
Kontinuierliche Verhaltenssteuerung eines autonomen Fahrzeugs in dynamischer Umgebung

Dem Fachbereich Informatik
der Universität Kaiserslautern
vorgelegte

Dissertation

zur Erlangung des Grades
eines Doktor rer. nat.

von

Dirk Reichardt

Januar 1996

Vorwort

Die vorliegende Arbeit beschreibt ein Konzept zur autonomen Fahrzeugführung für Kraftfahrzeuge auf Autobahnen sowie dessen Realisierung. Sie entstand in der Forschungsabteilung F1M/IA der Daimler-Benz AG, in der ich während dieser Arbeit tätig war.

Das hier vorgestellte Konzept stellt Methoden zur Verfügung, die anhand von Sensorinformationen eine Situationsbeschreibung aufbauen und adäquate Fahrmanöver ermitteln, die dann vom Fahrzeug ausgeführt werden. Das Versuchsfahrzeug ist dadurch in der Lage, im realen Autobahnverkehr ohne Eingriff des menschlichen Fahrers auf der Straße zu fahren, Sicherheitsabstände einzuhalten und sogar selbständig zu überholen und Ausweichmanöver einzuleiten.

Das im Rahmen dieser Arbeit entwickelte Steuerungssystem ist Bestandteil des Versuchsfahrzeugs VITA II (*Vision Technology Application*) der Daimler-Benz AG, das in der Endphase des EUREKA Projekts PROMETHEUS als Demonstrator für autonomes Fahren mittels Bildverarbeitung aufgebaut worden ist.

Ich möchte mich an dieser Stelle bei allen, die am Aufbau von VITA II beteiligt waren, für die sehr gute und erfolgreiche Zusammenarbeit bedanken. Sie haben es möglich gemacht, die entwickelten Konzepte auch im realen Straßenverkehr zu verifizieren.

Außerdem möchte ich Herrn Metzler und Herrn Hahn sowie allen Kollegen für die sehr gute Arbeitsatmosphäre danken, in der ich mich während der Erstellung meiner Arbeit sehr wohl gefühlt habe.

Zu Dank bin ich auch allen verpflichtet, die mir durch kurzfristiges Durchlesen und Korrigieren von Vorversionen meiner Arbeit geholfen haben.

Bei Herrn Professor Richter und Herrn Professor von Puttkamer möchte ich mich für die Unterstützung bedanken, die mir beim Verfassen meiner Arbeit eine große Hilfe war.

Mein besonderer Dank gilt Dr. Jens Schick, der durch zahlreiche anregende Diskussionen und durch konstruktive Kritik zum Erfolg dieser Arbeit beigetragen hat.

Inhaltsverzeichnis

1 EINLEITUNG	1
<hr/>	
1.1 Aufgabenstellung und Zielsetzung	1
1.2 Eingliederung in das Prometheus Programm	2
1.3 Gliederung und Überblick	5
<hr/>	
2 AUTONOMES FAHREN	7
<hr/>	
2.1 Aufbau eines autonomen mobilen Systems	7
2.1.1 Das Sensorsystem	8
2.1.2 Das Aktorsystem	10
2.1.3 Das Kontrollsystem	10
2.1.4 Das Kommunikationssystem	11
2.2 VITA II als Plattform für autonomes Fahren	12
2.2.1 Hardware- und Softwarearchitektur	14
2.2.1.1 Anwendungsrechner	14
2.2.1.2 Fahrzeugrechner	16
2.2.1.3 Plattformrechner	17
2.2.2 Das Sensorsystem	17
2.2.2.1 Straßenerkennung	17
2.2.2.2 Hinderniserkennung	19
2.2.2.3 Verkehrszeichenerkennung	22
2.2.3 Das Kontrollsystem	23
2.2.4 Die Fahrzeugansteuerung	23
2.2.5 Das Kommunikationssystem	24
<hr/>	
3 AUTONOME VERHALTENSSTEUERUNG	27
<hr/>	
3.1 Abgrenzung der Diskurswelt	32
3.2 Gefahrengebirge und elektrisches Feld	35
3.2.1 Der generalisierte Potentialfeldansatz	36
3.2.1.1 Die generalisierte Potentialfeldmethode nach Krogh	37
3.2.1.2 Diskussion von Erweiterungen der Potentialfeldmethode	39
3.2.2 Definition einer Gefahrenkarte	40
3.2.3 Die Gefahrenkarte als Grundlage der Verhaltenssteuerung	41
3.2.3.1 Modellierung der Straße	41
3.2.3.2 Modellierung anderer Verkehrsteilnehmer	43
3.2.3.3 Modellierung von Verkehrsregeln	45
3.2.3.4 Modellierung von Fahrerintentionen	46
3.2.4 Verknüpfung der Teilkarten	47
3.2.5 Interpretation der Gefahrenkarte als elektrisches Feld	48
3.2.6 Kritische Betrachtung des Ansatzes	49

3.3 Das Verhaltensmusterkonzept	52
3.3.1 Erklärungsmodelle des Fahrverhaltens	53
3.3.2 Architektur des Verhaltensmusterkonzepts	56
3.3.3 Wissensrepräsentation und Inferenz	58
3.3.3.1 Fahrer- und Fahrzeugparameter	59
3.3.3.2 Aufbau eines Verhaltensmusters	61
3.3.3.3 Repräsentation von Handlungsbeschreibungen	62
3.3.3.4 Repräsentation von Abhängigkeiten zwischen Verhaltensmustern	64
3.3.3.5 Algorithmus zur Verhaltenssteuerung	65
3.3.4 Verhaltensmuster für die Fahrzeugführung auf Autobahnen	72
3.3.4.1 Verhaltensmuster für Notreaktionen	73
3.3.4.2 Basisverhaltensmuster	80
3.3.4.3 Verhaltensmuster für schematisierte Handlungen	82
3.3.4.4 Integration der Fahrerintentionen	86
3.3.4.5 Verknüpfung der Verhaltensmuster	87
3.4 Aufbau einer Situationsbeschreibung aus Sensordaten	88
3.4.1 Aufgaben und Lösungsmethoden	88
3.4.1.1 Auftretende Fehlerarten	89
3.4.1.2 Die Fusionsaufgabe	90
3.4.1.3 Glättung von Sensordaten	92
3.4.2 Architektur des Situationsverwaltungssystems	94
3.4.3 Konzeptrahmen als Repräsentationsstruktur	96
3.4.3.1 Straßenkonzeptrahmen	97
3.4.3.2 Verkehrszeichenkonzeptrahmen	98
3.4.3.3 Hinderniskonzeptrahmen	99
3.4.4 Verwaltung des Szenenspeichers	100
3.4.4.1 Der Straßenverwaltungsprozeß	101
3.4.4.2 Der Verkehrszeichenverwaltungsprozeß	102
3.4.4.3 Der „Matching“-Prozeß	104
3.4.4.4 Der „Set & Forget“- Prozeß	106
3.4.4.5 Der „Merge & Split“-Prozeß	109
3.4.4.6 Der „Filtering & Extrapolation“-Prozeß	109
3.4.5 Schemabasierte Situationsanalyse	111
3.4.5.1 Anforderungen an ein Fehlermodell	111
3.4.5.2 Beispiele kritischer Verkehrsszenen	112
3.4.5.3 Erkennung von dynamischen Szenen	113
3.4.5.4 Korrektur erkannter Fehler	118
3.5 Simulation	119
3.5.1 Der Aufbau des Simulationssystems	120
3.5.1.1 Die Ein-/Ausgabeverwaltung	121
3.5.1.2 Die virtuelle Welt	123
3.5.1.3 Die Simulation von Agenten	124
3.5.2 Die Simulation von VITA II	127
3.5.2.1 Simulation des Sensorsystems von VITA II	128
3.5.2.2 Integration des Kontrollsystems von VITA II	130
3.5.2.3 Simulation des Aktorsystems von VITA II	131
3.5.3 Die Analysehilfsmittel der Simulationsumgebung	131
3.5.3.1 Generierung von Eingangsdaten	131
3.5.3.2 Darstellung und Auswertung	132

4 ERGEBNISSE IN SIMULATION UND REALITÄT	133
<hr/>	
4.1 Ergebnisse im realen Straßenverkehr	133
4.1.1 Laufzeitbetrachtungen	134
4.1.2 Ergebnisse aus Messfahrten	134
4.1.2.1 Randbedingungen bei autonomen Fahrten mit VITA II	134
4.1.2.2 Exemplarische Darstellung von Fahrmanövern	136
4.2 Simulationsergebnisse	140
4.2.1 Beispiel eines Überholvorgangs	140
4.2.2 Szenario für Notreaktionen	142
<hr/>	
5 ZUSAMMENFASSUNG UND AUSBLICK	145
<hr/>	
5.1 Zusammenfassende Kritik	145
5.2 Vergleich zu existierenden Verfahren	148
5.2.1 Das Kontrollsystem von VaMoRs-P	148
5.2.1.1 Zusammenfassung	149
5.2.1.2 Vergleich	150
5.2.2 Verwandte Arbeiten	150
5.3 Autonome Fahrzeuge im Straßenverkehr von morgen	151
<hr/>	
ANHANG	
<hr/>	

Abbildungsverzeichnis

Abbildung 1	: Der Daimler-Benz Versuchsträger VITA II.....	12
Abbildung 2	: Aufbau des autonomen Fahrzeugs VITA II.....	13
Abbildung 3	: Rechnerarchitektur von VITA II.....	14
Abbildung 4	: Prozessornetzwerk des Anwendungsrechners.....	15
Abbildung 5	: Prozessornetzwerk des Fahrzeugrechners.....	16
Abbildung 6	: Rückwärtige Kameraplattform in VITA II.....	18
Abbildung 7	: Kamerablickwinkel von VITA II.....	19
Abbildung 8	: VisionBumper Kamera in VITA II.....	21
Abbildung 9	: Fächerkamera des LOD-Moduls.....	21
Abbildung 10	: Kommunikationsstruktur in VITA II.....	25
Abbildung 11	: Informations- / Aktionszyklus des autonomen Fahrens.....	29
Abbildung 12	: Koordinatensystem der Diskurswelt.....	33
Abbildung 13	: Potentialfeld $P(x,v)$ mit $v = 12 \text{ m/s}$ und $\alpha = 5 \text{ m/s}^2$	38
Abbildung 14	: Beispiel einer Gefahrenkarte $G_{15,15}$	40
Abbildung 15	: Gefahrenkarte für eine einspurige Straße.....	42
Abbildung 16	: Modellierung einer Straße mit zwei Spuren.....	43
Abbildung 17	: Beispiel einer Sollgeschwindigkeitskurve.....	44
Abbildung 18	: Modellierung eines Hindernisses als Gefahrengelände.....	44
Abbildung 19	: Gefahrenrepräsentation für ein langsamer vorausfahrendes Fahrzeug.....	45
Abbildung 20	: Fahrerkenntlinie zur Einhaltung von Sollgeschwindigkeitsvorgaben.....	47
Abbildung 21	: Fusion einer Teilkarte zur Straßenbeschreibung (a) und einer zur Hindernisbeschreibung (b) mit dem Gewichtsverhältnis 2:3 (c) und 3:2 (d).....	48
Abbildung 22	: Drei-Ebenen Verhaltensmodell nach Rasmussen [Rasmussen 83].....	55
Abbildung 23	: Regelkreis "Mensch - Straße - Fahrzeug" nach Hehlen & Huguenin [Huguenin 88].....	56
Abbildung 24	: Architektur des Kontrollsystems basierend auf Verhaltensmustern.....	58
Abbildung 25	: Basisverhaltensmuster.....	63
Abbildung 26	: Übergangsnetz zur Repräsentation von Handlungsabläufen.....	64
Abbildung 27	: Übergangsnetz für Notreaktionen.....	64
Abbildung 28	: Gewichtung und Verknüpfung von Verhaltensmustern.....	70
Abbildung 29	: Übersicht über die Notreaktionen.....	74
Abbildung 30	: Modellierung der Fahrerreaktion für das Folgen vorausfahrender Fahrzeuge.....	81
Abbildung 31	: Übersicht über die Verhaltensmuster für schematisierte Handlungen.....	83
Abbildung 32	: Verhaltensmuster und Kantenschemata zum Aufbau eines Inhibitionsnetzes.....	86
Abbildung 33	: Funktionsschaubild des SA-Moduls.....	95
Abbildung 34	: Straßenkonzeptrahmen.....	98
Abbildung 35	: Aufbau des Situationsverwaltungssystems.....	101
Abbildung 36	: Einfache Ähnlichkeitsfunktion für Hindernispositionen.....	105
Abbildung 37	: Ähnlichkeitsfunktion für Verfolgungsfahrten.....	106
Abbildung 38	: Beispiele für Fehlerfälle.....	112
Abbildung 39	: Veranschaulichung unscharfer Begriffe für die Geschwindigkeit.....	114
Abbildung 40	: Automat zur Erkennung von Ereignissen.....	117
Abbildung 41	: Datenflußdiagramm der Simulation.....	121
Abbildung 42	: Straßenmodellierung in der virtuellen Welt.....	124
Abbildung 43	: Datenflußdiagramm für die Agentensimulation.....	125
Abbildung 44	: Datenflußschema der Sensorsimulation.....	129
Abbildung 45	: Benutzeroberfläche der Simulation.....	132
Abbildung 46	: Messung von Vision Bumper.....	135
Abbildung 47	: Automatischer Spurwechsel des VITA II Fahrzeugs.....	137
Abbildung 48	: Fusion der Entfernungsmessung von CT und ODT.....	138
Abbildung 49	: Schätzung der Hindernisgeschwindigkeit aus Messungen.....	138
Abbildung 50	: Automatische Abstandhaltung.....	139
Abbildung 51	: Abbremsen auf ein langsames Fahrzeug.....	140
Abbildung 52	: Distanzverläufe der Hindernisse eines Simulationslaufs.....	141
Abbildung 53	: Aktivierung des Überhol-Verhaltensmusters.....	141
Abbildung 54	: Geschwindigkeitsverlauf für das autonome Fahrzeug.....	142
Abbildung 55	: Simulierter Überholvorgang.....	142
Abbildung 56	: Notreaktionsszenario.....	143
Abbildung 57	: Steuervorgabe für die laterale Position.....	144
Abbildung 58	: Notbremsung bis zum Stillstand.....	144

Kurzfassung

Das Ziel der vorliegenden Arbeit ist es, ein Konzept zur Steuerung eines autonomen Fahrzeugs auf Autobahnen zu entwickeln und dieses anhand des Versuchsfahrzeugs VITA II (Vision Technology Application) der Daimler Benz AG auch zu realisieren.

Der hier vorgestellte Ansatz ist motiviert durch Erklärungsmodelle des menschlichen Fahrverhaltens sowie durch die Methode der künstlichen Potentialfelder zur Steuerung autonomer mobiler Roboter. Der direkte Zusammenhang zwischen dem Fahrverhalten und der subjektiv empfundenen Gefahr einer Verkehrssituation, bzw. des Risikos, das sich aus der Situation ergibt, wird genutzt, um ein Entscheidungskriterium für automatische Fahrmanöver zu erhalten.

Jedem Element der Umgebung wird ein Gefahrenwert zugeordnet. Aus dieser Gefahrenrepräsentation der Verkehrssituation werden dann Fahrzeugaktionen abgeleitet, die unter Berücksichtigung der Fahrerintentionen eine Gefahrenminimierung anstreben.

Zwei Ansätze werden vorgestellt, denen unterschiedliche Gefahrenrepräsentationen zugrunde liegen. In einer ersten Untersuchung wird eine Gefahrenkarte der Umgebung aufgebaut und Gefahr und Reaktion direkt gekoppelt, so daß sich die Steuerungsaufgabe als Pfadsuche-Problem in einem Gefahrengebirge darstellt. Der zweite Ansatz berücksichtigt durch diese Untersuchung gewonnene Erkenntnisse und trennt zwischen Gefahr und Reaktion. Das Wissen über Verhaltensweisen und Gefahren ist hier in konkurrierenden Verhaltensmustern repräsentiert. Diese Verhaltensmuster entsprechen Handlungsschemata, die Verhaltensweisen in verschiedenen Ebenen kognitiver Inanspruchnahme eines Fahrers beschreiben. Ein Inhibitionsnetzwerk verbindet die Verhaltensmuster und regelt den Grad ihrer Aktivierung über eine Risikoabwägung.

Mit dem Einsatz der Verhaltenssteuerungsmethoden im realen Umfeld ergibt sich die Problematik der Verarbeitung fehlerbehafteter, unvollständiger und inkonsistenter Information, die von den Sensormodulen des Versuchsfahrzeugs VITA II geliefert wird. Der zweite Teil der Arbeit befaßt sich daher mit einer Methode zur Sensorfusion und Filterung der durch Sensormodule erfaßten Umgebung.

Zur Vermeidung von Risiken wird eine Simulationsumgebung speziell für das Versuchsfahrzeug entwickelt und eingesetzt, in der beliebige Szenarien gestellt werden können, um die Reaktionen des Fahrzeugs zu testen.

Abstract

This thesis discusses an approach to automatic vehicle guidance on a motorway. The realization of the system has taken place on the Daimler-Benz research vehicle VITA II (Vision Technology Application).

The approach presented here is motivated by psychological models of human driving behaviour and by the artificial potential field approach which is known from robotics.

The close connection between driving behaviour and the subjective risk of a traffic situation is used to introduce a decision criterion for driving manoeuvres. Each element of the traffic scene is associated with a risk value. This leads to a risk based representation of traffic situations which is used to generate actions that intend to minimise the risk.

Two approaches are presented which are based on different risk representations. The first approach builds a risk map for representation of the environment. By computing the risk value and the reaction the vehicle guidance task is reduced to a search problem.

The second approach integrates the gained experience and separates risk and reaction. Knowledge about the driver's behaviour is represented by competing behavioural patterns. Behavioural patterns correspond to action schemes. They represent driving behaviour of different cognitive levels together with a risk value that describes the necessity of their execution. A set of behavioural patterns is arranged in an inhibition network structure. This network is used to combine their results with regard to their associated risk. The objective is the computation of a command vector for the autonomous vehicle.

In real world application the main problem is the noisy input data from different image processing modules. This renders autonomous driving very difficult. The main task to solve this problem is to build up a consistent situation representation in real-time. One part of this thesis deals with this problem and presents a sensor fusion method designed for VITA II.

As a supporting work a simulation system for system development and testing has been built up. It contains models of the used sensors and deterministic models of vehicle dynamics and provides an appropriate test bed for the subsequent use of the developed programs in real traffic.

Abkürzungen

AD/DA-Wandler	Analog D igital/ D igital A nalog - Wandler
AHS	A utomated H ighway S ystems
AICC	A utonomous I ntelligent C ruise C ontrol
AMR	A utonomer M obiler R oboter
AMV	A utonomous M obile V ehicle
AWR	A nwendungs r echner
BC	B ehaviour C ontrol
C40	TMS320 C40
CAN-Bus	C ontroller A rea N etwork
CCD-Kamera	C harge C oupled D evice
CED3	C ommon E uropean D emonstrators
CT	C ar T rack
DDB	D ynamic D ata B ase
FZG	Fahrzeug
I/O	I nput/ O utput
LOD	L ateral O bstacle D etection
MON	M onitoring Modul
ODT	O bstacle D etection and T racking.
PROMETHEUS	P rogramme for a E uropean T raffic with H ighest E fficiency and U nprecedented S afety
RDT/ RT	R oad D etection and T racking / R oad T racking
REC	R ecorder Modul
SA	S ituation A ssessment
SPARC	S calable P rocessor A rchitecture
STS	S tate T ransition S ystem
TIP	T ransputer I mage P rocessing
TLC	T ime to L ine C rossing
TSR	T raffic S ign R ecognition
TTC	T ime to C ollision
VB	V ision B umper
VC	V ehicle C ontrol
VITA	V ision T echnology A pplication

Mathematische Symbole

R	Reelle Zahlen
N	Natürliche Zahlen
Pot(M)	Potenzmenge einer Menge M

1 Einleitung

1.1 Aufgabenstellung und Zielsetzung

Das Thema der vorliegenden Arbeit ist Bestandteil einer Machbarkeitsstudie des autonomen Fahrens im Straßenverkehr mittels Bildverarbeitung. Basis dieser Studie ist das Versuchsfahrzeug VITA II (*Vision Technology Application*), welches innerhalb des europäischen Forschungsprogramms PROMETHEUS (vgl. Abschnitt 1.2) von der Daimler-Benz AG aufgebaut worden ist.

Die Aufgabenstellung der hier vorgestellten Arbeit umfaßt die Konzeption eines Systems zur Verhaltenssteuerung und dessen Realisierung im VITA II Fahrzeug. Die Verhaltenssteuerung beinhaltet Methoden zur Verarbeitung der fehlerbehafteten Informationen aus den Bildverarbeitungsmodulen und zur Generierung einer kontinuierlichen Steuervorgabe für die Längs- und Querführung dieses autonomen Fahrzeugs. Das Konzept ist auf die Anforderungen zugeschnitten, die durch den Einsatz im Versuchsfahrzeug VITA II gegeben sind. Die Arbeit ist daher praxisorientiert und durch die strenge Echtzeitanforderung geprägt.

Die innerhalb dieser Aufgabenstellung bearbeiteten Themengebiete lassen sich dem Bereich der praktischen Informatik zuordnen. Teilaufgaben beinhalten die Behandlung von Unsicherheit und Vagheit, die Modellierung zeitlichen Verhaltens, die Repräsentation von Wissen über Fahrer, Fahrzeug und Straßenverkehr und die echtzeitfähige Abbildung von kognitiven Prozessen auf einen Verhaltensentscheidungsalgorithmus.

Das autonome Fahrzeug ist damit in der Lage, im realen Autobahnverkehr zu agieren, ohne daß der Eingriff eines menschlichen Fahrers benötigt wird. Neben dem Folgen der Fahrspur und dem Einhalten von Sicherheitsabständen entscheidet das Kontrollsystem auch selbständig über die Durchführung von Überholmanövern und berücksichtigt Verkehrszeichen. Im Falle plötzlich auftretender Gefahrensituationen ist es weiterhin in der Lage, durch Ausweichen, Bremsen oder auch Beschleunigen Kollisionen zu vermeiden.

Dazu sind insbesondere die folgenden Anforderungen an das System zur Verhaltenssteuerung zu berücksichtigen:

- ***Reaktion auf beliebige Szenenkonstellationen***

Das Fahrzeug muß sich selbständig bewegen können. Dazu muß es in der Lage sein, auf alle sich ergebenden Situationen innerhalb seiner Domäne korrekt zu reagieren.

- ***Echtzeitfähigkeit***

Die Algorithmen des Kontrollsystems müssen auf der Zielhardware in Echtzeit lauffähig sein. Echtzeit bedeutet in diesem Fall, daß ein Verarbeitungszyklus so wenig Zeit beansprucht, daß das Gesamtsystem eine Reaktionszeit aufweist, die unter der eines menschlichen Fahrers liegt. Dabei sind auch die Latenzzeiten der Sensorsysteme und Aktorsysteme zu berücksichtigen. Da die verwendete Umgebungsinformation durch optische Sensoren erfaßt wird, bildet die in diesem Bereich verwendete Echtzeitdefinition, die eine Verarbeitungsrate von 25 Bildern pro Sekunde beschreibt, die Grundlage für den hier verwendeten Echtzeitbegriff.

- ***Erklärbarkeit***

Die Reaktionen des Fahrzeugs müssen für einen menschlichen Fahrer plausibel sein. Im besten Fall sollte das Fahrzeug in der gleichen Art reagieren wie ein menschlicher Fahrer.

Die Zielsetzung auf operationaler Ebene ist durch den Einsatz des Fahrzeugs im realen Straßenverkehr gegeben. Das Einsatzgebiet ist auf Autobahnen und autobahnähnliche Bundesstraßen ohne Gegenverkehr eingeschränkt.

1.2 Eingliederung in das PROMETHEUS Programm

Die in dieser Arbeit entwickelten Methoden und Algorithmen zur autonomen Fahrzeugführung werden nicht nur in der Simulation eingesetzt, sondern kommen auch im realen Straßenverkehr zum Einsatz. Da neben dem theoretischen Entwurf einer neuen Methode auch die echtzeitfähige Realisierung der Fahrzeugführungsalgorithmen und deren Einsatz im realen Straßenverkehr Inhalt der vorliegenden Arbeit ist, soll an dieser Stelle die Eingliederung der Arbeit in das PROMETHEUS Programm beschrieben werden.

Hintergrund der Initiierung des europäischen Forschungsprogramms PROMETHEUS im Jahre 1985 war ein ständig steigendes Verkehrsaufkommen in Westeuropa, die damit verbundenen 1,37 Millionen schweren Unfälle mit 60.600 Toten und 1,81 Millionen

Verletzten¹, aber auch die Angst vor einem wirtschaftlichen und technologischen Bedeutungsverlust Europas gegenüber Japan und den USA.

Das Hauptziel von PROMETHEUS war es, das Auto sicherer zu machen und den Verkehr besser zu organisieren. Dazu sollten die neuen Techniken, wie etwa die Mikroelektronik, die Sensorik sowie die Informations- und Kommunikationstechnik für das Automobil nutzbar gemacht werden. Die Themengebiete innerhalb PROMETHEUS waren

- das *Verkehrssystem-Management* mit den Aspekten duale Zielführung, Flottenmanagement und Verkehrsinformationssysteme,
- das *verkehrsangepaßte Fahren* mit den Aspekten kooperatives Fahren, autonome intelligente Geschwindigkeits- und Abstandsregelung sowie automatischer Notruf,
- das *sichere Fahren* mit den Aspekten der Sichtverbesserung, Überwachung des Kraftschlusses zwischen Rad und Straße sowie der verbleibenden Sicherheitsreserven, Unterstützung der Spurhaltung, Sichtweitenüberwachung, Überwachung des Fahrerzustands und Kollisionsvermeidung.

Der Name des Programms, PROMETHEUS, entstammt der griechischen Mythologie. Dort ist es der Name eines Titans, der den Menschen das Feuer und damit den Fortschritt bringt. Der für das Programm gewählte Name ist zugleich ein Akronym für „*Programme for a European Traffic with Highest Efficiency and Unprecedented Safety*“.

Im Herbst 1985 entschlossen sich die maßgeblichen Forschungsleiter der europäischen Automobilindustrie dazu, das PROMETHEUS Programm ins Leben zu rufen. Für die Dauer von acht Jahren wurden die Forschungsaktivitäten in anfangs 15 europäischen Automobilfirmen, über 70 externen Forschungsinstituten und etwa 200 Zulieferern sowie Universitäten koordiniert, um gemeinsam eine vorwettbewerbliche Technologieforschung zu betreiben.

In der Definitionsphase des Programms, vom 1. Oktober 1986 bis Ende 1987, wurden die Anforderungen an ein solches Projekt, die vorhandenen Technologien und der aktuelle Wissensstand untersucht. Anschließend wurden in einer einjährigen Startphase konkrete Forschungsprojekte festgelegt. Diese wurden von den europäischen Partnern während der Hauptphase von 1989 bis 1994 bearbeitet. Im Oktober 1994 wurden die Ergebnisse zum Projektabschluß in Paris demonstriert. Die PROMETHEUS-Forschung hat dabei einige neue Techniken für die Sicherheit des Autofahrens sowie Techniken für ein verbessertes Verkehrssystem-Management hervorgebracht.

Die Aufgabengebiete innerhalb des Programms umfaßten sieben Bereiche. Davon wurden die Bereiche PRO-CAR, PRO-NET und PRO-ROAD von den industriellen

¹ Die Zahlen gelten für Westeuropa im Jahr 1984 (vgl. [Franke 89]).

