optimierungsverfahren, welches unter Rückgriff auf die Rohdaten (Pixel des Videobildes) zu einer eindeutigen Zuordnung gefundener Kanten (Merkmals-hypothesen) zu beobachteten Objekthypothesen (3D-Modell eines Fahrzeugs) gelangt.<sup>22</sup>

Für die praktische Umsetzung in einem Versuchsträger, in dem Videodaten und Laserscannerdaten fusioniert werden, wählt er schließlich einen heuristischen Ansatz für die Assoziation der aus dem Videobild gewonnenen Merkmals-hypothesen:

„Für einen solchen Fall - und ähnliche 'Spezialfälle' - bietet es sich an, die Zuordnung nicht durch den intransparenten Mechanismus des Hopfield-Netzes vornehmen zu lassen, da der Aufwand für eine 'handverdrahtete' Zuordnungslogik begrenzt ist, und darüberhinaus gestattet, für derart relevante Szenarien ggf. Sonderregeln für bestimmte Ansichten zu integrieren.“<sup>23</sup>

Der 'Spezialfall' bezieht sich dabei auf Fahrten im Längsverkehr und eine bezüglich des Fahrzeugs feste Kameraanordnung. Für die Daten des Laserscanners verwendet er wiederum einen Nächster-Nachbar-Ansatz für die Hypothesenauswahl.<sup>24</sup>

## 4.3 Zusammenhang der Datenassoziation mit der Merkmalsextraktion und Objekthypothesenbildung

Sowohl der Prozess der Merkmalsextraktion als auch der Prozess der Objekthypothesenbildung können mit der Datenassoziation zusammenhängen. Abbildung 4.2 zeigt als Beispiel zur Verdeutlichung die Abstandsmessungen eines Laserscanners mit einer Winkelauflösung von  $1,5^\circ$ .<sup>25</sup> In der Szene wird ein beobachtetes Fahrzeug von einem anderen überholt.

<sup>19</sup>Stüker, D.: *Heterogene Sensordatenfusion* (2004), S. 84.

<sup>20</sup>Holt, V. v.: *Multisensorielle Fahrumgebungserfassung* (2005), S. 77-78.

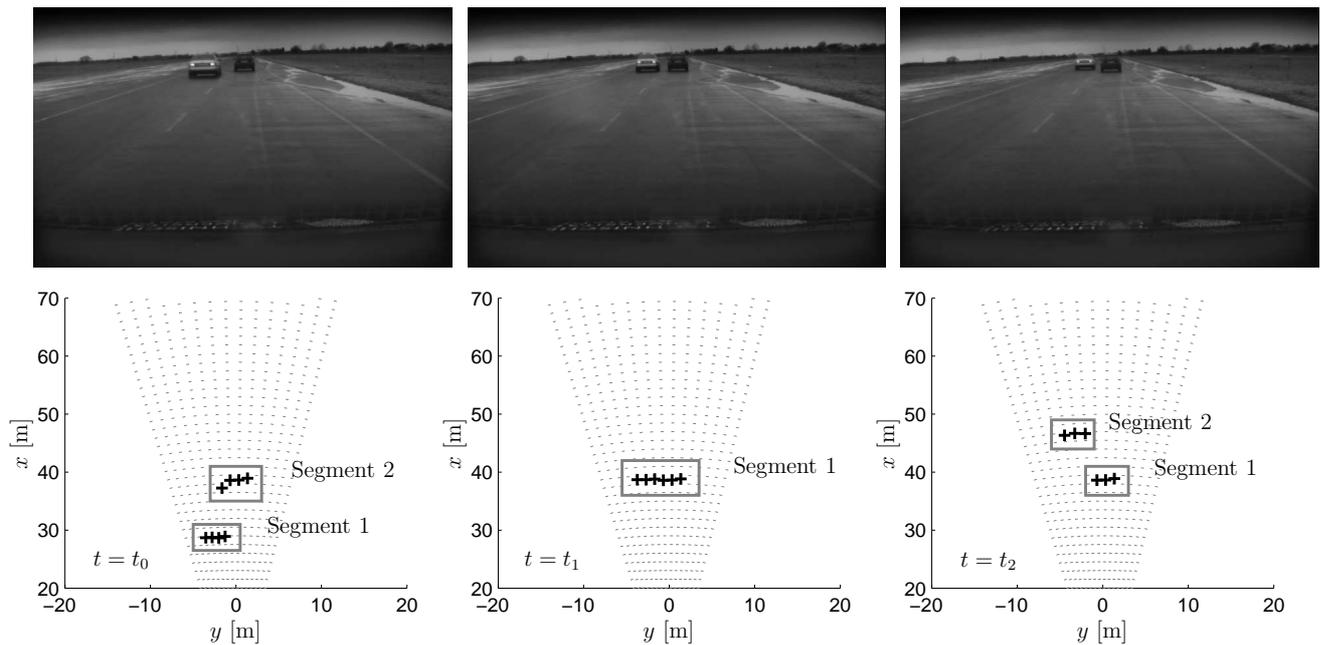
<sup>21</sup>Die Algorithmen gehören zur Klasse der neuronalen Netze, siehe Holt, V. v.: *Multisensorielle Fahrumgebungserfassung* (2005), S. 79.

<sup>22</sup>Holt, V. v.: *Multisensorielle Fahrumgebungserfassung* (2005), S. 83-92.

<sup>23</sup>Holt, V. v.: *Multisensorielle Fahrumgebungserfassung* (2005), S. 100.

<sup>24</sup>Siehe Holt, V. v.: *Multisensorielle Fahrumgebungserfassung* (2005), S. 106-108.

<sup>25</sup>Details zum verwendeten Sensor sind in Abschnitt 6.1 auf S. 72 beschrieben.



**Abbildung 4.2:** Segmentierung von Laserscannerdaten.

Eine in der Laserscannerdatenverarbeitung angewandte Methode der Merkmalsextraktion ist zunächst die Daten mittels einer Heuristik zu segmentieren, um zu Merkmals-hypothesen zu gelangen. Hierbei werden die einzelnen Entfernungsmessungen zu Clustern (Segmenten) zusammengefasst, die „idealerweise den selben Gegenstand vermessen“<sup>26</sup>, siehe Abbildung 4.2, Zeitpunkte  $t_0$  und  $t_2$ . Zur Bestimmung, welche Ziele zusammengehören, können z.B. geometrische Abstände zwischen den Messpunkten und Informationen über die Messgenauigkeit des Sensors herangezogen werden.<sup>27</sup>

Im Rahmen der Datenassoziation werden die Segmente dann Objekthypothesen zugeordnet, die auf den jeweiligen Messzeitpunkt des Laserscanners prädiert sind. Im gewählten Beispiel kommt es dabei zum Zeitpunkt  $t_1$  zu einer Fehlinterpretation in der Segmentierung: Aufgrund der Auflösung des verwendeten Laserscanners können die beiden Fahrzeuge durch die Heuristik nicht mehr getrennt werden. Dies zeigt sich spätestens während der Hypothesenauswahl, da weniger Merkmals-hypothesen als Objekthypothesen vorhanden sind.

Ähnliche Fehlinterpretationen können auftreten, wenn aufgrund von Sensorartefakten einzelne Messwerte ausfallen und so mehr Segmente erzeugt werden als auf Grund der Objekthypothesen zu erwarten sind.<sup>28</sup> Auf diesen Effekt hat sowohl die Auflösung des Laserscanners als auch die Heuristik bei der Segmentierung einen Einfluss.<sup>29</sup>

Auch der Einsatzbereich des Sensors spielt eine entscheidende Rolle. So schreibt Streller:

<sup>26</sup>Streller, D.: *Multi-Hypothesen-Ansatz* (2006), S. 29.

<sup>27</sup>Detaillierte Beschreibungen finden sich z.B. in Sparbert, J.; Dietmayer, K.; Streller, D.: *Laser Range Images* (2001); Kirchner, A.: *Sensordatenverarbeitung eines Laserscanners* (2000) S. 29-30; Streller, D.: *Multi-Hypothesen-Ansatz* (2006), S. 29.

<sup>28</sup>Abgesehen von neuen Objekten, die erstmals im Sichtbereich des Sensors auftauchen.

<sup>29</sup>In Streller, D.: *Multi-Hypothesen-Ansatz* (2006), S. 29-30 wird der Effekt diskutiert.

„Der Vorteil eines auf Multi-Hypothesen-Ansätzen basierenden Verfahrens wirkt sich bei übersichtlichen Situationen, beispielsweise auf Autobahnen mit schnell fließendem Verkehr, nicht besonders aus, da meistens ein Segment genau einem Gegenstand entspricht. In komplexen Szenarien mit dichten Abständen liefert die Segmentierung allerdings oft keine eindeutig zuweisbaren Ergebnisse.“<sup>30</sup>

Da der Effekt in Abbildung 4.2 auf die Auflösung des verwendeten Scanners zurückzuführen ist, lässt sich eine Trennung der Ziele beider Fahrzeuge nicht ohne weitere Informationen durchführen. Auch Algorithmen, die zunächst die Geschwindigkeit einzelner Ziele schätzen, können die Fahrzeuge dann nicht mehr trennen, wenn sie sich mit annähernd gleicher Geschwindigkeit bewegen.

Eine Segmentierung rein auf Basis des Vorwissens über prädizierte Objekte könnte die beschriebene Situation zwar lösen, ist aber auch nicht universell anwendbar, da so die Chance einer Korrektur einer Fehlinterpretation genommen wird. Die Hypothesen bewahrheiten sich sozusagen im Sinne einer „Self Fulfilling Prophecy“.

Zur Lösung des Problems schlägt Streller einen Multi-Hypothesen-Ansatz bzgl. der Objekthypothesen vor. Hierbei werden in jedem Zeitschritt „aus allen Segmenten und deren Segmentkombinationen neue Hypothesen [Anm.: Objekthypothesen] gebildet, die dann [...] über die Zeit verfolgt werden.“<sup>31</sup> In einem Hypothesen-Reduktionsschritt eliminiert ein Auswahlalgorithmus schließlich Objekthypothesen geringer Güte .

Ein Nachteil des Verfahrens ist der hohe Speicher- und Rechenzeitbedarf.<sup>32</sup> Geringere Anforderungen an die Ressourcen stellen Ansätze, die eine Änderung der Objekthypothesen bzw. Merkmals-hypothesen während der Datenassoziation zulassen.<sup>33</sup> Kirchner schlägt den im Folgenden beschriebenen Algorithmus vor.<sup>34</sup>

Die segmentierten Daten werden zu einem Merkmalsvektor verdichtet, bestehend aus Position des Schwerpunkts und Ausdehnung des jeweiligen Segments entlang der Koordinatenachsen. Auf Basis der prädizierten Objekthypothesen und Bewegungsgleichungen werden Bereiche (Gates) berechnet, in welchen gültige Merkmals-hypothesen für eine Objekthypothese liegen können.

Befinden sich innerhalb des Gültigkeitsbereiches einer Objekthypothese mehrere Merkmals-hypothesen, wird eine neue Merkmals-hypothese berechnet, die „sich aus der Vereinigungsmenge der einzelnen Punktmengen [Anm: aus den Segmenten] ergibt“<sup>35</sup>. Ist der geometrische Abstand dieser neu berechneten Merkmals-hypothese zur Objekthypothese kleiner als der kleinste Abstand einer der im Gültigkeitsbereich liegenden ursprünglichen Hypothesen, wird

<sup>30</sup>Streller, D.: *Multi-Hypothesen-Ansatz* (2006), S. 124.

<sup>31</sup>Streller, D.: *Multi-Hypothesen-Ansatz* (2006), S. 47.

<sup>32</sup>Siehe Streller, D.: *Multi-Hypothesen-Ansatz* (2006), S. 127.

<sup>33</sup>Siehe z.B. Kirchner, A.: *Sensordatenverarbeitung eines Laserscanners* (2000), S. 37 - 39; Holt, V. v.: *Multisensorielle Fahrumgebungserfassung* (2005), S. 106 - 108.

<sup>34</sup>Vgl. Kirchner, A.: *Sensordatenverarbeitung eines Laserscanners* (2000), S. 37 - 38.

<sup>35</sup>Kirchner, A.: *Sensordatenverarbeitung eines Laserscanners* (2000), S. 37.

die neue Merkmalshypothese zur Objekthypothese assoziiert. Dies stellt eine Revision der Merkmalsypothesen unter Rückgriff auf die Rohdaten der Messung dar.

Liegt umgekehrt eine Merkmalshypothese in den Gültigkeitsbereichen mehrerer Objekthypothesen, werden die Objekthypothesen zu einer neuen Objekthypothese fusioniert und der Abstand gegenüber der gemeinsamen Merkmalshypothese berechnet. Ist dieser Abstand kleiner als der zu den ursprünglichen Objekthypothesen, wird die Merkmalshypothese zur fusionierten Objekthypothese assoziiert. Dies stellt eine Revision der bestehenden Objekthypothesen im Assoziationschritt dar.

Das beschriebene Verfahren ist „ein Kompromiss zwischen Aufwand und Wirkung“<sup>36</sup>. Im vorgestellten Beispiel (Abbildung 4.2), führt es z.B. zu dem Ergebnis, dass zum Zeitpunkt  $t_1$  die Objekthypothesen fusioniert werden und zu einem späteren Zeitpunkt (nicht notwendiger Weise  $t_2$ ) ein neues Objekt entsteht.

Allerdings ist die in Abbildung 4.2 gezeigte Situation für den in der Arbeit Kirchner verwendeten Laserscanner weniger relevant, da die Auflösung mit  $0,35^\circ$  deutlich feiner ist als im hier gezeigten Beispiel.<sup>37</sup> Das Problem der Trennung zweier nebeneinander fahrender Fahrzeuge, insbesondere im Autobahn- bzw. Landstraßenverkehr, spielt hier eine untergeordnete Rolle. Entsprechend führt Kirchner in der Auflistung möglicher Probleme beim Objekt-Tracking den im Beispiel gezeigten Fall nicht auf.<sup>38</sup>

In Abschnitt 6.3.2<sup>39</sup> wird ein alternatives Verfahren vorgestellt, welches die Segmente bzw. Objekthypothesen zerteilt, anstatt sie zu fusionieren. Dieses Vorgehen liefert für einen Laserscanner mit gröberer Auflösung bessere Ergebnisse. Allerdings kann auch dieses Verfahren zu unerwünschten Ergebnissen führen, da z.B. einmal getrennte Objekte nicht wieder zusammengeführt werden.

## 4.4 Zusammenhang der Datenassoziation mit der Datenfilterung

Die Wahl der Algorithmen zur Datenassoziation beeinflusst die Wahl der Algorithmen zur Datenfilterung und umgekehrt. Eine vollkommen unabhängige Entwicklung beider Algorithmen ist daher nicht möglich.

---

<sup>36</sup>Kirchner, A.: *Sensordatenverarbeitung eines Laserscanners* (2000), S. 37.

<sup>37</sup>Für detaillierte Angaben zu diesem Laserscanner siehe Kirchner, A.: *Sensordatenverarbeitung eines Laserscanners* (2000), S. 12-17.

<sup>38</sup>Zum Vergleich: In 38 m Entfernung beträgt die räumliche Auflösung des im Beispiel in Abbildung 4.2 gezeigten Laserscanners ca. 1 m ( $1,5^\circ$  Segmentgröße).

<sup>39</sup>Siehe S. 80.

So existieren z.B. Filteralgorithmen, die eine explizite 1 : 1-Zuordnung von Merkmalshypothesen zu Objekthypothesen als Ergebnis der Datenassoziation benötigen, wie z.B. das Kalmanfilter.<sup>40</sup>

Daneben gibt es Filteralgorithmen, die ohne eine explizite Hypothesenauswahl auskommen und die Messungen stattdessen gewichtet mit einer auf der bayesischen Theorie beruhenden Assoziationswahrscheinlichkeit in den Schätzprozess einbringen. Ein Beispiel für solch einen Filteralgorithmus ist das Joint Probabilistic Data Association Filter (*JPDA*),<sup>41</sup> welches auch als „all-neighbour’s approach“<sup>42</sup> bezeichnet wird. Zur Berechnung der Assoziationswahrscheinlichkeiten existieren für diesen Ansatz auch Algorithmen, welche die Rohdaten explizit in die Berechnung der Wahrscheinlichkeiten einbeziehen und den Assoziationsprozess insbesondere in Umgebungen mit vielen Fehldetektionen verbessern können.<sup>43</sup>

Neben der Wahl der Algorithmen im Designprozess des Systems hängen die Prozesse der Datenassoziation und der Datenfilterung auch während der Laufzeit voneinander ab. Die geschätzten Zustandsgrößen aus der Datenfilterung sind die Grundlage für die Prädiktion der Merkmalshypothesen für die Assoziation. Das Zeitverhalten und die Genauigkeit des in der Datenfilterung verwendeten Filters können auf diesem Weg die Anzahl fehlerhafter Zuordnungen beeinflussen.

Zur weiteren Betrachtung wird wieder das Beispiel aus Kapitel 3.3 herangezogen.<sup>44</sup> Die Zustandsgrößen  $d_{\text{rel}}$ ,  $v_{\text{rel}}$  und  $a_{\text{rel}}$  eines vorausfahrenden Fahrzeugs werden über einen Kalmanfilter aus Abstandswerten bestimmt, welchen eine additive Störung  $n \sim \mathcal{N}(0, \sigma_n)$  überlagert ist. Betrachtet wird die eindeutige Zuordnung von Merkmalshypothesen zu Objekthypothesen unter Verwendung eines Gating-Verfahrens, wie dies im Bereich der Umfelddatenverarbeitung in der Fahrzeugtechnik auch eingesetzt wird.<sup>45</sup> Die Wahl der Gategröße beeinflusst dabei die Möglichkeit für das Entstehen von Fehlern sowohl des Typs I als auch des Typs II bei der Zuordnung.

Wird das Gate klein gewählt, ist es möglich, dass eine richtige Zuordnung schon im Schritt der Hypothesengenerierung ausgeschlossen wird. Dies kann soweit führen, dass aufgrund fehlender Informationen für die Track-Schätzung die Ungenauigkeit so weit ansteigt, dass es nicht mehr sinnvoll möglich ist, den Track weiter zu schätzen, der Track wird somit verloren.

<sup>40</sup>Siehe z.B. Hall, D. L.: *Handbook of multisensor data fusion* (2001), S. 153-154; Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 199-265; Kapitel 3, S. 24.

<sup>41</sup>Siehe z.B. Bar-Shalom, Y.; Li, X.-R.: *Multitarget-multisensor tracking* (1995), S. 307-371.

<sup>42</sup>Hall, D. L.; McMullen, S. A.: *Mathematical techniques in multisensor data fusion* (2004), S. 124.

<sup>43</sup>Siehe z.B. Bar-Shalom, Y.; Li, X.-R.: *Multitarget-multisensor tracking* (1995), S. 151, S. 190-198, S. 247-276.

<sup>44</sup>Siehe S. 31.

<sup>45</sup>Siehe z.B. Becker, J.-C.: *Fusion der Daten der objekterkennenden Sensoren* (2002), S. 34-36; Stüker, D.: *Heterogene Sensordatenfusion* (2004), S. 47-48; Kirchner, A.: *Sensordatenverarbeitung eines Laserscanners* (2000), S. 33-34.

Ist das Gate groß, kann zwar die Wahrscheinlichkeit für einen Fehler des Typs I minimiert werden, allerdings steigt damit wiederum die Komplexität bei der Hypothesenevaluierung und -auswahl sowie die Wahrscheinlichkeit für einen Fehler des Typs II.

Die Berechnung der Gategröße lässt sich beim Kalmanfilter auf eine theoretische Grundlage stellen.<sup>46</sup> Bei Gültigkeit der Modellannahmen, ist die normalisierte, quadrierte Innovation  $\epsilon$  mit

$$\epsilon = (z - \bar{z})' S (z - \bar{z}) \quad (4.1)$$

$\chi^2$ -verteilt mit  $n_z$  Freiheitsgraden.<sup>47</sup> Hierbei ist  $z$  der Messvektor (im Beispiel die gemessene Entfernung),  $\bar{z}$  der auf den Messzeitpunkt prädizierte Messvektor,  $n_z$  die Dimension des Messvektors (im Beispiel gleich eins) und  $S$  die Innovationskovarianzmatrix des Kalmanfilters, welche ein Maß für die Genauigkeit der Prädiktion darstellt. Über die Beziehung (4.1) lässt sich ein Volumen bzw. Gate

$$\mathcal{V}(\bar{z}, S, \gamma) = \{z : (z - \bar{z})' S (z - \bar{z}) < \gamma\} \quad (4.2)$$

definieren, in dem der Messwert mit einer von  $\gamma$  abhängigen Wahrscheinlichkeit liegt.  $\gamma$  wird dabei entsprechend der gewünschten Wahrscheinlichkeit aus der  $\chi^2$ -Verteilung entnommen.

Wird wie in Kapitel 3.3<sup>48</sup> ein DWPA-Modell für die Schätzung zugrunde gelegt, konvergiert  $S$  gegen einen konstanten Wert. Die Größe des Gates hängt dann nur noch vom Modellrauschen  $\sigma_v$ , dem Messrauschen<sup>49</sup>  $\sigma_w$  und der Abtastperiode  $T$  ab.

Durch eine Multiplikation von  $\sigma_v$  und  $\sigma_w$  mit dem gleichen Faktor ändert sich dabei der Manöverindex

$$\lambda = \frac{\sigma_v T^2}{\sigma_w} \quad (4.3)$$

und damit das dynamische Verhalten und die erzielte Messgenauigkeit des Kalmanfilters nicht.<sup>50</sup> Die Gategröße kann hierdurch allerdings beliebig beeinflusst werden, da  $\sigma_v$  und  $\sigma_w$  additiv in die Berechnung der Innovationskovarianzmatrix eingehen.<sup>51</sup>

Für die folgenden Betrachtungen wird  $\sigma_w = \sigma_n$  gesetzt. Zudem wird  $\gamma$  so gewählt, dass die Wahrscheinlichkeit für das Auffinden einer Merkmalshypothese im Gate basierend auf dem Modell des Kalmanfilters 99,7 % beträgt.<sup>52</sup>

<sup>46</sup>Siehe z.B. Bar-Shalom, Y.; Li, X.-R.: *Multitarget-multisensor tracking* (1995), S. 94-96.

<sup>47</sup>Siehe z.B. Bar-Shalom, Y.; Li, X.-R.: *Multitarget-multisensor tracking* (1995), S. 95.

<sup>48</sup>Siehe S. 31.

<sup>49</sup> $\sigma_w$  ist ein Modellierungsparameter des Kalmanfilters. Die Größe muss nicht gleich dem Sensorrauschen  $\sigma_n$  gewählt werden.

<sup>50</sup>Vgl. die Ausführungen in Abschnitt 3.2, S. 25.

<sup>51</sup>Siehe z.B. Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 208.

<sup>52</sup>Diese Festlegungen sind zunächst willkürlich, entsprechen aber einem in der Praxis angewandten Vorgehen, vgl. z.B. Bar-Shalom, Y.; Li, X.-R.: *Multitarget-multisensor tracking* (1995), S. 96; Stüker, D.: *Heterogene Sensordatenfusion* (2004), S. 49; Kirchner, A.: *Sensordatenverarbeitung eines Laserscanners* (2000), S. 36, hier wird z.B. der Wert  $\gamma = 99\%$  vorgeschlagen.

Die freie Größe  $\sigma_v$  kann dazu genutzt werden, das Verhalten des Kalmanfilters zu variieren. Durch eine entsprechende Wahl werden systematische Abweichungen vom Modell (Manöver wie z.B. ein plötzliches Bremsen des vorausfahrenden Fahrzeugs)<sup>53</sup> durch real nicht vorhandenes Rauschen („Pseudo Noise“) in das Modell aufgenommen. Dieses Vorgehen wird auch als „Filter Tuning“ bezeichnet.<sup>54</sup>

Die Wahrscheinlichkeit, mit der sich die Merkmalshypothese im Gate befindet, im folgenden als Gatewahrscheinlichkeit bezeichnet, lässt sich dann wie folgt berechnen:

$$\mathcal{P}(z \in \mathcal{V}(S, \gamma)) = \int \int_{\mathcal{V}(\bar{z}, S, \gamma)} f_{z\bar{z}}(z, \bar{z}) dz d\bar{z} = \int \mathcal{P}(z \in \mathcal{V}(\bar{z}, S, \gamma)) f_{\bar{z}}(\bar{z}) d\bar{z}. \quad (4.4)$$

Hierbei ist  $f_{z\bar{z}}(z, \bar{z}) = f_z(z) f_{\bar{z}}(\bar{z})$  die gemeinsame Dichte von  $z$  und  $\bar{z}$ , wobei die Variablen  $z$  und  $\bar{z}$  statistisch unabhängig sind, da die überlagerte Störung weißem Rauschen entspricht.