Partnern ausgeführt. Hierbei handelte es sich um fahrzeugorientierte Forschungs- und Entwicklungsaufgaben sowie um den Aufbau von Demonstrationsfahrzeugen.

Die vier weiteren Bereiche PRO-ART, PRO-GEN, PRO-CHIP und PRO-COM beinhalten größtenteils Themenbereiche der Grundlagenforschung und wurden hauptsächlich an Universitäten und Forschungsinstituten bearbeitet. Die einzelnen Bereiche können wie folgt skizziert werden:

- PRO-CAR** Entwicklung von Systemen im Fahrzeug, die den Fahrer unterstützen und entlasten sollen. Mit Rechnerunterstützung soll ein solches System den Fahrer vor Gefahren warnen und sogar selbst in die Fahrzeugführung eingreifen können.
- PRO-NET** Entwicklung von Systemen zur Kommunikation zwischen mehreren Fahrzeugen.
- PRO-ROAD** Entwicklung von Systemen zur Kommunikation des Fahrzeugs mit externen Informationssystemen, um ein optimales Verkehrsmanagement realisieren zu können.
- PRO-ART** Untersuchung der Einsatzmöglichkeiten der Künstlichen Intelligenz im Fahrzeug bzw. im Straßenverkehr.
- PRO-CHIP** Entwicklung geeigneter Mikroelektronik für den Einsatz im Fahrzeug.
- PRO-COM** Entwicklung von Standards für den Datenaustausch.
- PRO-GEN** Erarbeitung von Einführungsstrategien, Klärung von Rechts- und Bedarfsfragen bei der Einführung neuer Technologien in den Straßenverkehr.

In der bereichsübergreifenden Projektgruppe CED 3 (*Common European Demonstrators*) befaßte man sich mit der Fahrerunterstützung zur Vermeidung von Unfällen durch rechtzeitiges Erkennen von Hindernissen und kritischen Verkehrssituationen. Ein solches System unterstützt den Fahrer in Gefahrensituationen bzw. versucht, diese erst gar nicht entstehen zu lassen. Eine angemessene Unterstützung kann dabei aus Informationsbereitstellung, aber auch aus Intervention seitens des Rechners bestehen.

Bei der Konzeption des Versuchsfahrzeugs VITA II gingen die Forscher der Daimler-Benz AG mit ihren Partnern noch einen Schritt weiter. Dieses Fahrzeug wurde als Machbarkeitsstudie des autonomen Fahrens mittels Bildverarbeitung konstruiert.

Mit Hilfe von 18 Kameras ist das Fahrzeug in der Lage, Informationen über den Straßenverlauf, die Position und Geschwindigkeit anderer Fahrzeuge zu gewinnen, sowie Verkehrszeichen zu erkennen.

An dieser Stelle setzt die vorliegende Arbeit an. Die eingehenden Daten der Bildverarbeitung werden ausgewertet, um aus dieser Information Befehle für autonome Fahrmanöver zu ermitteln, die dann vom Fahrzeug ausgeführt werden. Das Fahrzeug wird dadurch in die Lage versetzt, im realen Autobahnverkehr ohne Eingriff des

menschlichen Fahrers auf der Straße zu fahren, Sicherheitsabstände einzuhalten und sogar selbständig zu überholen und Ausweichmanöver einzuleiten.

Auf der PROMETHEUS Abschlußdemonstration wurde der Daimler-Benz Versuchsträger VITA II schließlich erfolgreich im öffentlichen Verkehr auf der Autobahn vorgeführt.

Nähere Informationen zu PROMETHEUS sind [Zetsche 95, Beste 95, Franke 89] sowie [Sandewall et al. 90] zu entnehmen. Die Aktivitäten der Daimler-Benz AG sind zusätzlich in einer Sonderausgabe des High Tech Reports beschrieben [HighTechReport 94].

1.3 Gliederung und Überblick

Die hier vorgestellte Arbeit umfaßt die Konzeption und Realisierung eines Kontrollsystems für ein autonomes Fahrzeug.

Einleitend wird in Kapitel 2 zunächst der allgemeine Aufbau eines autonomen mobilen Fahrzeugs beschrieben und gleichzeitig eine Terminologie für die einzelnen Komponenten festgelegt. Anschließend wird der Aufbau des Versuchsträgers der Daimler-Benz AG, VITA II, beschrieben, der als Plattform für die hier entwickelten Algorithmen dient.

Kapitel 3 bildet den Hauptteil der Arbeit. Dieser gliedert sich in vier Themenbereiche.

- (1) *Gefahrenkarte und elektrisches Feld*
- (2) *Verhaltensmusterkonzept*
- (3) *Aufbau einer Situationsbeschreibung* und
- (4) *Simulation*

Nach Festlegung der Diskurswelt in Abschnitt 3.1 wird in Abschnitt 3.2 ein erster Ansatz zur Verhaltenssteuerung eines autonomen Fahrzeugs beschrieben, der auf der direkten Verknüpfung der Fahrzeugreaktion mit einer Gefahrenrepräsentation der Umwelt basiert. Die damit gewonnenen Erkenntnissen fließen in die anschließend konzipierte Methode der konkurrierenden Verhaltensmuster ein, die weiterhin eine Gefahrenrepräsentation verwendet, sich aber näher an der menschlichen Informationsverarbeitung orientiert. Diese Methode, die auch im realen Straßenverkehr Einsatz findet, wird in Abschnitt 3.3 diskutiert. Dieser Abschnitt stellt zugleich den Kern der vorliegenden Arbeit dar.

Um die Algorithmen zur Verhaltenssteuerung im realen Straßenverkehr einsetzen zu können, werden die fehlerbehafteten Daten der Sensormodule des Versuchsträgers VITA II aufbereitet. Die in Abschnitt 3.4 diskutierte Aufbereitungsmethode hat die Aufgabe, aus den eingehenden Sensorinformationen durch Datenfusion, Meßwertglättung und Extrapolation unter Nutzung von Hintergrundwissen über die Domäne

Straßenverkehr, eine Situationsbeschreibung aufzubauen. Diese Situationsbeschreibung dient den Verfahren zur Verhaltenssteuerung als Eingabe.

Die Verifikation der Algorithmen zur autonomen Fahrzeugführung stellt aufgrund des damit verbundenen Kostenaufwands, des Zeitaufwands, aber vor allem der Gefahren im realen Straßenverkehr ein nicht unwesentliches Problem dar. Die aus diesem Grund entwickelte Simulationsumgebung wird in Abschnitt 3.5 beschrieben. Sie umfaßt die Simulation der Sensormodule des Versuchsträgers VITA II sowie die Dynamik des Fahrzeugs und integriert die Fahrzeugsimulation in eine virtuelle Welt, in der beliebige Szenenfolgen zu Testzwecken generiert werden können.

Im Anschluß an den Hauptteil werden in Kapitel 4 Ergebnisse aus dem praktischen Einsatz der oben beschriebenen Methoden dargestellt.

Eine zusammenfassende Kritik und ein Vergleich zu anderen Verfahren sowie ein Ausblick auf mögliche Erweiterungen der Methoden und eine Betrachtung des Einsatzes von intelligenten Fahrerunterstützungssystemen im Straßenverkehr schließt die Arbeit ab.

2 Autonomes Fahren

Der Begriff „autonomes Fahren“ bedarf einer genaueren Abgrenzung. Der aus dem Griechischen stammende Begriff „autonom“ hat ursprünglich die Bedeutung „nach eigenem Gesetz lebend“. Zieht man diese Bedeutung zur Auslegung des Begriffs „autonomes Fahren“ heran, so wird sicherlich nicht die eigentliche Intention getroffen. Im allgemeinen Sprachgebrauch ist autonom mit *selbständig* und *unabhängig* gleichzusetzen. Autonomes Fahren bedeutet in unserem Kontext das selbständige, zielgerichtete Bewegen eines Fahrzeugs ohne Eingriff eines menschlichen Fahrers. Insbesondere wird durch den Begriff autonomes Fahren im Straßenverkehr eine aktive Infrastruktur ausgeschlossen. Das autonome Fahrzeug muß seine Eingangsdaten ausschließlich aus den Informationsquellen beziehen, die auch dem menschlichen Fahrer zur Verfügung stehen.

Entsprechend der an ein solches Fahrzeug gestellten Anforderungen bezüglich des Einsatzumfelds, der zu erledigenden Aufgaben und der Freiheitsgrade der Bewegung, kann man von dem Grad der Autonomie eines Systems sprechen. So wird in Veröffentlichungen zu diesem Themengebiet beispielsweise ein Fahrzeug, welches auf der Autobahn die Fahrspur erkennt und einen automatischen Lenkeingriff vornehmen kann, um auf dieser Fahrspur zu bleiben, bereits als autonom bezeichnet [Ulmer 92]. Einen weitaus höheren Grad der Autonomie erreicht das Daimler-Benz Versuchsfahrzeug VITA II. Es ist in der Lage neben der eigenen Fahrspur auch die Nachbarspuren, andere Verkehrsteilnehmer sowie Verkehrszeichen zu erkennen und auf seine Umgebung zu reagieren.

Im Folgenden wird zunächst der typische Aufbau eines autonomen mobilen Systems beschrieben. Anschließend werden die Anforderungen an ein autonomes Fahrzeug für den Straßenverkehr diskutiert, bevor VITA II als konkretes Beispiel eines solchen autonomen Fahrzeugs beschrieben wird.

2.1 Aufbau eines autonomen mobilen Systems

Rechnergesteuerte Fahrzeuge, die sich selbständig in teilweise unbekannter Umgebung bewegen und Aufgaben erfüllen können, für die aufgrund ihrer Komplexität kein a priori festgelegter Handlungsablauf programmiert werden kann, bezeichnet man als autonome mobile Systeme. Autonome mobile Systeme oder autonome mobile Roboter lassen sich in Landfahrzeuge, Wasserfahrzeuge und Luftfahrzeuge unterteilen, wobei im Folgenden nur die Landfahrzeuge betrachtet werden. Ein solches autonomes mobiles System besteht im wesentlichen aus drei Komponenten: dem *Sensorsystem* zur Erfassung der Umgebung,

dem *Aktorsystem* zur Fortbewegung des Fahrzeugs und dem *Kontrollsystem* zur Auswertung der aktuellen Umgebungsinformation in Bezug auf die vorgegebene Aufgabe und zur Aktionssteuerung [Knieriemen 91].

Neben diesen drei Hauptkomponenten wird eine *Kommunikationsstruktur* der einzelnen Komponenten untereinander sowie zur Verbindung mit der Außenwelt benötigt. Desweiteren benötigt das System eine eigene Energieversorgung und je nach Applikation verschiedene Aufbauten zur Erfüllung der gestellten Aufgaben. Die nachfolgende Beschreibung der einzelnen Komponenten eines autonomen mobilen Systems beschränkt sich auf Landfahrzeuge, deren Aufgabengebiet in der Kollisionsvermeidung und Navigation in unbekannter oder variierender Umgebung besteht.

2.1.1 Das Sensorsystem

Ein Roboter, der in teils unbekannter Umgebung selbständig agieren soll, muß seine Umgebung erkennen und interpretieren können. Dem Menschen stehen dazu seine Sinnesorgane zur Verfügung, so daß er seine Umwelt durch Sehen, Hören, Riechen und Fühlen wahrnehmen kann. Um einen autonomen mobilen Roboter mit einer Wahrnehmungskomponente auszustatten, wird der Mensch oft als Vorbild genommen. Die in einem autonomen mobilen System verwendeten Meßeinrichtungen lassen sich in *interne* und *externe* Sensoren gliedern.

Interne Sensoren werden zur Erfassung des Systemzustands benutzt, welcher unabhängig von Umgebungsinformationen ist. Zu nennen sind hier Neigungsmesser zur Bestimmung der Lage des Systems im Raum, sowie Beschleunigungsmesser und Drehsensoren (Kreiselinstrumente) zur Feststellung von einwirkenden Kräften. Desweiteren werden Sensoren benötigt, die Zustandsgrößen wie die aktuelle Geschwindigkeit, den Lenkwinkel oder Kenngrößen der Antriebskomponente erfassen. Eine wichtige Aufgabe der internen Sensoren ist die Überwachung von Systemkomponenten, um Fehlfunktionen erkennen und gegebenenfalls entsprechende Maßnahmen einleiten zu können

Externe Sensoren dienen zur Erfassung der Umgebung des autonomen mobilen Systems. Man unterscheidet hier zwischen *taktilen* und *berührungslosen* Sensoren. Taktile Sensoren stellen den unmittelbaren Kontakt mit Objekten fest.

Zur Messung von Distanzen werden hauptsächlich aktive Meßgeräte eingesetzt, aber auch bilderzeugende, passive Meßgeräte wie Kameras sind zu diesem Zweck einsetzbar.

Mit Hilfe von *Ultraschall* kann die Entfernung von Objekten bestimmt werden. Dazu wird ein Ultraschallimpuls gesendet, der vom Objekt reflektiert und anschließend wieder empfangen wird. Durch Laufzeitmessung wird die Entfernung des Objekts

bestimmt. Probleme entstehen bei dieser Methode durch Mehrfachreflexion an verschiedenen Objekten, sowie durch Ablenkung des Echos an Objekten, deren Oberfläche nicht senkrecht zur Strahlrichtung steht.

Mit Hilfe eines *Lasers* und eines Fotodetektors können Entfernungen ähnlich der Ultraschallmethode gemessen werden. Neben der Laufzeitmethode wird das Triangulationsprinzip zur Entfernungsmessung auf kürzeren Distanzen genutzt. Dazu werden ein Laser und eine Linse sowie eine CCD-Zeilenkamera in einem festen Abstand und einem festen Winkel zueinander angebracht. Durch Messung der Abweichung des reflektierten Strahls wird die Entfernung des Objekts berechnet.

Eine weitere Methode zur Detektion von Hindernissen stellt das *Millimeterwellen-Radar* dar. Mit Hilfe eines Hohlspiegels wird das Radarsignal gebündelt und ähnlich der Ultraschallmethode ausgewertet. Diese Methode liefert sehr zuverlässige Entfernungsangaben zu Hindernissen in größeren Entfernungen.

Benutzt man ausschließlich passive Sensoren wie CCD-Kameras, so kann man Entfernungen mit Hilfe der Korrelation von Kameraaufnahmen berechnen. Auch hier wird die Triangulationsmethode genutzt, wobei jetzt zwei in festem Abstand und mit festgelegten Winkeln installierte Kameras als Basis dienen und Bildpunkte oder Bitmuster auf den Kamerabildern als Referenz herangezogen werden.

Eine weitere Methode ist die modellbasierte Bildauswertung. Hier wird davon ausgegangen, daß Hindernisformen bekannt sind und im Kamerabild wiedererkannt werden können. Bei bekannter Hindernisgröße wird dann durch die Größe des erkannten Objekts im Bild die Entfernung bestimmt. Eine weitere Methode zur Abstandsbestimmung von Objekten ist die Nutzung der Information über die Anbringungs- und Öffnungswinkel der Kamera. Mit diesem Wissen läßt sich dann die Entfernung unter Annahme einer ebenen Grundfläche anhand der Höhe des erkannten Objekts im Bild bestimmen.

Um die durch Sensoren erfaßte Umgebungsinformation nutzen zu können, werden die gewonnenen Informationen aufbereitet. Dabei gilt es beispielsweise, Meßfehler der Sensorik durch geeignete Filtersysteme zu unterdrücken, die Datenmengen durch Kompressionsmethoden zu verringern und bestimmte Merkmale wie Linien, Objektkanten oder Flächen aus Kamerabildern zu extrahieren.

Die meisten Robotersysteme verfügen über mehrere verschiedene Sensorsysteme, die ihre Stärken in unterschiedlichen Bereichen haben. Beispielsweise ist die Kamera-bildauswertung geeignet zur Objektidentifikation, während Millimeterwellen-Radar zur Entfernungsmessung besser geeignet ist. Durch Datenfusion werden die Vorteile

beider Verfahren kombiniert. Detailliertere Informationen sind in [Knieriemen 91] und [Puttkamer 90] nachzuschlagen.

Das Sensorsystem umfaßt (in der hier verwendeten Terminologie) die physikalischen Sensoren sowie die ihnen zugeordneten Verfahren zur Sensordatenverarbeitung. Dies umfaßt beispielsweise ein Kamerasystem in Verbindung mit einem Bildverarbeitungsprogramm, welches auf dem Kamerabild Hindernisse erkennt. Die Ausgaben eines solchen *logischen Sensors* sind dann je nach Spezifikation Attribute des detektierten Objekts wie beispielsweise dessen Distanz vom Beobachter oder dessen Relativgeschwindigkeit.

2.1.2 Das Aktorsystem

Das Aktorsystem ist dem Aufgabenbereich angepaßt, für den das autonome mobile System konstruiert wird. In unebenen, unregelmäßigen Geländestrukturen kommen Gehmaschinen oder Raupen zum Einsatz. Diese Maschinen werden in den verschiedensten Ausführungen konstruiert, so z.B. mit einem bis acht Beinen, mehreren Gelenken und sehr komplexen Steuerungs- und Koordinationsmechanismen konstruiert.

Der Großteil der autonomen mobilen Systeme ist jedoch radgetrieben. Auch hier variiert die Anzahl der Räder, meist sind es jedoch drei oder vier. Weiterhin unterscheidet man zwischen angetriebenen, freilaufenden und gelenkten Rädern. Dabei kann eine Steuerung des Fahrzeugs sowohl durch explizite Drehung eines oder mehrerer Räder, aber auch implizit durch Gegeneinanderdrehen von Antriebsrädern erfolgen.

Zur Ansteuerung der Aktoren werden impulsgetriebene Schrittmotoren, drehzahlregulierbare Gleichstrommotoren und Verbrennungsmotoren unterschieden. Während Schrittmotoren bei jedem Impuls um einen konstanten Winkel weiterschalten, ermöglicht die Impulsbreitensteuerung bei Gleichstrommotoren eine kontinuierliche Regelung, beispielsweise des Lenkwinkels eines Fahrzeugs.

2.1.3 Das Kontrollsystem

Die Aufgaben des Kontrollsystems unterteilen sich in die *Sensordatenverarbeitung* und die *Aktionssteuerung*. Die Sensordatenverarbeitung dient der Erkennung von Hindernissen, der Bestimmung der Position des autonomen mobilen Systems sowie dem Aufbau einer internen Repräsentation der Umwelt. Die Aktionssteuerung interpretiert die von außen vorgegebene Aufgabenstellung, plant die zur Erfüllung dieser Aufgabe nötigen Bewegungen und überwacht die Ausführung der kommandierten Aktionen.

Die klassische Architektur eines autonomen mobilen Roboters ist hierarchisch aufgebaut. In konkreten Realisierungen von autonomen mobilen Systemen existieren mehrere Varianten horizontaler Kontrollstrukturen, exemplarisch wird hier die Struktur des Kontrollsystems des MOBOT-III Systems beschrieben [Puttkamer & Trieb 91, Puttkamer 90].

Die oberste Schicht bildet die *Planungsebene*. Diese Ebene verfügt über eine Gesamtkarte der Umgebung und plant anhand dieser Information Wege für das autonome mobile System. Der generierte Wegeplan wird an die darunterliegende Ebene, den *Navigator*, weitergegeben. Dieser kennt wesentlich weniger von der Gesamtumgebung, verfügt jedoch über detailliertere Umgebungsinformationen. Im Gegensatz zum zeitunkritischen Planer, dessen Berechnungen auch a priori angestellt werden können, muß der Navigator in Echtzeit laufen, um adäquat auf Hindernisse und dynamische Veränderungen der Umgebung reagieren zu können. Der Navigator übergibt die gewünschte Fahrtrichtung und Geschwindigkeit an den unterlagerten *Piloten*. Dieser sorgt für einen kontinuierlichen Bewegungsablauf und beherrscht zusätzlich grundlegende Fahrmanöver, zu deren Ausführung die übergeordneten Ebenen nicht benötigt werden.

Das zur Planung und Navigation benötigte Weltmodell ist eng mit den verwendeten Algorithmen gekoppelt. Man unterscheidet topologische, semantische und geometrische Modelle. Das topologische Modell beschreibt die Umwelt auf abstraktem Niveau anhand von Lagerrelationen und der Anordnung von Objekten. Dies kann geeignet durch Graphenstrukturen dargestellt und zur Wegeplanung verwendet werden. Das geometrische Modell wird direkt aus den Sensorinformationen gewonnen und zur lokalen Wegeplanung verwendet. Hindernisse lassen sich hier durch Polygone repräsentieren, deren Darstellung unter anderem von der Bewegungsrichtung und der Form des autonomen mobilen Systems abhängt [Lozano-Pérez & Wesley 79].

Im Gegensatz zu den klassischen Kontrollstrukturen unterscheidet Brooks keine Funktionsebenen, sondern Verhaltensebenen [Brooks 86]. Die Verhaltensebenen bestehen aus einer Menge von modellierten Verhaltensweisen, die jeweils eine bestimmte Reaktion auf die Umwelt festlegen. Durch diese Architektur werden eine hohe Parallelisierbarkeit und Fehlertoleranz erreicht.

2.1.4 Das Kommunikationssystem

Das Kommunikationssystem dient dem Datenaustausch des autonomen Systems mit der Außenwelt sowie der Kommunikation einzelner interner Systemkomponenten untereinander.

Eine Schnittstelle zur Außenwelt erlaubt es einem Programmierer, das System mit Programmen und Parametern zu versorgen, ihm eine Aufgabe zu geben oder auch das System per Hand zu steuern. Betriebszustände, Sensordaten und Karten können über diese Schnittstelle ausgelesen und zur Auswertung graphisch dargestellt werden.

Interne Kommunikationsstrukturen sind von der Systemarchitektur abhängig. Sie gewinnen an Bedeutung, sobald das System einen höheren Grad an Parallelität erreicht. Neben Punkt-zu-Punkt Verbindungen einzelner Prozesse sowie Broadcasting Kommunikation (gleichzeitiges Senden einer Nachricht an mehrere Empfänger) kommen insbesondere bei komplexen Systemen auch zentrale, gemeinsam verwendete Datenspeicher zum Einsatz.

2.2 VITA II als Plattform für autonomes Fahren

Das von der Daimler-Benz AG im Rahmen des PROMETHEUS Projekts aufgebaute autonome Fahrzeug VITA II (*Vision Technology Application*) bildet die Plattform für die in dieser Arbeit vorgestellten Methoden der autonomen Fahrzeugführung. Deshalb soll an dieser Stelle näher auf den Aufbau und die einzelnen Komponenten des Fahrzeugs eingegangen werden. In diesem Abschnitt wird ein kurzer Überblick über die in VITA II zur Verfügung stehenden Informationen, sowie über die Architektur des Systems gegeben, um die Rahmenbedingungen für das im Hauptteil der Arbeit beschriebene Kontrollsystem zu verdeutlichen (vgl. auch [Ulmer 94a, Ulmer 94b]).



Abbildung 1 : Der Daimler-Benz Versuchsträger VITA II

Der Versuchsträger VITA II ist eine *Machbarkeitsstudie* für autonomes Fahren mit Hilfe der Bildverarbeitung. Als externe Sensoren dienen aus diesem Grund ausschließlich

CCD-Kameras. Um eine Rundumsicht zu gewährleisten, sind insgesamt 18 Kameras an den Seiten und im Inneren des Fahrzeugs angebracht. Das Gas, die Bremse und die Lenkung des Fahrzeugs, eines Mercedes 500 SEL, sind elektronisch ansteuerbar und erlauben somit die rechnergestützte Ausführung autonomer Fahrmanöver.

Die zu bewältigenden Fahraufgaben für VITA II sind:

- *Spurhaltung*
- *Kommandierter automatischer Spurwechsel*
- *Abstandhalten zu vorausfahrenden Fahrzeugen*
- *Einhalten allgemeiner Verkehrsregeln*
- *Beachten von Verkehrszeichen*
- *Kollisionsvermeidung*
- *Selbständiges Überholen*

Die zur Bewältigung dieser Aufgaben benötigten Sensorinformationen und Datenströme in VITA II stehen im Vordergrund der nachfolgenden Beschreibung des Versuchsträgers. Die Rechnerhardware, ein Transputersystem bestehend aus über 70 Prozessoren, befindet sich im Kofferraum des Fahrzeugs. Als Benutzerschnittstelle dient der serienmäßige Tempomat sowie der Blinkerhebel, den der Fahrer zum Kommandieren eines Spurwechsels benutzen kann. Während der Fahrt können Rechnerbilder und Statusinformationen über Monitore in der Mittelkonsole und in den hinteren Kosmetikspiegeln dargestellt werden. Im Folgenden wird die Hard- und Softwarearchitektur des autonomen Fahrzeugs beschrieben, bevor näher auf das Sensorsystem, das Kontrollsystem und das Aktorsystem eingegangen wird.

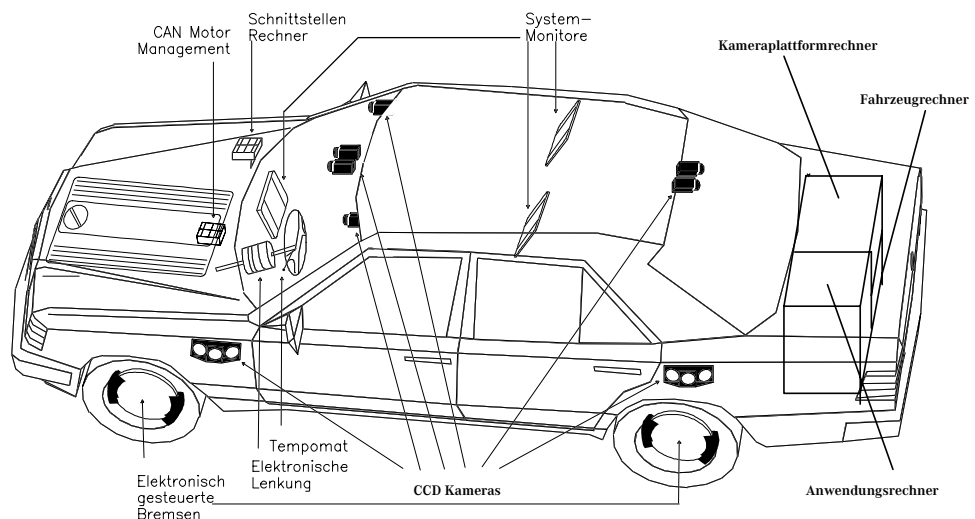


Abbildung 2 : Aufbau des autonomen Fahrzeugs VITA II

2.2.1 Hardware- und Softwarearchitektur

Die VITA II Hardware ist modular aufgebaut. Das Rechnersystem besteht aus drei voneinander unabhängigen Rechereinheiten:

- dem *Anwendungsrechner*, der für die Bildverarbeitung sowie für die Verhaltenssteuerung des Fahrzeugs zuständig ist;
- dem *Fahrzeugrechner*, der die unterlagerte Regelung für Gas, Bremse und Lenkung beherbergt und die Daten der internen Sensorik zur Verfügung stellt;
- dem *Plattformrechner*, der die Ansteuerung der beweglichen Kamera-Plattformen erledigt.

Die Basis der Rechner bildet jeweils ein Transputersystem¹. Die Verknüpfung der drei Rechner wird mittels Transputerlinks und eines festgelegten Kommunikationsprotokolls realisiert. Als Host für die Transputersysteme wird eine SUN SPARC Karte auf VME Basis eingesetzt, die ebenfalls im Kofferraum des Versuchsträgers eingebaut ist. Software-Erstellung und Konfiguration erfolgen auf dem Hostrechner bzw. auf einem über Ethernet mit diesem verbundenen zweiten Rechner.

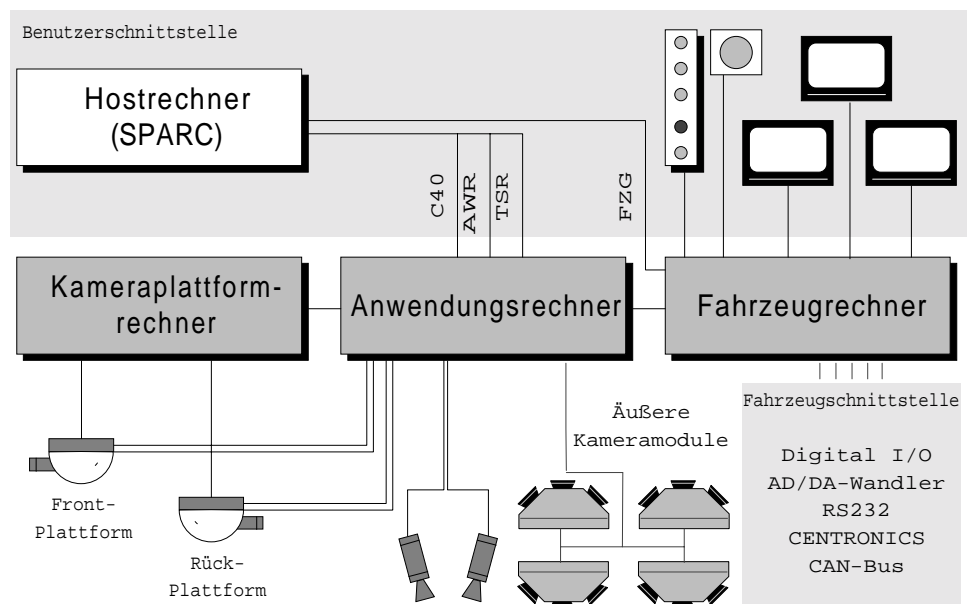


Abbildung 3 : Rechnerarchitektur von VITA II

2.2.1.1 Anwendungsrechner

Der Anwendungsrechner bildet die Plattform für die Bildverarbeitungsprozesse sowie für das Kontrollsystem des Fahrzeugs. Die Hardwarebasis des Anwendungsrechners ist ein Transputernetzwerk. In dieses Netzwerk sind MPC 601 Prozessoren integriert, deren Leistungsfähigkeit weit über der von derzeit erhältlichen Transputern liegt. Auf diesen Prozessoren laufen die Prozesse zur Verkehrszeichen-

¹ Nähere Ausführungen zu Transputern, deren Aufbau und Programmierung, sind im Anhang A nachzuschlagen.

erkennung. Über eine Dual Ported RAM Verbindung ist ein weiteres Prozessor-
netzwerk auf Basis von TMS320 C40 Signalprozessoren integriert. Dieses Subnetz
wird von den Applikationen zur lateralen Hinderniserkennung (Lateral Obstacle
Detection), sowie durch zwei Subsysteme zur Hinderniserkennung nach vorne
(Vision Bumper, CarTrack) verwendet. Es enthält neben acht C40 Prozessoren auch
Framegrabber und einen eigenen Hostzugang, so daß es unabhängig vom Gesamt-
system lauffähig ist.

Für die Bildverarbeitungsprozesse auf Transputer wird eine spezielle schnelle
Busstruktur, der TIP (*Transputer Image Processing*) Bus, benutzt. Diese Hardware
erlaubt die Verschiebung von größeren Datenblöcken, in diesem Fall digitalisierte
Bilder, zwischen verschiedenen Transputerknoten mit einer wesentlich höheren
Geschwindigkeit, als dies über Transputerlinks zu erreichen wäre. Dadurch wird die
parallele Verarbeitung von Bildern in Echtzeit ermöglicht.

Ein in der Daimler-Benz Forschung entwickeltes Werkzeug zur Software-
generierung (TRAPPER) ermöglicht die Beherrschung des komplexen Netzwerks
von über 200 Prozessen, die auf den 70 Prozessoren ablaufen. TRAPPER bietet eine
graphische Oberfläche zur Konfiguration von Transputernetzwerken auf
Hardwareebene und Softwareebene [Schäfers et al. 92].

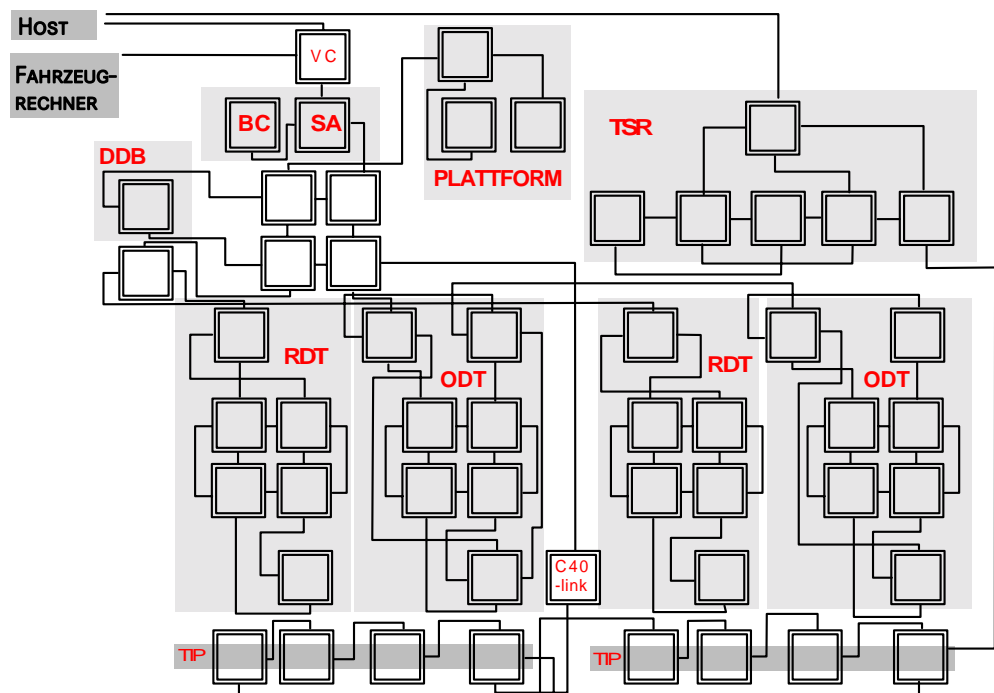


Abbildung 4 : Prozessornetzwerk des Anwendungsrechners

Die Software ist in Module unterteilt, die jeweils eine ihnen zugeordnete Aufgabe
erfüllen. Jedes Modul besteht aus mehreren parallel ausführbaren Prozessen, die auf
Prozessorgruppen des Netzwerks abgebildet werden. Innerhalb einer solchen

Prozessorgruppe ist die Kommunikationsform dem jeweiligen Modul überlassen. Um die Daten flexibel handhaben zu können, wird eine gemeinsame dynamische Datenbasis (DDB) eingeführt, mit der jedes Modul kommuniziert [Dickmanns 93a]. Diese Datenbank verwaltet den aktuellen Systemzustand. Es bestehen also keine direkten Verbindungen zwischen den Applikationsmodulen. Zur Kommunikation mit dem Fahrzeugrechner ist ein Steuerungsprozeß (VC - *Vehicle Control*) an diese gemeinsame Datenbasis angeschlossen.

2.2.1.2 Fahrzeugrechner

Der Fahrzeugrechner beinhaltet sämtliche Prozesse zur Ansteuerung des Fahrzeugs sowie zum Auslesen der internen Sensoren. Durch diesen modularen Aufbau ist es möglich, den Rechner mit wenigen Modifikationen in anderen Fahrzeugen mit anderen Applikationen einzusetzen.

Zur Überwachung der einwandfreien Funktion der einzelnen Komponenten werden Hardware-*"Watchdogs"* eingesetzt, die bei Fehlfunktionen warnen und einzelne Komponenten, wie beispielsweise die Lenkungsansteuerung, abschalten können.

Die Rechnerhardware besteht aus fünf Transputern des Typs T805 und weiteren vier Transputern des Typs T222. Zudem befinden sich in dem Fahrzeugrechner Analog-Digital Wandler und weitere Schnittstellenkarten zur Ansteuerung der Aktoren.

Auf dieser Hardware laufen mehrere Kommunikations- und Steuerungsprozesse

parallel ab. Die Verbindung zum Anwendungsrechner wird durch einen Kopplungsprozeß realisiert, der nach einem festen Protokoll im 40-Millisekunden-Takt Daten austauscht. Ein zentraler Steuerungsprozeß koordiniert den Zugriff auf die Fahrzeugbedienung und verteilt Daten an die Prozesse zur Längs- und Querregelung und gibt Informationen der internen Sensoren (Lenkwinkel, Geschwindigkeit, usw.) an den Kopplungsprozeß weiter. Weiterführende

Informationen zum Aufbau und zur Funktion des Fahrzeugrechners sind in [Sauerwein 94] nachzulesen.

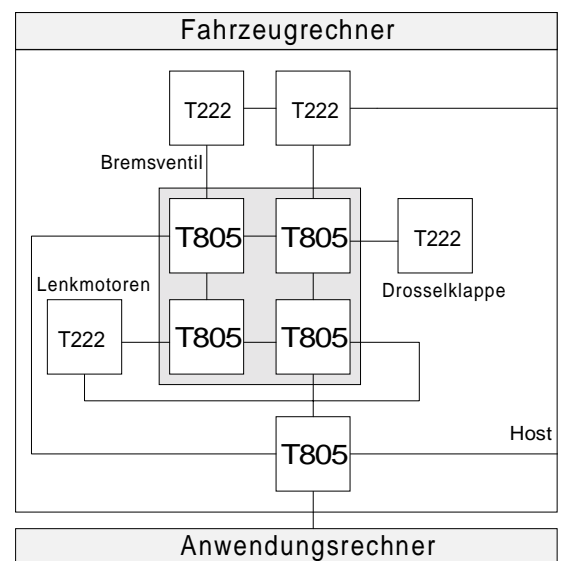


Abbildung 5 : Prozessornetzwerk des Fahrzeugrechners

2.2.1.3 Plattformrechner

Im Fahrzeuginnenraum sind vorne hinter der Windschutzscheibe sowie vor der Heckscheibe motorisch angetriebene Kameraplattformen angebracht. Der Anwendungsrechner versorgt den Plattformrechner mit Vorgaben für die Winkelstellungen der Plattform. Aufgabe des Plattformrechners ist es, diese Vorgaben einzuregeln. Durch Drehen der Kameraplattform kann ein Sichtbereich von ca. 120 Grad vor und hinter dem Fahrzeug erreicht werden. Die benötigte Hardware besteht aus einem Transputerknoten des Typs T805 sowie zwei Transputerknoten des Typs T222 zum Einlesen der aktuellen Winkelpositionen der Plattform und zur Ausgabe der Motorenmomente. Nähere Informationen zur beweglichen Kameraplattform sind [Schielen & Dickmanns 94] zu entnehmen.

2.2.2 Das Sensorsystem

Das Sensorsystem ist unterteilt in Subsysteme zur Erfassung von *internen* und *externen* Daten. Fahrzeuginterne Daten werden über den serienmäßigen CAN-Bus (Controller Area Network) zur Verfügung gestellt. Die elektronisch ansteuerbare Lenkung ist mit einem Lenkwinkelgeber ausgestattet, über den sich der aktuelle Lenkradwinkel, sowie Lenkrate und Lenkwinkel des Fahrzeugs ermitteln lassen. Diese internen Daten werden den Längs- und Querreglern im Fahrzeugrechner zur Verfügung gestellt. Über Beschleunigungsmesser werden die auf das Fahrzeug wirkenden Kräfte gemessen und zur Verbesserung der Regelung benutzt.

Um die gestellte Aufgabe zu realisieren und das Fahrzeug selbständig auf der Autobahn fahren zu lassen, müssen die Sensoren die Umgebung in Echtzeit und in ausreichendem Umfang erfassen und auswerten. Dazu gehört das Erkennen von Hindernissen rings um das Fahrzeug, das Erkennen der Straße und das Erkennen von Verkehrszeichen.

Mit Hilfe der in VITA II integrierten optischen Sensoren wird der Bereich rund um das Fahrzeug abgedeckt und durch Bildverarbeitungsmodule analysiert. Dazu stehen vier verschiedene Verfahren zur Hinderniserkennung, ein Bildverarbeitungsmodul zur Straßenerkennung und eines zur Verkehrszeichenerkennung zur Verfügung.

Alle Sensormodule arbeiten in Echtzeit. Ihre Zykluszeiten liegen, abhängig von der Konfiguration, zwischen 40 und 120 Millisekunden.

2.2.2.1 Straßenerkennung

Das Modul zur Straßenerkennung (*RDT - Road Detection and Tracking*) [Behringer 94, Behringer 95] nimmt unter den Sensormodulen eine Sonderstellung ein, da es als Grundlage für das benutzte Koordinatensystem des Fahrzeugs dient (siehe Abschnitt 2.2.5). Es liefert die zur Spurhaltung eines autonomen Fahrzeugs

notwendigen Zustandsinformationen. Einerseits wird die Lage des Fahrzeugs relativ zur Fahrspur ermittelt (Ablage zur Fahrspurmitte, Gier- und Schwimmwinkel), andererseits wird auch der Verlauf der Fahrspur selbst beschrieben. Das Modul verwaltet dazu ein internes Krümmungsmodell der Fahrspur in horizontaler und vertikaler Richtung.

Zur Erkennung der Fahrspur werden je zwei Kameras mit Blickrichtung nach vorne und nach hinten benutzt. Durch die Montage der Kamerapaare auf beweglichen Plattformen können diese während der Fahrt immer dem Straßenverlauf folgen und

so die Vermessung der Straße, insbesondere in Kurven, verbessern. Durch die gleichzeitige Benutzung zweier Kameras mit verschiedenen Brennweiten kann ein Bereich von bis zu 100 Metern vor und hinter dem Fahrzeug abgedeckt werden. Die benötigte Rechnerhardware besteht aus drei Transputern des Typs T805 sowie acht Transputern des Typs

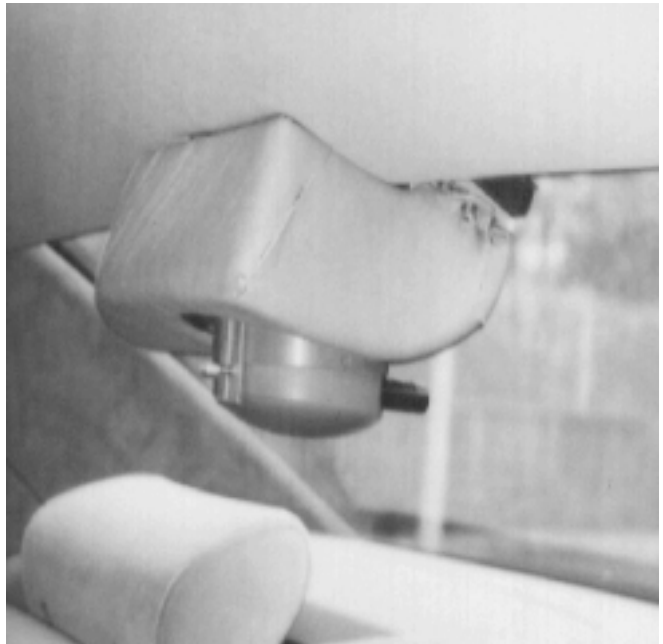


Abbildung 6 : Rückwärtige Kameraplattform in VITA II

T222, die durch das unterlagerte KRONOS Modul zur Merkmalsextraktion aus vier Videobildern belegt werden [Dickmanns 93b]. Mit diesen Hardwarevoraussetzungen wird eine Zykluszeit von 40 Millisekunden erreicht. Das unterlagerte KRONOS Modul liefert die Lage und Orientierung von Kanten im Bild mittels Maskenkorrelation in vorbestimmten Regionen des Videobildes. Mehrere Klienten des Road Tracker Moduls haben Zugriff auf das KRONOS System. Dabei handelt es sich um Detektions- und Verfolgungsprozesse für Straßenmarkierungen im zweidimensionalen Bild. Aufbauend auf diesen Informationen wertet ein 4D Schätzprozeß die 2D Daten unter Verwendung eines räumlich-zeitlichen Modells aus und führt eine rekursive Schätzung der Zustandsvektoren des Straßenmodells sowie des Fahrzeugmodells durch.

Durch das interne räumlich-zeitliche Modell des Fahrzeugs und der Fahrbahn wird dem Straßenverfolgungsmodul eine große Robustheit verliehen, die sich in der ruhigen Querführung des Fahrzeugs zeigt. Die nachfolgende Tabelle zeigt die gelieferten Daten.

Modul	Bezeichnung	Sichtbereiche	Gelieferte Daten
RT	Road Tracker	15-40 Meter vor dem Fahrzeug (bei 2 Kameras bis 100 Meter)	Spurbreite Horizontale Straßenkrümmung Krümmungsänderung der Straße Vertikale Straßenkrümmung Maximale Sichtweite Linke Spur existiert (Flag) Rechte Spur existiert (Flag) Spurwechselzähler Gierwinkel des Fahrzeugs

Tabelle 1: Das Straßenerkennungsmodul von VITA II

2.2.2.2 Hinderniserkennung

VITA II verfügt über vier Bildverarbeitungsmodule zur Hinderniserkennung, die mit unterschiedlichen Verfahren Hindernisse in teilweise überlappenden Sichtbereichen rund um das Fahrzeug erkennen. Als Ergebnisse werden die Distanz zum autonomen Fahrzeug, sowie die Geschwindigkeit und die seitliche Ablage des Fahrzeugs ermittelt. Während die Module ODT (Obstacle Detection and Tracking und CT (CarTrack) für die Verfolgung von Fahrzeugen ausgelegt sind, ist das Modul VB (Vision Bumper) als „optische Stoßstange“ zu sehen, die Hindernisse in unmittelbarer Nähe vor dem Fahrzeug mit Hilfe eines Stereoverfahrens detektiert. Aufgabe des Moduls LOD (Lateral Obstacle Detection) ist die Überwachung der Bereiche seitlich neben dem Fahrzeug. Abbildung 7 zeigt schematisch die Kameraanordnung von VITA II und die abgedeckten Bereiche.

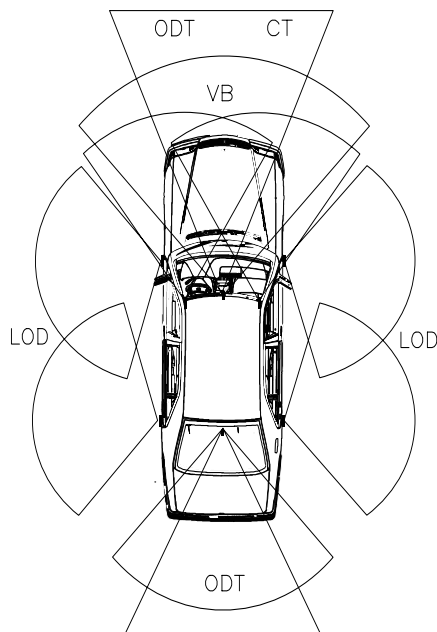


Abbildung 7 : Kamerablickwinkel von VITA II

ODT - Obstacle Detection and Tracking

Das Hinderniserkennungsmodul ODT (*Obstacle Detection and Tracking*) [Thomanek & Dickmanns 92, Thomanek et al. 94] verwendet dieselben Kameras wie das System zur Straßenerkennung. Durch die verwendeten Kameras wird ein Bereich von 100 Metern vor und hinter dem Fahrzeug abgedeckt. ODT ist in der Lage bei einer Zykluszeit von 80 Millisekunden gleichzeitig 10 Objekte zu verfolgen. Hardwareanforderungen dazu sind vier T805 Prozessoren und acht T222 Prozessoren, die für das unterlagerte Merkmalsextraktionsmodul KRONOS benötigt werden.

Zur Hinderniserkennung verwendet ODT 30 vertikale Suchfenster, die etwa ein Drittel des Bildes abdecken. Aus bis zu 180 darin erkannten Kanten werden bis zu 20 Konturen rekonstruiert. Wissen über die dreidimensionale Geometrie wird genutzt, um die extrahierten Konturen auf Kandidaten für Fahrzeuge hin zu untersuchen. Hauptmerkmal ist dabei der Schatten unterhalb des Fahrzeugs, der ein sehr signifikantes Merkmal zur Erkennung von Fahrzeugen darstellt. Zusätzlich werden die linke und rechte Fahrzeugbegrenzung bestimmt, um damit Fehldetektionen einzuschränken.

Mit Hilfe eines dynamischen Zustandsmodells können gute Schätzungen für den Abstand, die relative Ablage von Fahrzeugen zur eigenen Fahrspur sowie für die Geschwindigkeit der erkannten Fahrzeuge errechnet werden.

CarTrack

CarTrack [Brauckmann et al. 94] ist ein echtzeitfähiges, bifokales Bildverarbeitungsmodul zur Erkennung und Verfolgung von bis zu drei Fahrzeugen. Die erkannten Fahrzeuge werden nach dem Fahrzeugtyp klassifiziert. Es ist ausgelegt für den fließenden Verkehr auf Autobahnen und erkennt Fahrzeuge in einem Bereich von fünf bis 80 Metern. Ein erkanntes Objekt kann bis zu einer Entfernung von 100 Metern im Bild verfolgt werden. Das Hinderniserkennungsmodul CarTrack benutzt das Kamerapaar auf der vorderen Kameraplattform. Im Gegensatz zu dem Hauptsystem des VITA II Fahrzeugs wird ein Prozessornetzwerk bestehend aus C40 Signalprozessoren verwendet, welches über eine Dual Ported Memory Verbindung mit der Transputerhardware verknüpft ist. CarTrack läuft situationsabhängig alternativ mit den nachfolgend beschriebenen Stereoverfahren auf insgesamt acht C40 Prozessoren.

Eine der von diesem Modul verwendeten Methoden beruht auf der Nutzung von Symmetrieeigenschaften der rückwärtigen Ansicht von Fahrzeugen. Die zweite Methode ist ein Neuronales Netzwerk, welches Fahrzeuge als PKW oder LKW klassifizieren kann. Durch Bestimmung der Größe des verfolgten Objekts im Bild zieht CarTrack Rückschlüsse auf dessen Entfernung und kann somit mittels zeitlicher Ableitung auch dessen Relativgeschwindigkeit bestimmen.

VisionBumper

Das VisionBumper Modul [Brauckmann et al. 94, Bohrer et al. 95] dient als "erweiterte optische Stoßstange" des Fahrzeugs.

Es ist ein stereobasiertes Verfahren zur Detektion von Objekten im Nahbereich vor dem autonomen Fahrzeug und ausgelegt für Warnfunktionen im Stadtverkehr sowie Stop-And-Go Szenarien. Im Gegensatz zu den oben beschriebenen modellbasierten Ansätzen zur Hinderniserkennung, detektiert VisionBumper jede Erhebung, die sich im Nahbereich vor dem Fahrzeug befindet. Das System nutzt



Abbildung 8 : VisionBumper Kamera in VITA II

Wissen über die perspektivische Projektion einer ebenen Grundfläche, die gleichzeitig von zwei Kameras aus gesehen wird. Eine auf beiden Kameras ausgeführte invers-perspektivische Abbildung wird verglichen und Unterschiede werden als mögliche Hindernisse interpretiert.

LOD - Lateral Obstacle Detection

Zur Abdeckung der toten Winkel an den Seiten des Fahrzeugs dient das Modul LOD (*Lateral Obstacle Detection*) [Brauckmann et al. 94]. Der Blick zur Seite während der Fahrt stellt, aufgrund der hohen Geschwindigkeiten im Bild, hohe Anforderungen an die Bildverarbeitung. Speziell für diese Anwendung wurden von der C-VIS GmbH Fächerkamas entwickelt, die drei CCD-Kameras in einem Gehäuse integrieren. Vom Prinzip her verwendet LOD die gleiche Methode wie VisionBumper. In jeder Fächerkamera ist



Abbildung 9 : Fächerkamera des LOD-Moduls

jeweils eine Kamera aktiv, d.h. ein Kamerabild wird zur Auswertung weitergeleitet. Durch Kombination verschiedener Kamerapaare kann ein großer Bereich an den Seiten abgedeckt werden.

Modul	Bezeichnung	Sichtbereiche	Objekte	Gelieferte Daten
ODT	Obstacle Detection & Tracking	ca. 10-100 Meter vor und hinter dem Fahrzeug	10	Identifikationsnummer für ein Objekt Ablage des Objekts von der Fahrspurmitte Distanz des Objekts Objektbreite, -höhe Objektgeschwindigkeit Spurzähler
CT	Car Track	ca. 10-100 Meter vor dem Fahrzeug	3	Identifikationsnummer für ein Objekt Ablage des Objekts von der Fahrzeugmitte Distanz des Objekts Objektbreite,-höhe,-länge Objektgeschwindigkeit Objekttyp (PKW,LKW)
VB	Vision Bumper	ca. 5-35 Meter vor dem Fahrzeug	2	Identifikationsnummer für ein Objekt Ablage des Objekts von der Fahrspurmitte Distanz des Objekts Objektbreite
LOD	Lateral Obstacle Detection	ca. 0.3 - 5 Meter links und rechts des Fahrzeugs	4	Identifikationsnummer für ein Objekt Seitliche Distanz des Objekts Objektlänge

Tabelle 2: Hinderniserkennungsmodule von VITA II

2.2.2.3 Verkehrszeichenerkennung

Ein weiteres Sensormodul von VITA II ist die Verkehrszeichenerkennung (*TSR - Traffic Sign Recognition*) [Estable et al. 94]. Eine a priori spezifizierte Menge von Verkehrszeichen wird unter Nutzung der Farbinformation eines Kamerabildes aus natürlichen Szenen während der Fahrt erkannt. Dadurch kann sich das Fahrzeug u.a. an Geschwindigkeitsbeschränkungen halten und Überholverbote beachten.

Das Verkehrszeichenerkennungssystem läuft auf einem in das Transputernetz integrierten Subnetz bestehend aus drei MPC 601 Prozessoren. Zur Bildaufnahme und Verschickung werden Framegrabber und TIP-Bus Komponenten des Transputersystems verwendet. Auf dieser Hardware erreicht das System Zykluszeiten von unter 500 Millisekunden. Diese Verarbeitungsgeschwindigkeit ist ausreichend um Verkehrszeichen bei Geschwindigkeiten von etwa 120 km/h zu erkennen.

Erster Schritt zur Erkennung von Verkehrszeichen aus einem Videobild ist die Farbsegmentierung. Wird eine Region anhand ihrer Farbklasse als Kandidat für ein Verkehrszeichen eingeschätzt, so folgt eine Untersuchung der Form und schließlich

eine Piktogrammanalyse der Region in der ein mögliches Verkehrszeichen detektiert worden ist.

Zur Verifikation und Verbesserung der Erkennung, wird ein Verkehrszeichen im Bild verfolgt. Kandidaten, die in einer Entfernung detektiert wurden, in der eine sichere Piktogrammerkennung noch nicht möglich ist, werden weitergeführt und zu einem späteren Zeitpunkt dem Piktogrammerkennung erneut vorgelegt.

Detailliertere Informationen zur Verkehrszeichenerkennung sind [Besserer et al. 93, Zheng et al. 94 und Priese et al. 94] zu entnehmen.