Ist das Kalmanfilter eingeschwungen, geht es in das lineare  $\alpha\beta\gamma$ -Filter über.<sup>55</sup> Damit gilt für die Prädiktion  $\bar{z}$

$$\bar{z} \sim \mathcal{N}(\bar{z}|_{\sigma_n=0}, \sigma_{\bar{z}}). \quad (4.5)$$

Hierbei ist  $\bar{z}|_{\sigma_n=0}$  der Wert von  $\bar{z}$ , welcher sich einstellt, wenn  $\sigma_n = 0$  gilt, also rein deterministisches Verhalten vorliegt. Die Varianz  $\sigma_{\bar{z}}$  des zufälligen Anteils berechnet sich mit der Impulsantwort von Störung  $n$  zur Prädiktion  $\bar{z}$   $h_{n\bar{z}}(n)$  analog (3.5) zu:

$$\sigma_{\bar{z}}^2 = \sigma_n^2 \sum_{n=0}^{\infty} h_{n\bar{z}}(n)^2. \quad (4.6)$$

Für die Messung  $z$  gilt

$$z \sim \mathcal{N}(d_{\text{rel}}, \sigma_n). \quad (4.7)$$

Der Abstand  $\nu$  von Messung zu Prädiktion ergibt sich zu

$$\nu = z - \bar{z} \quad (4.8)$$

und ist damit die Differenz zweier normalverteilter, statistisch unabhängiger Zufallsvariablen. Die Varianz  $\sigma_\nu$  berechnet sich damit zu.<sup>56</sup>

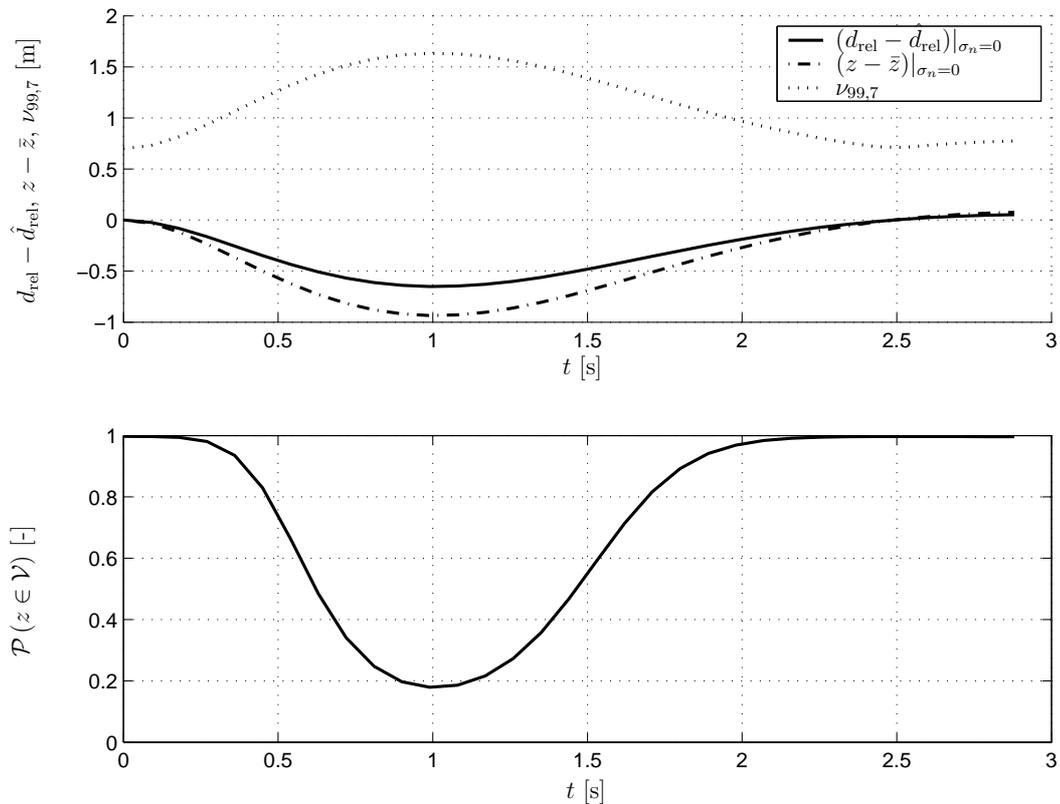
$$\sigma_\nu = \sqrt{\sigma_n^2 + \sigma_{\bar{z}}^2}, \quad (4.9)$$

so dass gilt:

$$\nu \sim \mathcal{N}(z|_{\sigma_n=0} - \bar{z}|_{\sigma_n=0}, \sigma_\nu). \quad (4.10)$$

99,7 % der Abstände zwischen Messung und Prädiktion sind damit kleiner als

$$\nu_{99,7} = |z|_{\sigma_n=0} - \bar{z}|_{\sigma_n=0}| + 3\sigma_\nu. \quad (4.11)$$



**Abbildung 4.3:** Unterschied der Abstandsschätzung und der Prädiktion der Messung zu den ungestörten Werten (oben), sowie Verlauf der maximalen Assoziationsdistanz und der Gatewahrscheinlichkeit bei einem Beschleunigungssprung von  $a_{\text{fw}} = 0$  auf  $a_{\text{fw}} = -8 \text{ m/s}^2$  (unten).

Dieser Wert wird im Folgenden als maximale Assoziationsdistanz bezeichnet.

Abbildung 4.3 zeigt den Verlauf der Gatewahrscheinlichkeit und der Prädiktion bei einem Sprung der Beschleunigung des vorausfahrenden Fahrzeugs von  $a_{\text{fw}} = 0$  bei  $t = 0$  auf  $a_{\text{fw}} = -8 \text{ m/s}^2$  für  $\sigma_n = \sigma_w = 0,2 \text{ m}$  und  $\lambda = \lambda_{\text{opt}}^{\text{acc}} \approx \lambda_{\text{opt}}^{\text{ttc},1 \text{ s}}$ , dem bei dieser Sensorgenauigkeit optimalen Manöverindex für ACC bzw. für eine  $t_{\text{TTC}}$ -basierte Anwendung mit  $t_{\text{TTC}} = 1 \text{ s}$ . Die Gategröße ergibt sich bei diesen Parametern zu  $\pm 0,7 \text{ m}$ . Bei einer Entfernung der Fahrzeuge von  $33 \text{ m}$  zum Zeitpunkt  $t = 0$ , kommt es am Ende des Verlaufs zu einer Kollision.

Aufgrund der Trägheit des Filters fällt die Gatewahrscheinlichkeit auf einen Wert unter  $18 \%$ . Das Maximum der maximalen Assoziationsdistanz ( $\max \nu_{99,7}$ ) liegt über  $1,5 \text{ m}$ , das Minimum ( $\min \nu_{99,7} = 3\sigma_\nu$ ) bei ca.  $0,7 \text{ m}$ . Dies ist zugleich der kleinste Wert, welchen  $\nu_{99,7}$  annehmen kann (der Absolutbetrag in (4.11) ist null).

Die Wahrscheinlichkeit, dass alle Messwerte ohne einen Fehler des Typs I über den gesamten gezeigten Verlauf assoziiert werden, beträgt null. Von daher ist es sehr wahrscheinlich, dass

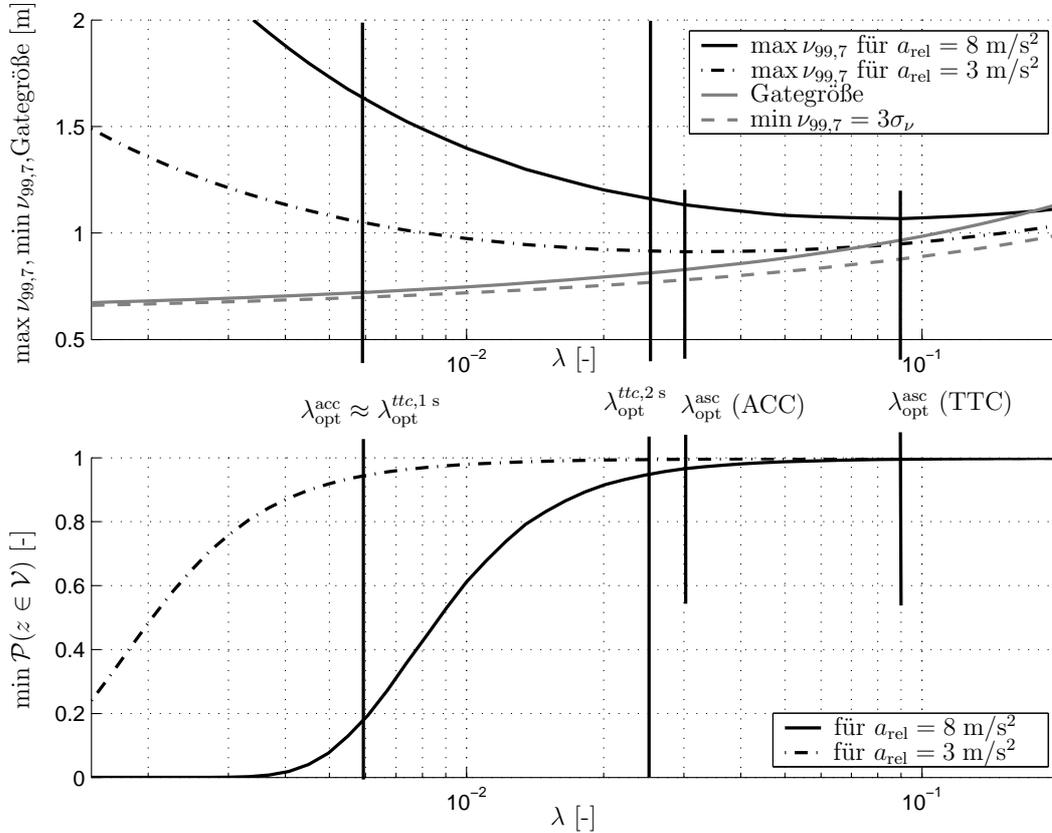
<sup>53</sup>Zum Begriff Manöver siehe z.B. Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 421.

<sup>54</sup>Siehe z.B. Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 243.

<sup>55</sup>Siehe auch S. 27.

<sup>56</sup>Siehe z.B. Hänsler, E.: *Statistische Signale* (1997), S. 97.

der Track verloren wird, so dass das Filter ungeeignet für die Datenassoziation für die  $t_{\text{TTC}}$ -basierte Anwendung mit  $t_{\text{TTC}} = 1$  s ist.



**Abbildung 4.4:** Einfluss des Manöverindex  $\lambda$  auf die Datenassoziation bei  $\sigma_w = \sigma_n = 0,2$  m.

Abbildung 4.4 zeigt den Einfluss des Manöverindex  $\lambda$  auf die Datenassoziation, wobei  $\sigma_w = \sigma_n$  gilt. Für jeden Wert von  $\lambda$  wurde wie in Abbildung 4.3 ein Beschleunigungsmanöver berechnet.<sup>57</sup> Hierbei wurden die Beschleunigungen  $a_{\text{rel}} = 8$  m/s<sup>2</sup>, die der maximalen Beschleunigung für die  $t_{\text{TTC}}$ -basierte Anwendung entspricht,<sup>58</sup> und  $a_{\text{rel}} = 3$  m/s<sup>2</sup>, die eine Mindestanforderung hinsichtlich der zu schätzenden Beschleunigungen für eine ACC-Anwendung darstellt, ausgewertet.<sup>59</sup>

Zu sehen ist, dass die Gategröße und das Minimum von  $\nu_{99,7}$  streng monoton mit  $\lambda$  ansteigen. Beide Größen sind unabhängig von der Beschleunigung  $a_{\text{rel}}$ . Auch die minimale Gatewahrscheinlichkeit  $\min \mathcal{P}(z \in \mathcal{V})$  steigt streng monoton mit  $\lambda$  an, ist allerdings abhängig von  $a_{\text{rel}}$ , da hier die Trägheit des Filters mit eingeht.

Die Kurve des Maximums der maximalen Assoziationsdistanz hat jeweils ein Minimum, welches deutlich rechts vom optimalen Manöverindex  $\lambda$  für die jeweilige Anwendung liegt. Mit diesem Manöverindex  $\lambda_{\text{opt}}^{\text{asc}}$  wird die Wahrscheinlichkeit eines Fehlers des Typs II für den dy-

<sup>57</sup>Aufgrund der Linearität des Modells spielt es keine Rolle, ob verzögert oder beschleunigt wird.

<sup>58</sup>Siehe S. 34.

<sup>59</sup>Der Beschleunigungswert für ACC wurde dabei unterschätzt, der für die  $t_{\text{TTC}}$ -basierte Anwendung überschätzt. Somit ist im Folgenden eine „Worst Case“ Betrachtung sichergestellt.

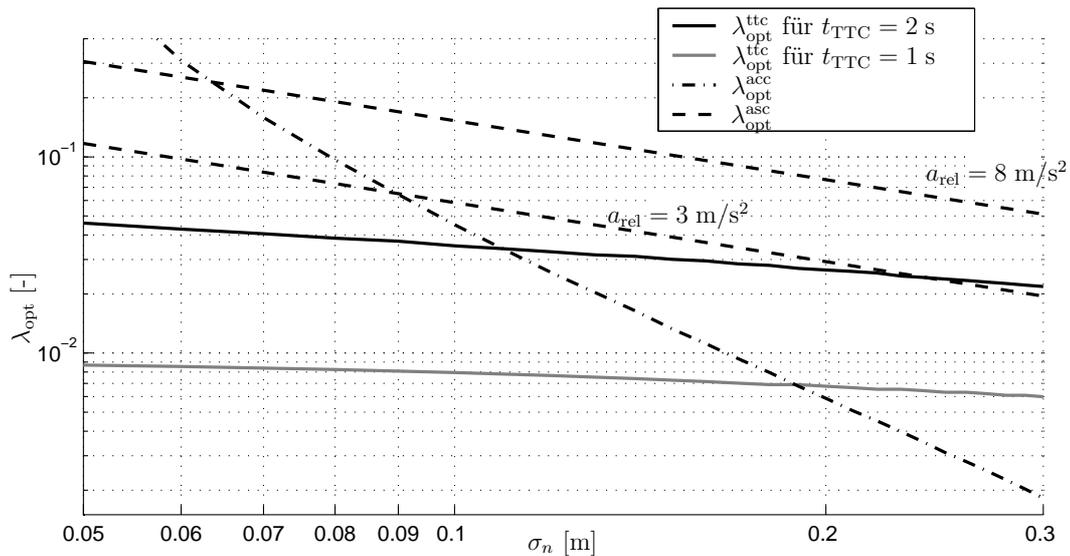
namischen Fall minimiert.<sup>60</sup> Der Wert für die minimale Assoziationswahrscheinlichkeit liegt dabei jeweils bei nahezu eins, wodurch ein Fehler des Typs I unwahrscheinlich wird. Tabelle 4.1 zeigt die entsprechenden Zahlenwerte.

**Tabelle 4.1:** Einfluss des Manöverindex  $\lambda$  auf die Datenassoziation ( $\sigma_n = 0,2$  m).

	$\lambda_{\text{opt}}^{\text{acc}} \approx \lambda_{\text{opt}}^{\text{ttc},1\text{ s}}$	$\lambda_{\text{opt}}^{\text{ttc},2\text{ s}}$	$\lambda_{\text{opt}}^{\text{asc}} (\text{ACC})$	$\lambda_{\text{opt}}^{\text{asc}} (\text{TTC})$
Gategröße [m]	0,72	0,81	0,83	0,96
min $\nu_{99,7}$ [m]	0,70	0,77	0,78	0,88
max $\nu_{99,7}$ [m] ( $a_{\text{rel}} = 3 \text{ m/s}^2$ )	1,04	0,92	0,91	0,94
max $\nu_{99,7}$ [m] ( $a_{\text{rel}} = 8 \text{ m/s}^2$ )	1,62	1,17	1,13	1,07
min $P(z \in \mathcal{V})$ [-] ( $a_{\text{rel}} = 3 \text{ m/s}^2$ )	0,94	0,99	1,00	1,00
min $P(z \in \mathcal{V})$ [-] ( $a_{\text{rel}} = 8 \text{ m/s}^2$ )	0,19	0,95	0,97	1,00

Wird zur Datenfilterung für die Assoziation bei der ACC-Funktion ( $a_{\text{rel}} = 3 \text{ m/s}^2$ ) anstatt des optimalen Manöverindex  $\lambda_{\text{opt}}^{\text{asc}}(\text{ACC})$  der für die  $t_{\text{TTC}}$ -basierte Anwendungen optimale Manöverindex  $\lambda_{\text{opt}}^{\text{asc}}(\text{TTC})$  verwendet, ändert sich  $\max \nu_{99,7}$  nur um 0,03 m bzw. 3 %. Ein Nachteil des größeren Manöverindex ist dabei ein größeres Gate und eine höhere Varianz der Prädiktion  $\bar{z}$ , die wiederum zu einem größeren Wert für  $\min \nu_{99,7}$  führt. Beides erhöht die Wahrscheinlichkeit für einen Fehler des Typs II im statischen Fall.

Da aber immer  $\min \nu_{99,7} \leq \max \nu_{99,7}$  gilt, wird das Verhalten im statischen Fall nie schlechter sein als im dynamischen Fall. Dieser stellt damit den „Worst Case“ für die Auslegung des Filters zur Assoziation dar. Die Wahl von  $\lambda_{\text{opt}}^{\text{asc}} (\text{TTC})$  wird daher zu keinem signifikant schlechteren Verhalten der ACC-Funktion führen.

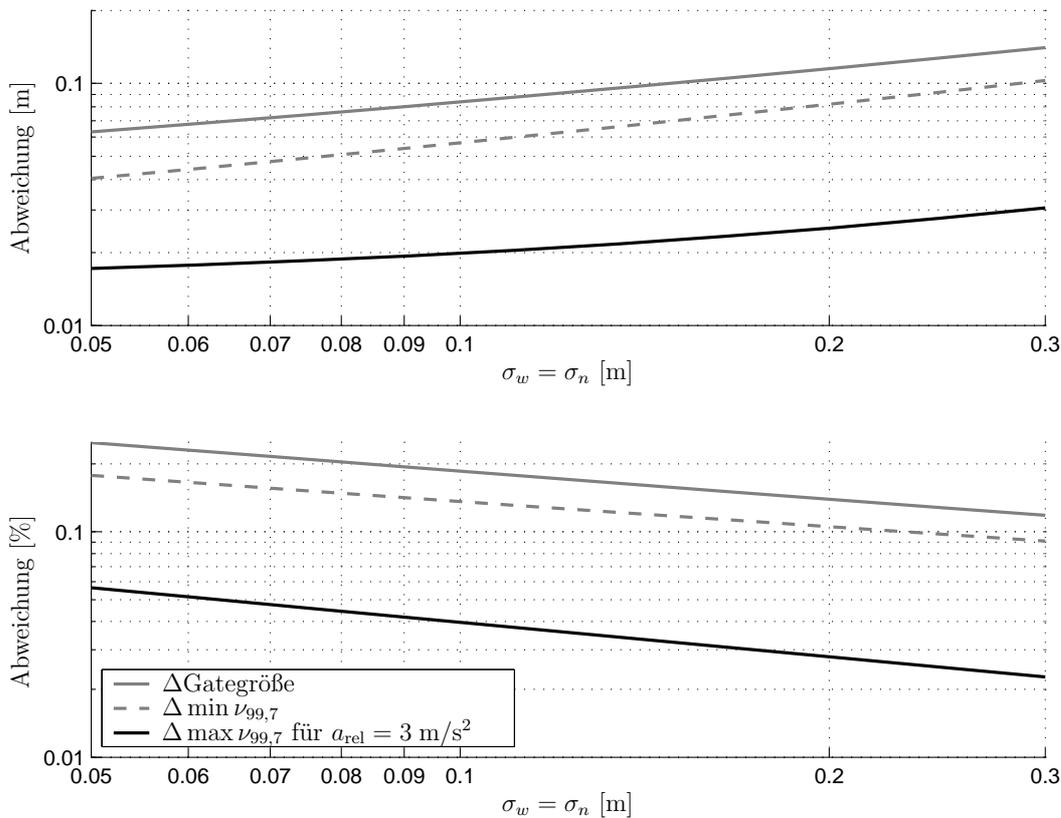


**Abbildung 4.5:** Optimaler Manöverindex  $\lambda$  für die Datenassoziation.

<sup>60</sup>Anmerkung: Im statischen Fall ist das Optimum für den Manöverindex  $\lambda_{\text{opt}}^{\text{asc}} = 0$ . Dieser Fall kann daher für eine Optimierungsaufgabe nicht herangezogen werden.

Abbildung 4.5 zeigt die optimalen Werte für den Manöverindex  $\lambda$  bzgl. der Datenassoziation in Abhängigkeit der Sensorgenauigkeit  $\sigma_n$  und der Beschleunigung  $a_{\text{rel}}$  im Vergleich zu den jeweiligen optimalen Werten des Manöverindex  $\lambda$  für ACC bzw. für eine  $t_{\text{TTC}}$ -basierte Anwendung.<sup>61</sup>

Der optimale Manöverindex für die Assoziation  $\lambda_{\text{opt}}^{\text{asc}}$  für die  $t_{\text{TTC}}$ -basierte Anwendung liegt im gesamten Bereich signifikant über dem optimalen Wert für die  $t_{\text{TTC}}$ -Berechnung. Auch für ACC ergibt sich fast über den gesamten Bereich ein deutlicher Unterschied.



**Abbildung 4.6:** Abweichungen der Gategröße und der Extremwerte der maximalen Assoziationsdistanz bei Verwendung des Manöverindex  $\lambda_{\text{opt}}^{\text{asc}}$  (TTC) anstatt des optimalen Manöverindex  $\lambda_{\text{opt}}^{\text{asc}}$  (ACC) zur Datenassoziation für die ACC-Funktion.

Abbildung 4.6 zeigt die absolute und prozentuale Änderung des Maximums und des Minimums der maximalen Assoziationsdistanz sowie der Gategröße bei Verwendung von  $\lambda_{\text{opt}}^{\text{asc}}$  (TTC) anstatt  $\lambda_{\text{opt}}^{\text{asc}}$  (ACC) für die ACC-Funktion ( $a_{\text{rel}} = 3 \text{ m/s}^2$ ). Selbst bei einem sehr geringen Sensorrauschen  $\sigma_n$  ist die Abweichung der maximalen Assoziationsdistanz vom Optimalwert kleiner als 6 %. Die absoluten Werte liegen dabei selbst bei  $\sigma_n = 0,3 \text{ m}$  bei weniger als 0,15 m, was als Größe bezogen auf Anwendungen im Straßenverkehr als gering angesehen werden kann.<sup>62</sup>

<sup>61</sup>Vgl. auch Abbildung 3.10, S. 35.

<sup>62</sup>Zum Vergleich: Einparkhilfen (z.B. ultraschallbasiert) geben ab ca. 0,3 m einen Dauerton für die dichteste Entfernung zu einem Hindernis ab.

## 4.5 Fazit zur Datenassoziation

Die sensorspezifischen Rohdaten können zur Optimierung des Assoziationsprozesses herangezogen werden. Dies gilt sowohl für den Prozess der Hypothesengenerierung als auch für die Hypothesenevaluierung. Auch im Schritt der Hypothesenauswahl spielen die sensorspezifischen Rohdaten eine Rolle. So können je nach Sensortechnologie, also z.B. Laserscanner oder Videosensor, verschiedene Algorithmen zur Hypothesenauswahl existieren, die auf die jeweilige Technologie angepasst sind.

Die Datenassoziation hängt mit der Merkmalsextraktion und Objekthypothesenbildung zusammen. Auch hierbei existieren verschiedene sensorspezifische Möglichkeiten, die einzelnen Prozesse zu optimieren bzw. aufeinander abzustimmen, um schließlich zu einer bestmöglichen<sup>63</sup> Zuordnung von Merkmalshypothesen zu Objekthypothesen zu gelangen.

Wissen über die Entstehung der Daten, wie z.B. mögliche Artefakte und typische Fehlinterpretationen, kann zur Optimierung der Algorithmen eingesetzt werden. Daneben können auch spezielle Ausprägungen einer Sensortechnologie, wie z.B. das Auflösungsvermögen, beim Entwurf der Algorithmen berücksichtigt werden. Das Design der Algorithmen zur Datenassoziation hängt daher mit dem Wissen über die Entstehung der Daten und damit der Hardware des eingesetzten Sensors zusammen.