Modul	Bezeichnung	Sichtbereiche	Gelieferte Daten
TSR	Traffic Sign Recognition	40-80 Meter vor dem Fahrzeug	Verkehrszeichencode (bis zu 5 Zeichen) Zusatzinformation für das Verkehrszeichen Hinweisstärke auf dieses Verkehrszeichen

Tabelle 3: Die Verkehrszeichenerkennung in VITA II

2.2.3 Das Kontrollsystem

Das Kontrollsystem von VITA II besteht aus den Teilmodulen SA (Situation Assessment) und BC (Behaviour Control), die Gegenstand des Hauptteils dieser Arbeit sind. Aufgabe von SA ist die Aufbereitung der Sensordaten für die Verhaltenssteuerung durch Fusion und Glättung. BC ermittelt die Vorgaben für die unterlagerten Längs- und Querregler zur Fahrzeugansteuerung. Ausgabe des Kontrollsystems sind Sollwerte für die Geschwindigkeit und Beschleunigung sowie für die Sollablage des Fahrzeugs von der aktuellen Fahrspur und die Quergeschwindigkeit.

Modul	Bezeichnung	Gelieferte Daten
SA	Situation Assessment	Gefilterte Daten des Straßenerkenners Gefilterte und fusionierte Daten der Hinderniserkennung Aktuell gültige Verkehrszeichen
BC	Behaviour Control	Kommandierte Geschwindigkeit Kommandierte Beschleunigung Kommandierte Ablage (Y-Position) Kommandierte Quergeschwindigkeit Blinker links, rechts

Tabelle 4: Ausgaben des Kontrollsystems von VITA II

2.2.4 Die Fahrzeugansteuerung

Das Modul zur Fahrzeugansteuerung (*VC - Vehicle Control*) stellt die Schnittstelle zwischen Anwendung und Fahrzeug dar. VC erhält die für die Fahrzeugansteuerung relevanten Daten von den Sensormodulen und dem Kontrollsystem und übermittelt diese an den Fahrzeugrechner. Von diesem empfängt VC Fahrzeuginformationen sowie Fahrerkommandos, die über Blinkerhebel sowie Tempomathebel erfolgen.

Der Fahrer kann durch Vorgabe einer Wunschgeschwindigkeit mit Hilfe des Serientempomats sowie durch Setzen des Blinkers, zur Kommandierung eines Spur-

wechsels, mit dem Fahrzeug kommunizieren. Die im Fahrzeugrechner integrierten Regler zur Längs- und Querverführung benötigen als Sollwerte die Geschwindigkeit und Beschleunigung sowie die aktuelle Ablage des Fahrzeugs von der Spurmitte, die Sollablage und Quergeschwindigkeit. Diese Vorgaben werden dann in Lenkwinkel (bzw. Lenkraten) und Drosselklappenöffnung sowie Bremsdruck transformiert und an die Aktoren gegeben.

Gleichzeitig stellt dieses Modul auch eine Sicherheitsschale dar. Es überwacht den Zustand des Straßenerkenners und der weiteren Objektmodule und verweigert gegebenenfalls die Aktivierung des autonomen Modus. Tabelle 5 zeigt die Daten, die in VC verwaltet und verschoben werden.

Modul	Bezeichnung	Anwendungsrechner zu FZG	FZG zu Anwendungsrechner
VC	Vehicle Control	Kommandierte Geschwindigkeit Kommandierte Beschleunigung Kommandierte Ablage von der Fahrspurmitte Kommandierte Quergeschwindigkeit Blinker setzen links Blinker setzen rechts Ablage des Fahrzeugs Gierwinkel des Fahrzeugs Spurbreite Horizontale Straßenkrümmung Krümmungsänderung der Straße Kommandomodus	Zurückgelegte Strecke Geschwindigkeit des Fahrzeugs Beschleunigung in Längsrichtung Beschleunigung in Querrichtung Gierrate des Fahrzeugs Wunschgeschwindigkeit des Fahrers (Eingabe über den Serientempomat) Monitorkodierung (Auswahl des Monitorbildes durch den Fahrer) Blinker links gesetzt/nicht gesetzt Blinker rechts gesetzt/nicht gesetzt Fahrzeug aktiv/inaktiv

Tabelle 5 : Kommunikationsschnittstelle zum Fahrzeugrechner

2.2.5 Das Kommunikationssystem

Das Gesamtsystem ist in drei Rechnersysteme unterteilt. Innerhalb des Fahrzeugrechners und des Plattformrechners gibt es aufgrund der geringen Prozessorzahl keine weitere modulare Untergliederung von Prozessorgruppen. Auf beiden Rechnern ist jeweils ein Kopplungsprozeß für den Datenaustausch mit anderen Rechnersystemen zuständig. Dagegen besteht der Anwendungsrechner aus einer großen Anzahl von Prozessoren und Prozessen, zu deren Verwaltung es einer geeigneten Kommunikationsstruktur bedarf.

Als Kommunikationsschnittstelle der einzelnen Module dient eine gemeinsame dynamische Datenbasis (*DDB - Dynamic Data Base*). Alle Softwaremodule des Anwendungsrechners sind an die DDB angeschlossen, beziehen Eingaben aus der DDB und liefern ihre Ergebnisse und Meßwerte dort ab. Die dynamische Datenbasis verwaltet alle im Anwendungsrechner zur Verfügung stehenden Daten. Sie enthält zu jedem Zeitpunkt ein Abbild des gesamten Systemzustands. Jedes Modul kann sich als Klient bei der Datenbasis anmelden und den lesenden oder schreibenden Zugriff auf bestimmte Datenpakete anfordern. Der DDB-Prozeß sendet dann die geforderten Daten in dem gewünschten Takt an den jeweiligen Klienten, bzw. empfängt Daten von diesem. Neben der Koordinierung der Kommunikation mit den einzelnen Modulen verwaltet die

dynamische Datenbasis auch den Zustand der Klienten. Somit ist es einem lesenden Prozeß möglich, zu überprüfen, ob die gelesenen Daten aktuell sind.

Untersuchungen zufolge kann die Zykluszeit der DDB von 40 Millisekunden mit der in VITA II benötigten Klientenzahl und den verschickten Datenmengen garantiert werden (vgl. [Kämmeler 93]).

Abbildung 10 zeigt schematisch den Aufbau des Kommunikationssystems von VITA II. Die Hilfsmodule MON (Monitor) und REC (DDB-Recorder) dienen zur Darstellung des internen Systemzustands. MON erstellt graphische Systemansichten, die den aktuellen Zustand einzelner Module, sowie die Weltansicht des autonomen Fahrzeugs auf den im Fahrzeug integrierten Monitoren darstellen.

Das Modul REC zeichnet die DDB Daten während einer Fahrt auf, so daß diese im Nachhinein untersucht werden können. Ein im Labor erstelltes Simulationsprogramm ermöglicht das Einspielen und Visualisieren dieser Daten (vgl. Abschnitt 3.5).

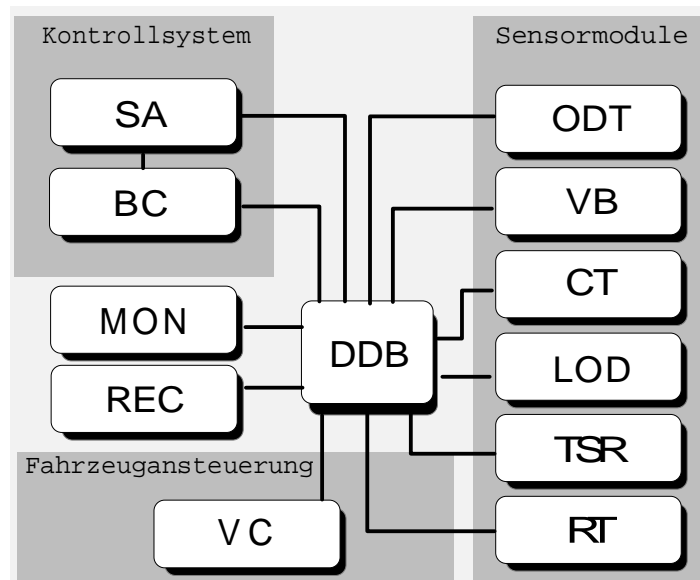


Abbildung 10 : Kommunikationsstruktur in VITA II

3 Autonome Verhaltenssteuerung

„Künstliche Intelligenz (KI) ist die Forschung darüber, wie man Computer Dinge ausführen lassen kann, die zur Zeit noch vom Menschen besser beherrscht werden.“

Elaine Rich [Rich 88]

Die vorliegende Arbeit beschreibt einen neuen Ansatz zur Steuerung eines autonomen Fahrzeugs im Straßenverkehr. Dieser umfaßt die Verarbeitung von Umgebungsinformationen in Autobahnszenarien und deren Repräsentation sowie die kontinuierliche Steuerung eines autonomen Fahrzeugs in diesem Umfeld. In Anlehnung an die in Kapitel 2 gewählte Terminologie für autonome mobile Fahrzeuge wird hier im Folgenden auch von einem *Kontrollsystem* gesprochen.

Die Zielsetzung der Verhaltenssteuerung ist in der Einleitung anhand von drei Kriterien beschrieben:

- Reaktion auf beliebige Szenenkonstellationen,
- Echtzeitfähigkeit der Algorithmen und
- Erklärbarkeit der Verhaltensweisen.

Der hier vorgestellte Ansatz zur kontinuierlichen *Verhaltenssteuerung* berücksichtigt diese drei Anforderungen. Dazu wird eine Repräsentation der Situation und des Wissens über die Domäne Straßenverkehr benötigt. Diese Repräsentation muß die Anforderung der weitestgehenden Allgemeinheit erfüllen, um alle sich ergebenden Situationen beschreiben zu können. Weiterhin muß ihre Komplexität gering gehalten sein, um eine Echtzeitfähigkeit zu gewährleisten. Da auch die Erklärbarkeit gegeben sein soll, liegt es nahe, die Motivation für eine solche Repräsentationsform im Bereich des menschlichen Fahrverhaltens zu suchen. Einen guten Ausgangspunkt bilden die *Risikokompensationstheorien* zur Erklärung menschlicher Verhaltensweisen im Straßenverkehr [Huguenin 88]. Hier wird ein direkter Zusammenhang des Fahrverhaltens mit der *Gefahr* bzw. mit dem sich aus der Situation für den Fahrer ergebenden *Risiko* hergestellt. Um den Risikobegriff etwas einzuschränken, wird hier nur der Aspekt der Gefahr beim Führen eines Fahrzeugs betrachtet. Weiterhin unterscheidet man zwischen objektivem und subjektivem Risiko. Dabei läßt sich objektives Risiko als Produkt von Auftretenswahrscheinlichkeit und Schwere eines Ereignisses definieren. Subjektives Risiko ist psychologisch definiert als die Erwartung eines Ereignisses, dessen Schwere und Auftretenswahrscheinlichkeit bestimmt, aber unbekannt

sind. Das subjektive Risiko ist also weniger durch die Situation als durch den Fahrer bestimmt. Je sicherer sich dieser fühlt, desto geringer ist sein subjektives Risiko.

Die Risikokompensationstheorien postulieren, daß ein Fahrer sein Fahrverhalten unter Gefahrenkompensation an die Umwelt anpaßt. Nach der Theorie der *Risikohomöostase*¹ vergleichen die Verkehrsteilnehmer das aktuell bestehende Maß an (subjektivem) Risiko mit dem Maß akzeptierten Risikos und tendieren dazu, bestehende Diskrepanzen zu eliminieren. Auch wenn die Risikokompensationstheorien zur Erklärung menschlichen Fahrverhaltens nicht unumstritten sind, so bietet der Risikobegriff doch eine geeignete Modellierungsgrundlage für eine allgemeine, echtzeitfähige und erklärbare Repräsentationsform der Domäne Straßenverkehr.

Aufgrund dieser Eigenschaften wird der Risikobegriff als Kern der Modellierung des hier vorgestellten Kontrollsystems gewählt. Der Aufbau des Kontrollsystems ist weiterhin durch Modelle aus der kognitiven Psychologie [Johannsen 90, Rasmussen 83] sowie durch die Methode der künstlichen Potentialfelder zur Steuerung autonomer mobiler Roboter [Krogh & Thorpe 86, Tilove 90] motiviert.

In den Abschnitten 3.2 und 3.3 werden zwei Ansätze zur autonomen Fahrzeugführung vorgestellt, denen eine unterschiedliche Interpretation des Gefahrenbegriffs zugrundeliegt.

- (1) Der in Abschnitt 3.2 diskutierte Ansatz bildet die Umgebung auf eine **Gefahrenkarte** ab. Diese Karte ist durch ein zweidimensionales Feld von Gefahrenwerten beschrieben. Die Karte ist egozentrisch aufgebaut, d.h. der Kartenmittelpunkt stellt die Position des autonomen Fahrzeug dar. Die Fahraufgabe wird in dieser Repräsentation auf eine Minimumsuche in dem durch die Umgebung aufgebauten Gefahrengebirge abgebildet. Zur Lösung dieses Problems wird eine Interpretation des Gefahrengebirges als **elektrisches Feld** vorgenommen. Die Anforderung der Reaktion auf beliebige Szenenkonstellationen wird durch diesen Ansatz weitgehend erfüllt. Die Echtzeitfähigkeit des Algorithmus und die Erklärbarkeit der Fahrzeugreaktionen (im Sinne menschlicher Verhaltensweisen) sind jedoch nicht zufriedenstellend.
- (2) Die in Abschnitt 3.3 vorgestellte Methode zur Verhaltenssteuerung verwendet ebenfalls eine Gefahrenrepräsentation der Umgebung. Hier wird im Gegensatz zum ersten Ansatz zwischen Gefahr und Reaktion unterschieden. Damit erreicht man eine bessere Anpassungsfähigkeit an Verhaltensweisen menschlicher Fahrer. Fahrerreaktionen werden in diesem Ansatz als *konkurrierende Verhaltensmuster* modelliert,

¹ Risikohomöostase (abgeleitet aus Homöostase = Gleichgewicht der physiologischen Körperfunktionen): Verkehrspsychologische Theorie des Fahrverhaltens, die zu erklären versucht, weshalb Fahrer bereit sind, ein bestimmtes Maß an Risiko zu akzeptieren. (nach Wilde; vgl. [Huguenin 88])

die über ein Inhibitionsnetzwerk miteinander verknüpft sind. Die Gefahrenwerte dienen der Gewichtung und Priorisierung der Verhaltensmuster untereinander.

Die Untersuchung der in Abschnitt 3.2 beschriebenen Methode stellt einen ersten Ansatz dar, der die Schwierigkeiten einer allgemeinen Gefahrenrepräsentation aufzeigt. Er ist in der Anfangsphase der Bearbeitung des Themas anzusiedeln. Seine Diskussion in Abschnitt 3.2 dient der Motivation für die Nutzung einer Gefahrenrepräsentation als Leitlinie für die autonome Fahrzeugführung und zeigt gleichzeitig deren Grenzen auf.

Die Methode der konkurrierenden Verhaltensmuster berücksichtigt die Kritik an diesem ersten Ansatz und trennt zwischen Gefahren und Fahrzeugreaktionen. Die dadurch erreichten Verbesserungen führen zu einem Kontrollsystem, das im realen Straßenverkehr eingesetzt wird.

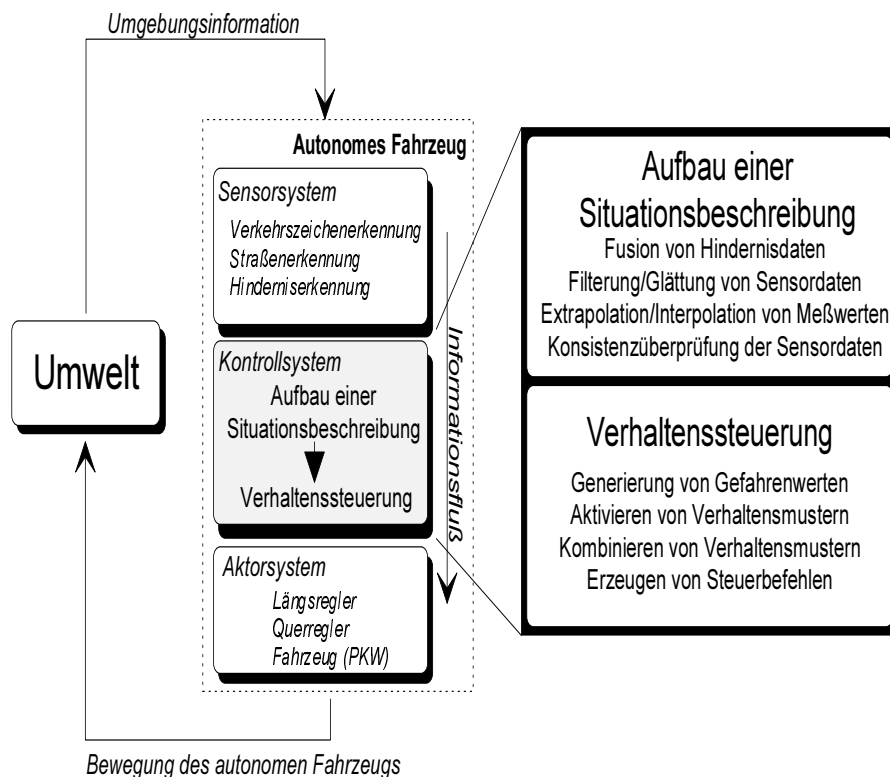


Abbildung 11 : Informations- / Aktionszyklus des autonomen Fahrens

Die Verhaltenssteuerung berücksichtigt nicht explizit das in realen Anwendungen vorkommende Rauschen von Meßwerten. Für den Einsatz der Algorithmen auf dem Versuchsträger VITA II ist deshalb ein Datenaufbereitungsmodul (SA - Situation Assessment) zum Aufbau einer Situationsbeschreibung vorgeschaltet.

Das Kontrollsystem besteht aus zwei Teilen:

- **Aufbau einer Situationsbeschreibung**

Das Modul zum Aufbau einer Situationsbeschreibung nimmt verrauschte Sensorinformationen über die Position und Geschwindigkeit von anderen Fahrzeugen, über den Straßenverlauf und über erkannte Verkehrszeichen auf. Diese Daten werden geglättet und zu einer gemeinsamen Situationsbeschreibung kombiniert. Zur Überbrückung von toten Winkeln und vorübergehenden Objektverlusten durch die logischen Sensoren werden die Meßwerte kurzfristig extrapoliert. Aufgabe dieses Datenaufbereitungsmoduls ist zusätzlich die Überwachung der Konsistenz der generierten Situationsbeschreibung. Abschnitt 3.4 beschreibt den zur Lösung dieser Aufgaben entwickelten Ansatz.

- **Verhaltenssteuerung**

Die Verhaltenssteuerung wertet die jeweils vorgelegte Situationsbeschreibung aus und generiert Vorgaben zur kontinuierlichen Steuerung des autonomen Fahrzeugs in Längs- und Querrichtung.

Abbildung 11 veranschaulicht den Aufbau des autonomen Fahrzeugs sowie die Wechselwirkung zwischen diesem und seiner Umwelt.

Sowohl zur Situationsanalyse als auch zur Verhaltenssteuerung wird *Hintergrundwissen* benötigt. Dieses Wissen ist in unterschiedlichen Repräsentationsformen in das Kontrollsystem integriert, die in den folgenden Abschnitten behandelt werden. Die folgenden Wissensgebiete werden unterschieden:

- ***Wissen über das autonome Fahrzeug selbst***

- a) Wissen über das Sensorsystem*

Zur Aufbereitung der über das Sensorsystem erhaltenen Informationen wird Wissen über die Funktionsweise der einzelnen logischen Sensoren mit einbezogen. Insbesondere zur Datenfusion und Filterung werden statistische Angaben über die zu erwarteten Meßwertabweichungen und die spezifischen Fehler der logischen Sensoren verwendet.

- b) Wissen über die Fahrzeugdynamik*

Um die Generierung von Steuerbefehlen zu verhindern, die nicht vom Fahrzeug durchführbar sind, wird ein dynamisches Modell des Fahrzeugs mitgeführt.

- ***Wissen über den Fahrer***

Der menschliche Fahrer wird hier durch ein Computerprogramm ersetzt. Da es verschiedene Fahrer und Fahrertypen mit unterschiedlichen Handlungspräferenzen, Sicherheitsbedürfnissen und Fahrstilen gibt, wird dies auch durch das Kontrollsystem des autonomen Fahrzeugs berücksichtigt. Mittels eines Fahrermodells wird die Verhaltensweise des autonomen Fahrzeugs entsprechend eines Fahrertyps festgelegt.

- ***Wissen über die Domäne***

- a) *Wissen über den Straßenverlauf*

Durch Einschränkung des Diskursbereichs auf die Autobahnumgebung (siehe Abschnitt 3.1) ist eine starke Vereinfachung eines Straßenmodells möglich. Unter anderem wird zur Meßwertfilterung ausgenutzt, daß die Straßenbreite keinen abrupten Änderungen unterliegt, und sich die Anzahl der Fahrspuren nur sehr niederfrequent ändert.

- b) *Wissen über andere Verkehrsteilnehmer*

Wie schon bei der Eigenbetrachtung wird auch hier ein Modell der Fahrzeugdynamik zugrundegelegt. Dieses Wissen wird vor allem zur Filterung der eingehenden Sensordaten sowie zur zeitlichen Fortschreibung von Objekten in toten Winkeln genutzt.

Neben diesem, auf physikalischen Gesetzen beruhenden Wissen, wird auch Wissen über typisches Verhalten im Straßenverkehr zur Generierung von Hypothesen eingesetzt.

- c) *Wissen über Verkehrsregeln und Verkehrszeichen*

Um ein adäquates Verhalten des autonomen Fahrzeugs im Straßenverkehr zu erreichen, müssen die geltenden Verkehrsregeln befolgt werden. Dazu wird Wissen über allgemeine und lokale, durch Verkehrszeichen angezeigte, Verkehrsregeln verwendet.

Nach der Konzeption des Kontrollsystems werden die Algorithmen vor dem Einsatz im realen Straßenverkehr in einer Simulation getestet. Für die Test- und Adaptionphase der Kontrollalgorithmen wird eine speziell für diese Aufgaben erstellte Simulationsplattform eingesetzt. Hier werden sowohl korrekte Umgebungsinformationen generiert und als Eingabe für die Verhaltenssteuerung benutzt als auch Sensorsimulationen eingesetzt, um das Verhalten des Systems bei verrauschten Eingangsdaten zu simulieren. Ein Modell der Fahrzeugdynamik dient der adäquaten Generierung von Fahrzeugreaktionen. Erst nach erfolgreichem Test in der Simulation wird das System im Versuchsträger VITA II im

realen Autobahnumfeld eingesetzt. Die zu diesem Zweck entwickelte Simulationsumgebung wird in Abschnitt 3.5 beschrieben.

Zur Festlegung des Einsatzgebietes der hier vorgestellten Methoden zur autonomen Fahrzeugführung wird in Abschnitt 3.1 zunächst eine Diskurswelt definiert. Auch wenn das Versuchsfahrzeug im realen Straßenverkehr eingesetzt wird, so muß eingestanden werden, daß die verwendete Umgebungsinformation auf das Wesentliche beschränkt ist. Ein korrektes Verhalten des autonomen Fahrzeugs ist nur im Rahmen seiner Eingangsinformationen und der Freiheitsgrade der Bewegung gewährleistet.

3.1 Abgrenzung der Diskurswelt

Betrachtet man eine Verkehrssituation als Ganzes, so kann diese beliebig kompliziert werden. Durch die vorläufige Einschränkung des Einsatzgebietes der Methoden zur autonomen Fahrzeugführung auf Autobahnen wird bereits eine Reduktion der Kompliziertheit erreicht. Es ist jedoch weiterhin nötig, für die nachfolgenden Betrachtungen eine Diskurswelt zu definieren, die eine Untermenge der Informationen der realen Welt beinhaltet. Unter einer Diskurswelt ist hier die Festlegung des zur Beschreibung der Domäne verwendeten Sprachgebrauchs zu verstehen. Diese Diskurswelt stellt die Basis für die Verhaltenssteuerung des autonomen Fahrzeugs dar.

Die Festlegung der Diskurswelt orientiert sich dabei hauptsächlich an den im Versuchsträger VITA II zur Verfügung stehenden Informationen, die durch das Sensorsystem erfaßt werden können. Diese Informationen reichen im realen Autobahnverkehr aus, um ein durch einen menschlichen Fahrer überwachtes, autonomes Fahren zu ermöglichen.

Betrachtet man das Kontrollsystem des autonomen Fahrzeugs als Mittelpunkt der Diskurswelt, so wird diese durch die Festlegung der *eingehenden* und *ausgehenden* Informationen bestimmt.

Die eingehende Information setzt sich aus fünf Teilbereichen zusammen, die im Folgenden die Diskursgegenstände darstellen:

- *Straße*
- *Hindernis*
- *Verkehrszeichen*
- *Zustand des autonomen Fahrzeugs*
- *Benutzereingabe*

Die Straßeninformation ist beschränkt auf die Straße, auf der sich das Fahrzeug befindet. Informationen über einmündende Querstraßen liegen nicht vor. Durch die Einschränkung des Einsatzbereichs des Fahrzeugs auf die Autobahn ist die vorliegende Information in den meisten Fällen ausreichend.

Als Hindernisse werden Fahrzeuge betrachtet, die sich in der Umgebung des autonomen Fahrzeugs befinden. Dabei wird zwischen Personenkraftwagen und Lastkraftwagen unterschieden. Motorräder, Fahrräder sowie Fußgänger und weitere mögliche Hindernisse werden nicht berücksichtigt.

Als weitere Eingangsinformation stehen die erkannten Verkehrszeichen zur Verfügung. Hier wird eine ausgewählte Menge von Verkehrszeichen betrachtet, die das Verhalten des autonomen Fahrzeugs beeinflusst.

Der aktuelle Zustand des autonomen Fahrzeugs selbst wird als vierte Eingangsinformation betrachtet. Zusätzlich wird eine Benutzereingabe in die Diskurswelt aufgenommen. Dem Fahrer wird damit eine Benutzerschnittstelle zur Verfügung gestellt, durch die er Kommandos an das Fahrzeug weitergeben kann. Neben dem Aktivieren und Deaktivieren kann er dem Fahrzeug eine gewünschte Geschwindigkeit vorgeben und einen Wechsel der Fahrspur einleiten.

Der vom Kontrollsystem generierte Steuervektor, die ausgehende Information, ist letzter Bestandteil der Diskurswelt.

Als Grundlage für die Kombination der Informationen dient ein gemeinsames egozentrisches Koordinatensystem, in dessen Ursprung das autonome Fahrzeug selbst steht. Die x-Achse dieses Koordinatensystems verläuft durch den Fahrzeugmittelpunkt entlang der Straße. Die y-Achse verläuft orthogonal dazu durch den Fahrzeugmittelpunkt. Die nachfolgenden Definitionen legen die Begriffe der Diskurswelt fest.

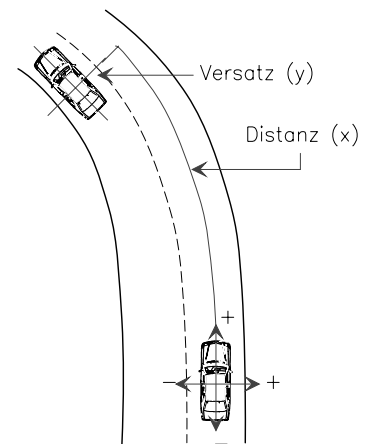


Abbildung 12 : Koordinatensystem der

Definition 3.1.1 Straße

Eine Straße Str wird relativ zum autonomen Fahrzeug durch ein n -Tupel der Form

$$Str = (Sp_{Links}, Sp_{Aktuell}, Sp_{Rechts}, Position_{Amv}, Krümmung)$$

beschrieben. Es wird jeweils die aktuell befahrene Fahrspur $Sp_{Aktuell}$ sowie eine linke und eine rechte Fahrspur betrachtet, sofern diese vorhanden sind. Eine Spur Sp wird beschrieben durch ein n -Tupel der Form

$$Sp = (Spurbreite, Spurmarkierung_{Links}, Spurmarkierung_{Rechts}).$$

Im Falle einer nicht vorhandenen linken oder rechten Spur wird deren Spurbreite auf Null gesetzt. Dabei gilt $Spurbreite \in \mathbb{R}^+$ und $Spurmarkierung \in \{durchgezogen, gestrichelt\}$.