Die Annahme zur Abhängigkeit der Datenassoziation von den Rohdaten<sup>64</sup> konnte damit bestätigt werden. Die gezeigten Beispiele verdeutlichen, dass hier ohne einen Zugriff auf die Rohdaten und eine Berücksichtigung der sensorspezifischen Gegebenheiten die Güte der Datenassoziation abnimmt.

Die Datenassoziation lässt sich nicht von der Datenfilterung trennen. Dies trifft zum einen auf den Designprozess zu, in dem zueinander kompatible Algorithmen gefunden werden müssen, zum anderen auf das Laufzeitverhalten, da die Dynamik der Datenfilterung die Güte des Assoziationsprozesses beeinflusst.

Auch hier führt die Optimierung des eingesetzten Schätzalgorithmus für unterschiedliche Anwendungsszenarien zu verschiedenen Filterparametern. Die Filter unterscheiden sich je nach Sensorrauschen signifikant von den auf die Anwendung optimierten Filtern.

Die starke und die schwache Annahme für getrennte Filter zur Datenassoziation<sup>65</sup> konnten damit belegt werden. Wie in Kapitel 3 wurden in der Untersuchung Anwendungen herangezogen, die bereits in Serie sind oder bald in Serie gehen werden sowie Sensorgenauigkeiten betrachtet, mit welchen die jeweiligen Anwendungen realisierbar sind, so dass eine Übertragbarkeit der Aussagen gegeben ist.

---

<sup>63</sup>Im Sinne vorhandener Ressourcen, wie z.B. vorhandenem Know How oder der Leistungsfähigkeit der Hardware.

<sup>64</sup>Siehe S. 23.

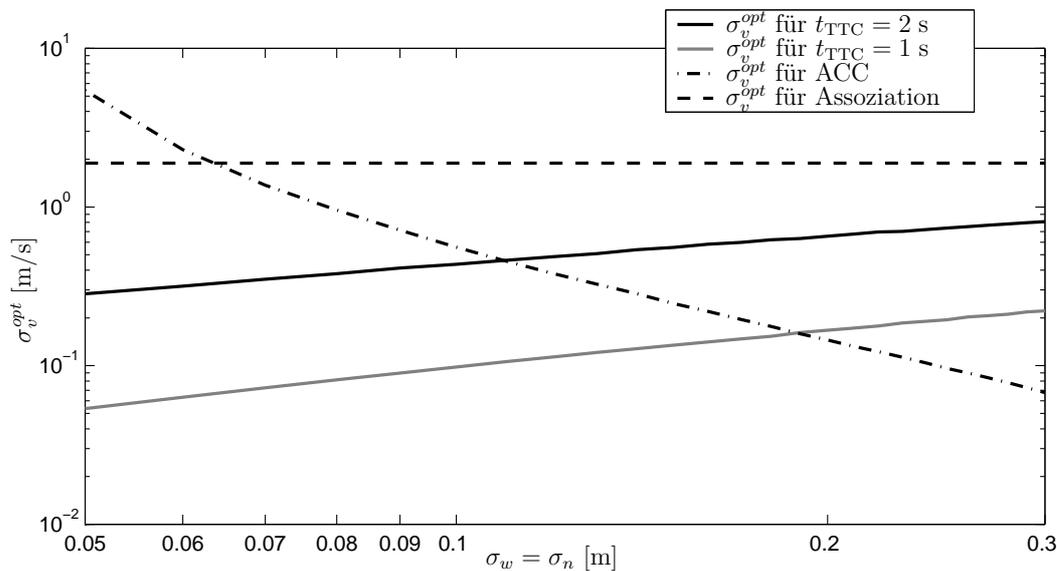
<sup>65</sup>Siehe S. 22.

Soll die Systemarchitektur für verschiedene Anwendungen und verschiedene Messungenauigkeiten geeignet sein, muss daher eine besondere Behandlung der Datenfilterung für die Datenassoziation innerhalb der Architektur vorgesehen werden.

Die schwache Annahme für getrennte Filter zur Datenassoziation<sup>66</sup> gilt im gezeigten Fall wiederum für nahezu alle Werte des Sensorrauschens  $\sigma_n$ . Die Betrachtung der  $t_{TTC}$ -basierten Anwendungen<sup>67</sup> zeigt, dass der Effekt für geringe Messungenauigkeiten sogar zunehmen kann.

Untereinander unterscheiden sich die auf die Assoziation optimierten Filter allerdings deutlich weniger als die Gruppe der auf die Anwendung optimierten Filter.<sup>68</sup> In Kaufnahme einer geringfügigen Abweichung vom Optimum kann daher das dynamischere Filter zur Assoziation eingesetzt werden.

Die Annahme für das Ausreichen eines Einzelfilters für die Datenassoziation<sup>69</sup> konnte somit nicht widerlegt werden. Dabei wurden Anwendungen mit stark unterschiedlichen Anforderungen an die Dynamik und Genauigkeit der Filteralgorithmen betrachtet. Selbst für ein sehr geringes Sensorrauschen ist hier der Nachteil der Nutzung des dynamischeren Filters zur Datenassoziation nur mit einer für die Anwendung unbedeutenden Verschlechterung der Assoziation verbunden.



**Abbildung 4.7:** Optimales Modellrauschen  $\sigma_v$  für Datenassoziation und Anwendungen.

Im Vergleich zur Optimierung eines Filters für die Ziele einer Anwendung, wie z.B. komfortables und kolonnenstabiles Folgeverhalten bei ACC, steht bei der Optimierung des Filters zur Datenassoziation die möglichst exakte Abbildung der Umgebung im Vordergrund. Dies spiegeln auch die Ergebnisse der unterschiedlichen Optimierungen wider.

<sup>66</sup>Siehe S. 23.

<sup>67</sup>Siehe Abbildung 4.5.

<sup>68</sup>Siehe Kapitel 3, S. 24.

<sup>69</sup>Siehe S. 23.

Abbildung 4.7 zeigt den optimalen Wert des Modellrauschens  $\sigma_v^{opt}$ , wenn für das Messrauschen  $\sigma_w = \sigma_n$  gewählt wird. Für ACC nimmt mit steigender Sensorgenauigkeit ( $\sigma_n \rightarrow 0$ ) das optimale Modellrauschen  $\sigma_v^{opt}$  zu, das Filter wird so dynamisch wie möglich eingestellt. Für die  $t_{TTC}$ -basierte Anwendung nimmt  $\sigma_v^{opt}$  hingegen ab, das Filter wird so träge wie möglich eingestellt, um das Sensorrauschen bestmöglich zu unterdrücken. Für die Datenassoziation ist  $\sigma_v$  annähernd konstant. Hier ist es das Ziel, über das Modellrauschen Manöver zu modellieren. Im optimalen Wert für  $\sigma_v$  spiegeln sich daher die Eigenschaften der Manöver wider.

# 5 Basis-Systemarchitektur zur Sensordatenfusion für Fahrerassistenzsysteme

## 5.1 Überblick zur Architektur

Im Folgenden wird die Basis-Systemarchitektur zur Sensordatenfusion von Umfoldsensoren für Fahrerassistenzsysteme vorgestellt.<sup>1</sup> Die Architektur baut auf den in Abschnitt 2.3 aufgestellten Grundannahmen auf. Die Gültigkeit der Annahmen wurde in den Kapiteln 3 und 4 dargelegt.<sup>2</sup> Sie bilden das Fundament für die hier beschriebene Architektur.<sup>3</sup> Die Robustheit der Architektur bzgl. der Gültigkeit der Grundannahmen wird in Abschnitt 5.4 diskutiert.

Abbildung 5.1 zeigt die Bausteinsicht der Basis-Systemarchitektur. Die Umsetzung der Laufzeitsicht wird soweit möglich offen gehalten.<sup>4</sup> Auswirkungen unterschiedlicher Gestaltungen der Laufzeitsicht auf die Bausteinsicht werden an entsprechender Stelle im Text diskutiert.

Die Architektur ist in drei Ebenen aufgeteilt: Die unterste Ebene bildet die Sensorebene, die oberste die Anwendungsebene, dazwischen liegt die Fusionsebene. Von Sensor- und Anwendungsebene können jeweils mehrere Instanzen existieren, die Fusionsebene wird nur einmal instantiiert.

Die Architektur ist auf orthogonale Sensoren optimiert. Für Sensoren, die mit Rohdaten im gleichen Format arbeiten, wird eine zentrale Fusion mit synchronisierten Sensoren auf Rohdatenebene vorgeschlagen. Diese Sensoren (z.B. Stereokamera) bilden dann im Rahmen der Architektur einen eigenen Rohdatensensor, der jeweils durch eine eigene Instanz einer Sensorebene repräsentiert wird. Abweichend davon ist die Architektur aber auch für nicht orthogonale Sensoren geeignet, wobei die Vorteile einer Fusion auf Rohdatenebene dann allerdings nicht realisiert werden können.

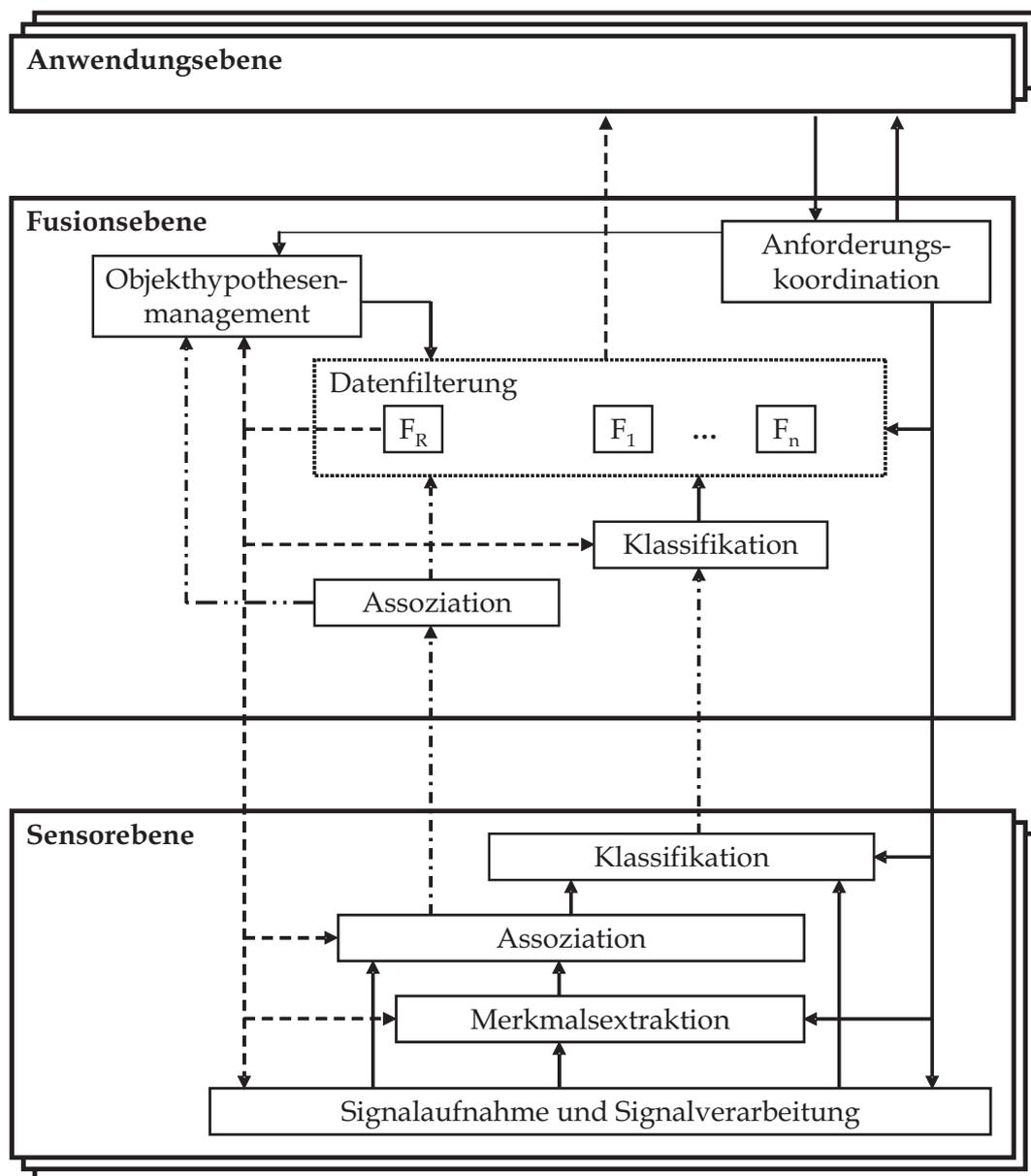
---

<sup>1</sup>Die Grundidee wurde erstmals in Darms, M.; Winner, H.: *Fusion von Umfelddaten* (2003) veröffentlicht. Siehe auch Darms, M.; Winner, H.: *Modular System Architecture* (2005).

<sup>2</sup>Die Gültigkeit der existentiellen Hypothesen wurde gezeigt, die universellen Hypothesen konnten nicht widerlegt werden.

<sup>3</sup>Anmerkung: Hiermit ist allerdings nicht ausgesagt, dass bei Gültigkeit der Annahmen die Umsetzbarkeit der Architektur nicht auf andere Weise widerlegt werden kann.

<sup>4</sup>Siehe hierzu Abschnitt 6.6, S. 85.



L2 Darstellung (sensorunabhängig) - - - - -

L1 Darstellung (sensorspezifisch) - . - . -

**Abbildung 5.1:** Bausteinsicht der Basis-Systemarchitektur mit Datenflüssen zwischen den Bausteinen.

Grundsätzlich werden auf Sensorebene alle Funktionen bzgl. der Umfelderkennung implementiert, die einen Zugriff auf die Rohdaten benötigen, bzw. die durch einen Zugriff auf die Rohdaten optimiert werden können. Diese werden im Folgenden als sensorspezifische Funktionen bezeichnet. Auf diese Weise werden alle direkt von der Sensorhardware abhängigen Funktionen in einem Baustein gekapselt.<sup>5</sup>

<sup>5</sup>Separation-Of-Concerns Prinzip, S. 14.

Von zwei Sensoren gleichen Typs<sup>6</sup> wird im Folgenden gesprochen, wenn beide die gleiche Funktion<sup>7</sup> zur Verfügung stellen können. Die betrachtete Funktion kann sich z.B. darauf beziehen, dass aus den Rohdaten bestimmte Merkmals- oder Klassifikationshypothesen extrahiert werden können, oder dass die Sensorhardware bestimmte Eigenschaften bzgl. der Konfiguration zulässt.<sup>8</sup> Die Typ-Einordnung ist dabei unabhängig von der zugrunde liegenden Sensortechnologie bzw. der sensorspezifischen Umsetzung.

Auf Anwendungsebene wird die eigentliche Funktion des Assistenzsystems realisiert. Diese basiert dabei auf den Objekt-Schätzungen, welche die Fusionsebene zur Verfügung stellt. In der Anwendungsebene gekapselt sind z.B. die Regler<sup>9</sup> und die Situationsanalyse. In der vorliegenden Arbeit wird diese Ebene als „Black-Box“ betrachtet, eine tiefer gehende Strukturierung dieses Bausteins wird daher nicht durchgeführt.<sup>10</sup>

Durch die über die Fusionsebene realisierte Trennung von Anwendungsebene und Sensorebene bleibt die Entstehung der Objekt-Schätzung für die Anwendungsebene unsichtbar.<sup>11</sup> Die verschiedenen Instanzen der Anwendungsebene greifen damit auf einen einzelnen virtuellen Umgebungssensor zu, welcher durch die Fusionsebene repräsentiert wird. Das Design der Anwendungen kann hierdurch unabhängig von den verwendeten Sensoren gestaltet werden.

Der Informationsgehalt der Objekt-Schätzung nimmt in der Architektur von der Sensorebene bis zur Anwendungsebene hin zu. Sind Entscheidungen innerhalb des Prozesses der Umgebungserfassung zu treffen, werden diese so weit nach oben verlagert, bis eine weitere Verlagerung keinen Informationsgewinn mehr für die Entscheidung mit sich bringt.<sup>12</sup>

In jeder Ebene wird die Entscheidung vorbereitet, allerdings nur soweit dies möglich ist und rein unter Hinzunahme der von niedrigeren Ebenen in die jeweilige Ebene gelangten Information. Von höheren Ebenen in die jeweilige Ebene zurückgeführte Informationen werden ausschließlich zur Vorkonditionierung von Algorithmen oder zur Erzeugung von Alternativhypothesen für eine Entscheidung genutzt. Auf diese Weise werden zyklische Abhängigkeiten<sup>13</sup> reduziert und es wird ausgeschlossen, dass fehlerhafte Informationen (z.B. aufgrund einer einmaligen Fehlinterpretation) dauerhaft im System bestehen bleiben.<sup>14</sup>

Im Folgenden werden die einzelnen Ebenen der Architektur im Detail dargestellt. Wichtig ist hierbei insbesondere, welche Dienste die einzelnen Ebenen an der Schnittstelle erbringen,

---

<sup>6</sup>Allgemeiner kann auch von gleicher Klasse gesprochen werden.

<sup>7</sup>Allgemeiner kann von Methoden gesprochen werden.

<sup>8</sup>Z.B. ein schwenkbarer Sichtbereich. Sensoren, die diese Eigenschaft haben, können dem Typ der schwenkbaren Sensoren zugeordnet werden.

<sup>9</sup>Wie in Abschnitt 3.2 dargelegt wurde, kann eine Anwendung auch aus mehreren Reglern bestehen. Zur Verallgemeinerung kann jeder Regler auch als eine eigenständige Assistenzfunktion (Subfunktion) angesehen werden.

<sup>10</sup>Siehe zu diesem Thema z.B. Maurer, M.: *Automatisierung von Straßenfahrzeugen* (2000).

<sup>11</sup>Information Hiding Prinzip, S. 14.

<sup>12</sup>Separation-of-Concerns Prinzip, S. 14.

<sup>13</sup>Vermeidung von zyklischen Abhängigkeiten, S. 15.

<sup>14</sup>Self Fulfilling Prophecy, siehe auch S. 50. Der Prozess bleibt „Wahrnehmungsgetrieben“.

bzw. welche Funktionen innerhalb der Ebene realisiert sind. Insofern es Vorteile bringt, kann die Struktur innerhalb einer Ebene auch anders als hier vorgestellt gestaltet werden. Nach außen hin wird dies nicht sichtbar, da diese Information in der jeweiligen Ebene gekapselt ist.<sup>15</sup>

## 5.2 Sensorebene

Auf Sensorebene werden die folgenden Funktionen realisiert:

- Signalaufnahme und Signalverarbeitung
- Merkmalsextraktion
- Datenassoziation (unter Einbeziehung von Rohdaten)
- Klassifikation auf Basis der Rohdaten

Die Signalaufnahme und Signalverarbeitung sowie die Merkmalsextraktion können mit Hilfe von bereits im System bekannten Objekthypothesen, die von der Fusionsebene an die Sensorebene zurückgeführt werden, vorkonditioniert werden. Beispielsweise lassen sich für die Merkmalsextraktion Suchbereiche in den Rohdaten auf Basis der Objekthypothesen definieren.<sup>16</sup>

Bei Verwendung der zurückgeführten Objekthypothesen im Block Merkmalsextraktion<sup>17</sup> dürfen die erstellten Merkmalshypothesen hierbei nicht alleine auf den zurückgeführten Daten basieren. Die zurückgeführten Daten dürfen nur dazu genutzt werden, um im Rahmen eines Multihypothesenansatzes alternative Merkmalshypothesen zu generieren.<sup>18</sup>

Unter Nutzung der gesamten auf Sensorebene vorhandenen Information findet die Datenassoziation der extrahierten Merkmalshypothesen zu den zurückgeführten Objekthypothesen statt. Hierbei können, wie in Kapitel 4 beschrieben, die Rohdaten zur Verbesserung des Prozesses hinzugezogen werden. Sollte eine Assoziation unter Hinzunahme der Rohdaten keine signifikanten Vorteile bringen, kann der Schritt der Datenassoziation auch weggelassen und erst auf Fusionsebene durchgeführt werden.

Die Klassifikation findet ausschließlich auf Basis der Rohdaten des Sensors statt. Jegliche weitere Information, die zur Klassifikation genutzt werden könnte (wie z.B. die geschätzten Zustände aus der Track-Schätzung), wird erst in der Fusionsebene hinzugezogen.

Ergebnis der Verarbeitung in der Sensorebene ist die L1 Darstellung. Diese ist abhängig vom Sensortyp. Die Beschreibung erfolgt in einer nicht sensorspezifischen Sprache in physikalischen Größen und ist unabhängig von der zugrunde liegenden Sensortechnologie.

---

<sup>15</sup>Information Hiding Prinzip, S. 14.

<sup>16</sup>Siehe hierzu auch Abschnitt 6.4, S. 82.

<sup>17</sup>Siehe hierzu die Ausführungen in Abschnitt 4.3, S. 48.

<sup>18</sup>Siehe hierzu auch Abschnitt 6.3.3, S. 81.

Auf diese Weise können unterschiedliche Sensortechnologien genutzt werden, um die gewünschten Informationen zu erlangen, ohne dass die Schnittstelle geändert werden muss. Die Qualität der Darstellung, also z.B. das Messrauschen, ist von der Leistungsfähigkeit des jeweiligen Sensors abhängig. Die Erweiterung der Sprache ist von der Entwicklung der Sensortechnologie getrieben. Die Darstellung beinhaltet die folgenden Informationen:

- Merkmals-hypothesen inkl. der zugehörigen Messfehler und Information zur Assoziation
- Klassifikations-hypothesen auf Basis der Rohdaten

## 5.3 Fusionsebene

Auf Fusionsebene werden die folgenden Funktionen realisiert:

- Datenassoziation (auf Basis der Zustandsvariablen)
- Klassifikation auf Basis der gesamten vorhandenen Information
- Anwendungsbezogene Datenfilterung
- Objekthypothesenmanagement
- Anforderungsmanagement

Noch nicht auf Sensorebene zugeordnete Merkmals-hypothesen werden auf Fusionsebene assoziiert. Da hier die Rohdaten nicht mehr zur Verfügung stehen, können nur noch die Zustandsvariablen, bzw. deren Transformation in den Messraum, zur Datenassoziation genutzt werden.

Dieser weitere Block zur Datenassoziation innerhalb der Architektur hat verschiedene Vorteile. So kann der Block auf Sensorebene weggelassen werden, insofern die Hinzunahme von Rohdaten keine signifikanten Vorteile mit sich bringt. Da für eine Assoziation auf Basis der Zustandsvariablen allgemein gültige Algorithmen existieren,<sup>19</sup> wird auf diese Weise der Algorithmus zur Datenassoziation nicht mehrfach im System implementiert.

Zudem wird die Gestaltung der Laufzeitsicht vereinfacht. So wird bei einem Szenario mit unsynchronisierten Sensoren und einer Berechnung der Fusion jeweils beim Eintreffen neuer Daten auf Fusionsebene folgendes Problem abgefangen: Mehrere Sensoren detektieren ein bisher nicht bekanntes Objekt und reichen diese Information in Form nicht assoziierter Daten an die Fusionsebene weiter. Durch die zweite Assoziation auf Fusionsebene werden die Daten zumindest auf Basis der Zustandsvariablen zueinander assoziiert.

Die Klassifikation wird auf Basis aller vorhandener Informationen auf Fusionsebene durchgeführt, d.h. den Informationen aus der Sensorebene und den Informationen aus der Track-Schätzung. Es findet somit eine dezentrale Fusion auf Entscheidungsebene bzgl. der Klassifikation statt. Diese ist optimal, insofern orthogonale Sensoren verwendet werden.<sup>20</sup>

<sup>19</sup>Siehe z.B. Bar-Shalom, Y.; Li, X.-R.: *Multitarget-multisensor tracking* (1995), S. 95.

<sup>20</sup>Siehe Tabelle 2.1, S. 19.

Die Frage, ob auf Fusionsebene eine abschließende bzw. harte Entscheidung für die Zuordnung in eine einzige Klasse durchgeführt werden kann<sup>21</sup> oder nur eine weiche Entscheidung in Form einer Maßzahl, welche die Zugehörigkeit zu einer Klasse widerspiegelt, berechnet werden sollte<sup>22</sup>, wird im Rahmen dieser Arbeit nicht behandelt. Die Architektur lässt prinzipiell beide Möglichkeiten zu.<sup>23,24</sup>

Aus Sicht der Datenfilterung findet eine zentrale Fusion auf Merkmalsebene statt. Auch diese ist optimal bei orthogonalen Sensoren.<sup>25</sup> Die Gültigkeit sowohl der schwachen als auch der starken Annahme für funktionsbezogene Mehrfachfilter konnte gezeigt werden. Daher wird für verschiedene Anwendungsgruppen, deren Anforderungen an Dynamik und Genauigkeit der Datenfilterung ähnlich ausfallen, jeweils ein eigenes Filter vorgesehen.