Das Attribut $Position_{Amv} \in \mathbb{R}$ bezeichnet die Ablage des autonomen Fahrzeugs von der Fahrspurmitte. Das letzte Attribut $Krümmung \in \mathbb{R}$ beschreibt die Straßenkrümmung als Kehrwert des Radius eines entsprechenden Kreissegments.

Definition 3.1.2 Hindernis

Ebenso wie die Straße wird ein Hindernis H relativ zum autonomen Fahrzeug definiert. Ein Hindernis wird durch ein n -Tupel der Form

$$H = (\text{Typ}, \text{Länge}, \text{Breite}, \text{Höhe}, \text{Distanz}_{\text{Long}}, \text{Distanz}_{\text{Lat}}, v_H, a_H)$$

beschrieben. Dabei wird durch das Attribut $\text{Typ} \in \{\text{PKW}, \text{LKW}\}$ die Fahrzeugart bestimmt und zusätzlich durch die Attribute $\text{Länge}, \text{Breite}, \text{Höhe} \in \mathfrak{R}^+$ die äußere Form des Fahrzeugs beschrieben. Das Attribut $\text{Distanz}_{\text{Long}} \in \mathfrak{R}$ gibt die Entfernung des Hindernisses vom autonomen Fahrzeug entlang der Straße an. Die relative Ablage des Hindernisses von der lateralen Position des autonomen Fahrzeugs wird mit $\text{Distanz}_{\text{Lat}} \in \mathfrak{R}$ angegeben. Referenzpunkte für die Entfernungswerte sind der Mittelpunkt des Hindernisses und der des autonomen Fahrzeugs. Die Attribute $v_H, a_H \in \mathfrak{R}$ beschreiben die Geschwindigkeit und Beschleunigung des Hindernisses entlang der Straße.

Definition 3.1.3 Verkehrszeichen

Ein Verkehrszeichen Vz wird durch ein Tupel der Form

$$Vz = (VzCode, \text{Zusatz})$$

beschrieben. Dabei wird $VzCode$ gemäß StVO (Straßenverkehrsordnung) bestimmt (vgl. [HAV 87]). Da die StVO durch die Kodierung der Verkehrszeichen nicht in allen Fällen eindeutige Codes liefert, wird die hier verwendete Kodierung durch Hinzufügen eines Zusatzcodes vervollständigt. Die zulässigen Verkehrszeichenkodierungen und ihre Bedeutungen sind im Anhang B beigefügt.

Definition 3.1.4 Fahrzeugzustand

Der Fahrzeugzustand Z wird durch ein n -Tupel der Form

$$Z = (v, a, v_{\text{Lat}}, a_{\text{Lat}}, \text{Blinker}_{\text{Links}}, \text{Blinker}_{\text{Rechts}})$$

beschrieben. Dabei bezeichnen $v, a \in \mathfrak{R}$ die Geschwindigkeit und Beschleunigung des autonomen Fahrzeugs entlang der Fahrbahn. Die Attribute $v_{\text{Lat}}, a_{\text{Lat}} \in \mathfrak{R}$ bezeichnen die laterale Geschwindigkeit und Beschleunigung quer zur Fahrbahn. Die Beschreibung des Fahrzeugzustands wird vervollständigt durch die Attribute $\text{Blinker}_{\text{Links}}, \text{Blinker}_{\text{Rechts}} \in \text{Boolean}$, die anzeigen, ob die Fahrtrichtungsanzeiger des autonomen Fahrzeugs gesetzt sind.

Definition 3.1.5 Benutzereingabe

Die Benutzereingabe Be wird beschrieben durch das Tupel

$$Be = (on, Sw_{\text{Links}}, Sw_{\text{Rechts}}, v_{\text{Soll}})$$

Dabei beschreibt das Attribut $on \in \text{Boolean}$ einen Schalter, den der Benutzer zum Aktivieren und Deaktivieren des Kontrollsystems benutzen kann. Durch Setzen der Attribute $Sw_{\text{Links}}, Sw_{\text{Rechts}} \in \text{Boolean}$ leitet der Benutzer einen automatischen Spurwechsel nach links oder rechts ein. Die gewünschte Geschwindigkeit wird dem Kontrollsystem durch das Attribut v_{Soll} übermittelt.

Aufgabe des Kontrollsystems des autonomen Fahrzeugs ist es, die aktuelle Situation zu einem Zeitpunkt t auszuwerten und eine adäquate Reaktion des autonomen Fahrzeugs in Form eines Steuervektors zu generieren. Eine auszuwertende Situation setzt sich aus den oben definierten Diskursgegenständen zusammen.

Definition 3.1.6 Situation

Eine Situation setzt sich aus der Umgebungsinformation, einer Benutzereingabe und dem Fahrzeugzustand zusammen. Eine Situation Sit_t zum Zeitpunkt t ist definiert als

$$Sit_t = \{Str_t\} \cup \{H_t\}^n \cup \{Vz_t\}^m \cup \{Z_t\} \cup \{Be_t\}.$$

Dabei bezeichnet n die Anzahl der Hindernisse in der Umgebung und m die Zahl der geltenden Verkehrszeichen. Zur Unterscheidung zwischen der Umgebung und den fahrzeuginternen Informationen wird darüberhinaus zwischen interner und externer Situation unterschieden:

$$SitExtern_t = \{Str_t\} \cup \{H_t\}^n \cup \{Vz_t\}^m \quad SitIntern_t = \{Z_t\} \cup \{Be_t\}$$

Definition 3.1.7 Steuervektor

Der Steuervektor Sv stellt die Ausgabe des Kontrollsystems dar. Er wird beschrieben durch ein n -Tupel der Form

$$Sv = (v_K, a_K, v_{KLat}, y_{KLat}, Blinker_{KLinks}, Blinker_{KRechts}).$$

Dabei bezeichnen $v_K, a_K \in \mathfrak{R}$ die longitudinale Geschwindigkeit und Beschleunigung. Die Vorgaben $y_{KLat}, v_{KLat} \in \mathfrak{R}$ bezeichnen die laterale Geschwindigkeit und den gewünschten seitlichen Versatz des autonomen Fahrzeugs gegenüber der Spurmitte der Ausgangspur. Durch Setzen der Attribute $Blinker_{KLinks}, Blinker_{KRechts} \in \text{Boolean}$ werden die Fahrrichtungsanzeiger des autonomen Fahrzeugs angesteuert.

Für die Verwendung der Begriffe im weiteren Verlauf der Arbeit werden folgende Konventionen vereinbart:

- Die aufgeführten Begriffe $Str, H, Vz, Be, Z, Sit, SitExtern$ und $SitIntern$ werden mit einem Index t versehen, um ihre Gültigkeit zum Zeitpunkt t auszudrücken.
- Ein Attribut a eines Begriffs A wird als $a(A)$ bzw. $a(A_t)$ referenziert.
- Ein Element $e \in Sit_t$ wird als Element der Situation bzw. Situationsbeschreibung bezeichnet.
- Ist e ein Hindernis $H \in Sit_t$, dann hat e den Typ H . Analog wird mit Be, Vz, Str und Z verfahren.

3.2 Gefahrengelände und elektrisches Feld

Die Idee des Potentialfeldansatzes ist die Überwindung des starren hierarchischen Konzepts eines klassischen Kontrollsystems für autonome mobile Roboter. Vorteile sind die kontinuierliche Modellierung der Umgebung sowie die höhere Leistungsfähigkeit² eines solchen Systems gegenüber hierarchisch organisierten Kontrollsystemen.

Das hier untersuchte Konzept greift diese Idee auf. Die Umgebung des autonomen Fahrzeugs wird in Form von *Gefahrenpotentialen* auf ein zweidimensionales Feld, die

² Leistungsfähigkeit bedeutet hier insbesondere die schnellere Informationsverarbeitung durch enge Kopplung von Wegplanung und Umsetzung der Fahrmanöver.

Gefahrenkarte, abgebildet. Es entsteht ein Gefahrengebirge, in welchem sich das Fahrzeug fortbewegt. Durch diese Abbildung wird die eigentliche Fahraufgabe auf die Minimierung der Gefahr entlang des gewählten Weges reduziert.

Klassische Lösungsansätze zur Suche eines Pfades in einer solchen Umgebung sind das *Hill-Climbing* Verfahren oder die *Gradientenmethode*. Bei beiden handelt es sich um lokale Verfahren. Da hier die gesamte Gefahrenkarte, d.h. ein größerer lokaler Ausschnitt der Umgebung bekannt ist, wird ein Verfahren untersucht, welches mehr Informationen nutzt.

Interpretiert man die Gefahrenpotentiale als Punktladungen, die abstoßend auf das Fahrzeug wirken, so induziert die Gefahrenkarte ein Kräftefeld. Dieses Kräftefeld bestimmt die Bewegung des autonomen Fahrzeugs, welches als geladenes Teilchen modelliert wird.

Die Anwendbarkeit dieses Konzepts auf die Problematik der autonomen Fahrzeugführung im Straßenverkehr wird in diesem Kapitel diskutiert. Im Folgenden wird zunächst der generalisierte Potentialfeldansatz beschrieben, der als Grundlage für die Gefahrenkarte dient. Anschließend wird das Verfahren in seiner Anwendung auf die autonome Fahrzeugsteuerung im Straßenverkehr diskutiert. Dabei werden Realisierungsdetails nur soweit einbezogen, daß die Vor- und Nachteile des Verfahrens veranschaulicht werden.

3.2.1 Der generalisierte Potentialfeldansatz

Das in dieser Arbeit vorgestellte Prinzip der Verhaltenssteuerung ist angelehnt an den Potentialfeldansatz zur Steuerung von autonomen mobilen Robotern (AMR), wie er von Bruce H. Krogh 1983 vorgestellt wurde [Krogh 83]. Auf dieses grundlegende Prinzip soll an dieser Stelle kurz eingegangen werden, um später auch Unterschiede und Erweiterungen aufzeigen zu können.

Der Potentialfeldansatz dient der Steuerung von autonomen mobilen Robotern. Mit Hilfe dieser Methode kann ein solches System Hindernissen ausweichen und zu einem vorgegebenen Ziel gesteuert werden.

In klassischen, hierarchisch aufgebauten Steuerungssystemen für autonome mobile Roboter stellt die Hindernisvermeidung eine Aufgabe für einen Pfadplanungsalgorithmus der höheren Steuerungsebene dar. Nach einer groben Pfadplanung wird der ermittelte Pfad auf niedrigerer Planungsebene verfeinert. Erst auf dieser Ebene wird auch die Dynamik des zu steuernden Fahrzeugs in die Trajektorienberechnung einbezogen. Eine dritte Steuerungsebene übernimmt schließlich die Ansteuerung der Aktoren.

Aufgrund der Komplexität der verwendeten Algorithmen, die sich insbesondere bei Veränderungen der Umgebung oder dem Erhalt von neuen Informationen bemerkbar machen, sind diese Algorithmen nur bedingt für Steuerung von Robotern in Echtzeit einsetzbar.

Im Gegensatz dazu bestimmt die Potentialfeldmethode direkt eine kollisionsfreie Trajektorie, die dynamischen Randbedingungen genügt und keine a priori Berechnung voraussetzt. Die Bewegungsrichtung des autonomen mobilen Roboters wird zu jedem Zeitpunkt durch ein Vektorfeld bestimmt, welches durch die Hindernisse und das Ziel erzeugt wird.

3.2.1.1 Die generalisierte Potentialfeldmethode nach Krogh

Man kann sich ein Potentialfeld als eine zu einem vorgegebenen Ziel hin geneigte Ebene vorstellen. Hindernisse werden als Berge repräsentiert und die Beschleunigungsrichtung des zu steuernden Systems wird durch den Gradienten des Feldes bestimmt.

Bruce H. Krogh erweitert diesen positionsgebundenen Ansatz durch Berücksichtigung der Geschwindigkeit und Bewegungsrichtung des autonomen mobilen Systems [Krogh 83, Krogh 84]. Nach Krogh wird jedes Hindernis in der Ebene als konvexe Menge definiert. Die Vereinigung dieser Mengen wird als *avoidance set* bezeichnet. Das zu steuernde System wird durch einen Punkt $x(t)$ repräsentiert. Die Dimension des zu steuernden Systems wird bei der Repräsentation der Hindernisse berücksichtigt. Die Aufgabe besteht darin, das autonome mobile System von einem Startpunkt $x(0) = x_0$ zu einer Zielposition $x(T) = x_g$ zu bewegen. Dabei sollten die Start- sowie die Endgeschwindigkeit gleich Null sein. Als Randbedingung darf kein Punkt $x(t)$ der Trajektorie innerhalb des *avoidance sets* A liegen.

Zur Steuerung des autonomen Systems wird zu jedem Zeitpunkt ein Beschleunigungsvektor c berechnet, der sich aus einer Linearkombination von Vektorfeldern ergibt, die durch die vorhandenen Hindernisse und das vorgegebene Ziel erzeugt werden.

$$c = c_o + \gamma \cdot c_g$$

c	:	Beschleunigungs-Richtungsvektor
c_o	:	Summe der Hindernisvektoren $c_o = c_{o,1} + \dots + c_{o,n}$
c_g	:	Zielvektor
γ	:	Zielgewichtungsfaktor

Jeder Hindernisvektor $c_{o,i}$ ist einem Hindernis A_i zugeordnet. Die Länge des Vektors entspricht der Notwendigkeit einer Reaktion des autonomen Systems.

Das dem Hindernis A_i zugeordnete generalisierte Potentialfeld wird durch die Funktion $P_i(x,v)$ beschrieben. Bewegt sich das System von Hindernis A_i weg, so ist $P_i(x,v)$ Null. Ansonsten ist die Notwendigkeit der Beschleunigung in Richtung n_i abhängig von der Entfernung d_i des Systems zum Hindernis sowie der Annäherungsgeschwindigkeit v_i unter Berücksichtigung des festgelegten Beschleunigungslimits α . Der Vektor n_i bezeichnet dabei einen Einheitsvektor, der vom nächsten Punkt von A_i ausgehend nach x zeigt.

Zur Bestimmung von $P_i(x,v)$ werden zunächst zwei Extremfälle betrachtet. Verringert das System mit maximaler Verzögerung α seine Geschwindigkeit ab dem Zeitpunkt t_i , so ergibt sich die *minimum avoidance time* zu

$$\tau_i = \frac{v_i(t_0)}{\alpha}.$$

Dagegen berechnet sich die *maximum avoidance time* bei konstanter Verzögerung über die Strecke d_i als

$$T_i = \frac{2d_i}{v_i(t_0)}.$$

Das generalisierte Potentialfeld $P_i(x,v)$ ist als die Inverse der *reserve avoidance time* definiert, welche sich aus $T_i - \tau_i$ ergibt. Der Hindernisvektor $c_{o,i}$ ist als der Gradient des generalisierten Potentialfelds $P_i(x,v)$ bezüglich der Position x definiert.

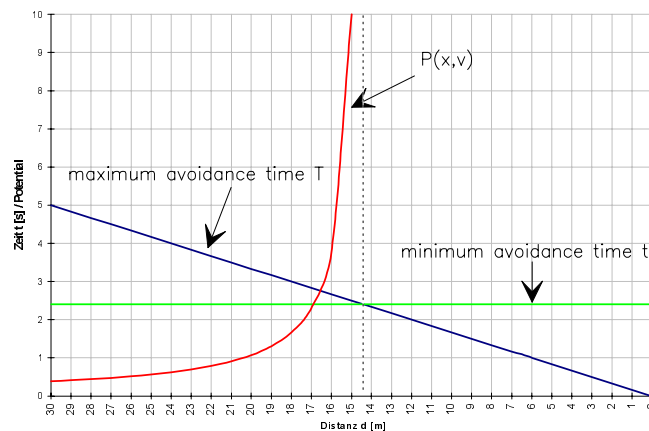


Abbildung 13 Potentialfeld $P(x,v)$ mit $v = 12 \text{ m/s}$ und $\alpha = 5 \text{ m/s}^2$

Zur Erzeugung des Zielvektors c_g benutzt Krogh kein Potentialfeld. Im Fall einer hindernisfreien Umgebung wird c_g durch Minimierung der benötigten Zeit von Startpunkt x_0 zu Ziel x_g bestimmt. Sind Hindernisse vorhanden, so werden Zwischenziele eingeführt. Bezeichne e_g den nächsten Punkt eines Hindernisses zwischen x und x_g , so wird das einzuführende Zwischenziel x_g' durch $x_g' = e_g + \epsilon s_n$ definiert, wobei s_n ein zu der Linie zwischen x und e_g orthogonaler Einheitsvektor

ist, der vom Hindernis weggerichtet ist. Der Parameter ϵ beeinflusst die Entfernung der gewünschten Trajektorie zu den Hindernissen.

3.2.1.2 Diskussion von Erweiterungen der Potentialfeldmethode

Bei seinen Untersuchungen über Potentialfelder unterscheidet Tilove zwischen lokalen und globalen sowie zwischen klassischen und generalisierten Potentialfeldern [Tilove 90]. Während lokale Methoden mit aktuell gewonnenen Sensorinformationen auskommen, benötigen globale Methoden Wissen über die Struktur der gesamten Umgebung, welches a priori aufgebaut werden muß. Durch das Einbeziehen von Geschwindigkeit und Bewegungsrichtung des zu steuernden Systems, kann im Gegensatz zu ortsgebundenen klassischen Potentialfeldern unterschieden werden, wie kritisch die Nähe eines Hindernisses für die Bewegung des Systems ist. Eine unnötige Beeinflussung der Trajektorie kann damit durch die generalisierte Potentialfeldmethode verhindert werden. Einen wichtigen Kritikpunkt an der Potentialfeldmethode stellt die Existenz lokaler Minima dar, in denen das System zum Stehen kommen kann, ohne das eigentliche Ziel erreicht zu haben. Zur Lösung dieses Problems trägt die oben beschriebene Methode zur Erzeugung des Zielvektors bei, die von Krogh und Thorpe um eine a priori Methode der Pfad-Relaxation erweitert wurde [Krogh & Thorpe 86, Thorpe 84]. Neben lokalen Minima kann es auch zu Situationen kommen, in denen das System oszillierendes Verhalten zeigt. Die Häufigkeit dieses unerwünschten Phänomens kann mit Hilfe eines Korrekturterms eingeschränkt werden. Während Krogh und Khatib [Khatib 85] eine Kraft oder Beschleunigung des zu steuernden Systems berechnen, wird das Potentialfeld in anderen Ansätzen als Kostenfunktion interpretiert. Die von Tilove beschriebene Hill-Climbing Methode zur Pfadgenerierung, die auf einer Interpretation des Potentialfeldes als Kostenfunktion aufbaut, berücksichtigt keine dynamischen Randbedingungen des zu steuernden Systems. Dadurch wird zwar im Gegensatz zu den beschleunigungsbasierten Potentialfeldern nie ein Pfad generiert, der ein Hindernis berührt, jedoch kann nicht garantiert werden, daß ein solcher Pfad von einem zu steuernden System realisierbar ist.

Anwendungen der Potentialfeldmethode sowie Weiterentwicklungen sind [Puttkamer et al. 91] sowie [Sandler & Gilles 95, Mezger 88] zu entnehmen. Mit einer Interpretation der Potentialfelder als Kostenfunktion befassen sich [Arkin 87] und [Chattergy 85].

3.2.2 Definition einer Gefahrenkarte

Für die gesamte Umgebung wird eine Gefahrenkarte erstellt, die jedem Punkt der Umgebung einen Gefahrenwert zuordnet. Den Mittelpunkt der Gefahrenkarte bildet die Position des autonomen Fahrzeugs selbst. Um den Begriff der Gefahrenkarte eindeutig festzulegen, wird die folgende Definition zugrundegelegt.

Definition 3.2.1 Gefahrenkarte

Eine Gefahrenkarte zu einem Zeitpunkt t ist diskret definiert durch ein zweidimensionales Feld von Gefahrenwerten der Form

$$G_{m,n}^t = \{x | x \in [-m, m], m \in N_0^+\} \times \{y | y \in [-n, n], n \in N_0^+\} \times \{g | g \in \mathfrak{R}\}.$$

Einer Gefahrenkarte $G_{m,n}^t$ ist eine Potentialfunktion der Form

$$P_{m,n}^t : \{x | x \in [-m, m], m \in N_0^+\} \times \{y | y \in [-n, n], n \in N_0^+\} \rightarrow \{g | g \in \mathfrak{R}\}$$

zugeordnet. Das Gefahrenpotential am Mittelpunkt der Gefahrenkarte $G_{m,n}^t$ wird als Gefahrenniveau bezeichnet. Das Gefahrenniveau $N_{G_{m,n}^t}^t$ zum Zeitpunkt t ergibt sich aus

$$N_{G_{m,n}^t}^t = P_{m,n}^t(0,0).$$

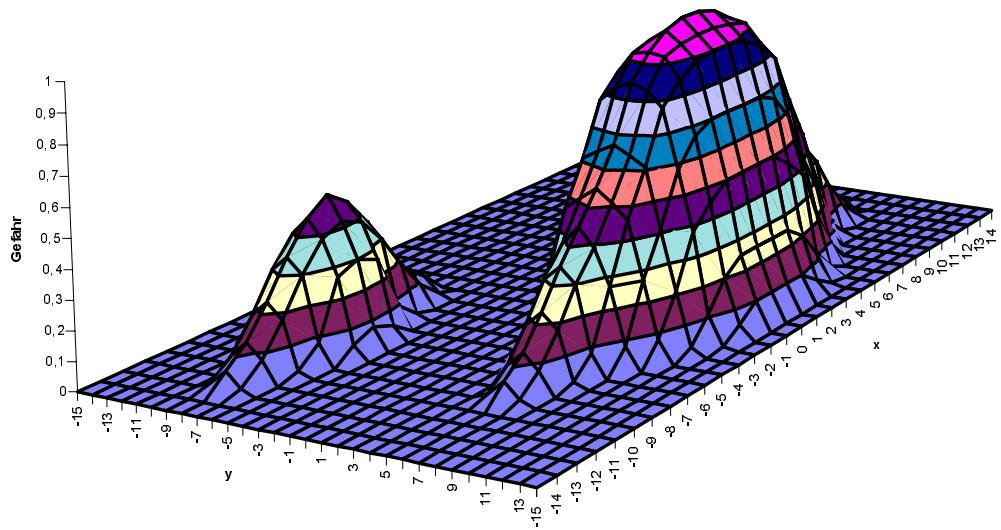


Abbildung 14 : Beispiel einer Gefahrenkarte $G_{15,15}$

Die Semantik eines Gefahrenpotentials $P_{m,n}^t(x,y)$ ist die folgende: Das Gefahrenpotential $P_{m,n}^t(x,y)$ beschreibt die subjektive Gefährlichkeit der Position (x,y) als Aufenthaltsort für das autonome Fahrzeug zum Zeitpunkt t .

Die Modellierung der Umgebung in Form einer Gefahrenkarte stellt eine Abstraktionsabbildung dar. Die Betrachtung der Umgebung wird auf den Aspekt *Gefahr* eingeschränkt. Dadurch erhält man eine stark vereinfachte

Szenenbeschreibung und damit die Möglichkeit der Echtzeitauswertung auf der vorgegebenen Zielhardware. Die Abbildungen der Umwelt auf die Gefahrenkarte sind wie folgt beschrieben.

Definition 3.2.2 Gefahrenabbildung

Eine Gefahrenabbildung A^t liefert zu einer zum Zeitpunkt t vorliegenden Situationsbeschreibung eine zugehörige Gefahrenkarte. Eine Gefahrenabbildung A^t ist definiert als $A^t: Sit^t \rightarrow G_{m,n}^t$.

3.2.3 Die Gefahrenkarte als Grundlage der Verhaltenssteuerung

Die Aufgabe der durch künstliche Potentialfelder gesteuerten autonomen mobilen Roboter besteht hauptsächlich in der Vermeidung von Kollisionen. Dies ist natürlich auch Bestandteil der Fahraufgabe im Straßenverkehr. Darüber hinaus sind jedoch noch weitere Faktoren wie beispielsweise Verkehrsregeln und Fahrkomfort zu berücksichtigen.

Zur Repräsentation der Umgebung in einer Gefahrenkarte werden zunächst mehrere Teilkarten generiert, die anschließend zu einer gemeinsamen Karte fusioniert werden. Die einzelnen Teilkarten repräsentieren jeweils einen Aspekt der Umgebung. Die folgenden Aspekte werden unterschieden:

- Intentionen des Fahrers/Benutzers
- Straße und Fahrspur
- Andere Verkehrsteilnehmer
- Verkehrsregeln/Verkehrszeichen

Dabei ist es in manchen Fällen hilfreich, Aspekte bei der Modellierung einer Teilkarte zu kombinieren. Im Folgenden wird zunächst der Aufbau von Teilkarten beschrieben und anschließend eine Methode zu deren Kombination vorgestellt. Zu beachten ist dabei, daß es sich bei der Modellierung der Umgebung als Gefahren stets um eine subjektive Interpretation handelt. Die nachfolgend aufgeführten Teilkarten sind demnach als eine mögliche Modellierung zu sehen, nicht jedoch als Modellierungsvorschrift.

3.2.3.1 Modellierung der Straße

Geht man zunächst vereinfachend von einer Situation aus, in der keine Hindernisse vorhanden sind, so besteht die Aufgabe des autonomen Fahrzeugs darin, die Spur zu halten. Um die Betrachtung weiter zu vereinfachen, wird für diesen Fall zunächst die Längsbewegung ausgeklammert.

Modell mit nur einer Fahrspur

Ist nur eine einzige Fahrspur vorhanden, so ist es das Ziel, in der Mitte dieser Spur zu fahren. Driftet das Fahrzeug nur leicht von der Spurmittle ab, so ist das tolerierbar und bedarf nur einer leichten Korrektur mit niedriger Dringlichkeit. Driftet es jedoch stärker in Richtung Straßenrand, so stellt das eine größere Gefahr dar. Intuitiv ergibt sich aus dieser Situation eine Gefahrenkarte, in der die Straße in Form einer „Gefahrenwanne“ dargestellt werden kann. Abbildung 15 veranschaulicht eine Gefahrenkarte für eine einspurige Straße. Dazu wird die Abbildungsfunktion A^t so gewählt, daß die Spurbreite der Breite der Karte entspricht. Im Fall einer Spurbreite von drei Metern entspricht eine Einheit der y-Achse der Karte zehn Zentimetern in der Realität.

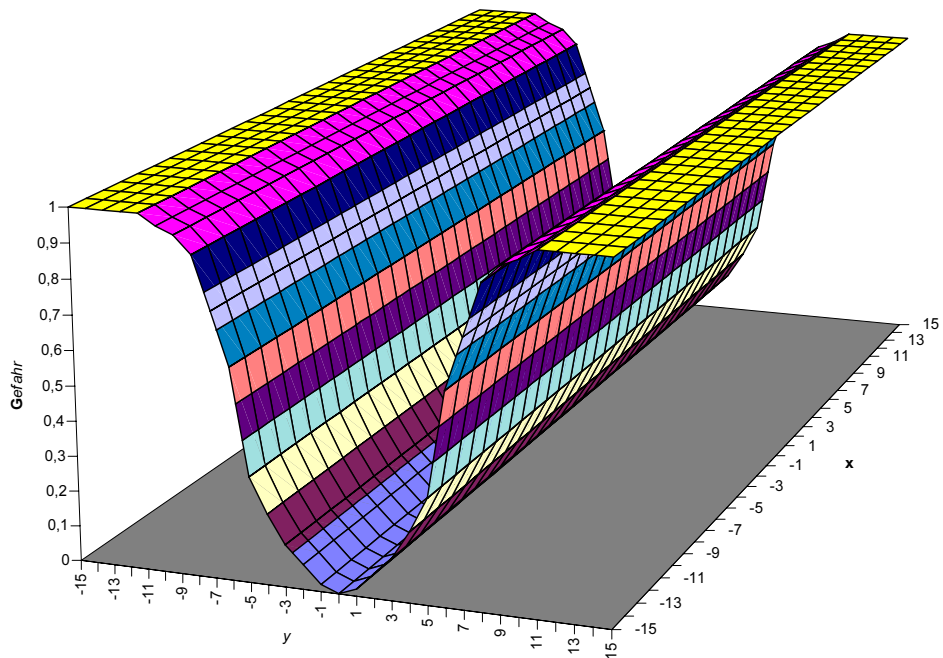


Abbildung 15 : Gefahrenkarte für eine einspurige Straße

Modell mit mehreren Fahrspuren

Sind zwei Fahrspuren vorhanden, so ist es das Ziel, auf der Mitte einer Fahrspur zu fahren. Das Überschreiten der Trennlinie der Fahrspuren ist mit einer Gefahr verbunden, die jedoch nicht so hoch zu wählen ist, wie das Verlassen der Straße.

Das Beispiel einer Straße mit zwei Fahrspuren wird in Abbildung 16 veranschaulicht. Es ergeben sich zwei gleichwertige Wege der geringsten Gefahr. Die Wahl des Weges ist demnach nur von der Ausgangsposition des autonomen Fahrzeugs abhängig.

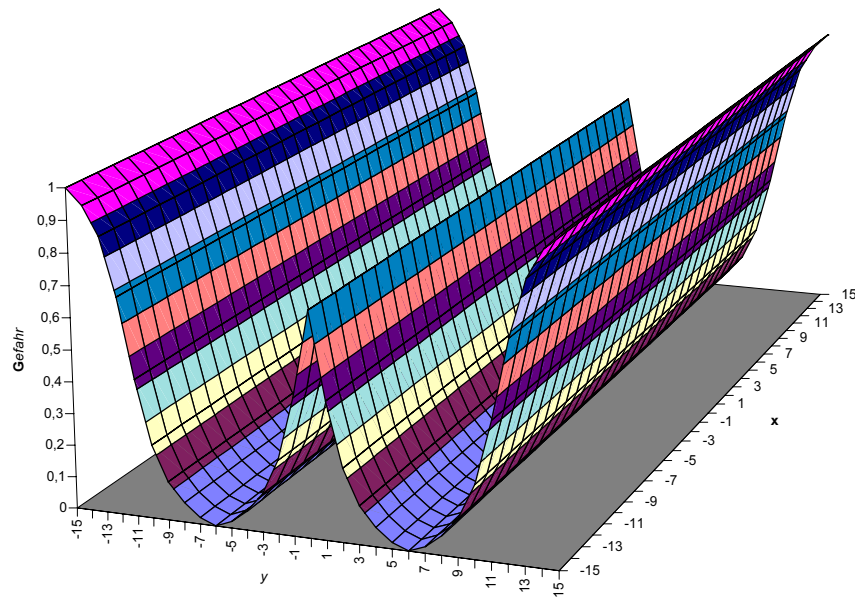


Abbildung 16 : Modellierung einer Straße mit zwei Spuren

3.2.3.2 Modellierung anderer Verkehrsteilnehmer

In vielen Potentialfeldmodellen werden Hindernisse als zum Hindernisrand ansteigende Gebirge modelliert. Krogh und Thorpe beschreiben ein Hindernis in Abhängigkeit von dessen Position zur Fahrtrichtung des zu steuernden Fahrzeugs [Krogh & Thorpe 86]. Liegt das Objekt in Fahrtrichtung, so ist das induzierte Potential zur aktuellen oder geplanten Position proportional zur „reserve avoidance time“ (siehe Abschnitt 3.2.1.1).

Betrachtet man zunächst nur die longitudinale Komponente, so ergeben sich die nachfolgend beschriebenen Überlegungen zur Gefährlichkeit.

Neben der Kollisionsvermeidung gehört auch ein „natürliches“ Fahrverhalten zur Zielsetzung. Dazu gehört die Einhaltung eines geschwindigkeitsabhängigen Sicherheitsabstands zum vorausfahrenden Fahrzeug. Die Gefährlichkeit einer Situation läßt sich zunächst unter dem Aspekt der Kollisionsvermeidung interpretieren:

Die Gefährlichkeit ist proportional zu der Verzögerung, die dazu benötigt wird, noch vor dem Hindernis zum Stehen zu kommen bzw. die Relativgeschwindigkeit auf Null zu reduzieren.

Diese Interpretation läßt sich nicht direkt auf das Einhalten eines Sicherheitsabstands übertragen. Hier ist eine Toleranz bezüglich des (kurzfristigen) Eindringens in den Sicherheitsabstand zu berücksichtigen. An dieser Stelle ist eine Gewichtung der Verzögerung gegenüber dem Einhalten des Sicherheitsabstands abhängig vom Fahrer zu wählen. Dazu wird eine Sollgeschwindigkeitskurve als Vergleich verwendet, die zu jeder Entfernung von einem Hindernis die vom Fahrer gewünschte maximale

Differenzgeschwindigkeit festlegt. Abweichungen von dieser Kurve werden als Gefahren interpretiert. Zusätzlich wird ein Relevanzfaktor eingeführt, mit dem die Abweichung abhängig von der Distanz zum Hindernis gewichtet wird. Abbildung 18 zeigt beispielhaft eine Sollgeschwindigkeitskurve und eine daraus abgeleitete Gefahrenkurve. Die Sollgeschwindigkeitskurve gilt für eine Objektgeschwindigkeit von 20 m/s und einen Sicherheitsabstand von 1,5 Sekunden. Ausgehend von einer Eigengeschwindigkeit von 25 m/s und einem Relevanzfaktor $r = 1/\text{Distanz}$, ergibt sich die in Abbildung 18 dargestellte Gefahrenkurve.

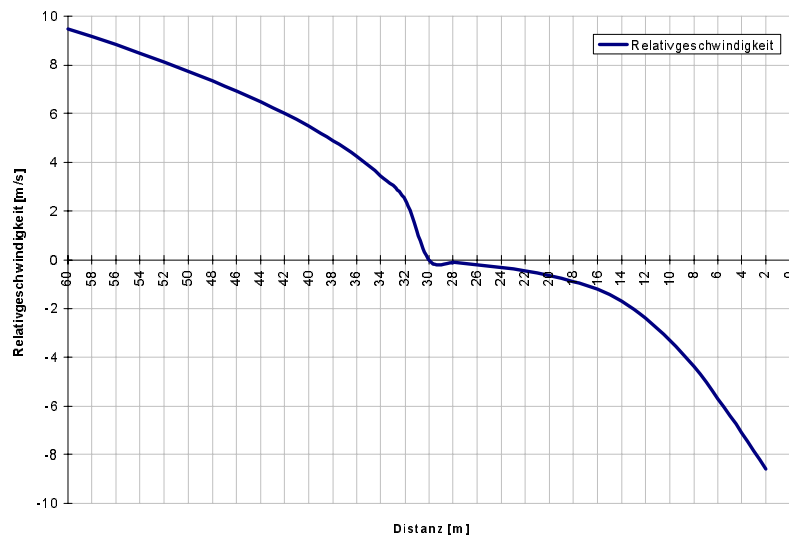


Abbildung 17 : Beispiel einer Sollgeschwindigkeitskurve

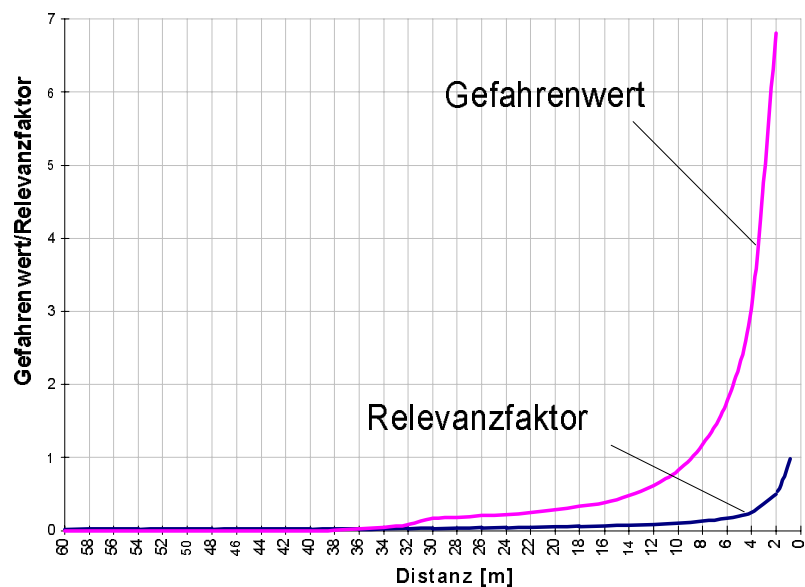


Abbildung 18 : Modellierung eines Hindernisses als Gefahrengelände

Analog werden Hindernisse modelliert, die sich hinter dem autonomen Fahrzeug befinden. Zur Modellierung von Hindernissen, die sich seitlich vom autonomen Fahrzeug befinden, wird ähnlich vorgegangen wie zur Modellierung der Straße. Hier wird ein fahrerspezifischer, seitlicher Sicherheitsabstand, sowie ein Sollabstand bestimmt und eine abstandsabhängige Gefahr berechnet.

Berücksichtigt man, daß jeweils die Wahl zwischen longitudinaler und lateraler Kollisionsvermeidung besteht, so läßt sich das ebenfalls in die Gefahrenrepräsentation integrieren. Ein seitlicher Versatz des Hindernisses gegenüber dem autonomen Fahrzeug stellt damit eine Reduktion der Gefährlichkeit dar, da dadurch das Ausweichen erleichtert wird. Legt man zusätzlich einen maximalen Gefahrenwert fest, so läßt sich ein typisches Hindernis als ein Gefahrenberg wie in Abbildung 19 modellieren.

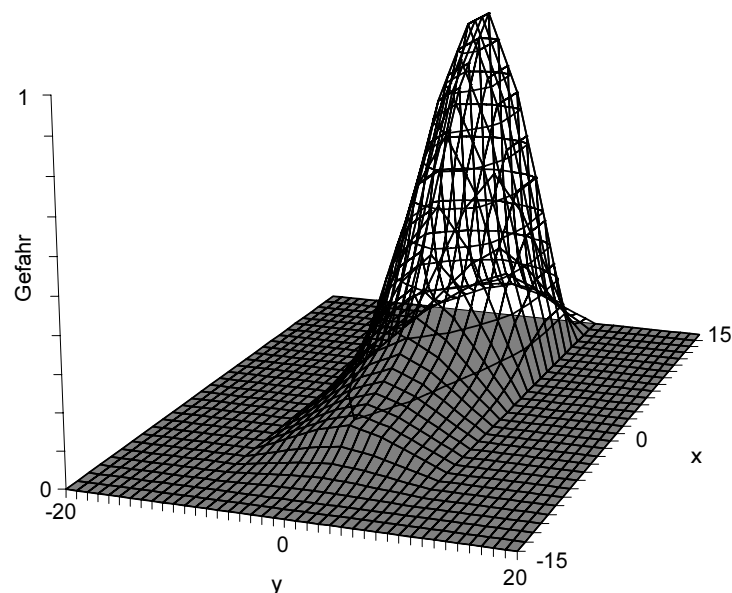


Abbildung 19 : Gefahrenrepräsentation für ein langsamer vorausfahrendes Fahrzeug

3.2.3.3 Modellierung von Verkehrsregeln

Zur Modellierung der Straße und der Hindernisse kann man den Begriff "Gefahr" plausibel verwenden. Bei der Integration von Verkehrsregeln handelt es sich dagegen nicht um eine konkrete Gefahr, sondern vielmehr um eine Einschränkung. Diese Einschränkungen sind durch die Straßenverkehrsordnung vorgegeben, da so eine Verringerung der *potentiellen* Gefahr erreicht werden soll. Insofern stellt eine Überschreitung der Verkehrsregeln eine Gefahr im erweiterten Sinne dar. Die Berücksichtigung solcher Verkehrsregeln in der Gefahrenrepräsentation wird durch zusätzliche Teilkarten, aber auch durch eine Modifikation der Straßen- bzw. Hindernisrepräsentation erreicht.

Verkehrsregeln sind entweder von permanenter Gültigkeit oder werden lokal durch Verkehrszeichen angezeigt. Zur ersten Kategorie gehören das Rechtsfahrgebot auf deutschen Autobahnen sowie das Rechtsüberholverbot. Zur Modellierung des Rechtsfahrgebots werden die linken Fahrspuren mit einer etwas höheren Gefahr belegt, so daß bei freier rechter Spur ein Spurwechsel nach rechts durchgeführt wird. Das Rechtsüberholverbot wird durch eine Erweiterung des Gebirges eines links fahrenden Fahrzeugs auf die rechte Spur erreicht. Dadurch verhält sich das autonome Fahrzeug in seiner Längsführung, als handele es sich um ein vorausfahrendes Fahrzeug auf der eigenen Spur. Da es unnötig ist, auch hier den gleichen Sicherheitsabstand einzuhalten, wird die Erweiterung des Gefahrengebirges soweit nach vorne geschoben, daß das autonome Fahrzeug bis auf gleiche Höhe heranfährt. Alternativ zu dieser Verhaltensweise kann das Gebirge auch so geformt werden, daß das autonome Fahrzeug die Spur wechselt und dem zweiten Fahrzeug im Sicherheitsabstand folgt. Ist eine dritte Spur vorhanden, so wechselt es erneut die Spur und überholt, wie oben beschrieben, falls keine weiteren Hindernisse dies verhindern.

Aus der zweiten Gruppe der Verkehrsregeln werden die Streckenverbote *Überholverbot* und *Geschwindigkeitsbeschränkung* betrachtet. Eine Geschwindigkeitsbeschränkung wird analog zur Modellierung von Fahrerintentionen durch Kippen der Gefahrenkarte in Längsrichtung modelliert.

Zur Einhaltung eines Überholverbots wird die linke Fahrspur mit einer erhöhten Gefahr belegt. Dadurch wird ein Spurwechsel als Reaktion auf ein vorausfahrendes Hindernis verhindert. Befindet sich das vorausfahrende Fahrzeug auf einer Fahrspur, die rechts der Fahrspur des autonomen Fahrzeugs liegt, so wird das diesem Fahrzeug zugeordnete Gefahrengebirge, analog zum Rechtsüberholverbot, nach links erweitert, so daß ein Vorbeifahren verhindert wird³.

3.2.3.4 Modellierung von Fahrerintentionen

Der Fahrer - im vorliegenden Fall ist dieser eher als *Benutzer* zu bezeichnen - kann das Verhalten des autonomen Fahrzeugs während der Fahrt beeinflussen. Er bestimmt die Geschwindigkeit, die das Fahrzeug einhalten soll und kann einen Spurwechsel kommandieren. Ist die aktuelle Geschwindigkeit zu niedrig, so wird automatisch beschleunigt, ist sie zu hoch, wird abgebremst. Dieses Verhalten wird erreicht, indem man die Gefahrenkarte in Längsrichtung nach vorne (bzw. hinten) über den Karten-

³ Die Integration der Verkehrszeicheninformation und der resultierenden Fahrerreaktion wird im Rahmen einer Diplomarbeit näher untersucht [Gachelin 94].

mittelpunkt kippt. In Querrichtung geht man analog vor, um einen Spurwechsel zu modellieren.

Der oben erläuterte Potentialfeldansatz unterscheidet zwischen Hindernisvektoren und dem Zielvektor. Die Rolle des Zielvektors übernehmen in der hier diskutierten Gefahrenrepräsentation die Teilkarten zur Modellierung der Fahrerintentionen.

Die Umsetzung des Fahrerkommandos in Gefahrenwerte geschieht anhand von fahrerspezifischen Verhaltensfunktionen, die auch als „Fahrerkennlinien“ bezeichnet werden können.

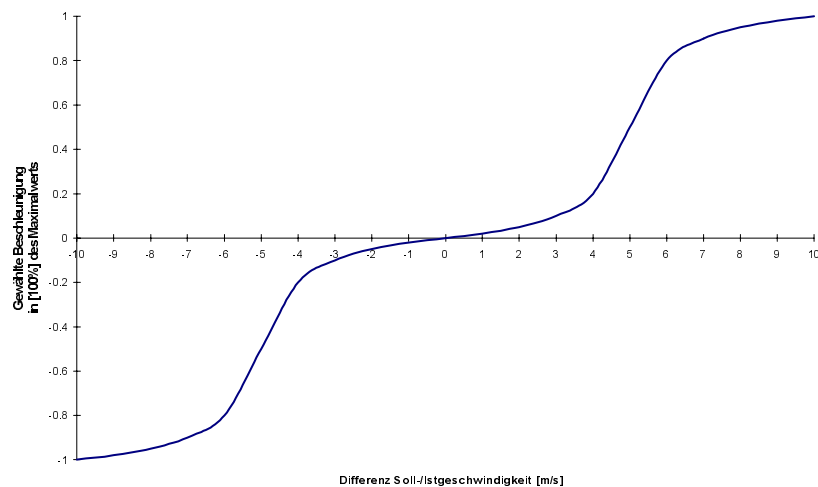


Abbildung 20 : Fahrerkennlinie zur Einhaltung von Sollgeschwindigkeitsvorgaben

Abbildung 20 zeigt ein Beispiel einer Fahrerkennlinie zum Folgen einer Sollgeschwindigkeitsvorgabe. In diesem Beispiel ist der absolute Maximalwert der vom Fahrer gewünschten Beschleunigung gleich dem der gewünschten Verzögerung. Diese Extremwerte, sowie der Kennlinienverlauf sind jedoch abhängig vom modellierten Fahrertyp.

Der Grad der Neigung der Gefahrenkarte in Längsrichtung ist proportional zu der aktuell gewünschten Beschleunigung, die anhand der Fahrerkennlinie ermittelt wird.

Auf ähnliche Weise wird bei einem kommandierten Spurwechsel vorgegangen. Hier wird ein fahrerspezifischer Sollverlauf der Querbeschleunigung (bzw. der Quergeschwindigkeit) zugrundegelegt, zu dem eine Querneigung der Gefahrenkarte bestimmt wird.

3.2.4 Verknüpfung der Teilkarten

Der vorangehende Abschnitt beschreibt die Erstellung von Gefahrenkarten für einzelne Aspekte der Umgebung. Diese Teilkarten müssen nun zu einer gemeinsamen Karte fusioniert werden. Dazu wird zwischen den Teilkarten, die Gefahren repräsentieren und den Karten, die der Repräsentation von Fahrerintentionen dienen,

unterschieden. Erstere werden durch eine gewichtete Maximumsbildung zusammengefügt, während letztere additiv mit dem Ergebnis verknüpft werden. Die Gewichtung der Maximumverknüpfung stellt einen Teil des Fahrermodells dar. Hierdurch kann eine Präferenz des Fahrers gesetzt werden, mit der beispielsweise dem Ausweichen der Kollision mit einem vorausfahrenden Fahrzeug ein höheres Gewicht gegeben wird als dem Überschreiten des Straßenrands. Um eine Kombination der Gefahrenkarten zu vereinfachen, wird eine Normierung festgelegt. Demnach sind die Gefahrenwerte $P_{m,n}^t(x, y)$ für alle x, y bei einer Gefahrenabbildung A zwischen 0 und 1 zu wählen.

Es ist zu beachten, daß es sich jeweils um eine absolute subjektive Gefahr handelt, die noch nicht in Relation zu einer anderen Gefahrenquelle betrachtet wird. Dieser Aspekt wird erst durch die Gewichtung der Teilkarten hinzugefügt. Abbildung 21 verdeutlicht dies anhand einer Querschnittsbetrachtung der Teilkarten für eine zweispurige Straße und ein Hindernis.

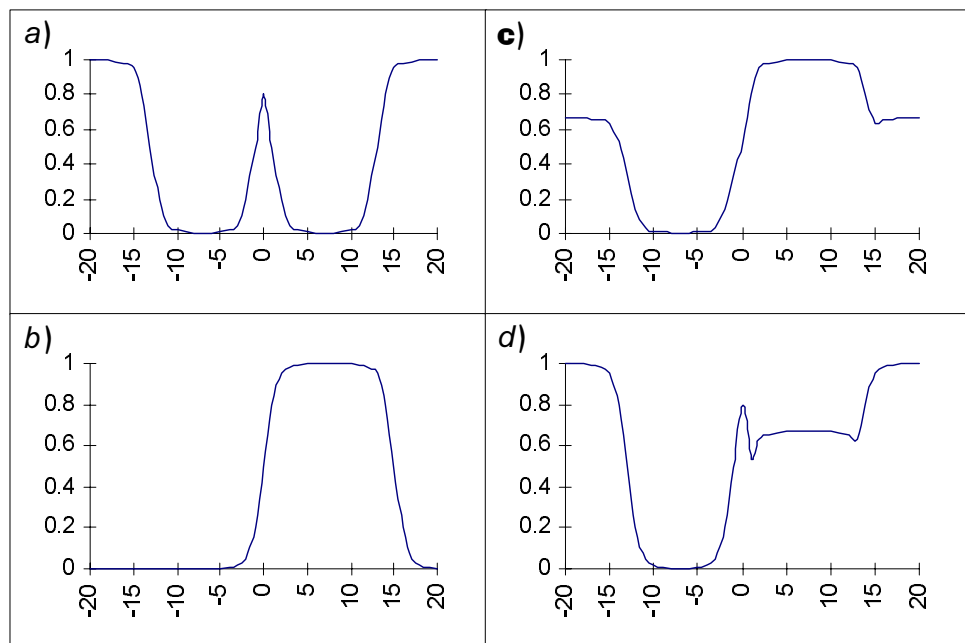


Abbildung 21 : Fusion einer Teilkarte zur Straßenbeschreibung (a) und einer zur Hindernisbeschreibung (b) mit dem Gewichtsverhältnis 2:3 (c) und 3:2 (d)

3.2.5 Interpretation der Gefahrenkarte als elektrisches Feld

Nach dem Aufbau der Gefahrenkarte gilt es, aus der so repräsentierten Umgebungsinformation einen Steuervektor für das autonome Fahrzeug abzuleiten. Dazu wird eine Interpretation der erzeugten Gefahrenkarte als elektrisches Feld untersucht. Interpretiert man dazu die Gefahrenpotentiale als Punktladungen, die abstoßend auf

das als geladenes Teilchen betrachtete autonome Fahrzeug wirken, so induziert die Gefahrenkarte ein Kräftefeld.

Für jedes Gefahrenpotential wird die von ihm ausgehende Feldstärke bezüglich des Kartenmittelpunkts berechnet, und anhand des Superpositionsprinzips wird die auf den Kartenmittelpunkt wirkende Kraft bestimmt.

In Anlehnung an elektrische Gesetzmäßigkeiten berechnet sich der Vektor der Feldstärke $E_p^t = (E_{px}^t, E_{py}^t)$ für den Kartenmittelpunkt bezogen auf eine Punktladung $P_{m,n}^t(x, y)$ näherungsweise wie folgt:

$$E_p^t = P_{m,n}^t(x, y) \cdot \frac{(x, y)}{\left(\sqrt{x^2 + y^2}\right)^3}$$

Unter Verwendung des Superpositionsprinzips wird das aus der Menge der Einzel-ladungen resultierende Feld E^t bestimmt:

$$E^t = \sum_{x=-m}^m \sum_{y=-n}^n P_{m,n}^t(x, y) \cdot \frac{(x, y)}{\left(\sqrt{x^2 + y^2}\right)^3}$$

Das Feld E^t übt auf das geladene Teilchen am Kartenmittelpunkt eine Kraft aus, die eine zur Feldstärke E^t proportionale Beschleunigung dieses Teilchens bewirkt.

Die Feldkraft, die auf den Kartenmittelpunkt wirkt, wird auf Sollwerte für unterlagerte Längs- und Querregler des Fahrzeugs abgebildet.

3.2.6 Kritische Betrachtung des Ansatzes

Das hier kurz skizzierte Konzept der autonomen Fahrzeugführung nutzt eine abstrakte Beschreibung der Umgebung in Form von Gefahrenwerten. Die Vorteile dieser Gefahrenrepräsentation und der Interpretation als elektrisches Feld, aber auch die sich daraus ergebenden Probleme und Unzulänglichkeiten werden im Folgenden angeführt.

- *Die Betrachtung als elektrisches Feld stellt ein gutes Kriterium zur Pfadsuche zur Verfügung.*

Was ist der Grundgedanke des oben beschriebenen Ansatzes? Interpretiert man ein Feld von Gefahrenpotentialen als Ladungspotentiale, so ergibt sich ein elektrisches Feld. Definiert man nun ein Ziel, d.h. einen Pol auf diesem Feld, so ergeben sich, sofern es einen Weg gibt, Feldlinien, die von der aktuellen Position zum Ziel führen. Man braucht jedoch nicht den kompletten Pfad bis zum Ziel zu berechnen, sondern es genügt die Berechnung der lokalen Feldkraft und Richtung. Dabei wird jedoch nicht beachtet, daß es sich bei der autonomen Fahrzeugführung nicht in erster Linie um ein Suchproblem handelt. Auch geht obige Betrachtung von einer

statischen Umgebung aus, die im vorliegenden Fall nicht gegeben ist. Die ermittelten Feldlinien stellen jedoch eine brauchbare Heuristik für die Pfadsuche dar.

- *Kompliziertheit der Modellierung durch die direkte Verknüpfung von Gefahr und Reaktion.*

Auch wenn ein Zusammenhang zwischen der subjektiven Gefahr und der Verhaltensweise eines Fahrers besteht, so ist die direkte Proportionalität von Gefahrenwerten zu Fahrzeugbeschleunigungen im Sinne einer „natürlichen“ Fahrweise nicht gegeben.

Insbesondere bei Verwendung der Interpretation als elektrisches Feld ist die Form der Gefahrengebirge für die einzelnen Objekte der Umgebung dem Operator anzupassen, um die gewünschte Fahrzeugreaktion zu erzeugen. Dies gestaltet sich durch den globalen Charakter des Operators schwierig, da jeweils das gesamte Gefahrengebirge betrachtet wird. Dabei sind Einflüsse irrelevanter Objekte ohne eine zusätzliche Logik nicht auszuschließen. Beispielsweise würden zwei in Reihe vorausfahrende Fahrzeuge eine größere abstoßende Kraft und damit eine größere Verzögerung des autonomen Fahrzeugs hervorrufen als ein Fahrzeug. Damit vergrößert sich unsinnigerweise auch der Sicherheitsabstand, den das Fahrzeug einhält. Ein weiteres Beispiel ist der Überholvorgang. Dieser wird durch ein langsamer vorausfahrendes Fahrzeug ausgelöst. Dasselbe Fahrzeug bewirkt in der oben beschriebenen Modellierung jedoch gleichzeitig eine Verzögerung des autonomen Fahrzeugs aufgrund der gleichen Gefahrenpotentiale. Ein Überholvorgang ist demnach entgegen dem natürlichen Fahrverhalten zwangsläufig mit einer Verzögerung verbunden.