Auch die Gültigkeit sowohl der schwachen als auch der starken Annahme für getrennte Filter zur Datenassoziation konnte gezeigt werden. Zudem konnte die Annahme für das Ausreichen eines Einzelfilters für die Assoziation nicht widerlegt werden. Von daher wird in der Architektur ein einzelnes Filter  $F_R$  vorgesehen, das auf die Aufgabe der Datenassoziation abgestimmt ist. Da es Ziel des Filters zur Datenassoziation ist, die Umgebung möglichst exakt abzubilden,<sup>26</sup> wird dieses Filter auch für die Rückführung der Daten zur Klassifikation und zur Vorkonditionierung der Algorithmen in der Sensorebene verwendet.

Die einzelnen spezialisierten Filter werden in einer Filterbank zusammengefasst, welche durch einen gepunkteten Kasten in Abbildung 5.1 dargestellt ist. Alle Filteralgorithmen erhalten als Eingangsgrößen die gleichen Merkmals hypothesen und Informationen über die Klassifikation der Objekte.

Das komplette Objekthypothesenmanagement wird auf Fusionsebene durchgeführt. Dies umfasst das Verwalten der Objekthypothesen, inklusive des Löschens nicht mehr relevanter Hypothesen und des Erstellens neuer Objekthypothesen auf Basis nicht assoziierter Merkmals hypothesen. Gegenüber einer Initialisierung neuer Tracks in der Sensorebene hat dies den Vorteil, dass verschiedene Verfahren zur Initialisierung eines neuen Tracks implementiert werden können, ohne dass die Sensorebene geändert werden muss. Auch können so

---

<sup>21</sup>Eine so genannte „Hard Decision“, siehe z.B. Hall, D. L.: *Handbook of multisensor data fusion* (2001), S. 206.

<sup>22</sup>Eine so genannte „Soft Decision“, siehe z.B. Hall, D. L.: *Handbook of multisensor data fusion* (2001), S. 206. Auf S. 208-209 dieser Quelle werden verschiedene Methoden zur Berechnung einer solchen Maßzahl aufgeführt.

<sup>23</sup>Anmerkung: Nach Ansicht des Autors ist nur eine weiche Entscheidung zulässig. Deutlich wird dies bei der Betrachtung verschiedener  $t_{TTC}$ -basierter Anwendungen. Soll nur eine Aktion ausgeführt werden, wenn das relevante Objekt der Klasse Fahrzeug zugeordnet wird, kann für eine Warnfunktion eine niedrigere Schwelle für die harte Entscheidung als für eine automatische Notbremsfunktion gewählt werden, da sich die Auswirkungen der Aktionen deutlich unterscheiden. Diese Information ist aber erst auf Anwendungsebene vorhanden.

<sup>24</sup>Eine weitere offene Frage ist, ob Szenenwissen mit in die Entscheidung einbezogen werden sollte. Da dieses Wissen erst im Rahmen der Situationsanalyse auf Anwendungsebene generiert wird, sollte es nach Ansicht des Autors auch erst hier in den Prozess der Klassifikation einfließen.

<sup>25</sup>Siehe hierzu Tabelle 2.1, S. 19.

<sup>26</sup>Siehe Abschnitt 4.5, S. 59.

für verschiedene Filteralgorithmen bzw. Anwendungsgruppen unterschiedliche Verfahren zur Track-Initialisierung verwendet werden.<sup>27</sup>

Von der Anwendungsebene können Anforderungen an die Fusionsebene gestellt werden. Dies kann z.B. den Sichtbereich des virtuellen Sensors betreffen, oder eine bestimmte Priorisierung der Objekte bzgl. der Klassifikation. Entscheidungen dieser Art können nicht auf Fusionsebene getroffen werden, da sie originär mit der Anwendung zusammenhängen. So kann beispielsweise der Sichtbereich des Sensors für die Anwendung ACC in Kurven in Richtung des relevanten Objekts ausgerichtet werden. Die Definition des relevanten Objekts findet allerdings erst auf Anwendungsebene im Rahmen der Situationsanalyse statt.

Verschiedene Anwendungen können unterschiedliche und sich widersprechende Anforderungen an die Fusionsebene stellen. Diese Konflikte werden im Block Anforderungskoordination gelöst.<sup>28</sup> Die in einer sensortypunabhängigen Beschreibungssprache gehaltenen Anforderungen werden innerhalb der Fusionsebene in eine sensortypabhängige Anforderung an die Sensorebene gewandelt und hier schließlich sensorspezifisch umgesetzt. Zudem können die Anforderungen Auswirkungen auf das Objekthypothesenmanagement (z.B. Bewertung der Relevanz von Objekthypothesen zur Entscheidung welche Objekthypothesen gelöscht werden können) und die Datenfilterung (z.B. Modifikation der Filterparameter) haben.

Ergebnis der Verarbeitung in der Fusionsebene ist die abgeschlossene Objekt-Schätzung in der L2 Darstellung. Im Gegensatz zur L1 Darstellung ist die Definition dieser Sprache von den Anforderungen der Anwendungen abhängig. Die Erweiterung der Darstellung wird von der Entwicklung neuer Anwendungsgruppen und deren Anforderungen begründet. Bestehende Elemente der Sprache werden hierbei nicht abgeändert, die Darstellung wird um die benötigten Attribute erweitert. Die Darstellung beinhaltet die folgenden Informationen:

- Schätzung der Zustände beobachteter Objekthypothesen mit Angaben zu Schätzfehlern bezogen auf das der Schätzung zugrunde liegende Modell
- Klassifikationshypothesen auf Basis aller in der Fusionsebene vorhandenen Informationen

## 5.4 Robustheit bezüglich der Grundannahmen

Sollte die starke Annahme für funktionsbezogene Mehrfachfilter nicht zutreffen, ist das Vorhalten verschiedener, anwendungsbezogener Filter in Frage gestellt. Solange die schwache Annahme für funktionsbezogene Mehrfachfilter noch gültig ist, garantiert die Filterbank dennoch einen optimalen Betrieb der Anwendungen. Sollte auch die schwache Annahme nicht

<sup>27</sup>Zu verschiedenen Methoden der Track-Initialisierung und deren Vor- und Nachteilen siehe z.B. Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 245-248.

<sup>28</sup>Hierzu könnte beispielsweise jeder Anwendung eine Priorität zugeordnet werden. Die Anforderung der Anwendung mit höchster Priorität wird schließlich umgesetzt.

zutreffen, ist die Filterbank nicht notwendig. Die Architektur würde hierdurch nur in geringem Maße beeinflusst, da eine Datenfilterung auf Fusionsebene bei orthogonalen Sensoren optimal ist und die Filterbank nur auf einen Einzelfilter reduziert werden würde.

Würde die starke Annahme für getrennte Filter zur Datenassoziation nicht zutreffen, würde das Abspalten des Assoziationsfilters  $F_R$  in Frage gestellt. Gilt zumindest die schwache Annahme, garantiert das Abspalten des Filters dennoch eine optimale Datenassoziation. Sollte auch die schwache Annahme für getrennte Filter zur Datenassoziation nicht zutreffen, wird kein abgekoppeltes Filter zur Datenassoziation mehr benötigt. Auch dies hat keine signifikanten Auswirkungen auf die Struktur der Architektur.

Sollte hingegen die Annahme für das Ausreichen eines Einzelfilters für die Assoziation widerlegt werden, müssten für verschiedene Anwendungen separate Assoziationsfilter vorgesehen werden. Da die Informationen der Fusionsebene über das Assoziationsfilter in die Sensorebene zurück gekoppelt werden, kann dies eine umfangreiche Überarbeitung der Struktur auf Bausteinsicht nach sich ziehen. Abhängig ist dies von der Realisierung der Laufzeitsicht.

Eine Lösung ließe sich z.B. durch einen Multihypothesenansatz bzgl. der Zustandsgrößen umsetzen, welcher im Rahmen der Track-Initialisierung ggf. ohnehin benötigt wird.<sup>29</sup> Dieser lässt sich unter Beibehaltung der Struktur auf Bausteinsicht durch eine entsprechende Gestaltung der Laufzeitsicht (Vorschrift zur Abarbeitung der verschiedenen Hypothesen) realisieren.

Würde schließlich die Annahme zur Abhängigkeit der Datenassoziation von den Rohdaten widerlegt werden, könnte die Datenassoziation komplett auf die Fusionsebene verlegt werden. Da ein entsprechender Block bereits auf der Fusionsebene vorgesehen ist, würde auch dies nur geringe Änderungen in der Architektur nach sich ziehen.

## 5.5 Zusammenfassung der Eigenschaften

Durch die Filterbank auf Fusionsebene werden die Abhängigkeiten von Anwendungsebene und Sensorebene minimiert.<sup>30</sup> Das Verhalten der anwendungsbezogenen Filter kann optimiert werden, ohne dass es eine Rückwirkung auf die Sensorebene oder den Prozess der Klassifikation gibt. Umgekehrt kann die Datenassoziation optimiert werden, ohne dass ein Einfluss auf eine bereits validierte Regler-Beobachter Kombination stattfindet.

Auch die Abhängigkeiten in der Anwendungsebene, welche bei Verwenden eines gemeinsamen Filters bestehen würden, werden aufgehoben. Eine neue Anwendung kann hinzugefügt werden, ohne dass die Funktion der anderen Anwendungen erneut validiert werden muss.<sup>31</sup>

<sup>29</sup>Zu Methoden der Track-Initialisierung mit einem Multihypothesenansatz siehe z.B. Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 245 - 248.

<sup>30</sup>Vermeidung von zyklischen Abhängigkeiten, siehe S. 15.

<sup>31</sup>Prinzip des Entwurfs für Veränderungen, siehe S. 14.

Durch die Kapselung der auf die Hardware des Sensors optimierten Algorithmen in der Sensorebene und der Definition einer allgemeinen sensortypabhängigen Beschreibungssprache (L1 Darstellung) lassen sich Sensoren des gleichen Typs tauschen. Hierbei können die internen Bausteine (z.B. Datenassoziation, Klassifizierung) unterschiedlich realisiert werden, ohne dass eine Rückwirkung auf die bestehende Struktur der Architektur stattfindet.<sup>32</sup> Insbesondere lassen sich die Bausteine sukzessive verbessern bzw. erweitern, ohne bestehende Strukturen ändern zu müssen.<sup>33,34</sup> Über entsprechende Tests lassen sich unterschiedliche Realisierungen der Sensorebene zudem objektiv bewerten (z.B. Anzahl der Fehlinterpretationen, Kosten des Moduls).

Werden hierbei Sensoren mit der gleichen oder einer besseren Spezifikation hinsichtlich der Messgenauigkeit eingesetzt, ist auch kein Eingriff in die Filteralgorithmen auf Fusionsebene notwendig.<sup>35</sup> Die dynamischen Eigenschaften der Regler-Beobachter Kombinationen werden sich nicht ändern, nur die statistischen Größen, wie z.B. das Beschleunigungsrauschen bei ACC (siehe Gleichung (3.5)), werden beeinflusst.

Das Hinzufügen eines neuen Sensortyps zu einer bestehenden Implementation erfordert einen Eingriff in die Algorithmen der Fusionsebene, da die neue L1 Darstellung realisiert werden muss. Die Struktur der Architektur ändert sich hierdurch allerdings nicht.

Hinsichtlich der hardwareseitigen Realisierung der Architektur sind verschiedene Verteilungssichten möglich. So können die verschiedenen Ebenen z.B. alle gemeinsam auf einem einzelnen Prozessor laufen, wobei diese dann unterschiedlichen Softwaremodulen entsprechen.<sup>36</sup> Genau so ist aber auch ein Betrieb auf verschiedenen Prozessoren möglich, wobei jede Instanz einer Schicht auf einem eigenen Steuergerät laufen kann. Die L1 und L2 Darstellungen können dabei auf einem Bus-System übermittelt werden, wobei für die L1 Darstellung eine deutlich niedrigere Bus-Kapazität vorgesehen werden muss, als bei einer Übertragung der Rohdaten. Auf diese Weise werden Algorithmen nicht nur in Software- sondern auch in Hardware-Modulen gekapselt, was die herstellerübergreifende Zusammenarbeit vereinfachen kann.

Obwohl die in Abschnitt 2.3 diskutierten Hypothesen die Grundlage der hier vorgestellten Architektur sind, ist die Architektur dennoch robust gegenüber einem Irrtum in den Annahmen. So fallen ggf. zentrale Elemente der Architektur, wie z.B. die Filterbank, weg. Dennoch können bestehende Algorithmen weiter verwendet werden. Einzige Ausnahme bildet eine Widerlegung der Annahme für das Ausreichen eines Einzelfilters für die Assoziation. Das Risiko kann hier allerdings durch eine entsprechende Gestaltung der Laufzeitsicht minimiert werden.

---

<sup>32</sup>Offen-Geschlossen-Prinzip, S. 14.

<sup>33</sup>Prinzip des Entwurfs für Veränderungen, siehe S. 14.

<sup>34</sup>Siehe auch S. 48, S. 49.

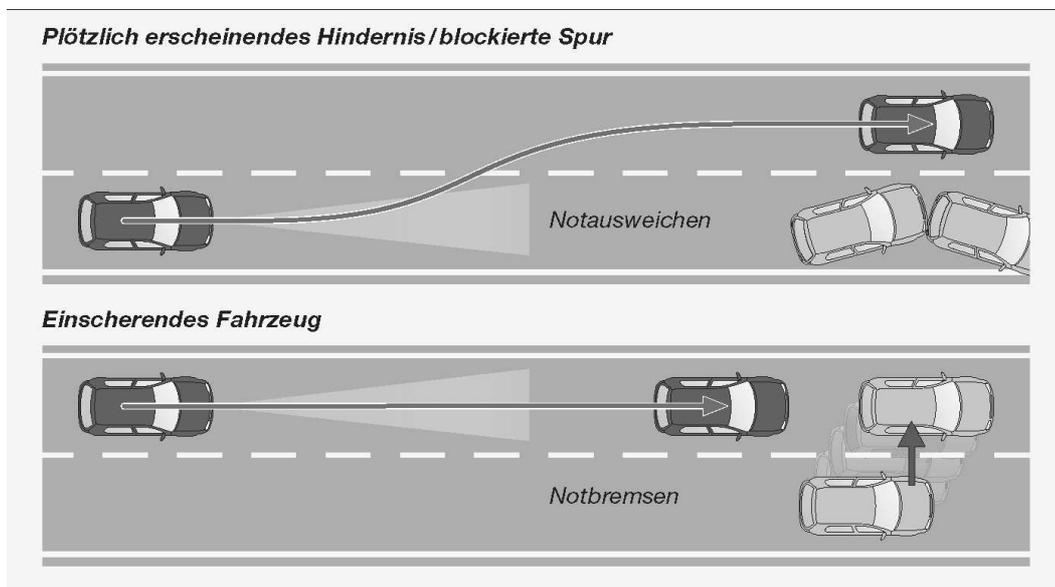
<sup>35</sup>So müssten z.B. die Parameter eines hierfür verwendeten Kalmanfilters nicht erneut angepasst werden. Beim Kalmanfilter setzt dies aber voraus, dass das Messrauschen  $\sigma_w$  als Modellierungsparameter des Filters gesehen wird und nicht dem Sensorrauschen  $\sigma_n$  gleichgesetzt wird. Der Assoziationsfilter könnte sich ggf. automatisch justieren, siehe die Ausführungen auf S. 60.

<sup>36</sup>Siehe hierzu auch Abschnitt 6.3.3, S. 81.

## 6 Praktische Umsetzung der Basis-Systemarchitektur

### 6.1 Versuchsträger und Umfoldsensoren

Die in Kapitel 5 vorgestellte Systemarchitektur wurde innerhalb des Projekts PRORETA auf einem Versuchsträger implementiert. Ziel des Projekts PRORETA war die Entwicklung von Algorithmen zur Unfallvermeidung auf Basis von Informationen aus dem Fahrzeugumfeld. Der Fokus für das Einsatzgebiet der Algorithmen lag auf Situationen im Längsverkehr auf außerstädtischen Straßen. Zwei typische Anwendungsszenarien zeigt Abbildung 6.1.<sup>1</sup>



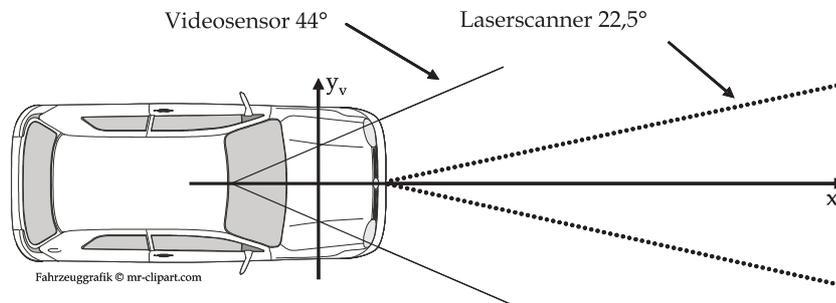
**Abbildung 6.1:** Anwendungsszenarien aus dem Projekt PRORETA.

(Entnommen aus dem Flyer zur Abschlusspräsentation des Projekts)

Im Szenario *Notausweichen* wird per Umfoldsensorik ein stehendes Hindernis auf der Fahrbahn erkannt. Reagiert der Fahrer nicht auf das Hindernis, wird im fahrdynamisch spätest möglichen Zeitpunkt ein Notausweichmanöver um das Hindernis herum eingeleitet. Im Szenario *Notbremsen* wird ein einscherendes Fahrzeug erkannt. Bei ausbleibender Fahrerreaktion,

<sup>1</sup>Für Details zum Projekt PRORETA siehe Bender, E.; Landau, K.: *PRORETA: Fahrerverhalten* (2006); Darms, M.; Winner, H.: *PRORETA: Umfelderfassung* (2006); Stählin, U.; Schorn, M.; Isermann, R.: *PRORETA: Notausweichen* (2006).

wird im fahrdynamisch spätest möglichen Zeitpunkt ein Notbremsmanöver ausgeführt, so dass das Fahrzeug noch vor dem Hindernis zum Stehen kommt.<sup>2</sup>

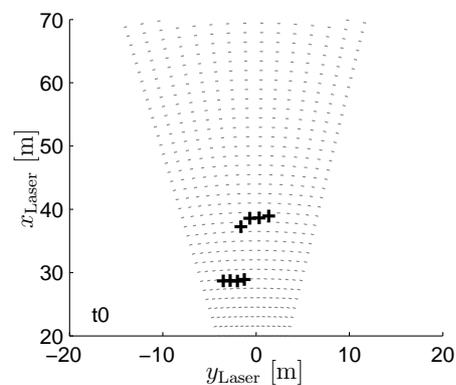


**Abbildung 6.2:** Konfiguration der Umfelsesensoren des Versuchsträgers aus dem Projekt PRORETA.

Je genauer die Position des Hindernisses in beiden Szenarien bekannt ist, desto präziser kann der Startzeitpunkt der Manöver bzw. die zu fahrende Trajektorie berechnet werden.<sup>3</sup> Abbildung 6.2 zeigt die Konfiguration der zur Umfelderkennung eingesetzten Sensoren, die unsynchronisiert betrieben werden.<sup>4</sup> Ein Beispiel für die Rohdaten der verwendeten Sensoren zeigt Abbildung 6.3.<sup>5</sup> Als Koordinatensystem der Fusionsebene wird das Fahrzeugkoordinatensystem nach DIN 70000 verwendet. Die y-Achse liegt dabei auf der Vorderachse des Versuchsträgers.



(a) Videosensor



(b) Laserscanner

**Abbildung 6.3:** Rohdaten der Sensoren.

Der Laserscanner ist ein Prototyp von ContinentalTemic, der in der für das Projekt verwendeten Konfiguration im 90 ms Zyklus einen Bereich von  $22,5^\circ$  erfasst, welcher in 15 Sektoren unterteilt ist. Dies führt zu einer Diskretisierung in  $1,5^\circ$  Schritten in der xy-Ebene des Erfassungsbereiches. Die gepunkteten Linien in Abbildung 6.3(b) markieren die Mitte

<sup>2</sup>Details zur Eingriffsentscheidung und zur regelungstechnischen Umsetzung der Manöver finden sich in Stählin, U.; Schorn, M.; Isermann, R.: *PRORETA: Notausweichen* (2006).

<sup>3</sup>Siehe auch Stählin, U.; Schorn, M.; Isermann, R.: *PRORETA: Notausweichen* (2006).

<sup>4</sup>Abbildung D.1 auf S. 97 zeigt ein Foto des Versuchsträgers.

<sup>5</sup>Siehe auch Abbildung 4.2 auf S. 49.

der Sektoren. Der Ursprung der Sektoren liegt im Punkt  $(0, 0)$  des Koordinatensystems des Laserscanners. Die Entfernung zu Objekten wird durch eine Laufzeitmessung von Lichtimpulsen bestimmt. In jedem der Sektoren können dabei bis zu drei Messungen liegen. Die Liste aller Messungen wird im Folgenden als Zielliste bezeichnet, eine einzelne Messung als Ziel, sie werden in Abbildung 6.3(b) durch Kreuze dargestellt. Der Erfassungsbereich des Sensors lässt sich per Software um  $\pm 4^\circ$  schwenken. Typischer Einsatzbereich dieses Sensors ist die Anwendung Adaptive Cruise Control (ACC).

Auch der Video-Sensor ist ein Prototyp von ContinentalTemic. Er arbeitet mit einem monochromen CMOS Bildsensor. Dieser liefert alle 40 ms ein komplettes Bild und hat eine Auflösung von  $640 \times 496$  Bildpunkten, eine Farbtiefe von 8 Bit und eine Pixelgröße von  $7,5 \mu\text{m} \times 7,5 \mu\text{m}$ . Die Brennweite des Objektivs beträgt 6 mm, so dass sich ein Erfassungsbereich von ca.  $44^\circ$  ergibt. Die Auflösung in der xy-Ebene ist mit ca.  $0,07^\circ$  deutlich höher als beim eingesetzten Laserscanner, siehe Abbildung 6.3(a). Eine direkte Entfernungsmessung ist - anders als beim Laserscanner - nicht möglich. Typischer Einsatzbereich des Sensors ist die Anwendung Fahrstreifenverlassenswarnung.<sup>6</sup>

Im Folgenden werden die entsprechenden Algorithmen der Fusions- und Sensorebene sowie die jeweilige L1- und die L2-Schnittstelle vorgestellt. In Abschnitt 6.7 werden schließlich die erzielten Vorteile der Fusion für das Projekt PRORETA aufgezeigt.

## 6.2 Umsetzung der Fusionsebene

### 6.2.1 L2-Darstellung

Abbildung 6.4(a) zeigt die allgemeine Definition der L2-Darstellung einer wahrgenommenen Szene. Da bei der Messung und Wahrnehmung auf Sensorebene Annahmen gemacht werden und somit Fehlinterpretationen nicht ausgeschlossen werden können,<sup>7</sup> wird eine wahrgenommene Szene durch verschiedene Szenenhypothesen repräsentiert.

Jede Szenenhypothese ergibt sich aus einer Menge von Objekthypothesen und stellt eine mögliche Interpretation der Sensorsignale dar. Eine Objekthypothese kann dabei mehreren verschiedenen Szenenhypothesen zugeordnet sein, so dass sich eine  $n : m$  Beziehung zwischen Objekthypothesen und Szenenhypothesen ergibt. Sich gegenseitig ausschließende Objekthypothesen werden dabei unterschiedlichen Szenenhypothesen zugeordnet.