- *Durch die Fusion werden die Teilkarten verzerrt und damit die Reaktion verändert.*

Eine Abbildungsfunktion, die auf die Auswertung der Gefahrenkarte mit Hilfe eines elektrischen Feld Operators ausgelegt ist, generiert eine Teilkarte in der Form, daß die gewünschte Reaktion hervorgerufen wird. Zur Verdeutlichung des Problems betrachte man das in Abbildung 21 d) angeführte Beispiel einer gewichteten Verknüpfung zweier Teilkarten für eine Straßen- und Hindernisrepräsentation. Die Veränderung der Gefahrenwerte durch Gewichtung der Teilkarten kann durch einen Faktor wieder kompensiert werden.

Treten bei der Fusion verschiedene Gewichtungen auf, so läßt sich die Gewichtung nicht mehr wie oben beschrieben kompensieren, da unterschiedliche Faktoren

nötig wären. Es bleibt die Grundtendenz der Reaktion erhalten, eine Korrektheit der Reaktion ist jedoch nicht mehr zu gewährleisten. Eine mögliche Lösung innerhalb des Konzepts ist der Verzicht auf die explizite Gewichtung und die Einbeziehung der Gewichtung in die Generierung der Teilkarten.

- *Durch die Fusion der Teilkarten kommt es zu Veränderungen der modellierten Reaktion.*

In manchen Fällen sind diese Veränderungen erwünscht, sie treten jedoch auch auf, wenn zwei Teilkarten unabhängige Reaktionen hervorrufen sollen.

Beispielsweise ist die Straßenrepräsentation so ausgelegt, daß sich die Kräfte in Längsrichtung gegeneinander aufheben. Die durch die Hindernis-Teilkarte erzeugten Feldkräfte für die Längsrichtung bleiben jedoch bei der Kombination der Karten nicht erhalten. Durch die Maximumverknüpfung der Hindernisrepräsentation mit der Straßenrepräsentation werden sie verändert, sobald sich Gefahrenwerte für das Hindernis mit Gefahrenwerten für die Straße überlagern.

- *Durch die Interpretation als elektrisches Feld wird die Dynamik des Fahrzeugs vernachlässigt.*

Die Interpretation des Fahrzeugs als masseloses Teilchen in einem elektrischen Feld ist nur bedingt zur Generierung von Steuervorgaben in Form von Beschleunigungen zu verwenden. Die Fahrzeugdynamik muß näherungsweise in die Form der Gefahrenkarte integriert werden.

- *Durch den Aufbau der Gefahrengebirge ergeben sich Rechenzeitprobleme.*

Die eigentliche Entscheidung über die Fahrzeugreaktion wird durch eine einfache Berechnung aus dem elektrischen Feld abgeleitet. Dies darf aber nicht darüber hinwegtäuschen, daß die Generierung der zugrundeliegenden Gefahrenkarte in einer Qualität, die ein adäquates Fahrzeugverhalten garantiert, mit enorm großem rechnerischen Aufwand verbunden ist. Die Berechnungszeit der Gefahrenkarte liegt auf der verwendeten Hardware um einiges über der erforderlichen Echtzeit.

- *Das Konzept umfaßt keine schematischen Handlungen des Fahrers.*

Kurz vor einem Überholvorgang betätigt der Fahrer eines Fahrzeugs beispielsweise den Fahrtrichtungsanzeiger, um seine Absicht anzuzeigen. Eine solche geplante schematische Handlung wird durch das Konzept, welches allein auf der direkten Reaktion auf Umwelteinflüsse basiert, nicht abgedeckt.

Zusammenfassend läßt sich feststellen, daß die Gefahrenkarte in der Form, wie sie in diesem Abschnitt aufgebaut und interpretiert wird, einen guten Anhaltspunkt für die

Steuerung eines Fahrzeugs liefert. Um eine bessere Einbeziehung der Fahrzeugdynamik zu erreichen, gleichzeitig aber auch eine echtzeitfähige Implementierung zu ermöglichen, wird eine andere Methode gewählt, die eine Trennung von Gefahr und Reaktion vorsieht. Die im folgenden Abschnitt diskutierte Methode der Verhaltensmuster beruht weiterhin auf einer Gefahrenrepräsentation und integriert das Konzept der künstlichen Potentialfelder sowie gewonnene Erkenntnisse aus dem oben beschriebenen Ansatz zur Verhaltenssteuerung.

3.3 Das Verhaltensmusterkonzept

Bei der Fahraufgabe handelt es sich um eine Aufgabenstellung, die dem Menschen nach einer Lernphase prinzipiell keine großen Probleme bereitet. Es ist daher naheliegend, sich bei der Konzeption einer Verhaltenssteuerung für autonome Fahrzeuge an menschlichem Fahrverhalten zu orientieren. Das nachfolgend beschriebene Verhaltensmusterkonzept zur Steuerung eines autonomen Fahrzeugs ist durch psychologische Verhaltensmodelle und Betrachtungen des Menschen als Fahrzeugführer motiviert. Das Konzept beruht auf der Verknüpfung einer Menge von Handlungsschemata mittels eines Risikobegriffs.

Stellt man einen Vergleich zu der oben beschriebenen Gefahrenkarte an, so entsprechen die hier verwendeten Verhaltensmuster in etwa den Teilkarten, aus denen sich die Gefahrenkarte zusammensetzt. Das Verhaltensmusterkonzept verwendet ebenfalls die Gefahr als Verknüpfungsgrundlage, enthält aber einige Erweiterungen, die insbesondere zur Behebung der in Abschnitt 3.2.6 beschriebenen Probleme führen:

- Durch *Trennung von Längs- und Querführung* wird ein adäquates Durchführen eines Überholmanövers ermöglicht.
- Durch *Trennung von Gefahr und Reaktion* wird das Modellierungsproblem erleichtert und eine bessere Integration der Fahrzeugdynamik und schematischer Handlungen ermöglicht.

Einige Verfahren zur Steuerung autonomer mobiler Systeme orientieren sich an der menschlichen Informationsverarbeitung. Der hier vorgestellte Ansatz läßt sich der Gruppe der verhaltensbasierten Verfahren (behavioural architectures) zur Steuerung eines autonomen mobilen Systems zuordnen. Die bekannteste ist die von Rodney A. Brooks vorgestellte *Subsumptionsarchitektur*. Im Gegensatz zu traditionellen Architekturen von Kontrollsystemen unterteilt Brooks das Steuerungsproblem in „aufgabenerfüllende Verhaltensweisen“ anstelle von funktionalen Einheiten [Brooks 86]. Durch diese horizontale Architektur werden Sensoren enger mit Aktoren verknüpft, so daß u.a. schnelle Reaktionen ermöglicht werden. Zentrales Problem dieser Verfahren ist

die Entscheidung darüber, welche Verhaltensweise zu welchem Zeitpunkt aktiviert wird und welche die Kontrolle über die Aktoren bekommt.

Im Folgenden werden in Abschnitt 3.3.1 zunächst die psychologischen Erklärungsmodelle beschrieben, an denen sich das Verhaltensmusterkonzept orientiert. An dieser Stelle ist festzuhalten, daß nicht der Versuch unternommen wird, die menschliche Informationsverarbeitung auf einem technischen System nachzubilden, sondern diese ausschließlich als Motivation zu sehen ist.

In Abschnitt 3.3.2 folgt die Vorstellung des Konzepts der konkurrierenden Verhaltensmuster. Am Beispiel der Fahrzeugführung auf Autobahnen wird dieses Konzept dann in Abschnitt 3.3.3.5 detailliert beschrieben und auf die einzelnen Verhaltensmuster und deren Verknüpfung eingegangen.

3.3.1 Erklärungsmodelle des Fahrverhaltens

Auch wenn eingangs erwähnt wurde, daß die Fahraufgabe kein prinzipielles Problem für den Menschen darstellt, so ist diese Aussage im Hinblick auf eine Verwendung der menschlichen Verhaltensweisen als Vorbild für eine computergestützte autonome Fahrzeugführung näher zu betrachten. In Anbetracht der nicht geringen Zahl an Unfällen erscheint sie zumindest auf den ersten Blick als fragwürdig. Betrachtet man die Unfallstatistiken [StatJB 1991], so sind die fünf häufigsten Unfallursachen auf Fehlverhalten des Fahrers zurückzuführen:

- (1) Nicht angepaßte Geschwindigkeit
- (2) Fehler beim Abbiegen, Wenden etc.
- (3) Vorfahrtsfehler
- (4) Ungenügender Sicherheitsabstand
- (5) Mangelnde Verkehrstüchtigkeit

Es stellt sich die Frage, wodurch das menschliche Fehlverhalten ausgelöst wird. Onken bezeichnet die Informationsaufnahme und speziell die mangelnde Fähigkeit, aus der Fülle der Informationen die relevanten herauszufiltern, als Hauptdefizit des Menschen (vgl. [Onken 93]). Betrachtet man insbesondere den nicht ausreichenden Sicherheitsabstand und die nicht angepaßte Geschwindigkeit, so sind diese auf eine zu große Diskrepanz zwischen subjektivem und objektivem Risiko zurückzuführen. Die Kognition des Risikos hängt von dessen Wahrnehmungsmöglichkeit ab. Das Fehlverhalten kann also durch fehlerhafte oder zu geringe Wahrnehmungsmöglichkeit einerseits und durch eine zu hoch angesetzte Grenze des akzeptierten Risikos andererseits begründet werden. Eine Theorie des Risikoverhaltens besagt, daß sich Unfälle ereignen, weil das subjektive Risiko zu gering angesehen wird [Huguenin 88].

Klammert man die für die eigentliche Fahraufgabe irrelevanten und störenden Einflußfaktoren bei der Konzeption eines Systems zur autonomen Fahrzeugführung aus, so bietet sich die menschliche Informationsverarbeitung dennoch als Vorbild an. Dies gilt insbesondere, wenn man berücksichtigt, daß ein Großteil der damit ausgeschlossenen Einflußfaktoren als Ursache für Fehlverhalten zu sehen ist.

Betrachtet man den Fahrer eines Kraftfahrzeugs als Operateur eines Mensch-Maschine-Systems, so kann man seinem Verhalten das von Rasmussen [Rasmussen 83] vorgestellte Verhaltensmodell einer Bedienperson zugrunde legen. Rasmussen unterscheidet drei Arten menschlichen Verhaltens anhand des Grades der kognitiven Inanspruchnahme des Menschen:

- *Fertigkeiten* (skill-based behaviour),
- *regelbasiertes Verhalten* (rule-based behaviour)
- und *wissensbasiertes Verhalten* (knowledge-based behaviour).

Als Fertigkeiten im Sinne gewohnheitsmäßigen Handelns werden erlernte, stark automatisierte sensomotorische Verhaltensweisen bezeichnet. Sie erfordern keine willentliche Aufmerksamkeit. In bekannten Situationen ist das Verhalten zielgerichtet und es werden Handlungsschemata und a priori festgelegte Regeln verwendet. Diese Verhaltensweisen bezeichnet Rasmussen als regelbasiertes Verhalten. In unbekanntem Situationen, in denen keine vorgefertigten Handlungsschemata existieren, wird das Verhalten auf einer höheren Ebene von Verhaltensweisen bestimmt. Diese Verhaltensweisen sind zielgerichtet und werden als wissensbasiert bezeichnet.

Die verschiedenen Ebenen des Verhaltens erfordern auch unterschiedliche Formen der Information. Die Ebene der Fertigkeiten verwendet Informationen in Form von *Signalen*. Diese beschreiben das räumlich-zeitliche Verhalten der Umwelt in kontinuierlicher, quantitativer Form. Auf der Ebene des regelbasierten Verhaltens werden *Zeichen* als Informationsträger verwendet. Diese dienen zum Aktivieren und Modifizieren vordefinierter Handlungsschemata. Die Informationsträger der wissensbasierten Ebene sind *Symbole*. Abbildung 22 veranschaulicht das Drei-Ebenen Modell des menschlichen Verhaltens.

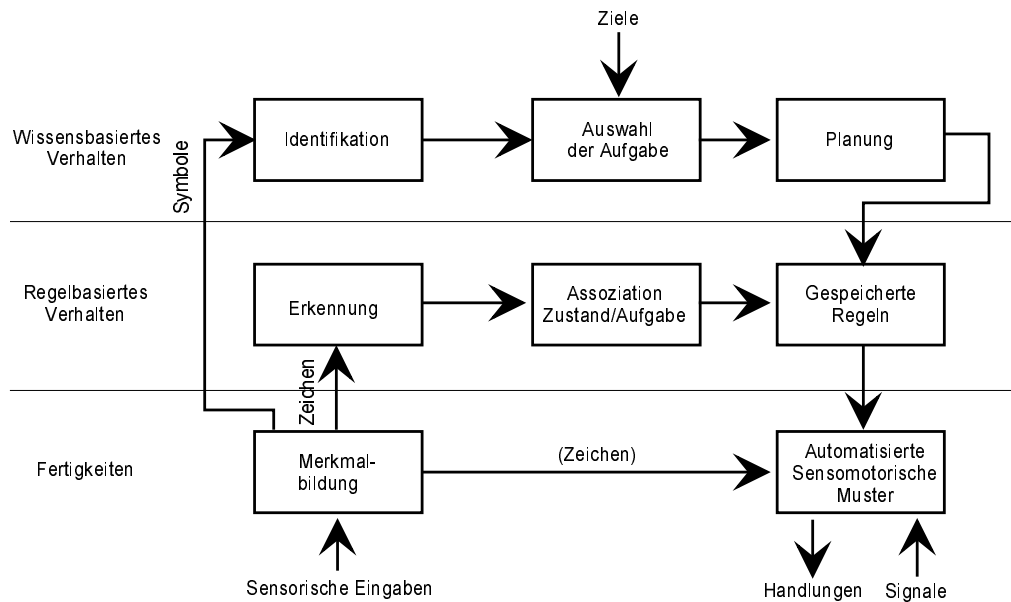


Abbildung 22 : Drei-Ebenen Verhaltensmodell nach Rasmussen [Rasmussen 83]

Johannsen [Johannsen 90] beschreibt eine hierarchische Mehrebenenstruktur der Fahrzeugführung. Er betrachtet die Navigationsaufgabe als oberste Ebene und die Stabilisierungsaufgaben als unterste Ebene. Die hierarchische Struktur ergibt sich aus ineinander verschachtelten Wirkungskreisen, wobei die Ziele eines Wirkungskreises vom jeweils übergeordneten vorgegeben werden. Insbesondere können die Aufgaben eines Wirkungskreises erst dann fortgeführt werden, wenn die nächstniedrigere Ebene ihre Aufgaben erfüllen kann. Johannsens Untergliederung der psychischen Prozesse bezüglich des Grades der kognitiven Inanspruchnahme des Menschen entspricht dem von Rasmussen beschriebenen Drei-Ebenen Modell.

Um eine adäquate Verhaltensweise im Straßenverkehr zu zeigen, wird Wissen über die Domäne benötigt. Nach Piaget [Piaget 71] ist Wissen mit einem Assimilationsprozeß und einem Handlungsschema verbunden. Ein Objekt oder ein Geschehen zu kennen wird nach Piagets Betrachtungsweise mit seiner Verwendung durch Assimilation auf ein Handlungsschema gleichgesetzt.

Betrachtet man unter diesem Gesichtspunkt die Handlungsschemata in Rasmussens Drei-Ebenen Modell, so umfaßt dieses auch die für das Handeln benötigte Wissensrepräsentation.

Neben den psychologischen Verhaltensmodellen ist die kybernetische Beschreibung des Verkehrssystems anzuführen. Ein kybernetisches System beschreibt die Prozesse innerhalb eines Verkehrssystems mit Hilfe von Begriffen des Regelkreises 'Führungsgröße', 'Störgröße', 'Regler' und 'Regelstrecke'. Dabei wird in diesem Fall die Sollage des Fahrzeugs bezüglich der Straße als Führungsgröße bezeichnet. Störgrößen sind andere Verkehrsteilnehmer, die Witterung und andere Faktoren, die

eine Verkehrssituation beschreiben. Der Fahrer selber wird als Regler betrachtet und das Fahrzeug als Regelstrecke [Huguenin 88, Bösch 91]. Abbildung 23 veranschaulicht diesen Regelkreis.

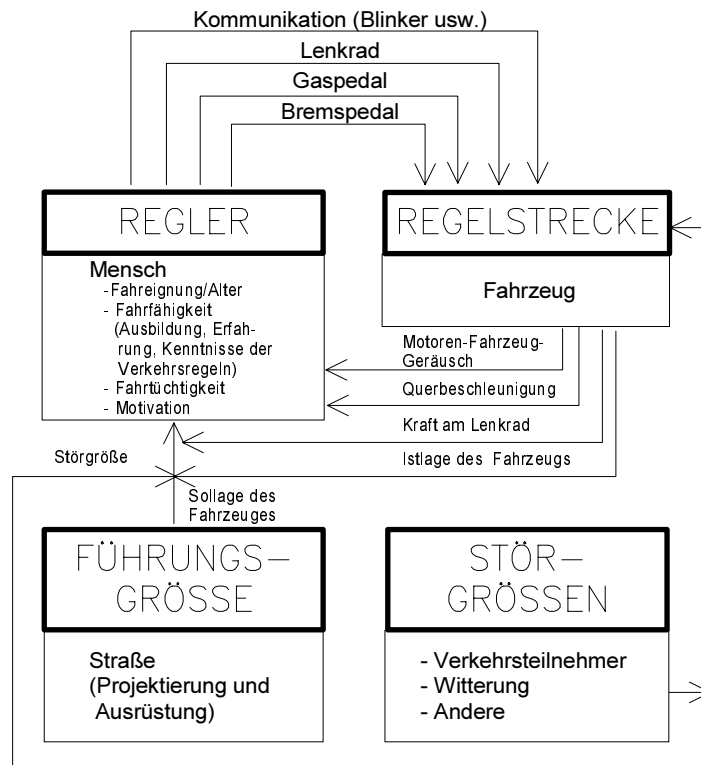


Abbildung 23 : Regelkreis "Mensch - Straße - Fahrzeug" nach Hehlen & Huguenin [Huguenin 88]

Die kybernetische Betrachtungsweise hat den Vorteil, dynamische Aspekte und funktionale Verknüpfungen darzustellen, jedoch lassen sich mit ihrer Hilfe keine Fahrerverhaltensmuster erklären [Huguenin 88].

3.3.2 Architektur des Verhaltensmusterkonzepts

Das Verhaltensmusterkonzept verbindet die Idee der Gefahrenmodellierung mit einem psychologisch motivierten Verhaltensmodell. Die vom Kontrollsystem durchführbaren Reaktionen auf die Umwelt sind in *Verhaltensmustern* organisiert. Unter Berücksichtigung eines bestimmten Aspekts der aktuellen Situation bestimmt ein Verhaltensmuster eine durch diesen Aspekt hervorgerufene, subjektive Gefährlichkeit und eine Reaktion des autonomen Fahrzeugs.

Nach paralleler Auswertung aller Verhaltensmuster wird durch Abwägung der mit den Verhaltensmustern verbundenen Gefährlichkeit das endgültige Verhalten in Form eines Steuerbefehls ermittelt. Dazu wird ein *Inhibitionsnetzwerk* verwendet, welches Abhängigkeiten und Gewichtungen zwischen den einzelnen Verhaltensmustern festlegt.

Jedes der Verhaltensmuster hat Zugriff auf ein *Fahrzeugmodell* und ein *Fahrermodell*. Das Gesamtfahrermodell setzt sich aus diesem Fahrermodell und den Gewichtungen des Inhibitionsnetzwerks zusammen.

Sind die Verhaltensmuster auch von der äußeren Form her gleich, so lassen sie sich semantisch in drei Hauptgruppen unterteilen:

- *Basisverhaltensmuster*,
- *Schematisierte Handlungen* und
- *Notreaktionen*.

Eine Navigationskomponente ist nicht integriert, ließe sich aber leicht hinzufügen (siehe Kapitel 5). Die Zielvorgabe beschränkt sich im hier betrachteten Fall auf eine vom Fahrer gegebene Wunschgeschwindigkeit sowie einen Spurwechselwunsch.

Als *Basisverhaltensmuster* werden Handlungen bezeichnet, die mit den stark automatisierten sensomotorischen Verhaltensweisen nach Rasmussen vergleichbar sind. Sie entsprechen der untersten Ebene der Fahrzeugführung, den Stabilisierungsaufgaben. Sie beschränken sich auf die Spurhaltung und die Abstandshaltung bei Kolonnenfahrt.

Die Gruppe der *schematisierten Handlungen* stellt den Hauptanteil der Verhaltensmuster dar. Zu dieser Gruppe gehören die Reaktion auf Verkehrszeichen, das Befolgen von Verkehrsregeln, das Überholen und alle weiteren Reaktionen auf Ereignisse in der Umgebung.

Die letzte Gruppe, die *Notreaktionen*, orientiert sich weniger an einem Fahrermodell als an den dynamischen Grenzen des Fahrzeugs. Sie stellen eine Rückfallebene dar, um in kritischen Situationen sehr schnell in einer Form zu reagieren, die nicht mehr der Befriedigung von Komfortansprüchen des Fahrers, sondern nur noch dem Vermeiden einer Kollision gilt.

Zieht man auch hier wieder den Vergleich zum Menschen, so zeigen sich sehr ähnliche Verhaltensweisen. Gerade in besonders kritischen Situationen würde man vom Menschen ein angemessenes wissensbasiertes oder regelbasiertes Verhalten erwarten. In der Praxis ist das Verhalten des Menschen in solchen Situationen, insbesondere unter Zeitdruck, jedoch stereotyp und damit eher auf unterster kognitiver Ebene angesiedelt [Johannsen 90].

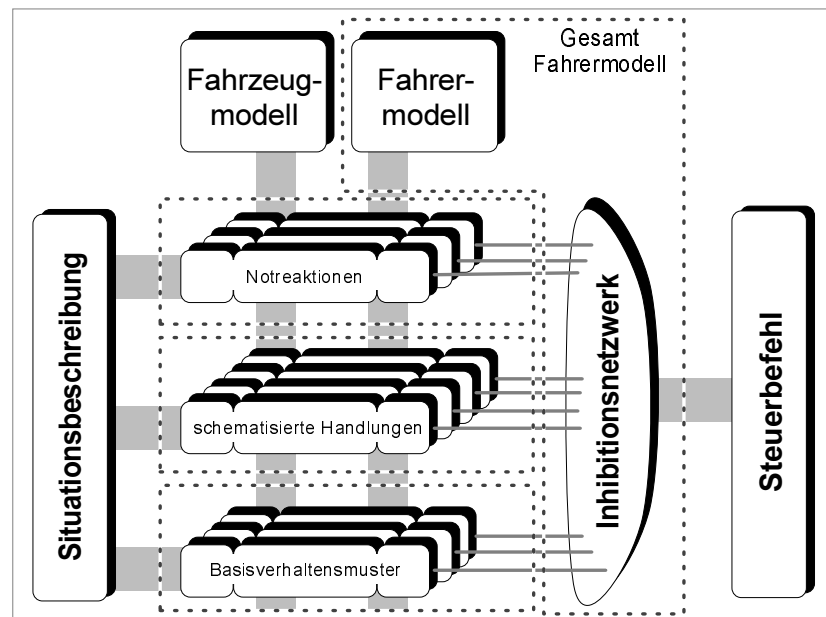


Abbildung 24 : Architektur des Kontrollsystems basierend auf Verhaltensmustern

Die Realisation der Notreaktionen ist an die in Abschnitt 3.2 diskutierte Gefahrenkarte bzw. an den zugrundeliegenden Potentialfeldansatz angelehnt. Damit ist eine Verhaltensebene gegeben, die jederzeit eine weitgehende Absicherung der übergeordneten Verhaltensweisen gegen Kollisionen bietet. Es kann jedoch keine Kollisionsfreiheit garantiert werden, da das Verhalten der anderen Verkehrsteilnehmer zu Situationen führen kann, in denen eine Kollision nicht vom eigenen Verhalten abhängt, bzw. eine Kollision im Rahmen der physikalischen Möglichkeiten nicht vermeidbar ist.

Abbildung 24 stellt die Architektur des Kontrollsystems schematisch dar. Aufgrund der flexiblen Struktur des Kontrollsystems ist es möglich, einzelne Verhaltensweisen zu deaktivieren oder zu verändern, ohne das Gesamtsystem zu blockieren. Damit sind einfache Anpassungen an den jeweiligen Kontext realisierbar. Beispielsweise ist das Rechtsfahrgebot im Kontext 'Innenstadtverkehr' zu deaktivieren. Von größter Wichtigkeit ist die sichere Realisierung der Notreaktionen. Fehler in den anderen Verhaltensmustern sollen durch die Notreaktionen abgefangen werden, so daß sie zwar zu unkomfortablem Verhalten des Fahrzeugs, nicht jedoch zu Kollisionen führen.

3.3.3 Wissensrepräsentation und Inferenz

Zur Lösung der Fahraufgabe wird Wissen über das Umfeld, das Fahrzeug und den zu modellierenden Fahrer benötigt. Der hier verwendeten Repräsentationsform liegt die These zugrunde, daß Wissen mit einem Assimilationsprozeß und einem Handlungsschema verbunden ist (vgl. [Piaget 71]). Jedes Verhaltensmuster enthält

Wissen über ein Element oder einen Aspekt einer Gesamtsituation, indem es diesem eine Handlung des Fahrers zuordnet.

Die Kombination der Verhaltensweisen führt dann zu einem Fahrverhalten, das aus dem verteilten Wissen abgeleitet ist. Interpretiert man die Handlung als Wissensträger, so läßt sich die Verhaltensmusterfusion als Inferenzschritt interpretieren. Insbesondere erfordert die Echtzeitanforderung eine strenge Begrenzung der Inferenztiefe, die durch die Verhaltensmusterfusion eingehalten wird. Die folgenden „Wissenspakete“ sind in Verhaltensmustern repräsentiert:

- Wissen über das autonome Fahrzeug, seine Freiheitsgrade der Bewegung und seine dynamischen Grenzen,
- Verkehrsregeln und
- heuristisches Wissen über Verhaltensweisen von anderen Verkehrsteilnehmern.

Gleichzeitig wird in den Verhaltensmustern berücksichtigt, wie ein Fahrer mit diesem Wissen *umgeht*, d. h. welche Konsequenzen es für sein Handeln hat. Je nach Art des modellierten Verhaltens werden klassische Repräsentationstechniken innerhalb eines Verhaltensmusters eingesetzt.

Allgemein führt jedoch die strenge Echtzeitanforderung der Applikation zu einer stark algorithmisch geprägten Repräsentationsform, die Fahrer- und Fahrzeugparameter berücksichtigt. Im Folgenden werden zunächst Fahrer- und Fahrzeugparameter vorgestellt. Anschließend wird der Aufbau eines Verhaltensmusters mit den dort verwendeten Repräsentationsformen beschrieben, und die Methode zur Verknüpfung von Verhaltensweisen erläutert.

3.3.3.1 Fahrer- und Fahrzeugparameter

Das Verhalten von Fahrern im Straßenverkehr ist sehr unterschiedlich. Fahrer unterscheiden sich u.a. in ihrem Sicherheitsempfinden, das sich hauptsächlich im Einhalten von Sicherheitsabständen zu anderen Verkehrsteilnehmern ausdrückt. Auch liegen unterschiedliche Handlungspräferenzen bei verschiedenen möglichen Verhaltensweisen vor. Beispielsweise führt die Entscheidung, in einer bestimmten Situation ein vorausfahrendes Fahrzeug zu überholen oder ihm in sicherem Abstand zu folgen, bei verschiedenen Fahrern zu unterschiedlichen Ergebnissen.