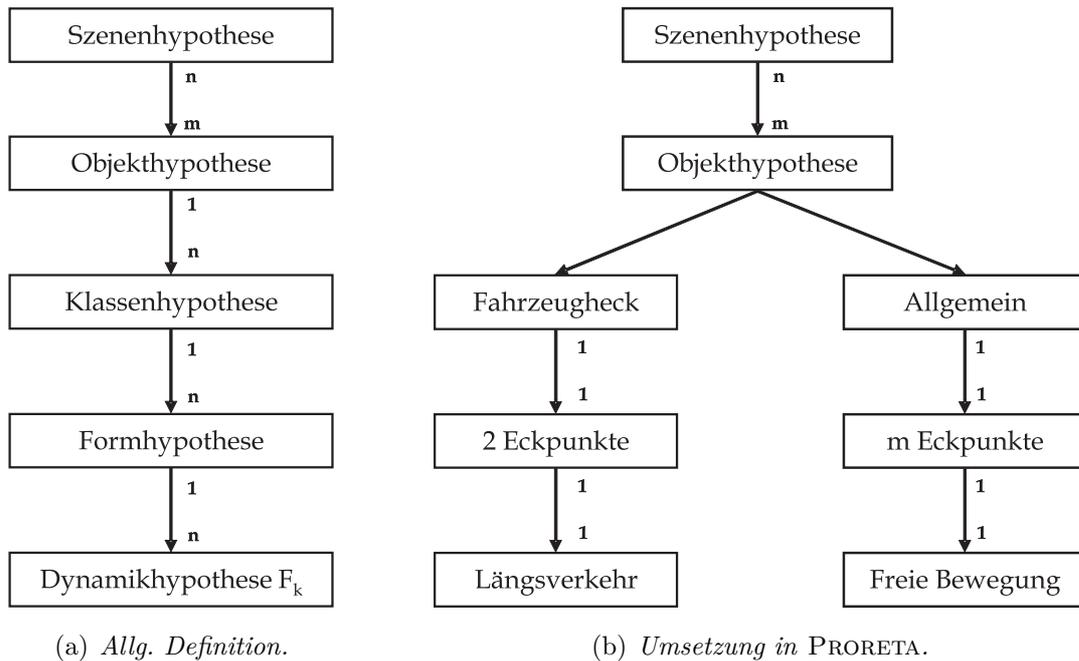
Eine Objekthypothese beschreibt ein wahrgenommenes Objekt in der Szene.<sup>8</sup> Da die Annahme der Existenz des Objekts auf einem Fehler in der Wahrnehmung bzw. der Messung beruhen kann, stellt die Objekthypothese eine Hypothese bzgl. der Existenz des Objekts dar.

---

<sup>6</sup>Lane Departure Warning.

<sup>7</sup>Siehe die Ausführungen in Abschnitt 2.1 auf S. 9.

<sup>8</sup>In einer weiteren Verallgemeinerung kann auch das Ego-Fahrzeug selbst als ein solches Objekt betrachtet werden. Siehe auch den Kommentar auf S. 44.



**Abbildung 6.4:** *L2-Darstellung einer wahrgenommenen Szene.*

Ein typisches Beispiel für eine fehlerhafte Objekthypothese ist eine Fehldetektion verursacht durch ein Artefakt in der Messung oder dem Rauschen bei der Signalaufnahme.

Unter der Annahme, dass die Objekthypothese zutrifft, können für dieses Objekt verschiedene Hypothesen bzgl. der Klassenzugehörigkeit aufgestellt werden. Es besteht damit eine  $1 : n$  Beziehung zwischen Objekthypothese und den Klassenhypothesen.

Auf Basis der Annahme der Klassenzugehörigkeit können Hypothesen über die Form des Objekts aufgestellt werden. Auch hier ist wiederum ein Multihypothesenansatz mit einer  $1 : n$  Beziehung möglich, so dass für jede Klasse unterschiedliche Formen zutreffen können.

Ist schließlich die Form festgelegt, kann eine Annahme über die Dynamik des beobachteten Objekts getroffen werden. Auch hier sind, insbesondere zur Initialisierung des Tracking,<sup>9</sup> mehrere Hypothesen über die Dynamik des Objekts möglich, so dass sich erneut eine  $1 : n$  Beziehung ergibt. Für jeden Filter  $F_i$  auf Fusionsebene wird dabei ein eigener Satz an Dynamikhypothesen verwaltet.

Abbildung 6.4(b) zeigt die Umsetzung im Projekt PROETA, welche auf das Einsatzgebiet Längsverkehr ausgerichtet ist. Nur für Objekt- und Klassenhypothesen wird ein Multihypothesenansatz gewählt, ansonsten gelten jeweils  $1 : 1$  Beziehungen. Als Klassenhypothesen gibt es hierbei die zwei Klassen *Allgemein* und *Fahrzeugheck*.

Die Form eines Objekts wird immer durch einen Polygonzug mit  $n$  Eckpunkten  $\xi_i$  im Fahrzeugkoordinatensystem beschrieben. Für eine Objekthypothese OH gilt damit:

$$\text{OH} \mapsto (\xi_1, \dots, \xi_n) \text{ mit } \xi_i = (x_i, y_i), i = 1, \dots, n. \quad (6.1)$$

<sup>9</sup>Siehe Abschnitt 5.3, S. 68.

Für Objekte der Klasse *Fahrzeugheck* ist die Anzahl der Eckpunkte auf  $n = 2$  festgelegt, für die Klasse *Allgemein* ist die Anzahl der Eckpunkte abhängig von der Wahrnehmung und kann sich dynamisch ändern.<sup>10</sup>

## 6.2.2 Datenfilterung

Zur Schätzung der Zustandsvariablen wird ein erweitertes Kalmanfilter erster Ordnung mit einem linearen Systemmodell für die beobachteten Objekte und nichtlinearen Gleichungen für die Abbildung der Zustandsvariablen in den Messraum eingesetzt. Für die Änderung des Zustandsvektors  $x$  einer Objekthypothese

$$x^T = (x_1, \dots, x_n, v_{\text{rel},x}, a_{\text{rel},x}, y_1, \dots, y_n, v_{\text{rel},y}, a_{\text{rel},y}) \quad (6.2)$$

vom Zeitschritt  $t_k$  zu  $t_{k+1} = t_k + \Delta T_k$  gilt dabei:

$$x(k+1) = F(k)x(k) + n_d(k) + n_v(k). \quad (6.3)$$

$v_{\text{rel},x}$  und  $v_{\text{rel},y}$  geben dabei die Relativgeschwindigkeit,  $a_{\text{rel},x}$  und  $a_{\text{rel},y}$  die Relativbeschleunigung der Eckpunkte  $(x_i, y_i)$  der Objekthypothese im Fahrzeugkoordinatensystem an, wobei diese für alle Eckpunkte gleich ist. Für die Zustandsübergangsmatrix  $F(k)$  gilt:

$$F(k) = \begin{pmatrix} F_x(k) & 0 \\ 0 & F_y(k) \end{pmatrix} \quad \text{mit} \quad F_{x,y}(k) = \begin{pmatrix} 1 & \Delta T_k & \frac{1}{2}\Delta T_k^2 \\ \ddots & \vdots & \vdots \\ & 1 & \Delta T_k & \frac{1}{2}\Delta T_k^2 \\ & & 1 & \Delta T_k \\ & & & 1 \end{pmatrix}. \quad (6.4)$$

$n_d(k)$  ist ein mittelwertfreier Zufallsprozess zur Modellierung der Dynamik und  $n_v(k)$  ein mittelwertfreier Zufallsprozess, der ein künstliches Rauschen zur Schätzung der Form in das System einbringt. Die Prozesse sind statistisch unabhängig. Für das Systemrauschen  $Q(k) = E\{n_d(k)n_d(k)^T\} + E\{n_v(k)n_v(k)^T\}$  gilt:<sup>11</sup>

$$Q(k) = \begin{pmatrix} Q_x(k) & 0 \\ 0 & Q_y(k) \end{pmatrix} \quad (6.5)$$

<sup>10</sup>Hier wäre z.B. ein Multihypothesenansatz bzgl. der Form möglich.

<sup>11</sup>Siehe auch Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 271 - 272.

mit

$$Q_{x,y}(k) = \begin{pmatrix} \frac{1}{20}\Delta T_k^5 & \dots & \frac{1}{20}\Delta T_k^5 & \frac{1}{8}\Delta T_k^4 & \frac{1}{6}\Delta T_k^3 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \frac{1}{20}\Delta T_k^5 & \dots & \frac{1}{20}\Delta T_k^5 & \frac{1}{8}\Delta T_k^4 & \frac{1}{6}\Delta T_k^3 \\ \frac{1}{8}\Delta T_k^4 & \dots & \frac{1}{8}\Delta T_k^4 & \frac{1}{3}\Delta T_k^3 & \frac{1}{2}\Delta T_k^2 \\ \frac{1}{6}\Delta T_k^3 & \dots & \frac{1}{6}\Delta T_k^3 & \frac{1}{2}\Delta T_k^2 & \Delta T_k \end{pmatrix} q_{x,y} + \begin{pmatrix} \Delta T_k & & & & \\ & \ddots & & & \\ & & \Delta T_k & & \\ & & & 0 & \\ & & & & 0 \end{pmatrix} m_{x,y}. \quad (6.6)$$

Die Bewegung in x- und y-Richtung wird somit unabhängig voneinander modelliert. Eine Gierbewegung wird nicht modelliert, sondern nur durch die Formschätzung berücksichtigt. Die Leistungsdichte  $q_{x,y}$  beeinflusst die Dynamik des Modells,  $m_{x,y}$  die Formschätzung.

Für ein Objekt der Klasse *Fahrzeugheck* ist  $q_y$  auf die optimale Erkennung des Einschermanövers abgestimmt,  $q_x$  auf die optimale Bestimmung des Zeitpunkts zum Auslösen des Ausweich- bzw. Notbremsmanövers. Für ein Objekt der Klasse *Allgemein* gilt  $q_x = q_y$ , da keine weitere Information über die Dynamik des Objekts bekannt ist.

Die Messgleichungen und das modellierte Sensorrauschen sind Teil des jeweiligen Filters  $F_k$  und damit Teil der Fusionsebene. Das modellierte Messrauschen muss dabei nicht dem realen Messrauschen entsprechen. Wie in Abschnitt 3 auf S. 24 beschrieben, bestimmen das modellierte Systemrauschen und das modellierte Messrauschen<sup>12</sup> die Dynamik und die Schätzgenauigkeit des Filters. Die Vorgaben für diese Größen hängen von der jeweiligen Anwendung ab.

Durch die Verankerung der Größen in den anwendungsbezogenen Filtern  $F_k$  wird die Erweiterbarkeit des Gesamtsystems erleichtert. Wird ein Sensor durch einen neuen Sensor mit geringerem Messrauschen ersetzt (z.B. aufgrund gestiegener Anforderungen einer neuen, dem Gesamtsystem hinzugefügten Anwendung), wird so sichergestellt, dass sich die Dynamik der vorhandenen Anwendungen nicht ändert. Durch die Verringerung des realen Sensorrauschens wird lediglich der zufällige Fehler verringert, das bereits validierte dynamische Verhalten der vorhandenen Anwendungen wird nicht beeinflusst.<sup>13</sup>

Ein weiterer Vorteil ist die Möglichkeit verschiedene, voneinander unabhängige Filteralgorithmen für verschiedene Anwendungen einzusetzen. So benötigt z.B. das erweiterte Kalmanfilter erster Ordnung lediglich eine Näherung erster Ordnung der Modellgleichungen, das erweiterte Kalmanfilter zweiter Ordnung eine Näherung zweiter Ordnung.<sup>14</sup> Durch die Berechnung der benötigten Größen im jeweiligen Filter  $F_k$  können so für verschiedene Anwendungen unterschiedliche Algorithmen eingesetzt werden – unabhängig von der Sensorebene.

<sup>12</sup>In Abschnitt 3 wurde hieraus der Manöverindex  $\lambda$  abgeleitet.

<sup>13</sup>In Abschnitt 3.2 verringert sich z.B. die maximale Amplitude der Beschleunigung  $a_{\text{ego}}$  im stationären Folgebetrieb, die Dynamik der Anwendung bleibt erhalten.

<sup>14</sup>Siehe z.B. Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 382.

### 6.2.3 Klassifikation

Zur Klassifikation von Objekten wurde im Projekt PRORETA eine Heuristik implementiert, welche die Klassifikationsentscheidungen der Sensorebene auswertet. Wie weiter unten beschrieben, trägt hierbei nur der Videosensor Informationen zur Klassifikation bei, Informationen aus den Zustandsvariablen werden nicht mit in die Klassifikation einbezogen.

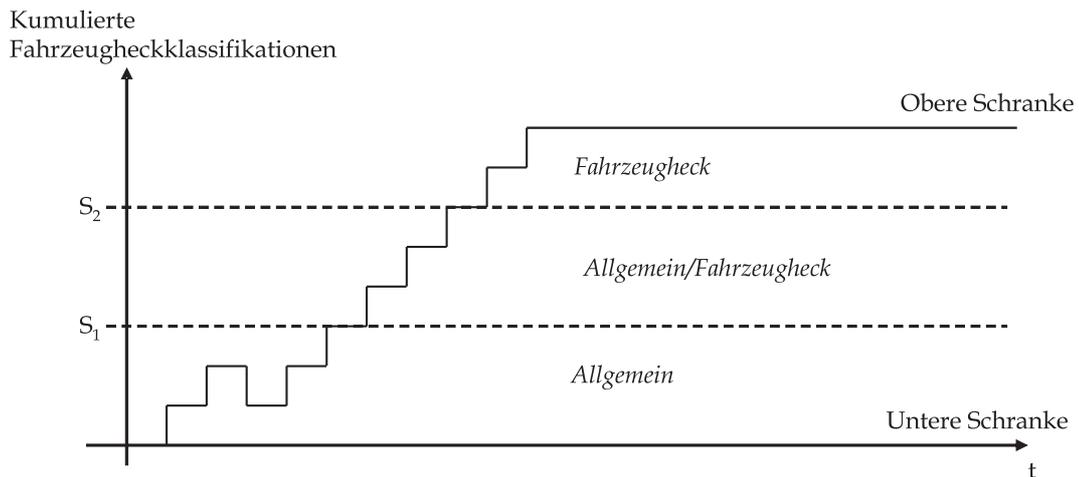


Abbildung 6.5: Heuristik zur Klassifikation.

Abbildung 6.5 stellt die Heuristik graphisch dar. Bzgl. der Hypothese *Fahrzeugheck* trägt die Sensorebene die Klassifikationsinformation *Als Fahrzeugheck klassifiziert*, *Nicht als Fahrzeugheck klassifiziert* bzw. *Keine Klassifikationsinformation* bei.

Die Informationen werden kumuliert, wobei die Information *Als Fahrzeugheck klassifiziert* positiv, die Information *Nicht als Fahrzeugheck klassifiziert* negativ und die Information *Keine Klassifikationsinformation* neutral summiert wird. Eine obere bzw. untere Schranke wird dabei nicht über- bzw. unterschritten. Ist der Wert kleiner als die Schwelle  $S_1$  wird die Hypothese *Allgemein* instantiiert, ist der Wert größer als  $S_2$  die Hypothese *Fahrzeugheck*. Im Zwischenbereich wird die Hypothese *Allgemein* gewählt, wenn sie Schwelle  $S_1$  überschritten wird bzw. die Hypothese *Fahrzeugheck*, wenn die Schwelle  $S_2$  unterschritten wird. Auf diese Weise wird eine Hysterese realisiert.

Als empirisches Gütemaß für das Zutreffen der Hypothese *Fahrzeugheck* werden die letzten  $n$  Klassifikationen betrachtet und der Quotient

$$\mathcal{P}(\text{Fahrzeugheck}) = \frac{\sum_n \text{Als Fahrzeugheck klassifiziert}}{\sum_n \text{Als Fahrzeugheck klassifiziert} + \text{Nicht als Fahrzeugheck klassifiziert}} \quad (6.7)$$

ausgewertet. Bis zum Erreichen der  $n$  Klassifikationen wird keine Aussage über die Güte gemacht.

Die Schwellen und Schranken sind fest konfiguriert und wurden im Rahmen von Fahrversuchen auf die Anforderungen des Projekts abgestimmt. Sie spiegeln daher zum einen die

Leistungsfähigkeit des verwendeten Sensors, zum anderen die Anforderungen der Anwendung wider.<sup>15</sup> Der Algorithmus ist damit nicht sensorneutral, in einem allgemeineren Ansatz müssten die Qualitätsangaben der Sensorebene berücksichtigt werden. Das Löschen der Hypothese *Allgemein* ist nicht notwendig, reduziert allerdings den Rechenaufwand, wenn die Hypothese *Fahrzeugheck* eine hohe Güte hat.

Das vorgestellte Vorgehen erlaubt es, parallel weitere Klassifikationshypothesen hinzuzufügen, ohne die übrigen Klassifikationen zu beeinflussen. Schließlich kann auf Anwendungsebene entschieden werden, welche Hypothese als wahr angenommen wird.

### 6.2.4 Objekthypothesenmanagement

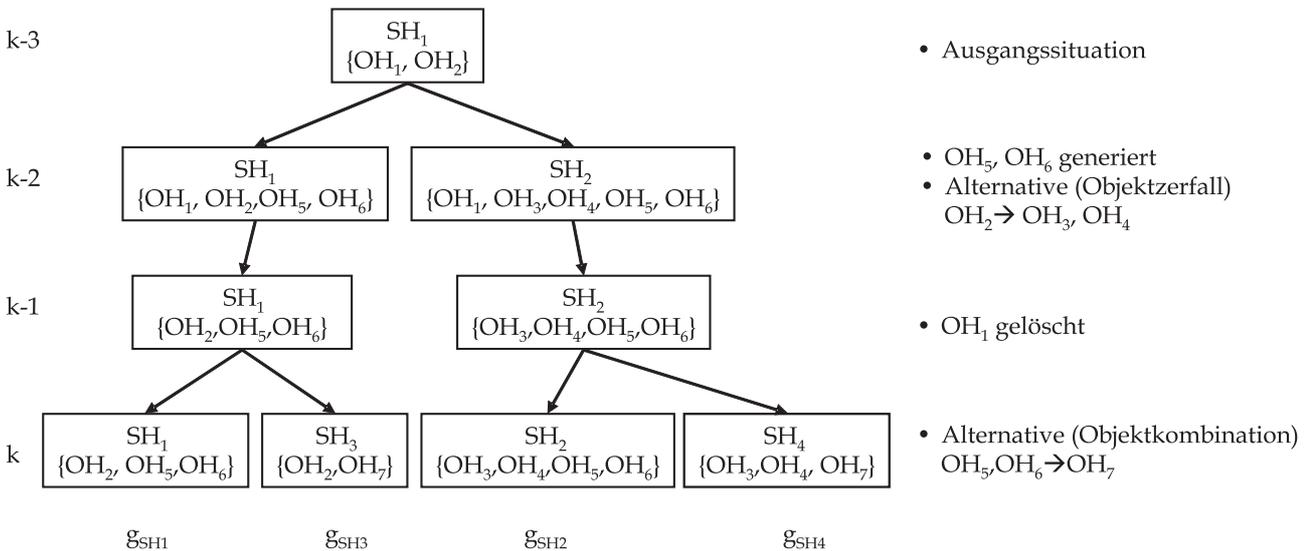


Abbildung 6.6: Hypothesenbaum zur Verwaltung der Szenenhypothesen.

Jeder Objekthypothese  $OH_i$  mit  $i = 1, \dots, n_{OH}$  wird eine Güte  $g_i$  mit  $0 \leq g_i \leq 1$  zugewiesen.  $n_{OH}$  ist dabei die Gesamtzahl der Objekthypothesen zu einem Zeitschritt  $k$ . Die Güte berechnet sich aus einem sensorspezifischen Anteil  $g_{SL}$  und einem nicht-sensorspezifischen Anteil  $g_{FL}$ . Im sensorspezifischen Anteil  $g_{SL}$  spiegelt sich wider, mit welcher Qualität die Objekthypothese auf Sensorebene gefunden wurde. Die Güte  $g_{SL}$  ist dabei höher, wenn die Objekthypothese direkt im Schritt der Merkmalsextraktion, d.h. ohne Informationen der Fusionsebene, gefunden werden konnte. Sie ist niedriger, wenn es sich um eine Alternativhypothese handelt, die unter Zuhilfenahme der Informationen der Fusionsebene generiert wurde.<sup>16</sup> Im nicht-sensorspezifischen Anteil  $g_{FL}$  steckt die Information über die Lebensdauer der Hypothese.<sup>17</sup>

<sup>15</sup>Siehe hierzu auch die Anmerkung auf S. 67.

<sup>16</sup>Siehe zur Verdeutlichung das Beispiel in Abschnitt 6.3.3.

<sup>17</sup>Eine Ausführliche Auseinandersetzung zum Thema der Güteberechnung von Hypothesen sowie weitere Verfahren zur Güteberechnung finden sich in Streller, D.: *Multi-Hypothesen-Ansatz* (2006), S. 73-83.

Aus den Güten  $g_i$  der Objekthypothesen wird eine Güte  $g_{SH}$  der Szenenhypothesen SH berechnet:

$$g_{SH} = f(g_1, \dots, g_{n_{SH}}). \quad (6.8)$$

Hierbei stellt  $n_{SH}$  die Anzahl der Objekthypothesen einer Szene  $SH_j$  dar. Als Funktion  $f$  wurde im Projekt PRORETA das arithmetische Mittel verwendet.

Zur Reduktion der Hypothesen wird das so genannte  $N$ -Scan-Pruning Verfahren eingesetzt.<sup>18</sup> Abbildung 6.6 verdeutlicht das Vorgehen. In einem so genannten Hypothesenbaum wird der Verlauf der Szenenhypothesen der letzten  $N + 1$  Zeitschritte ausgehend von einem Wurzelknoten gespeichert (im Beispiel gilt für die Baumgröße  $N = 2$ ).

Zum Zeitschritt  $k$  wird dann anhand der Güten der Szenenhypothesen  $g_{SH}$  die beste Hypothese bestimmt. Von dem Knoten, der diese Hypothese enthält, wird der Baum  $N$  Schritte zurückverfolgt. Der Knoten, der erreicht wird, ergibt den neuen Wurzelknoten für den Zeitschritt  $k + 1$ . Alle Szenenhypothesen unterhalb des neuen Wurzelknotens werden beibehalten, alle übrigen werden zusammen mit den Objekthypothesen, die zu keiner Szenenhypothese mehr zugeordnet sind, gelöscht.

Hat im gezeigten Beispiel die Szenenhypothese  $SH_3$  zum Zeitschritt  $k$  die höchste Güte, wird der Knoten  $SH_1$  des Zeitschritts  $k - 2$  zum neuen Wurzelknoten. Die Szenenhypothesen  $SH_2$  und  $SH_4$  sowie die Objekthypothesen  $OH_3$ ,  $OH_4$  werden gelöscht.

Zu beachten ist, dass neue Objekthypothesen, die nicht im Widerspruch zu bestehenden Objekthypothesen stehen, zu jeweils allen bestehenden Szenenhypothesen hinzugefügt werden. Nicht weiter verfolgte Objekthypothesen werden ebenfalls aus allen bestehenden Szenenhypothesen gelöscht. Basis zur Generierung von Objekthypothesen sind dabei allein die nicht assoziierten Merkmalshypothesen der Sensorebene, Basis zum Löschen einer Hypothese ist allein die Güte  $g_i$  einer einzelnen Objekthypothese  $OH_i$ .

## 6.2.5 Anforderungsmanagement

Im Projekt PRORETA wird von der Anwendungsebene eine Prioritätsliste zu den Objekthypothesen an die Fusionsebene zurückgemeldet.<sup>19</sup> Über diese Liste kann der Erfassungsbereich des Laserscanners gesteuert und die Bearbeitungspriorität der detektierten Objekte im Videosensor festgesetzt werden. Zudem wird anhand der Prioritätsliste entschieden, welche Objekthypothesen gelöscht werden können, wenn deren Anzahl eine bestimmte Obergrenze übersteigt.

<sup>18</sup>Siehe z.B. Blackman, S.: *Multiple hypothesis tracking* (2004).

<sup>19</sup>Siehe Darms, M.; Winner, H.: *PRORETA: Umfelderkennung* (2006).

## 6.3 Umsetzung der Sensorebene für den Laserscanner

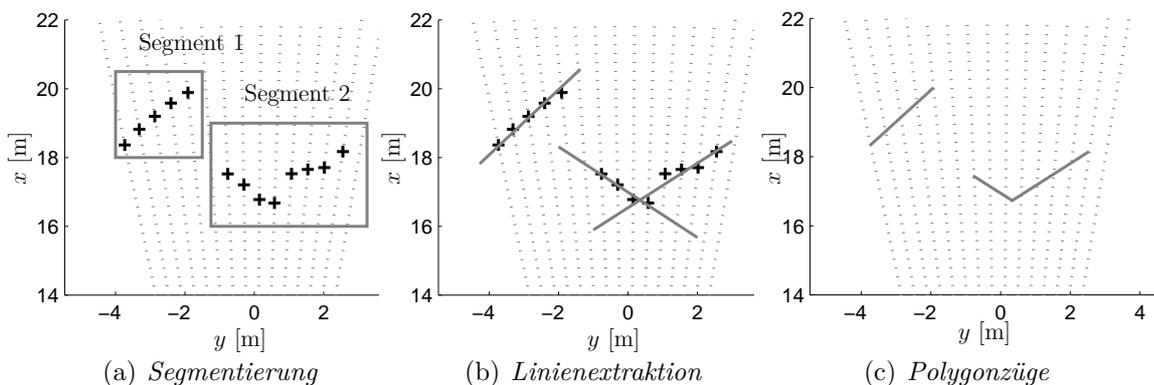
### 6.3.1 L1-Darstellung

Die Merkmalsannahmen MH des Laserscanners werden durch einen Polygonzug in Sensorkoordinaten dargestellt:

$$\text{MH} \mapsto (\xi_1, \dots, \xi_n) \text{ mit } \xi_i = (x_i, y_i) \text{ in Sensorkoordinaten, } i = 1, \dots, n. \quad (6.9)$$

Eine Merkmalsannahme ist dabei zu höchstens einer Objekthypothese auf Sensorebene assoziiert. Zu jedem Eckpunkt wird zusätzlich die Information abgespeichert zu welchem Eckpunkt der jeweiligen Objekthypothese er assoziiert wurde. Die Daten des Laserscanners werden nicht zur Klassifikation genutzt. Von daher enthält die L1-Darstellung des Laserscanners im Projekt PRORETA keine Information zur Klassifikation.

### 6.3.2 Merkmalsextraktion



**Abbildung 6.7:** Merkmalsextraktion für Laserscannerdaten.

Abbildung 6.7 verdeutlicht das Vorgehen zur Merkmalsextraktion. Im Segmentierungsschritt (Abbildung 6.7(a)) wird zunächst die Zielliste mit Hilfe einer Heuristik segmentiert.<sup>20</sup> Im Anschluss findet für jedes Segment die Extraktion eines Polygonzugs statt, aus welchen sich die Merkmalsannahmen ergeben. Die Ziele eines Segments werden dazu mithilfe eines Algorithmus zur Linienerkennung in Untergruppen unterteilt. Für jede Untergruppe wird dann eine Ausgleichsgerade ermittelt<sup>21</sup> (Abbildung 6.7(b)). Die Schnittpunkte der Ausgleichsgeraden definieren schließlich die inneren Polygonpunkte. Die äußeren Polygonpunkte ergeben sich aus den Schnittpunkten der Ausgleichsgeraden mit einer Geraden, die durch den Koordinatenursprung und den jeweils äußersten Punkt des Segments gelegt wird (Abbildung 6.7(c)).

<sup>20</sup>Siehe Abschnitt 4.3, S. 48.

<sup>21</sup>Siehe z.B. Einsele, T.: *Localization in Indoor Environments* (2002), S. 42-50.

### 6.3.3 Assoziation

Die ermittelten Eckpunkte der Merkmalshypothesen werden zu den Eckpunkten der auf den Messzeitpunkt prädierten Objekthypothesen mit Hilfe eines Nächster-Nachbar Verfahren mit Gating<sup>22</sup> eindeutig assoziiert. Dies wird für jede Szenenhypothese getrennt durchgeführt, ein Eckpunkt einer Merkmalshypothese kann damit mehreren Objekthypothesen verschiedener Szenenhypothesen zugeordnet sein.

Wurden die Eckpunkte einer Merkmalshypothese  $m$  verschiedenen Objekthypothesen derselben Szenenhypothese zugeordnet, wird das der Merkmalshypothese zugrunde liegende Segment (Ursprungssegment) in  $m$  neue Segmente geteilt. Dieser Vorgang wird als Segment-Split bezeichnet. Hierzu werden die einzelnen Ziele des Ursprungssegments der am nächsten liegenden Objekthypothese zugeordnet, so dass  $m$  neue Segmente entstehen. Für jedes der neuen Segmente wird die Extraktion des Polygonzugs durchgeführt. Die Eckpunkte der so entstandenen  $m$  neuen Merkmalshypothesen werden zu den entsprechenden Eckpunkten der  $m$  Objekthypothesen assoziiert. Die neuen Merkmalshypothesen bekommen dabei einen niedrigeren Wert für die Güte  $g_{SL}$  zugewiesen als die Ursprungsmerkmalshypothese.

Ist die Ursprungsmerkmalshypothese in keiner anderen Szenenhypothese eindeutig einer Objekthypothese zugeordnet, wird auf Fusionsebene eine neue Szenenhypothese generiert, in der die  $m$  Objekthypothesen durch eine neu instantiierte Objekthypothese ersetzt werden (Objektkombination). Die Initialisierung erfolgt auf Basis der Ursprungsmerkmalshypothese, die Zustandsschätzung der neuen Objekthypothese ergibt sich aus einem mit den Varianzen gewichteten Mittelwert der  $m$  Objekthypothesen.<sup>23</sup>

Wurden die Eckpunkte einer Objekthypothese  $n$  verschiedenen Merkmalshypothesen zugeordnet, werden die den Merkmalshypothesen zugrunde liegenden  $n$  Segmente zusammengefasst. Dieser Vorgang wird als Segment-Kombination bezeichnet. Hierzu werden die Ziele der  $n$  Segmente zu einem neuen Segment zusammengefasst und für das neue Segment die Extraktion des Polygonzugs durchgeführt. Die Eckpunkte der so neu entstandenen Merkmalshypothese werden zu den Eckpunkten der Objekthypothese assoziiert. Die neue Merkmalshypothese bekommt wieder einen niedrigeren Wert für die Güte  $g_{SL}$  zugewiesen als die Ursprungsmerkmalshypothesen.

Sind die  $n$  Ursprungsmerkmalshypothesen in keiner anderen Szenenhypothese eindeutig jeweils einer Objekthypothese zugeordnet, wird auf Fusionsebene eine neue Szenenhypothese generiert, in der die Ursprungsobjekthypothese durch  $n$  neu instantiierte Objekthypothesen ersetzt wird (Objekt-Split). Die Initialisierung erfolgt jeweils auf Basis der  $n$  Merkmalshypothesen, die Zustandsschätzung wird von der Ursprungsobjekthypothese übernommen.

Eine konservative Methode, die ohne den Multihypothesenansatz bzgl. der Objekthypothesen auskommt, führt immer den Segment- bzw. den Objekt-Split durch. Ein Nachteil ist, dass eine

---

<sup>22</sup>Siehe Abschnitt 4.2, S. 45.

<sup>23</sup>Messvektorfusion, siehe auch Becker, J.-C.: *Fusion der Daten der objekterkennenden Sensoren* (2002), S. 86.

einmal geteilte Objekthypothese dann niemals wieder kombiniert werden kann.<sup>24</sup> Allerdings wird auf diese Weise verhindert, dass eine „Lücke“ übersehen wird.<sup>25</sup>

Wurde mindestens ein Eckpunkt einer Merkmalshypothese zu einer Objekthypothese assoziiert und wurden andere Eckpunkte dieser Merkmalshypothese zu keinem Eckpunkt der entsprechenden Objekthypothese assoziiert, werden für die Formschätzung auf Fusionsebene die neuen Polygonpunkte zu dieser Objekthypothese hinzugefügt. Wird kein Eckpunkt einer Merkmalshypothese zu einer Objekthypothese assoziiert, wird eine neue Objekthypothese instantiiert.

Ist umgekehrt mindestens ein Eckpunkt einer Objekthypothese einer Merkmalshypothese zugeordnet, wird jeder nicht zugeordnete Eckpunkt dieser Objekthypothese aus der Formschätzung entfernt. Wird kein einziger Eckpunkt einer Objekthypothese zu einer Merkmalshypothese assoziiert gilt die Objekthypothese als nicht bestätigt, was zu einer Verringerung der Güte  $g_{FL}$  der Objekthypothese führt.

## 6.4 Umsetzung der Sensorebene für den Videosensor

### 6.4.1 L1-Darstellung

Die Merkmalsypothesen des Videosensors werden durch die Koordinaten  $x, y$  und die Breite  $b$  gefundener Fahrzeugheckansichten im Videobild in Bildkoordinaten beschrieben:

$$MH \mapsto (x, y, b) \text{ in Bildkoordinaten.} \quad (6.10)$$

Zu jeder Merkmalsypothese wird gespeichert, zu welcher Objekthypothese sie assoziiert wurde. Zudem wird zu jeder Objekthypothese, die von der Fusionsebene zurückgeführt wurde, die Klassifikationsinformation *Als Fahrzeugheck klassifiziert*, *Nicht als Fahrzeugheck klassifiziert* bzw. *Keine Klassifikationsinformation* abgelegt.

### 6.4.2 Merkmalsextraktion & Klassifikation

Zur Merkmalsextraktion wurde im Projekt PRORETA ein Algorithmus eingesetzt, der in einem vorgegebenen Bildausschnitt auf Basis einer Heuristik nach den Merkmalen einer Fahrzeugheckansicht und dem Schatten unter dem Fahrzeug sucht.<sup>26</sup> Da der Algorithmus auf der eingesetzten Hardware zu aufwändig zur Bearbeitung des gesamten Videobildes in Echtzeit ist, kann jeweils nur ein Ausschnitt des Bildes bearbeitet werden. Dabei wird die von der

<sup>24</sup>Theoretisch betrachtet wird sich eine statische Szene für  $t \rightarrow \infty$  in so viele Objekthypothesen zerteilen wie Ziele vorhanden sind.

<sup>25</sup>Vgl. hierzu auch das in Abschnitt 4.3 auf S. 50 vorgestellte Verfahren.

<sup>26</sup>Siehe hierzu z.B. Thomanek, F.: *Visuelle Erkennung und Zustandsschätzung von Straßenfahrzeugen* (1996), S. 23-44; Tzomakas, C.; von Seelen, W.: *Vehicle Detection Using Shadows* (1998).

Fusionsebene in die Sensorebene zurückgeführte Information über bereits bekannte Objekthypothesen genutzt.

Zunächst werden die bestehenden Objekthypothesen auf den Messzeitpunkt prädiert. Aus den Eckpunkten wird dann die Position eines potentiellen Fahrzeugs berechnet. Zur Ermittlung der potentiellen Ablage  $\tilde{y}$  wird der Mittelwert der  $y$ -Koordinaten der beiden äußersten Polygonpunkte gebildet. Für die potentielle Entfernung  $\tilde{x}$  wird der dem eigenen Fahrzeug am nächsten liegende Polygonpunkt herangezogen.<sup>27</sup>

Der Punkt  $(\tilde{x}, \tilde{y})$  wird über die Abbildungsgleichungen der Kameraoptik<sup>28</sup> in das Bildkoordinatensystem transformiert. Hierbei wird die Annahme zugrunde gelegt, dass sich alle Objekte inkl. des eigenen Fahrzeugs auf einer flachen Ebene befinden.<sup>29</sup> Auf Basis des so gefundenen Punktes und einer vordefinierten Annahme zur maximalen Breite eines Fahrzeugs wird ein Suchfenster im Videobild definiert, in welchem nach den Merkmalen einer Fahrzeugheckansicht gesucht wird.

Die Anzahl der so bearbeiteten Objekte ist aufgrund der Echtzeitbedingungen nach oben beschränkt.<sup>30</sup> Zur Bestimmung der zu bearbeitenden Objekte wird der Sensorebene von der Fusionsebene eine Prioritätsliste vorgegeben. Im Projekt PRORETA wurde diese vorher auf Anwendungsebene erzeugt und an die Fusionsebene weitergegeben.

### 6.4.3 Assoziation

Wird über den Suchalgorithmus keine Fahrzeugheckansicht im entsprechenden Bildausschnitt gefunden, wird die Information *Nicht als Fahrzeugheck klassifiziert* zur entsprechenden Objekthypothese assoziiert. Die gefundenen Merkmalshypothesen können nun allerdings nicht direkt den zugrunde liegenden Objekthypothesen zugeordnet werden. Das Beispiel in Abbildung 6.8 verdeutlicht dies.

Die Pylonen und das Fahrzeug in Abbildung 6.8(a) werden über den Laserscanner detektiert, so dass jeweils eine Objekthypothese existiert.<sup>31</sup> Abbildung 6.8(b) zeigt die entsprechenden Suchfenster. Fenster (1) und (3) enthalten jeweils die gleiche Fahrzeugheckansicht, wobei aufgrund der groben Diskretisierung des Laserscanners das Suchfenster (3) leicht nach links deplatziert ist. Die Fahrzeugheckansicht wird jeweils vom Suchalgorithmus gefunden, es entsteht daher eine Mehrfachmessung des gleichen, real existierenden Objekts. Aufgrund der unterschiedlichen Suchfenster können sich die gefundenen Merkmalshypothesen hierbei geringfügig unterscheiden.

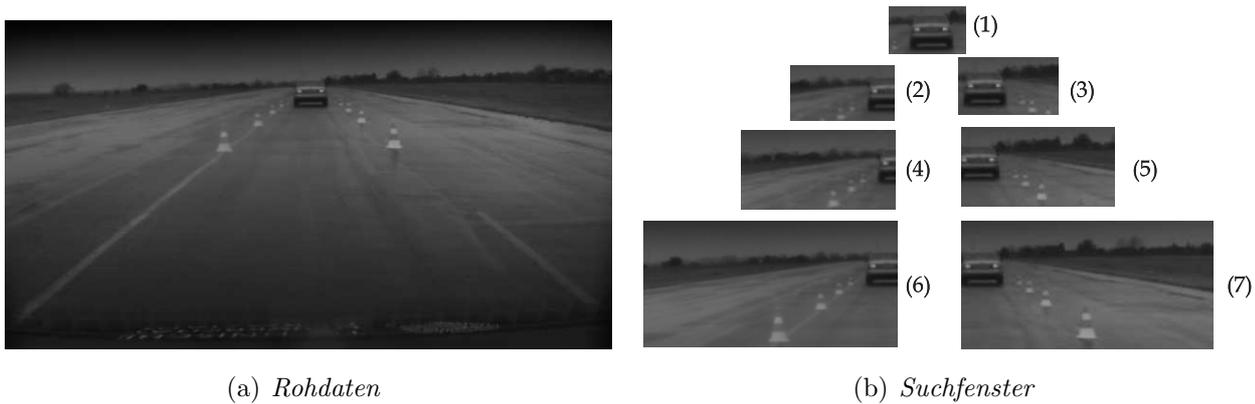
<sup>27</sup>Dies stellt eine sehr einfache Heuristik dar, welche z.B. schräg stehende Fahrzeuge nicht berücksichtigt.

<sup>28</sup>Es wurde ein Lochkameramodell verwendet.

<sup>29</sup>Der Nickwinkel des Fahrzeugs wird über Federwegsensoren ermittelt.

<sup>30</sup>Vgl. die Ausführungen in Abschnitt 6.6, S. 85.

<sup>31</sup>Die Pylonen direkt neben dem Fahrzeug können aufgrund der Auflösung des Laserscanners nicht vom Fahrzeug getrennt werden.



**Abbildung 6.8:** Mehrfachmessung einer Fahrzeugheckansicht.

Zur Lösung der Situation können wiederum verschiedene Strategien eingesetzt werden.<sup>32</sup> Für die Szenarien des Projekts PRORETA konnte eine einfache und robuste Heuristik gefunden werden. Die gefundenen Merkmals-hypothesen werden zunächst auf Mehrfachmessungen untersucht. Hierzu wird aufgrund von Schwellwerten entschieden, ob die extrahierten Merkmals-hypothesen dasselbe Objekt beschreiben. Bei Vorliegen einer Mehrfachmessung wird der Mittelwert der Einzelmessungen gebildet, die Ursprungshypothesen werden gelöscht.

Die Merkmals-hypothesen werden dann durch einen Nächster-Nachbar Algorithmus der am nächsten liegenden, in das Videobild projizierten Prädiktion zugeordnet, wobei die Objekthypothesen, zu welchen in diesem Messzyklus bereits das Attribut *Nicht als Fahrzeugheck klassifiziert* assoziiert wurde, nicht mehr berücksichtigt werden.<sup>33</sup> Die übrigen Objekthypothesen erhalten schließlich das Attribut *Keine Klassifikationsinformation* und es wird dementsprechend keine Merkmals-hypothese zugewiesen.

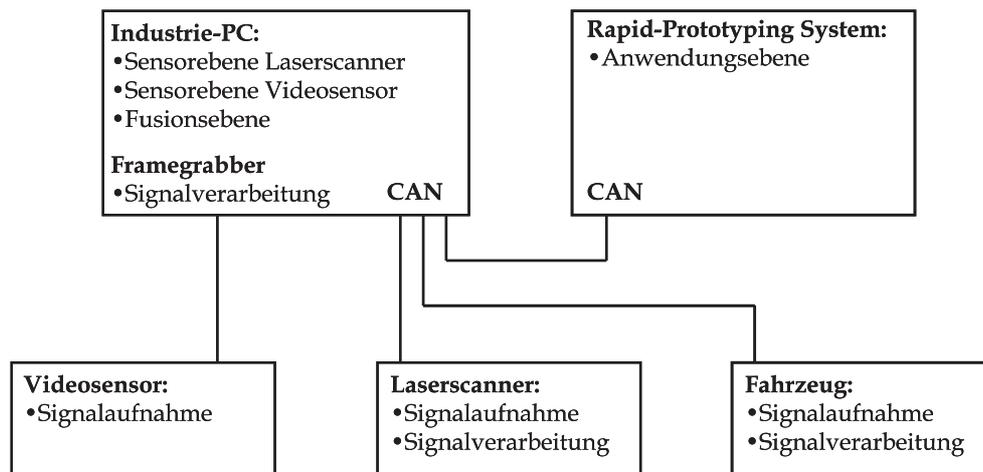
## 6.5 Verteilungssicht

Abbildung 6.9 zeigt auf welche Hardwarebausteine die einzelnen Komponenten der Bausteinsicht verteilt sind. Die Algorithmen der Anwendungsebene<sup>34</sup> laufen auf einem eigenen Rapid-Prototyping System, um die Echtzeitanforderungen der Regelungsalgorithmen einhalten zu können. Die Algorithmen zur Umfelddatenverarbeitung laufen auf einem Industrie-PC mit dem Betriebssystem Windows 2000. Beide Rechner sind über einen privaten CAN-Bus miteinander verbunden, auf welchem die L2-Darstellung und die Prioritätsliste übertragen werden. Zudem läuft über diesen Bus ein Protokoll zur Synchronisation der Systemzeiten.

<sup>32</sup>Die gewählte Strategie bestimmt schließlich die Qualität der Umsetzung der Sensorebene für einen bestimmten Videosensor.

<sup>33</sup>Zur weiteren Stabilisierung können Objekthypothesen, welche bereits vor der Messung als Fahrzeugheck klassifiziert wurden, bei der Zuordnung bevorzugt werden.

<sup>34</sup>Siehe Stählin, U.; Schorn, M.; Isermann, R.: *PRORETA: Notausweichen* (2006).



**Abbildung 6.9:** Realisierung der Verteilungssicht im Projekt PRORETA.

Für den Laserscanner findet die Signalaufnahme und Signalverarbeitung bereits in der Sensorhardware statt, so dass direkt die Zielliste über den CAN-Bus übertragen wird. In der Hardware des Videosensors findet zunächst nur die Signalaufnahme statt, die Signalverarbeitung dann schließlich auf einer Framegrabber Karte im PC. Schließlich stehen noch die Daten der Federwegsensoren über den Fahrzeug-CAN-Bus zur Verfügung.

Die Sensorebenen und die Fusionsebene laufen auf dem gleichen Prozessor. Die Ebenen sind hierbei im Rahmen eines Multi-Thread Programms in der objektorientierten Programmiersprache C++ umgesetzt.

## 6.6 Laufzeitsicht

Abbildung 6.10 zeigt die Umsetzung der Laufzeitsicht. Sobald neue Rohdaten eintreffen, wird von der entsprechenden Sensorebene eine Prädiktion der aktuellen Zustandsschätzung<sup>35</sup> auf den Messzeitpunkt von der Fusionsebene angefordert. Die Fusionsebene stellt diese in der L2-Darstellung bereit, so dass auf Sensorebene die Merkmalsextraktion, die sensorspezifische Klassifikation und die Assoziation durchgeführt werden können. Sobald die Daten in L1-Darstellung vorliegen, finden auf Fusionsebene die Klassifikation, das Objekthypothesenmanagement und die Datenfilterung statt. Nach Abschluss der Operationen wird die aktuelle L2-Darstellung inkl. des Zeitstempels der Signalaufnahme an die Anwendungsebene weitergeleitet. Neue Anforderungen für eine Prädiktion werden erst nach Abschluss eines kompletten Zyklus, beginnend mit der Anforderung der Prädiktion bis zum schließlichen Senden der neuen L2-Darstellung, beantwortet. Die Rohdaten werden in der Zwischenzeit auf Sensorebene zwischengespeichert. Sollten neue Rohdaten vor der Abarbeitung des aktuellen Datensatzes auf Sensorebene anfallen, werden auch diese in einem Puffer auf Sensorebene gespeichert und

<sup>35</sup>Es werden sowohl die Schätzungen zu den Objekten aus dem Umfeld des Fahrzeugs sowie der Nickwinkel an die Sensorebene geleitet.

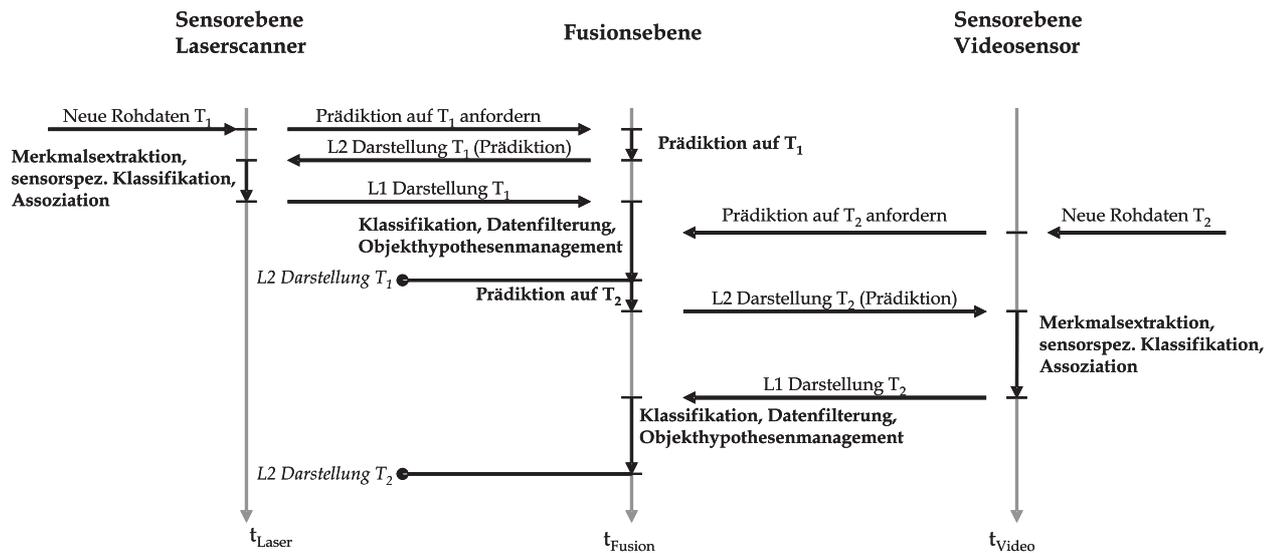


Abbildung 6.10: Realisierung der Laufzeitsicht im Projekt PRORETA.

direkt im Anschluss an die Bearbeitung des ersten Datensatzes verarbeitet. Das System ist dabei so abgestimmt, dass die Wahrscheinlichkeit für das Auftreten eines dritten Überlaufs gering ist.<sup>36</sup> Im Ausnahmefall wird zur Wahrung der Echtzeitbedingungen der jeweils letzte Datensatz verworfen.

## 6.7 Vorteile der Fusion

Die Szenarien des Projekts PRORETA lassen sich prinzipiell auch mit den Informationen des Laserscanners alleine bewältigen. Durch die Hinzunahme der Informationen des Videosensors lassen sich allerdings zwei wichtige Vorteile erzielen:

- Objekthypothesen können als Fahrzeugheckansichten klassifiziert werden. Diese Information kann bei der Entscheidung, ob ein Notmanöver durchgeführt werden soll, berücksichtigt werden. Die Gefahr einer Fehlalarmlösung, z.B. durch ein Objekt am Fahrbahnrand, kann auf diese Weise verringert werden. Mit dem Laserscanner alleine kann hier nur die Information über die Größe einer Objekthypothese herangezogen werden, welche je nach Beschaffenheit der Oberfläche des realen Objekts, verfälscht sein kann.<sup>37</sup>
- Aufgrund der höheren lateralen Auflösung des Videosensors lassen sich die Quer-Position und die Quer-Geschwindigkeit eines beobachteten Fahrzeugs präziser schätzen als mit dem Laserscanner alleine. Hierdurch ist eine exaktere Berechnung des Eingriffszeitpunkts und eine genauere Bestimmung der Ausweichtrajektorie möglich.

<sup>36</sup>Das eingesetzte Betriebssystem ist nicht echtzeitfähig, eine deterministische Aussage kann daher nicht getroffen werden.

<sup>37</sup>Letztendlich wird ein Seriensystem konservativ ausgelegt werden müssen, d.h. es muss akzeptiert werden, dass nicht jede kritische Situation zu einer Auslösung des Systems führt, um damit die Wahrscheinlichkeit einer Fehlalarmlösung zu reduzieren.



**Abbildung 6.11:** Versuchsaufbau zur Sensordatenfusion.

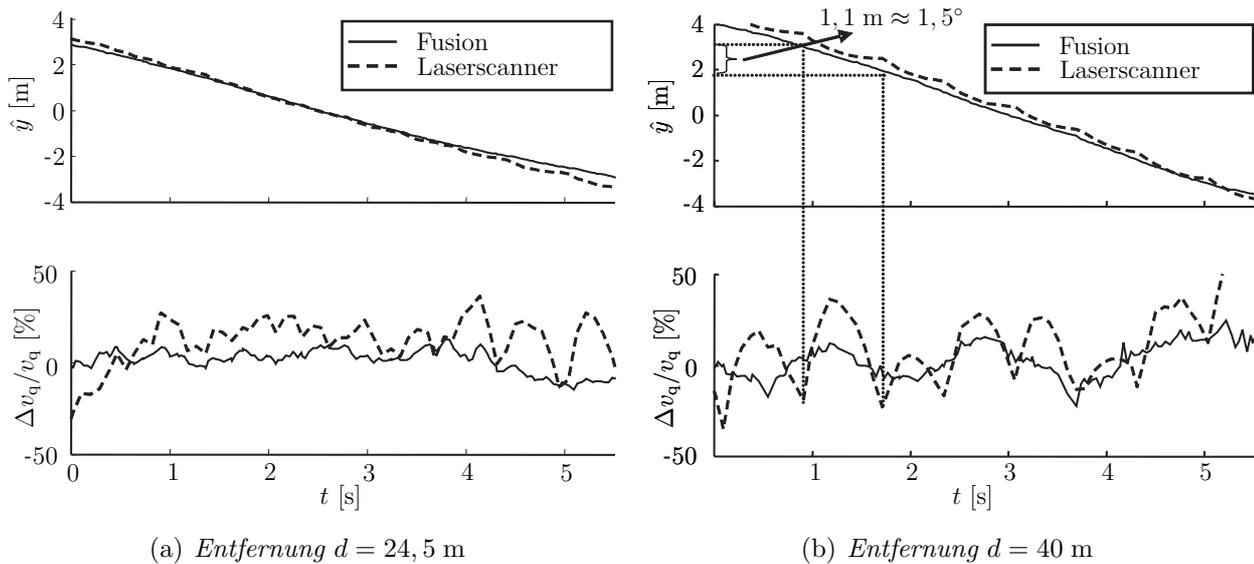
Abbildung 6.11 zeigt einen Versuchsaufbau zur Untersuchung der Schätzgenauigkeit der Quergeschwindigkeit. In einem Abstand  $d$  bezogen auf das Fahrzeugkoordinatensystem wird eine Fahrzeugattrappe senkrecht mit konstanter Geschwindigkeit zur Fahrzeugmittellachse bewegt. Die Geschwindigkeit  $v_q$  der Fahrzeugattrappe wird dabei mittels eines Correvit-Sensors gemessen und dient als Referenzgröße für die Schätzung. Die Schätzung der Quergeschwindigkeit durch den Laserscanner wird mit  $\hat{v}_{q,\text{Laser}}$  bezeichnet, die aus der Fusion ermittelte Schätzung mit  $\hat{v}_{q,\text{Fusion}}$ . Der Schätzfehler  $\Delta v_q$  berechnet sich damit zu:

$$\Delta v_q = \hat{v}_q - v_q. \quad (6.11)$$

Die Güte der Schätzung wird über die Wurzel der mittleren quadratischen Abweichung der Schätzung von der Messung bestimmt:

$$s = \sqrt{\frac{1}{T} \int_T (\hat{v}_q - v_q)^2 dt}. \quad (6.12)$$

$T$  bezeichnet dabei die Dauer der Messung und ist für beide Varianten identisch.



**Abbildung 6.12:** Schätzung der Querposition  $\hat{y}$  und Schätzfehler der Quergeschwindigkeitsschätzung mit und ohne Sensordatenfusion.

Abbildung 6.12 zeigt die Ergebnisse von zwei Experimenten, die bei unterschiedlichen Entfernungen  $d$  durchgeführt wurden. Aufgrund der größeren Diskretisierung des Laserscanners ist der Fehler in der Schätzung ohne die Informationen des Videosensors schlechter. Der Effekt verstärkt sich dabei mit größeren Entfernungen  $d$ . In den periodischen Schwankungen der Messfehler bei Verwendung des Laserscanners spiegelt sich zudem die Auflösung des Sensors wider, wie in Abbildung 6.12(b) zu sehen ist. Tabelle 6.1 zeigt die Werte für die Güte der Schätzung.

**Tabelle 6.1:** Vergleich der Güte der Quergeschwindigkeitsschätzung.

	$s_{Laser}/m/s$	$s_{Fusion}/m/s$
$d = 24,5$ m	0,18	0,06
$d = 40$ m	0,30	0,12

In beiden Fällen ist die mittlere quadratische Abweichung für die Fusion um ca. den Faktor 3 kleiner als bei Verwendung des Laserscanners alleine.

## 6.8 Fazit zur Umsetzung

Mit der Implementierung der Basis-Systemarchitektur im Projekt PRORETA konnte deren praktische Umsetzbarkeit nachgewiesen werden. Die nach der vorgestellten Architektur umgesetzte Umfelddatenverarbeitung wurde dabei für ein Fahrerassistenzsystem zur Unfallvermeidung eingesetzt und der Öffentlichkeit präsentiert.<sup>38</sup>

Für die einzelnen Bausteine der Architektur wurden z.T. nicht allgemeingültige Heuristiken realisiert, welche allerdings den praktischen Anforderungen des Projekts gerecht wurden. Durch den Aufbau der Architektur ist eine Weiterentwicklung bzw. ein Austausch dieser Algorithmen möglich, ohne dass die zugrunde liegende Struktur geändert werden muss.

Als in der praktischen Umsetzung vorteilhaft hat sich die unterschiedliche Implementation der Datenassoziation für die beiden Sensortypen in den jeweiligen Sensorebenen erwiesen. Auf diese Weise konnte den verschiedenen Charakteristiken der Sensoren Rechnung getragen werden. In beiden Sensorebenen fand dabei eine Verknüpfung der Merkmalsextraktion mit der Assoziation statt, wobei dieser Schritt für die Fusionsebene unsichtbar gekapselt in den jeweiligen Modulen blieb.

Durch Einsatz eines entsprechend schnellen Bussystems können die Sensorebenen und die Fusionsebene auf verschiedenen ggf. auf die Verarbeitung spezialisierten Prozessoren laufen. Auf diese Weise können die einzelnen Prozesse parallelisiert und so die Latenzzeit des Systems reduziert werden.

---

<sup>38</sup>Die offizielle Abschlussveranstaltung des Projekts PRORETA fand am 11.05.2006 auf dem Gelände der Continental Teves AG & Co. oHG in Frankfurt statt. Den Teilnehmern, vom Unternehmen eingeladenen Gästen inkl. Vertretern der Presse, wurde dabei die Möglichkeit geboten, das Versuchsfahrzeug selbst zu fahren und die Funktionen auszuprobieren. Siehe auch Bender, E. et al.: *Antikollisionssystem PRORETA - Teil 1* (2007a), S. 336-341; Bender, E. et al.: *Antikollisionssystem PRORETA - Teil 2* (2007b).

## 7 Fazit & Ausblick

Bei Fahrerassistenzsystemen mit Umfelderkennung werden in Zukunft die Daten von mehreren umfelderfassenden Sensoren fusioniert. Dies geschieht, um Kosten zu sparen und die Leistungsfähigkeit der Umfelderkennung zu verbessern. Die fusionierte Information bildet dabei die Basis für mehrere Anwendungen gleichzeitig.

Die Gestaltung der Hard- und Softwarearchitektur der Sensordatenfusion zu Beginn des Entwicklungsprozesses beeinflusst dabei die Kosten maßgeblich. Es existiert hierbei kein Verfahren, welches erlaubt, direkt aus den Anforderungen an das System eine optimale Architektur zu entwickeln. Vielmehr ist der Architekturentwurf ein iterativer Prozess, der durch die Anwendung von Heuristiken und bewährten Prinzipien optimiert werden kann.

In jedem Algorithmus zur Sensordatenfusion kommen die Schritte der Datenassoziation, d.h. der räumlichen und zeitlichen Zuordnung der gemessenen Daten, und der Datenfilterung, also der Trennung von Signal und Störung bzw. der Rekonstruktion von nicht direkt messbaren Größen, vor. In der Arbeit wurden für Fahrerassistenzsysteme allgemeingültige Grundannahmen bzgl. dieser Schritte formuliert, die beim Entwurf der Architektur beachtet werden müssen, bzw. zur Optimierung herangezogen werden können. Die Gültigkeit der Annahmen wurde anhand von existierender Literatur und von Simulationen dargelegt, so dass sich die folgenden Schlussfolgerungen ziehen lassen.

Die Gestaltung der Datenfilterung ist von den Anforderungen der Anwendungen abhängig. Diese können sich dabei so sehr unterscheiden, dass je nach Messgenauigkeit des verwendeten umfelderfassenden Sensors ein gemeinsamer Betrieb der Anwendungen mit einem einzigen Filteralgorithmus nicht, bzw. nicht optimal möglich ist. Soll die Architektur offen sein für die Erweiterung um neue Anwendungen und flexibel bzgl. des Einsatzes von Sensoren unterschiedlicher Messgenauigkeit, so besteht die Forderung, für unterschiedliche Anwendungen, bzw. Anwendungsgruppen, eine eigene Datenfilterung vorzusehen.

Die Anforderungen der Anwendungen an die Dynamik und Genauigkeit der Datenfilterung unterscheiden sich von den Anforderungen der Datenassoziation an die Datenfilterung. Je nach Genauigkeit der eingesetzten Sensoren existieren Fälle, bei denen ein gemeinsamer Betrieb von Datenassoziation und Anwendung mit einem einzigen Filter nicht, bzw. nicht optimal möglich ist. Für verschiedene Anwendungen lässt sich allerdings immer ein gemeinsames Filter zur Datenassoziation finden. Für eine auf verschiedene Anwendungen erweiterbare Architektur besteht daher die Forderung, die Datenfilterung zur Datenassoziation getrennt von

der Datenfilterung für die Anwendungen zu realisieren. Hierbei ist es hinreichend, ein einziges Filter zur Datenassoziation für alle Anwendungen gemeinsam vorzusehen.

Die Datenassoziation selbst ist ein Prozess, der von den Eigenschaften der verwendeten Sensoren abhängt. Durch auf die Sensoren abgestimmte Algorithmen zur Datenassoziation lässt sich eine geringere Zahl von fehlerhaften Zuordnungen erreichen als durch die Verwendung eines sensorunabhängigen Algorithmus. Um diesen Vorteil nutzen zu können, besteht daher die Forderung, in der Architektur zur Sensordatenfusion sensorspezifische Assoziationsalgorithmen zu ermöglichen.

Die in der Arbeit vorgestellte Basis-Systemarchitektur berücksichtigt diese Schlussfolgerungen. Sie stellt eine Vorversion einer endgültigen Architektur dar, die im Rahmen weiterer Iterationsschritte vervollständigt werden kann.

Die Architektur ist in eine Anwendungsebene, eine sensorunabhängige Fusionsebene und eine sensorspezifische Sensorebene untergliedert. Sie erlaubt sowohl das Hinzufügen neuer Anwendungen als auch neuer Sensoren mit unterschiedlichen Charakteristiken, ohne dass die Struktur der Sensordatenfusion geändert werden muss. Zudem lassen sich Sensoren gleichen Typs leicht austauschen, bzw. die in der Sensorebene gekapselten Algorithmen zur Merkmalsextraktion, Datenassoziation und Klassifikation sukzessive verbessern, ohne dass die bestehende Struktur geändert werden muss.

Durch die praktische Implementierung im Projekt PRORETA wurde die Umsetzbarkeit der Architektur nachgewiesen. Zudem wurden die Vorteile einer Sensordatenfusion von zwei nahezu orthogonalen, auf verschiedene Anwendungen optimierten Sensoren gezeigt.

Die Beschreibung der vorgestellten Basis-Systemarchitektur bezieht sich auf die Bausteinsicht. Der Aufbau der Laufzeit- und Verteilungssicht wurde so weit wie möglich offen gehalten. Die Gestaltung dieser Sichten ist dabei eine Fragestellung, die im Rahmen der Vervollständigung des Architekturentwurfs beantwortet werden muss, ebenso wie die Definition allgemein anerkannter Schnittstellensprachen zwischen den Architekturebenen. Die vorliegende Arbeit hat die Lösung dieser Punkte beispielhaft anhand der Umsetzung im Projekt PRORETA aufgezeigt.

Die Frage nach der Umsetzung der Laufzeitsicht hat dabei Auswirkungen auf die Entwicklung der Filteralgorithmen auf Fusionsebene. Daneben beeinflusst die Realisierung der Verteilungssicht die Gestaltung der Sensorhardware und der Bussysteme zur Kommunikation sowie die Möglichkeit, Algorithmen parallel ausführen zu können. Neue Technologien wie das Bussystem Flexray bieten hierbei Ansatzpunkte, auch für den Einsatz synchronisierter Sensoren.

Die vorliegende Basis-Systemarchitektur kann als Grundlage für eine weitere Untersuchung der genannten Fragestellungen dienen und bietet Architekten einer Serien-Systemarchitektur eine validierte Grundlage, auf der sich aufbauend schließlich ein Gesamtsystem durch eine feinere Ausgestaltung der einzelnen Komponenten konzipieren und umsetzen lässt.

## A Filter

### A.1 Discrete White Noise Acceleration Modell (DWNA)

Für die Zustandsgleichung des DWNA Modells gilt:<sup>1</sup>

$$x(k+1) = Fx(k) + \Gamma v(k) \quad (\text{A.1})$$

mit

$$x = \begin{pmatrix} d_{\text{rel}} \\ v_{\text{rel}} \end{pmatrix}, \quad F = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix}, \quad \Gamma = \begin{pmatrix} \frac{1}{2}T^2 \\ T \end{pmatrix}. \quad (\text{A.2})$$

Hierbei gibt  $v(k) \sim \mathcal{N}(0, \sigma_v)$  eine konstante Beschleunigung im  $k$ -ten Zeitintervall an und  $T$  die Dauer eines Zeitintervalls.

### A.2 Discrete Wiener Process Acceleration Modell (DWPA)

Für die Zustandsgleichung des DWPA Modells gilt:<sup>2</sup>

$$x(k+1) = Fx(k) + \Gamma v(k) \quad (\text{A.3})$$

mit

$$x = \begin{pmatrix} d_{\text{rel}} \\ v_{\text{rel}} \\ a_{\text{rel}} \end{pmatrix}, \quad F = \begin{pmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{pmatrix}, \quad \Gamma = \begin{pmatrix} \frac{1}{2}T^2 \\ T \\ 1 \end{pmatrix}. \quad (\text{A.4})$$

Hierbei gibt  $v(k) \sim \mathcal{N}(0, \sigma_v)$  das Inkrement der Beschleunigung im  $k$ -ten Zeitintervall  $T$  an.

### A.3 Übertragungsfunktion des $\alpha\beta\gamma$ -Filters

Das zeitdiskrete  $\alpha\beta\gamma$ -Filter ist ein lineares zeitinvariantes System mit einem Eingang ( $d_{\text{rel}}$ ) und drei Ausgängen ( $\hat{d}_{\text{rel}}, \hat{v}_{\text{rel}}, \hat{a}_{\text{rel}}$ ). Die z-Transformierten dieser Größen werden im Folgenden

<sup>1</sup>Siehe Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 273.

<sup>2</sup>Siehe Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 274.

mit  $D_{\text{rel}}, \hat{D}_{\text{rel}}, \hat{V}_{\text{rel}}, \hat{A}_{\text{rel}}$  bezeichnet, die Argumente werden zur Erhöhung der Übersichtlichkeit weggelassen. Aus den Grundgleichungen des Kalmanfilters<sup>3</sup> lässt sich ein lineares Gleichungssystem aufstellen, welches die Eingangs- und Ausgangsgrößen im  $z$ -Bereich zueinander in Bezug setzt:

$$(a_1 a_2 a_3) \begin{pmatrix} \hat{D}_{\text{rel}} \\ \hat{V}_{\text{rel}} \\ \hat{A}_{\text{rel}} \end{pmatrix} = w_\infty D_{\text{rel}} \quad (\text{A.5})$$

mit

$$a_1 = \begin{pmatrix} z^{-1}(1 - \alpha) - 1 \\ -z^{-1}\frac{\beta}{T} \\ -z^{-1}\frac{\gamma}{2T} \end{pmatrix}, a_2 = \begin{pmatrix} z^{-1}T(1 - \alpha) \\ z^{-1}(1 - \beta) - 1 \\ -z^{-1}\frac{\gamma}{2T} \end{pmatrix}, a_3 = \begin{pmatrix} z^{-1}\frac{1}{2}T^2(1 - \alpha) \\ z^{-1}T(1 - \frac{\beta}{2}) \\ z^{-1}(1 - \frac{\gamma}{4}) - 1 \end{pmatrix} \quad (\text{A.6})$$

und

$$w_\infty = \begin{pmatrix} -\alpha \\ -\frac{\beta}{T} \\ -\frac{\gamma}{2T^2} \end{pmatrix}. \quad (\text{A.7})$$

Die Größen  $\alpha, \beta, \gamma$  lassen sich dabei aus dem Manöverindex  $\lambda$  berechnen<sup>4</sup>,  $T$  ist die Abtastperiode. Mit der Cramerschen Regel<sup>5</sup> lassen sich nun die Übertragungsfunktionen berechnen:

$$H_{d\hat{d}} = \frac{\hat{D}_{\text{rel}}}{D_{\text{rel}}} = \frac{\Delta_1}{\Delta}, H_{d\hat{v}} = \frac{\hat{V}_{\text{rel}}}{D_{\text{rel}}} = \frac{\Delta_2}{\Delta}, H_{d\hat{a}} = \frac{\hat{A}_{\text{rel}}}{D_{\text{rel}}} = \frac{\Delta_3}{\Delta}. \quad (\text{A.8})$$

Hierbei bezeichnet  $\Delta$  die Determinante der Matrix  $(a_1 a_2 a_3)$  und  $\Delta_i$  die Determinante der Matrix, die sich ergibt, wenn die  $i$ -te Spalte durch  $w_\infty$  ersetzt wird.

<sup>3</sup>Siehe z.B. Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 208.

<sup>4</sup>Die Berechnungsvorschrift ist z.B. in Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: *Estimation* (2001), S. 291 zu finden. Vorsicht: Gleichung 6.5.5-21 in dieser Quelle enthält einen Fehler, die Potenzzahl von  $p$  muss 3 lauten und nicht 2.

<sup>5</sup>Siehe z.B. Lipschutz, S.: *Lineare Algebra* (1977), S. 177.

# B Berechnungen zur adaptiven Fahrgeschwindigkeitsregelung

## B.1 Modellierung des Fahrzeugverhaltens

Das Fahrzeug im Regelkreis gemäß Abbildung 3.1 wird als  $PT_1$ -Glied mit  $T_1 = 0,15$  s modelliert.<sup>1</sup> Dieses wird durch eine bilineare  $z$ -Transformation<sup>2</sup> mit

$$s = C \frac{1 - z^{-1}}{1 + z^{-1}} \quad (\text{B.1})$$

in den  $z$ -Bereich transformiert. Die Übertragungsfunktion von Sollbeschleunigung  $a_{\text{set}}$  zur Fahrzeugbeschleunigung  $a_{\text{ego}}$  ergibt sich damit zu

$$H_{a_{\text{set}}a_{\text{ego}}}(z) = \frac{1 + z^{-1}}{k_2 + k_1 z^{-1}}. \quad (\text{B.2})$$

Hierbei gilt

$$k_1 = 1 - T_1 C \text{ und } k_2 = 1 + T_1 C. \quad (\text{B.3})$$

Für die Simulationen wurde  $C = 19,2$  gewählt, was eine gute Übereinstimmung der kontinuierlichen und diskreten Repräsentation bewirkt (exakte Übereinstimmung für  $f = 2,2$  Hz).

## B.2 Übertragungsfunktion

Die Übertragungsfunktion der Beschleunigung von vorausfahrendem Fahrzeug  $a_{\text{fw}}$  zu betrachtetem Fahrzeug  $a_{\text{ego}}$  auf Seite 28 ergibt sich gemäß Abbildung 3.1 zu

$$H_{a_{\text{fw}}a_{\text{ego}}}(z) = \frac{H_{ax}(z)H_{d_{\text{rel}}a_{\text{ego}}}(z)}{1 + H_{ax}(z)H_{d_{\text{rel}}a_{\text{ego}}}(z)}. \quad (\text{B.4})$$

Hierbei ist  $H_{d_{\text{rel}}a_{\text{ego}}}(z)$  die Übertragungsfunktion von  $d_{\text{rel}}$  nach  $a_{\text{ego}}$  des am Addierer aufbrochenen Kreises

$$H_{d_{\text{rel}}a_{\text{ego}}}(z) = \frac{H_{dd}(z)\tau_d^{-1}\tau_v^{-1} + H_{dv}(z)\tau_v^{-1}}{H_{a_{\text{set}}a_{\text{ego}}}^{-1}(z) + H_{av}(z)\tau_0\tau_d^{-1}\tau_v^{-1}} \quad (\text{B.5})$$

<sup>1</sup>Siehe z.B. Witte, S.: *Simulationsuntersuchungen ACC* (1996), S. A-20.

<sup>2</sup>Siehe z.B. Proakis, J. G.; Manolakis, D. G.: *Digital signal processing* (2007), S. 712-729.

mit  $H_{d\hat{d}}(z)$ ,  $H_{d\hat{v}}(z)$  nach (A.8),  $H_{a_{\text{set}}a_{\text{ego}}}(z)$  nach (B.2) und den Übertragungsfunktionen von Beschleunigung zu Geschwindigkeit

$$H_{av}(z) = \frac{Tz^{-1}}{1 - z^{-1}} \quad (\text{B.6})$$

bzw. von Beschleunigung zu Ort

$$H_{ax}(z) = \frac{\frac{1}{2}T^2z^{-1} + Tz^{-1}H_{av}(z)}{1 - z^{-1}}. \quad (\text{B.7})$$

## C Berechnungen zur Time To Collision

Die partiellen Ableitungen in Gleichung 3.10 auf Seite 32 ergeben sich zu:

$$\frac{\partial t_{\text{TTC}}}{\partial d_{\text{rel}}} = \frac{1}{\sqrt{v_{\text{rel}}^2 - 2a_{\text{rel}}d_{\text{rel}}}}, \quad (\text{C.1})$$

$$\frac{\partial t_{\text{TTC}}}{\partial v_{\text{rel}}} = \frac{1}{a_{\text{rel}}} \left( -1 - \frac{v_{\text{rel}}}{\sqrt{v_{\text{rel}}^2 - 2a_{\text{rel}}d_{\text{rel}}}} \right), \quad (\text{C.2})$$

$$\frac{\partial t_{\text{TTC}}}{\partial a_{\text{rel}}} = \frac{d_{\text{rel}}}{a_{\text{rel}}\sqrt{v_{\text{rel}}^2 - 2a_{\text{rel}}d_{\text{rel}}}} + \frac{v_{\text{rel}} + \sqrt{v_{\text{rel}}^2 - 2a_{\text{rel}}d_{\text{rel}}}}{a_{\text{rel}}^2}. \quad (\text{C.3})$$

# D Projekt Proreta

## D.1 Versuchsträger



**Abbildung D.1:** *Foto des Versuchsträgers aus dem Projekt PROETA.*

## D.2 Projektteam



**Abbildung D.2:** *Foto des PRORETA Projekt Teams*

*(Von links: Eva Bender, Michael Darms, Matthias Schorn, Ulrich Stählin).*

# Literaturverzeichnis

1. **Bar-Shalom, Y. [Update with out-of-sequence measurements in tracking]:** Update with out-of-sequence measurements in tracking: exact solution. Aerospace and Electronic Systems, IEEE Transactions on, 2002 Nr. 3, 769 – 777, ISBN 0018-9251
2. **Bar-Shalom, Y. [MMT Vol. 1: Advanced applications]:** Multitarget-multisensor tracking - Vol. 1: Advanced applications. Boston [u.a.]: Artech House, 1990, Artech House radar library, ISBN 0-89006-377-X
3. **Bar-Shalom, Y. [MMT Vol. 2: Applications and advances]:** Multitarget-multisensor tracking - Vol. 2: Applications and advances. Boston [u.a.]: Artech House, 1992, Artech House radar library, ISBN 0-89006-517-9
4. **Bar-Shalom, Y. [MMT Vol 3: Applications and advances]:** Multitarget-multisensor tracking - Vol 3: Applications and advances. Boston [u.a.]: Artech House, 2000, Artech House radar library, ISBN 1-580-53091-5
5. **Bar-Shalom, Y.; Fortmann, T. E. [Tracking and data association]:** Tracking and data association. Boston: Academic Press, 1988, Mathematics in science and engineering, ISBN 0120797607
6. **Bar-Shalom, Y.; Li, X.-R. [Multitarget-multisensor tracking]:** Multitarget-multisensor tracking - principles and techniques. [Storrs, Conn.]: YBS, 1995, ISBN 0-9648312-0-1
7. **Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T. [Estimation]:** Estimation with applications to tracking and navigation - theory, algorithms and software. New York, NY [u.a.]: Wiley, 2001, A Wiley-Interscience publication, ISBN 096483121X
8. **Bass, L.; Clements, P.; Kazman, R. [Software Architecture in Practice]:** Software Architecture in Practice. 2. Auflage. Boston; San Francisco; New York: Addison-Wesley, 2003, ISBN 0-321-15495-9
9. **Bauer, H. [Adaptive Fahrgeschwindigkeitsregelung ACC]:** Adaptive Fahrgeschwindigkeitsregelung ACC. Robert Bosch GmbH, 2002, Gelbe Reihe
10. **Bauer, H.; Zabler, E. [Sensoren im Kraftfahrzeug]:** Sensoren im Kraftfahrzeug. 1. Auflage. Stuttgart: Robert Bosch GmbH, 2001, 148, ISBN 3-7782-2031-4
11. **Beck, K. [Extreme Programming]:** Extreme Programming - Die revolutionäre Methode für Softwareentwicklung in kleinen Teams. Boston [u.a.]: Addison-Wesley, 2003, ISBN 3-8273-2139-5

12. **Becker, J.-C. [Fusion der Daten der objekterkennenden Sensoren]:** Fusion der Daten der objekterkennenden Sensoren eines autonomen Straßenfahrzeugs. Düsseldorf: VDI-Verl., 2002, ISBN 3-18-394808-7
13. **Bender, E. et al. [Antikollisionssystem PRORETA - Teil 1]:** Antikollisionssystem PRORETA - Teil 1: Grundlagen des Systems. ATZ, 2007 Nr. 4, 336-341
14. **Bender, E. et al. [Antikollisionssystem PRORETA - Teil 2]:** Antikollisionssystem PRORETA - Teil 2: Ergonomie und Fahrversuch. ATZ, 2007 Nr. 5, 456-463
15. **Bender, E.; Landau, K. [PRORETA: Fahrerverhalten]:** Fahrerverhalten bei automatisierten Lenk- und Bremsingriffen eines Fahrerassistenzsystems zur Unfallvermeidung. In Steuerung und Regelung von Fahrzeugen und Motoren - AUTOREG 2006. Band VDI-Berichte 1931 Düsseldorf: VDI-Verlag, 2006
16. **Besada, J. et al. [Design of IMM filter for radar tracking]:** Design of IMM filter for radar tracking using evolution strategies. Aerospace and Electronic Systems, IEEE Transactions on, 41 2005 Nr. 3, 1109 - 1122, ISBN 0018-9251
17. **Bishop, R. [Vehicle Technology and Trends]:** Intelligent Vehicle Technology and Trends. Bosten, London: Artech House Publishers, 2005, 345, ISBN 1580539114
18. **Blackman, S. S.; Popoli, R. [Design and analysis of modern tracking systems]:** Design and analysis of modern tracking systems. Boston [u.a.]: Artech House, 1999, Artech House radar library, ISBN 1-58053-006-0
19. **Blackman, S. [Multiple hypothesis tracking]:** Multiple hypothesis tracking for multiple target tracking. Aerospace and Electronic Systems Magazine, IEEE, 19 2004 Nr. 1 , Part 2, 5 - 18, ISBN 0885-8985
20. **Brooks, R. R.; Iyengar, S. S. [Multi sensor fusion]:** Multi sensor fusion. Upper Saddle River, NJ, USA: Prentice Hall, 1998, ISBN 0-13-901653-8
21. **Chen, Z. [Local observability]:** Local observability and its application to multiple measurement estimation. Industrial Electronics, IEEE Transactions on, 38 1991 Nr. 6, 491 - 496, ISBN 0278-0046
22. **Cramer, H. [Modelle zur multisensoriellen Erfassung des Fahrzeugumfeldes]:** Modelle zur multisensoriellen Erfassung des Fahrzeugumfeldes mit Hilfe von Schätzverfahren. Aachen: Shaker, 2006, 167, ISBN 978-3-8322-5165-9 3-8322-5165-0
23. **Darms, M.; Winner, H. [Fusion von Umfelddaten]:** Fusion von Umfelddaten für Fahrerassistenzsysteme. In **Stiller, C.; Maurer, M. (Hrsg.):** Workshop Fahrerassistenzsysteme FAS2003. Karlsruhe: Fmrt, 2003, 13-16
24. **Darms, M.; Winner, H. [Modular System Architecture]:** A modular system architecture for sensor data processing of ADAS applications. In Intelligent Vehicles Symposium, 2005. Proceedings. IEEE. Las Vegas, USA, 6.-8. Juni 2005 2005, 729 - 734

25. **Darms, M.; Winner, H. [PRORETA: Umfelderkennung]:** Umfelderkennung für ein Fahrerassistenzsystem zur Unfallvermeidung. In Steuerung und Regelung von Fahrzeugen und Motoren - AUTOREG 2006. Band VDI-Berichte 1931 Düsseldorf: VDI-Verlag, 2006, 207–218
26. **Dietmayer, K.; Kirchner, A.; Kämpchen, N. [Fusionsarchitekturen]:** Fusionsarchitekturen zur Umfeldwahrnehmung für zukünftige Fahrerassistenzsysteme. In **Mauerer, M. (Hrsg.):** Fahrerassistenzsysteme mit maschineller Wahrnehmung. New York: Springer, 2005, 59–87
27. **Domschke, W.; Drexl, A. [Operations-Research]:** Einführung in Operations-Research. 4. Auflage. Berlin [u.a.]: Springer, 1998, Springer-Lehrbuch, ISBN 3–540–64587–X
28. **Einsele, T. [Localization in Indoor Environments]:** Localization in Indoor Environments Using a Panoramic Laser Range Finder. München: Online-Publikation, 2002 (URL: <http://deposit.ddb.de/cgi-bin/dokserv/idn=963995553>)
29. **Fardi, B.; Scheunert, U.; Wanielik, G. [Pedestrian Detection in Infrared Images]:** Shape and Motion-based Pedestrian Detection in Infrared Images: A Multi Sensor Approach. In IEEE Intelligent Vehicles Symposium. Las Vegas, USA, 6.-8. Juni 2005 2005
30. **Föllinger, O. [Regelungstechnik]:** Regelungstechnik - Einführung in die Methoden und ihre Anwendung. 6. Auflage. Heidelberg: Hüthig Buch Verlag, 1990
31. **Fuerstenberg, K.; Dittmer, M. [Ego-motion estimation]:** Data fusion and ego-motion estimation using laserscanners. Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE, 2 2003, 1224 – 1228
32. **Gränicher, H. [Grundlagen physikalischer Messungen]:** Grundlagen physikalischer Messungen. Zürich: Physik-Departement, ETH-Hönggerberg, 1978
33. **Gränicher, H. [Messung beendet - was nun?]:** Messung beendet - was nun? - Einführung und Nachschlagewerk für die Planung und Auswertung von Messungen. 2. Auflage. Zürich [u.a.]: vdf [u.a.], 1996, Getr. Zählung, ISBN 3–519–13659–7 ; 3–7281–2258–0
34. **Hall, D. L. [Handbook of multisensor data fusion]:** Handbook of multisensor data fusion. Boca Raton [u.a.]: CRC Press, 2001, The electrical engineering applied signal processing series (URL: [http://www.electricalengineeringnetbase.com/ejournals/books/book\\_km.asp?id=491](http://www.electricalengineeringnetbase.com/ejournals/books/book_km.asp?id=491)), ISBN 0–8493–2379–7
35. **Hall, D. L.; McMullen, S. A. [Mathematical techniques in multisensor data fusion]:** Mathematical techniques in multisensor data fusion. 2. Auflage. Boston[u.a.]: Artech House, 2004, ISBN 1–58053–335–3

36. **Hall, D.; Llinas, J. [An introduction to multisensor data fusion]:** An introduction to multisensor data fusion. Proceedings of the IEEE, 85 1997 Nr. 1, 6 – 23, ISBN 0018–9219
37. **Hänsler, E. [Statistische Signale]:** Statistische Signale: Grundlagen und Anwendungen. 2. Auflage. Berlin [u.a.]: Springer, 1997, ISBN 3–540–61816–3
38. **Hilberg, W. [Grundlagen elektronischer Schaltungen]:** Grundlagen elektronischer Schaltungen. 2. Auflage. München [u.a.]: Oldenbourg, 1992, Grundlagen der Schaltungstechnik, ISBN 3–486–22422–0
39. **Holt, V. v. [Multisensorielle Fahrumgebungserfassung]:** Integrale multisensorielle Fahrumgebungserfassung nach dem 4D-Ansatz. München, Univ. der Bundeswehr, Diss., 2004 (Online Publikation), 2005 (URL: urn:nbn:de:bvb:706-1072)
40. **Horn, M.; Dourdoumas, N. [Regelungstechnik]:** Regelungstechnik: Rechnerunterstützter Entwurf zeitkontinuierlicher und zeitdiskreter Regelkreise. München [u.a.]: Pearson Studium, 2004, et - Regelungstechnik, 457, ISBN 3–8273–7059–0 ; 978–3–8273–7059–4
41. **Huang, B. [Regelkonzepte zur Fahrzeugführung]:** Regelkonzepte zur Fahrzeugführung unter Einbeziehung der Bedienelementeigenschaften. Dissertation, München, 2004
42. **Isaacson, D. L.; Madsen, R. W.; Quenues, W. [Markov chains]:** Markov chains, theory and applications. New York: Wiley, 1976, A Wiley publication in mathematical statistics, ISBN 0–471–42862–0
43. **ISO-Norm [Adaptive Cruise Control systems]:** ISO 15622 - Transport information and control systems - Adaptive Cruise Control systems - Performance requirements and test procedures. 2002
44. **ISO-Norm [Forward vehicle collision warning systems]:** ISO 15623 - Transport information and control systems - Forward vehicle collision warning systems - Performance requirements and test procedures. 2002
45. **Jacobson, I.; Booch, G.; Rumbaugh, J. [The unified software development process]:** The unified software development process - The complete guide to the unified process from the original designers. 1. Auflage. Reading, Mass. [u.a.]: Addison-Wesley, 1999, The Addison-Wesley object technology series, ISBN 0–201–57169–2
46. **Kaempchen, N. et al. [IMM object tracking]:** IMM object tracking for high dynamic driving maneuvers. 2004, 825 – 830
47. **Kämpchen, N.; Dietmayer, K. [Data synchronization strategies for multi-sensor fusion]:** Data synchronization strategies for multi-sensor fusion. In 10th World Congress on Intelligent Transport Systems. Band Proceedings of ITS 2003 Madrid, Spain, September 2003 2003

48. **KBSt [V-Modell XT]:** V-Modell XT Dokumentation. 13.09.2006 2006 (URL: <http://www.v-modellxt.de/>)
49. **Kirchner, A. [Sensordatenverarbeitung eines Laserscanners]:** Sensordatenverarbeitung eines Laserscanners für autonome Fahrfunktionen von Kraftfahrzeugen. Düsseldorf: VDI-Verl., 2000, ISBN 3-18-343812-7
50. **Klaus, F. [Techniken und Methoden der Multisensor-Datenfusion]:** Einführung in Techniken und Methoden der Multisensor-Datenfusion. Siegen, Univ., Habil.-Schr., Online-Publikation, 2004 (URL: <urn:nbn:de:hbz:467-575>)
51. **Klein, L. A. [Sensor and data fusion]:** Sensor and data fusion concepts and applications. Band 35, 2. Auflage. Bellingham, Wash: SPIE, 1999, XXII, 226, ISBN 0-8194-3231-8
52. **Knothe, F. et al. [Mercedes Benz CL-Klasse]:** Mercedes Benz CL-Klasse. ATZ, 108 2006 Nr. 10, 801-810
53. **Kodaka, K. et al. [Rear-End Collision Velocity Reduction System]:** Rear-End Collision Velocity Reduction System. In SAE World Congress. Detroit (Michigan), 3. März 2003 2003
54. **Lipschutz, S. [Lineare Algebra]:** Lineare Algebra: Theorie u. Anwendung. Düsseldorf, New York: MacGraw-Hill, 1977
55. **Lunze, J. [Regelungstechnik 2]:** Mehrgrößensysteme, digitale Regelung. Band 2, Regelungstechnik. 4. Auflage. Berlin [u.a.]: Springer, 2006, XXIII, 651 S., ISBN 354032335X
56. **Maurer, M. [FAS2005]:** 3. Workshop Fahrerassistenzsysteme: FAS2005. Karlsruhe: Fmrt, 2005, 163, ISBN 3-9809121-1-6
57. **Maurer, M. [Automatisierung von Straßenfahrzeugen]:** Flexible Automatisierung von Straßenfahrzeugen mit Rechnersehen. Als Ms Auflage. Düsseldorf: VDI-Verl., 2000, ISBN 3-18-344312-0
58. **Mauthener, M. et al. [Out-of-Sequence Measurements]:** Out-of-Sequence Measurements Treatment in Sensor Fusion Applications: Buffering versus Advances Algorithms. In **Stiller, C.; Maurer, M. (Hrsg.):** 4. Workshop Fahrerassistenzsysteme, FAS2006. Karlsruhe: fmrt, 2006, 20-30
59. **Mayr, R. [Regelungsstrategien für die automatische Fahrzeugführung]:** Regelungsstrategien für die automatische Fahrzeugführung: Längs- und Querregelung, Spurwechsel- und Überholmanöver. Tokio: Springer, 2001, ISBN 3-540-67518-3
60. **Mayr, R.; Bauer, O. [Safety issues in intelligent cruise control]:** Safety issues in intelligent cruise control. In IEEE/IEEEJ/JSAI International Conference on Intelligent Transportation Systems. Tokyo, 5-8 October 1999 1999, 970 - 975

61. Meyer-Gramcko, F. [Verkehrsunfall und Fahrzeugtechnik, Nr. 3, 1990]: Gehörsinn, Gleichgewichtssinn und andere Sinnesleistungen im Straßenverkehr. Verkehrsunfall und Fahrzeugtechnik, 1990 Nr. 3, 73–76
62. Mori, S. [Random sets in data fusion]: Random sets in data fusion: formalism to new algorithms. Band 1, 2000, TUC4/24 – TUC4/31 vol.1
63. Naab, K. [Sensorik- und Signalverarbeitungsarchitekturen]: Sensorik- und Signalverarbeitungsarchitekturen für Fahrerassistenz und Aktive Sicherheit. In Aktive Sicherheit durch Fahrerassistenz. Garching bei München, 11. März 2004 2004
64. Narbe, B. et al. [Datennetzkonzepte - Teil 1]: Datennetzkonzepte für die Sensordatenfusion - Teil 1. Elektronik Automotive, 2003 Nr. 4, 54–59
65. Narbe, B. et al. [Datennetzkonzepte - Teil 2]: Datennetzkonzepte für die Sensordatenfusion - Teil 2. Elektronik Automotive, 2003 Nr. 5, 40–44
66. Niehsen, W.; Müller, M. [IMM-Tracking-Filter für Fahrerassistenzsysteme]: IMM-Tracking-Filter für Fahrerassistenzsysteme. In Stiller, C.; Maurer, M. (Hrsg.): Workshop Fahrerassistenzsysteme FAS2003. fmrt, 2003, ISBN 3–9809121–0–8, 37–41
67. N.N. [www.prevent-ip.org]: PREVENT. 01.01.2007 2007 <URL: [www.prevent-ip.org](http://www.prevent-ip.org)>
68. Proakis, J. G.; Manolakis, D. G. [Digital signal processing]: Digital signal processing. 4. Auflage. Upper Saddle River, NJ: Pearson Prentice Hall, 2007, ISBN 0–13–187374–1
69. Robinson, G.; Aboutalib, A. [Trade-off analysis of multisensor fusion levels]: Trade-off analysis of multisensor fusion levels. Proceedings of the 2nd National Symposium on Sensors and Sensor Fusion, 1990 Nr. 2, 21–34
70. Schulz, H.-G. [Modellgestützte Fehlerdiagnose]: Modellgestützte Fehlerdiagnose der Sensoren für die Fahrzeug-Querodynamik. Dissertation, Duisburg, 2006., Online-Ressource <URL: [urn:nbn:de:hbz:464-duett-04102006-1154363](http://nbn-resolving.org/urn:nbn:de:hbz:464-duett-04102006-1154363)>
71. Singh, R.-N.; Bailey, W. [Fuzzy logic applications]: Fuzzy logic applications to multisensor-multitarget correlation. Aerospace and Electronic Systems, IEEE Transactions on, 33 1997 Nr. 3, 752 – 769, ISBN 0018–9251
72. Sparbert, J.; Dietmayer, K.; Streller, D. [Laser Range Images]: Lane Detection and Street Type Classification using Laser Range Images. In IEEE Intelligent Transportation Systems Conference. Oakland (CA), USA, 25.-29. August 2001 2001
73. Specht, G.; Beckmann, C. [F&E-Management]: F&E-Management. Stuttgart: Schäffer-Poeschel, 1996, ISBN 3–8252–8112–4 (UTB) 3–7910–6012–0 (Schäffer-Poeschel)
74. Stählin, U.; Schorn, M.; Isermann, R. [PRORETA: Notausweichen]: Notausweichen für ein Fahrerassistenzsystem zur Unfallvermeidung. In Steuerung und Rege-

- lung von Fahrzeugen und Motoren - AUTOREG 2006. Band VDI-Berichte 1931 Düsseldorf: VDI-Verlag, 2006, 197–206
75. **Starke, G. [Effektive Software-Architekturen]:** Effektive Software-Architekturen - Ein praktischer Leitfaden. 2. Auflage. Wien: Hanser, 2005, ISBN 3-446-22846-2
  76. **Steinberg, A.; Bowman, C. [Revisions to the JDL Data Fusion Model]:** Revisions to the JDL Data Fusion Model. In **Hall, D. L.; Llinas, J. (Hrsg.):** Handbook of multisensor data fusion. Boca Raton [u.a.]: CRC Press, 2001, ISBN 0-8493-2379-7
  77. **Steinberg, A.; Bowman, C.; White, F. [Revisions to the JDL Data Fusion Model]:** Revisions to the JDL Data Fusion Model. Quebec City, Canada, 1998
  78. **Stiller, C. [FAS2003]:** Workshop Fahrerassistenzsysteme: FAS 2003. Karlsruhe: Fmrt, 2003, 44, ISBN 3-9809121-0-8
  79. **Stiller, C.; Maurer, M. [FAS2006]:** Workshop Fahrerassistenzsysteme: FAS 2006. Karlsruhe: Fmrt, 2006, ISBN 3-9809121-2-4
  80. **Streller, D.; Fürstenberg, K.; Dietmayer, K. [Vehicle and Object Models]:** Vehicle and Object Models for Robust Tracking in Traffic Scenes using Laser Range Images. In International Conference on Intelligent Transportation Systems. Singapore, 3.-6. September 2002 2002
  81. **Streller, D. [Multi-Hypothesen-Ansatz]:** Multi-Hypothesen-Ansatz zur Erkennung und Verfolgung von Objekten in Verkehrsszenen mit Laserscannern. Düsseldorf: VDI-Verl., 2006, ISBN 3-18-361412-X
  82. **Stüker, D. [Heterogene Sensordatenfusion]:** Heterogene Sensordatenfusion zur robusten Objektverfolgung im automobilen Straßenverkehr. Oldenburg, Univ., Diss. (Online Publikation), 2004 (URL: <http://deposit.d-nb.de/cgi-bin/dokserv?idn=972494464>), Online-Ressource
  83. **Thomanek, F. [Visuelle Erkennung und Zustandsschätzung von Straßenfahrzeugen]:** Visuelle Erkennung und Zustandsschätzung von mehreren Straßenfahrzeugen zur autonomen Fahrzeugführung. Düsseldorf: VDI-Verl., 1996, ISBN 3-18-327212-1
  84. **Tzomakas, C.; von Seelen, W. [Vehicle Detection Using Shadows]:** Vehicle Detection in Traffic Scenes Using Shadows. Institut für Neuroinformatik, Ruhr-Universität Bochum: Internal Report IRINI 98-06, 1998
  85. **Vogel, O. [Software-Architektur]:** Software-Architektur - Grundlagen - Konzepte - Praxis. 1. Auflage. München [u.a.]: Elsevier, Spektrum, Akad. Verl., 2005, XVI, 550, ISBN 3-8274-1534-9
  86. **Wallentowitz, H. [Sicherheitsanalyse zum automatischen Fahren]:** Sicherheitsanalyse der Systeme zum automatischen Fahren. Bremerhaven: Wirtschaftsverl. NW, Verl. für neue Wiss., 2001, ISBN 3-89701-688-5

87. **Waltz, E.; Llinas, J. [Multisensor data fusion]:** Multisensor data fusion. Boston u.a.: Artech House, 1990, Artech House radar library, ISBN 0-89006-277-3
88. **Weiss, K.; Kaempchen, N.; Kirchner, A. [Multiple-model tracking]:** Multiple-model tracking for the detection of lane change maneuvers. In Intelligent Vehicles Symposium, 2004 IEEE. Parma, Italien, 2004, 937 – 942
89. **Winner, H. [Adaptive Cruise Control]:** Adaptive Cruise Control. In **Jurgen, R. K. (Hrsg.):** Automotive Electronics Handbook. 2. Auflage. New York, London: McGraw-Hill, 1999, ISBN 0070344531, 30.1–30.30
90. **Witte, S. [Simulationsuntersuchungen ACC]:** Simulationsuntersuchungen zum Einfluß von Fahrerverhalten und technischen Abstandsregelsystemen auf den Kolonnenverkehr. Karlsruhe, Techn. Hochsch., Diss., 1996

# Lebenslauf

Dipl.-Wirtsch.-Ing. Michael Darms

104 S 24th Street

Pittsburgh, PA 15203 (USA)

Geburtsdatum, -ort 19.10.1976, Berlin-Wilmersdorf

Familienstand verheiratet

Staatsangehörigkeit deutsch

## Schule

1982-1986 Anne Frank-Schule (Grundschule), Großostheim

1986-1995 Friedrich Dessauer Gymnasium, Aschaffenburg

Abschluss: Allgemeine Hochschulreife (Abitur)

## Studium

1996-2002 Technische Universität Darmstadt

Wirtschaftsingenieurwesen mit Fachrichtung Elektrotechnik

Abschluss: Diplom

## Berufstätigkeit

09/2002-07/2006 Wissenschaftlicher Mitarbeiter am Fachgebiet Fahrzeugtechnik der Technischen Universität Darmstadt

seit 10/2006 Projektingenieur im Bereich Fahrerassistenzsysteme bei Continental Inc., Auburn Hills, MI (USA). Als Embedded Engineer an das Robotics Institute der Carnegie Mellon University in Pittsburgh, PA (USA) entsandt.

Pittsburgh, den 23. April 2007