Dem Verhalten des autonomen Fahrzeugs liegt daher ein Fahrermodell zugrunde, das sich durch die Realisierung der Verhaltensmuster und der Verknüpfungsmethode, aber auch durch Fahrerparameter ausdrückt, die den Verhaltensmustern zur Initialisierung übergeben werden. Einige Fahrerparameter werden in nachstehender Tabelle aufgeführt und kurz beschrieben. Sie werden in den Verhaltensmustern als Orientierungsgrößen für die Generierung von Handlungen genutzt.

Attribut	Bedeutung
a+,a-	Maximal gewünschte Beschleunigungen zum Einhalten der Wunschgeschwindigkeit
a- annähern	Gewünschte maximale Verzögerung bei der Annäherung an ein vorausfahrendes Fahrzeug
a- folgen	Gewünschte maximale Verzögerung während dem Folgen eines vorausfahrenden Fahrzeugs
v überholen	Gewünschte Überschreitung der Wunschgeschwindigkeit während eines Überholvorgangs
v schwelle	Benötigte Differenz zwischen Wunschgeschwindigkeit und Geschwindigkeit eines vorausfahrenden Fahrzeugs zur Einleitung eines Überholvorgangs
t warten	Gewünschte Wartezeit bis zum Einleiten eines Spurwechsels
t _{spurwechsel(v)}	Gewünschte Zeit für einen Spurwechsel, abhängig von der Eigengeschwindigkeit
y toleranz	Toleranzbereich für die Abweichung von der Spurmitte
a _{quer}	Maximale Querschleunigung während der Spurführung
d _{fix}	Fixer Anteil des Sicherheitsabstands
d _{var}	Geschwindigkeitsabhängiger Anteil des Sicherheitsabstands, angegeben durch die Zeit bis zum Erreichen der aktuellen Position des vorausfahrenden Fahrzeugs
d _{lat}	Seitlicher Sicherheitsabstand zu anderen Fahrzeugen

Tabelle 1 : Attribute des Fahrermodells

Zur Anpassung der Steuerung des Fahrzeugs an dessen dynamische Grenzen wird ein Fahrzeugmodell integriert. Dieses besteht aus der algorithmischen Integration von physikalischen Zusammenhängen in die Verhaltensmuster und der expliziten Vorgabe von Grenzwerten anhand von Fahrzeugparametern. Die Fahrzeugparameter werden dazu verwendet, dieses allgemeine Wissen an das zugrundegelegte Fahrzeug anzupassen. Tabelle 2 beschreibt Fahrzeugattribute und ihre Bedeutungen.

Attribut	Bedeutung
l _{fzg} , b _{fzg}	Länge und Breite des autonomen Fahrzeugs
t _{sensoren}	Latenzzeit der Sensormodule (Durchschnittswert)
t _{gas}	Latenzzeit der Ansteuerung der Drosselklappe
t _{bremse}	Latenzzeit der Ansteuerung der Bremse
t _{lenkung}	Latenzzeit der Ansteuerung der Lenkung
t _{spurwechsel(v)}	Minimale Spurwechselzeit, abhängig von der Eigengeschwindigkeit
da _{Max} ⁻	Maximal zulässige Verzögerungsänderung
da _{Max} ⁺	Maximal zulässige Beschleunigungsänderung
a _{-max} , a _{+max}	Maximal zulässige Verzögerung und Beschleunigung
aq _{max} (v)	Betrag der maximal zulässigen Querschleunigung, abhängig von der Eigengeschwindigkeit
v _{max}	Zulässige Höchstgeschwindigkeit des autonomen Fahrzeugs

Tabelle 2 : Attribute des Fahrzeugmodells

3.3.3.2 Aufbau eines Verhaltensmusters

Ein Verhaltensmuster stellt einen allgemeinen Rahmen für die Repräsentation einer Verhaltensweise dar und beinhaltet keine feste innere Repräsentationsstruktur für die Handlung selbst. Als allgemeine Randbedingungen werden die Eingangsdaten und die Ausgangsdaten festgelegt. Eine semantische Unterteilung der Verhaltensmuster in drei Gruppen erscheint in Anlehnung an Rasmussens Verhaltensmodell sinnvoll. Entsprechend werden in den nachfolgend beschriebenen Verhaltensmustern unterschiedliche Interpretationen der eingehenden Datenströme zugrundegelegt. Während Basisverhaltensmuster die Eingangsdaten als Signale betrachten, werden sie durch die beiden anderen Gruppen auf symbolischer Ebene betrachtet.

Desweiteren werden die Verhaltensmuster in zwei Kategorien unterteilt: die Verhaltensmuster zur Bestimmung der Vorgaben für die *Längsführung* und die *Querführung* des autonomen Fahrzeugs.

Für die weitere Verwendung des Begriffs Verhaltensmuster soll hier eine formale Definition gegeben werden. Dazu wird zunächst der Begriff der *Projektion* eingeführt. Er bezeichnet im Folgenden die Bindung von Objektreferenzen an Elemente der Situationsbeschreibung (vgl. [Krüger 91]). Eine *Projektionsrestriktion* stellt eine Einschränkung der zulässigen Bindungen dar, die durch ein Prädikat auf der Menge der Objekte definiert ist.

Definition 3.3.1 : Verhaltensmuster

Ein Verhaltensmuster ist ein instanzierbares Schema. Es wird repräsentiert durch ein 4-Tupel (K,O,S,H) , welches aus einer Klassenbezeichnung K , einer Menge von Objektreferenzen O , einem Zustand S und einer Handlungsbeschreibung H besteht.

- *Die Klassenbezeichnung K ist durch eine a priori festgelegte Natürliche Zahl gegeben.*
- *Eine Objektreferenz $o \in O$ beinhaltet einen Zeiger auf ein Element der Situation und eine Typrestriktion sowie eine Projektionsrestriktion.*
- *Der Zustands S verwaltet die Attribute 'Aktiv', 'Blinker_Links', 'Blinker_Rechts' mit Wertebereich $\{0,1\}$, 'Basis', 'Gewicht', 'Hemmung' mit Wertebereich $[0,1]$, 'Intensität' mit Wertebereich \mathfrak{R} .*
- *Die Handlungsbeschreibung H beinhaltet Berechnungsvorschriften, die der Bestimmung der Attributwerte dienen. Als Parameter werden n Objektreferenzen sowie der Zustandsspeicher S übergeben.*

Es ist zu beachten, daß ein Verhaltensmuster ein zu instanzierendes Schema darstellt. Eine Instanz eines Verhaltensmusters wird durch die Bindung von Objekten der Situation an Objektreferenzen eines Verhaltensmusters gebildet. Es wird daher in Anlehnung an die Terminologie objektorientierter Darstellungen von einer *Verhaltensmusterinstanz* gesprochen, wenn es sich nicht um das allgemeine Schema handelt.

Für nachfolgende Verweise auf Verhaltensmuster wird folgende Konvention festgehalten:

- Ist v eine Verhaltensmusterinstanz und a ein Attribut des Speicherbereiches, dann bezeichnet $a(v)$ den Wert des Attributs a .
- Ist v eine Verhaltensmusterinstanz, dann bezeichnet $\text{Klasse}(a)$ die Klasse des Verhaltensmusters (in Form einer Natürlichen Zahl).

Durch die Klassenbezeichnung wird eine Priorität auf der Menge der Verhaltensmuster festgelegt. Notreaktionen erhalten höchste Priorität. Für die Basisverhaltensmuster und schematisierten Handlungen ist keine Priorität vorgeschrieben. In der Regel sind jedoch Basisverhaltensmuster niedriger einzustufen. Die *Hemmung* wird durch den Einfluß anderer Verhaltensmuster hervorgerufen, d. h. der Wert des Attributs wird von außen gesetzt. Der *Intensitätswert* beschreibt je nach Kategorie des Verhaltensmusters einen Längsbeschleunigungswert oder eine Quergeschwindigkeitsvorgabe. Die Werte für die Intensität der Vorgabe (d. h. die Höhe der Werte für die Beschleunigung und Quergeschwindigkeit) werden nur dann in Betracht gezogen, wenn die Verhaltensmusterinstanz in *aktivem* Zustand ist. Dagegen ist das Hemmen anderer Verhaltensmusterinstanzen auch dann möglich, wenn keine eigene Aktion generiert wird, die Verhaltensmusterinstanz also in *passivem* Zustand ist.

Der *Basiswert* und der *Gewichtswert* eines Verhaltensmusters dienen der Fusion der generierten Fahrvorgaben. Sie spiegeln die Risikokompensation wider, die durch Ausführen eines Verhaltensmusters erreicht wird. Dabei bezieht sich der Basiswert auf die Klasse des Verhaltensmusters. Er gibt an, bis zu welchem Grad die der Klasse zugehörige Priorisierung der Reaktion in Anspruch genommen werden soll. Es ist damit möglich, niedriger priorisierten Verhaltensmustern den Vorrang zu lassen. Die Nutzung dieser Option wird später erläutert. Neben dem Basiswert gibt die Gewichtung die Wichtigkeit der Einhaltung der Steuervorgaben an, die durch das Verhaltensmuster generiert werden.

3.3.3.3 *Repräsentation von Handlungsbeschreibungen*

Die in Definition 3.3.1 nicht näher spezifizierte Handlungsbeschreibung H läßt sich durch ein Übergangsnetz mit endlicher Zustandsanzahl repräsentieren. Dieser Repräsentation liegt eine Unterteilung der Handlung in aufeinanderfolgende Phasen zugrunde. Jede Phase entspricht einem Zustand des Übergangsnetzes. Dem Zustand ist eine Prozedur zugeordnet, welche die Handlung für die zugehörige Phase bestimmt.

Ein Zustandsübergang ist an eine Übergangsbedingung geknüpft. Die Übergangsbedingungen werden durch drei Arten von Prädikaten bestimmt:

- *Situationsprädikate*, die sich auf die Elemente der Situationsbeschreibung beziehen (z. B. Distanz zum vorausfahrenden Fahrzeug),
- *Zustandsprädikate*, die sich auf die Zustandsattribute des Verhaltensmusters beziehen (z. B. Basis(v)-Hemmung(v) > 0) und
- *Zeitprädikate*, die sich auf die Verweildauer in einem Zustand des Übergangnetztes beziehen.

Bei Instanziierung eines Verhaltensmusters wird ein vorbestimmter Anfangszustand gewählt. In jedem Zyklus werden zunächst die Übergangsbedingungen zu Folgezuständen geprüft. Solange diese erfüllt sind, wird das Netz durchlaufen. Ist ein Zustand erreicht, in dem keine Übergangsbedingungen mehr erfüllt sind, so wird die ihm zugeordnete Prozedur ausgeführt, um die Attribute des Verhaltensmusters zu berechnen. Anhand dieser Repräsentationsform für Handlungsbeschreibungen lassen sich die Verhaltensmustergruppen strukturell unterscheiden.

Basisverhaltensmuster

Die Gruppe der Basisverhaltensmuster ist dadurch gekennzeichnet, daß ihre Instanzen immer aktiv sind. Die Handlungsbeschreibung ist nicht in Phasen untergliedert. Ein Basisverhaltensmuster besteht demnach aus nur einem Zustand. Abbildung 25 stellt den Aufbau eines Basisverhaltensmusters graphisch dar.

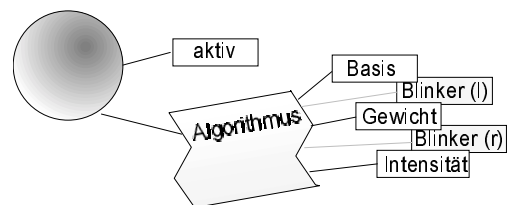


Abbildung 25 : Basisverhaltensmuster

Schematisierte Handlungen

Im Gegensatz zu den Basisverhaltensmustern lassen sich schematisierte Handlungen in Phasen unterteilen. Abbildung 26 zeigt ein Übergangnetz für schematisierte Handlungen. Der erste Zustand ist der Eingangszustand. Ihm ist keine Handlung zugeordnet. Ist eine Qualifikationsbedingung Q erfüllt, die kein Zeitprädikat enthält, so wird in Zustand 2 gewechselt. Dieser Zustand stellt die erste Phase der repräsentierten Handlung dar. Der Übergang zur Folgephase (Zustand 3) ist an eine Übergangsbedingung A geknüpft, die ein Zeitprädikat enthalten kann. Durch Anfügen weiterer Zustände ist eine Handlung mit mehreren Phasen darstellbar. Eine

Rückkehr in eine vorangehende Phase ist möglich (hier dargestellt durch Bedingung RA). Insbesondere wird in Zustand 1 zurückgegangen, sobald die Qualifikationsbedingung nicht mehr erfüllt ist.

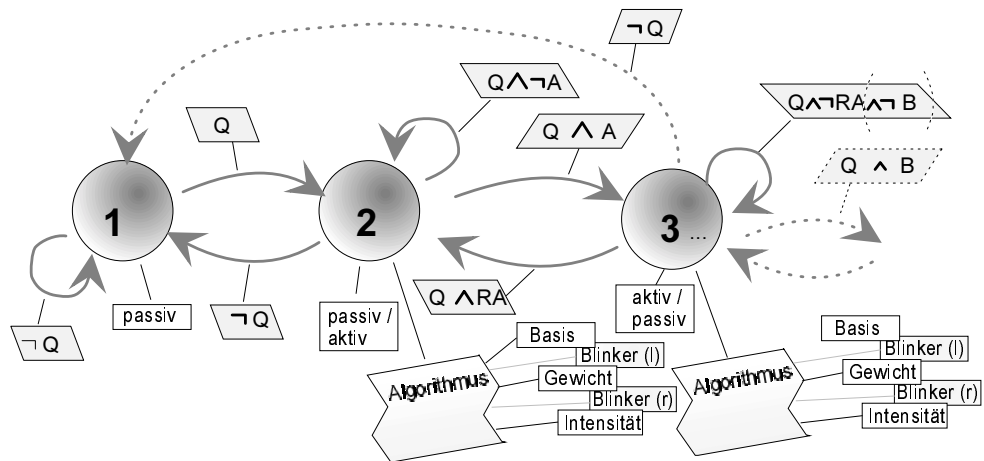


Abbildung 26 : Übergangsnetz zur Repräsentation von Handlungsabläufen

Notreaktionen

Die Verhaltensmuster für Notreaktionen sind nicht in Handlungsphasen unterteilt. Ist eine Qualifikationsbedingung Q erfüllt, so wird die Notreaktion durchgeführt. Dabei entscheidet das Verhaltensmuster selbst, ob die Reaktion aktiv oder passiv (durch Hemmung anderer Verhaltensmuster) erfolgt.

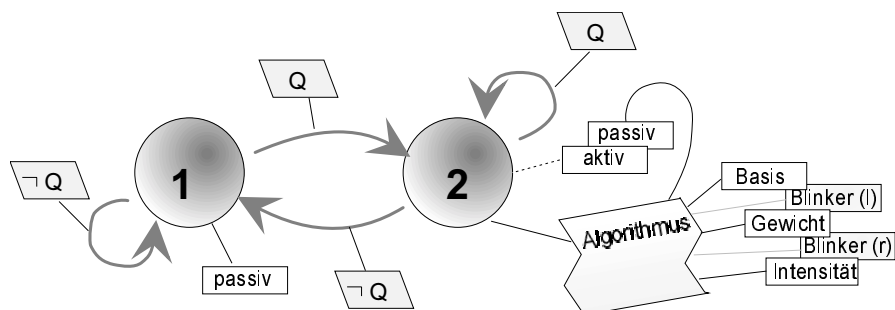


Abbildung 27 : Übergangsnetz für Notreaktionen

3.3.3.4 Repräsentation von Abhängigkeiten zwischen Verhaltensmustern

Zur Realisierung der Verknüpfung von Verhaltensmustern wird ein *Inhibitionsnetzwerk* aufgebaut. Die Knoten des Netzwerks repräsentieren die Verhaltensmusterinstanzen. Die Kanten des Netzwerks sind gerichtet und stellen hemmende Verbindungen zwischen zwei Verhaltensmusterinstanzen dar. Beispielsweise wirkt sich die Aktivierung einer Verhaltensmusterinstanz zum Überholen hemmend auf eine zur Einhaltung des Rechtsfahrgebots aus.

Jeder Kante ist ein Faktor zugeordnet, der die Stärke der Hemmung bestimmt. Das Inhibitionsnetzwerk ist wie folgt definiert:

Definition 3.3.2 : Inhibitionsnetzwerk

Das Inhibitionsnetzwerk IN besteht aus einer Menge von Knoten N_{IN} und einer Menge von gerichteten Kanten E_{IN} . Die Menge der Knoten entspricht der Menge der generierten Verhaltensmusterinstanzen. Die Menge E_{IN} ist definiert als

$$E_{IN} \subseteq \{(v_a, v_b, w, k) \mid v_a, v_b \in N_{IN}, w \in \mathfrak{R}_0^+, k \in N_0^+\}$$

Dabei bezeichnet w den Hemmungsfaktor zwischen den Verhaltensmustern v_a und v_b . Als Einschränkung gilt:

$$\forall e_1, e_2 \in E_{IN}: (e_1 = (a, b, w_1) \wedge e_2 = (a, b, w_2)) \rightarrow e_1 = e_2$$

Auf der Menge der Kanten E_{IN} ist eine reflexive Quasiordnung \leq_E definiert durch $e_1 = (a_1, b_1, w_1, k_1) \leq_E e_2 = (a_2, b_2, w_2, k_2) \Leftrightarrow k_1 \leq k_2$.

Die Kanten werden in n -Äquivalenzmengen gruppiert. Eine n -Äquivalenzmenge enthält alle Kanten $e = (a, b, w, k) \in E_{IN}$ mit $k = n$. Es bezeichnet $AQ(n)$ eine n -Äquivalenzmenge und $n_{MAX} = \max \{ k \mid e = (v_1, v_2, w, k) \in E_{IN}(t) \}$.

Durch das Inhibitionsnetzwerk wird insbesondere eine Verbindung zwischen Längs- und Querführung hergestellt. Der erzielbare Effekt läßt sich am besten durch das Überholen verdeutlichen. In Abschnitt 3.2 beinhaltet ein Kritikpunkt an der dort diskutierten Methode die Unvermeidbarkeit eines Verzögerns beim Überholvorgang. Dieses Problem ist hier durch eine hemmende Verbindung zwischen dem Verhaltensmuster zum Überholen und dem Verhaltensmuster zum Abstandhalten lösbar. Durch die Hemmung wird das Verhaltensmuster zum Abstandhalten erst später aktiviert, nämlich dann, wenn das abstandsbedingte Risiko größer als die Hemmung ist.

Das Inhibitionsnetzwerk alleine ist nicht ausreichend, um alle Konflikte zwischen Verhaltensmustern zu lösen. Es gibt Verhaltensmuster, die alternativ verwendet werden müssen, um ihren Zweck zu erfüllen. Ein Beispiel dafür ist gegeben, wenn das autonome Fahrzeug exakt auf ein Hindernis zufährt, so daß dem Ausweichen nach links und rechts die gleiche Gewichtung zukommt.

Um diese Eigenschaft zu repräsentieren, wird eine Menge von Exklusivpaaren $EV = \{(a, b) \mid a, b \in V\}$ verwaltet.

3.3.3.5 Algorithmus zur Verhaltenssteuerung

Das Verhalten des autonomen Fahrzeugs wird durch das Zusammenspiel der Menge der konkurrierenden Verhaltensmuster bestimmt. Nach der parallelen Auswertung der Verhaltensmusterinstanzen erfolgt die Kombination der Verhaltensweisen in zwei Schritten. Der erste Schritt beschreibt den Wettbewerb der Verhaltensmuster um die Systemkontrolle. Diejenigen Verhaltensmusterinstanzen, die nach diesem Wettbewerb noch aktiv sind, werden gewichtet verknüpft, um einen Steuervektor zu erhalten.

Der Algorithmus zur Verhaltenssteuerung ist in drei voneinander trennbare Phasen unterteilt, die in jedem Programmzyklus sequentiell durchlaufen werden. In der weiteren Beschreibung des Algorithmus bezeichnet

- VM die Menge der Verhaltensmuster,
- VMI(t) die Menge der Verhaltensmusterinstanzen zum Zeitpunkt t und
- $E_{IN}(t)$ die Menge der Kanten des Inhibitionsnetzwerks zum Zeitpunkt t.

Stimulation

Die Verhaltensmuster werden mit der aktuell vorliegenden Situationsbeschreibung konfrontiert. Durch Bindung der Objektreferenzen an Elemente der Situationsbeschreibung werden die Verhaltensmuster instanziiert.

Betrachtet man die Zahl der Objektreferenzen eines Verhaltensmusters, so ergibt sich die Problematik der kombinatorischen Komplexität. Bei n Objektreferenzen und m möglichen Elementen der Situationsbeschreibung berechnet sich die Zahl der möglichen Projektionen p zu

$$p = \prod_{i=0}^{n-1} (m - i).$$

Es sind daher Einschränkungen vorzunehmen, um die Projektion effizienter zu gestalten. Eine erste Einschränkung ist durch die Typrestriktion der Objektreferenzen gegeben. Da in einer Situationsbeschreibung nur jeweils ein Element des Typs Straße (Str), Benutzereingabe (Be) und Fahrzeugzustand (Z) existiert, ergibt sich die Problematik nur bei Betrachtung der Hindernisse (H) und der Verkehrszeichen (Vz).

Die Idee des Verhaltensmusterkonzepts ist es, die Beziehungen zwischen Objekten durch Verknüpfung der von ihnen ausgehenden Gefahren und Handlungsanweisungen darzustellen. Dadurch wird eine effiziente Projektion ermöglicht. Für die Modellierung der Verhaltensweisen werden hier Verhaltensmuster verwendet, die jeweils nur maximal *eine* Objektreferenz des Typs Hindernis und des Typs Verkehrszeichen enthalten. Dadurch wird eine lineare Komplexität der Projektion erreicht.

Wird eine Verhaltensmusterinstanz erzeugt, so wird sie in die Menge VMI(t) eingetragen. Eine Verhaltensmusterinstanz bleibt bestehen, bis ein referenziertes Objekt nicht mehr existiert. Bei Auflösung einer Referenz wird es aus der Menge VMI(t) gelöscht. Zusätzlich werden zwei temporäre Mengen EIN(t) und AUS(t) verwaltet, die zum Zeitpunkt t erzeugte und gelöschte Verhaltensmusterinstanzen beinhalten.

Die *Stimulationsphase* besteht nun aus der Instanziierung von Verhaltensmustern und der Bestimmung der Attribute durch die Berechnungsvorschrift der Verhaltensmusterinstanzen.

Hemmung

Als *Hemmungsphase* werden das Durchlaufen des Inhibitionsnetzwerks sowie die anschließende Elimination von Exklusivpaaren zusammengefaßt.

Durch die Veränderlichkeit der Menge $VMI(t)$ ist auch das Netzwerk *dynamisch*. Es bedarf daher einer Konstruktionsvorschrift. Diese besteht aus einer Menge von *Kantenschemata* $KS \subset VM \times VM \times \mathfrak{R}^+_0 \times \mathbb{N}^+_0 \times \{\text{alle, andere, gleiche}\}$. Elemente von KS sind 5-Tupel der Form $k = (v_1, v_2, w, k, \text{typ})$. Für Elemente der Menge $EIN(t)$ werden neue Kanten in das Netzwerk eingefügt, die es mit den bestehenden Verhaltensmusterinstanzen verbinden. Wird eine Verhaltensmusterinstanz gelöscht, so werden alle mit ihr verbundenen Kanten entfernt.

Die Einfügung erfolgt nach folgendem Algorithmus:

- (1) Sei $v \in EIN(t)$ eine neue Verhaltensmusterinstanz und bezeichne $M(v)$ das zu einer Instanz v gehörende Verhaltensmuster. Ist $EIN(t)$ leer, so wird keine Änderung vorgenommen, und der Algorithmus ist beendet.

Ansonsten wird zunächst die Menge der relevanten Kantenschemata KS_v bestimmt. Diese ist die Vereinigung der Menge der ausgehenden (KSA_v) und der eingehenden Kantenschemata (KSE_v):

$$KSA_v = \{ks \in KS \mid \exists vm \in VMI(t), \exists k \in KS, \exists w, k, \text{typ}: ks = (M(v), M(vm), w, k, \text{typ})\}$$

$$KSE_v = \{ks \in KS \mid \exists vm \in VMI(t), \exists k \in KS, \exists w, k, \text{typ}: ks = (M(vm), M(v), w, k, \text{typ})\}$$

- (2) Im zweiten Schritt werden die Instanzen (Kanten) der Kantenschemata aus KS_v erzeugt. Dabei bezieht sich die Typangabe des Kantenschemas auf die Bindung einer Objektreferenz an ein Element des Typs Hindernis und bestimmt, welche Kanten generiert werden. Enthält ein Verhaltensmuster keine Hindernisreferenz, so ist das zugehörige Kantenschema vom Typ 'alle'.

Folgende Fallunterscheidung wird zur Instanziierung eines Kantenschemas für eine Verhaltensmusterinstanz v vorgenommen:

typ = 'alle' : Ist $ks = (v_1, v_2, w, k, \text{typ})$ ein Kantenschema aus KSA_v , so wird für alle $vm \in VMI(t)$ mit $M(vm) = v_2$ eine Kante $e = (v, vm, w, k)$ in $EIN(t)$ eingefügt.

typ = 'andere' : Bezeichnet $rh(v)$ das durch v referenzierte Hindernis, so wird für jedes Kantenschema $ks = (v_1, v_2, w, k, typ) \in KSA_v$ für alle $vm \in VMI(t)$ mit $M(vm) = v_2$ und $rh(v_1) \neq rh(v_2)$ eine Kante $e = (v, vm, w, k)$ in $E_{IN}(t)$ eingefügt.

typ = 'gleiche' : Zu jedem Kantenschema $ks = (v_1, v_2, w, k, typ) \in KSA_v$ wird für alle $vm \in VMI(t)$ mit $M(vm) = v_2$ und $rh(v_1) = rh(v_2)$ eine Kante $e = (v, vm, w, k)$ in $E_{IN}(t)$ eingefügt.

Analog wird mit den Kantenschemata der Menge KSE_v verfahren.

Nach eventuellen Veränderungen des Netzwerks werden die Hemmungen der Verhaltensmusterinstanzen berechnet. Die Hemmung wird nach folgendem Algorithmus bestimmt:

(1) Setze $Hemmung(v) = 0$ für alle $v \in VMI(t)$. Setze $n = n_{MAX}$.

(2) Berechne die Hemmungen der n -Äquivalenzmenge *parallel*:

Es bezeichne $w_e(a, b) \in \mathfrak{R}^+$ den Hemmungsfaktor einer Kante $e = (a, b, w, k)$ zwischen Verhaltensmuster a und b . Für jede Kante $e = (v_1, v_2, w, k) \in AQ(n)$ wird die Hemmung $h(v_1, v_2)$ berechnet: $h(v_1, v_2) = w_e(v_1, v_2) \cdot \max((Basis(v_1) - Hemmung(v_1), 0)$.

(3) Für alle $v \in VMI(t)$ wird $Hemmung(v) = \max \{ x \mid x = h(a, v), a \in VMI(t) \}$ berechnet.

(4) Ist $n > 0$, so setze $n = n - 1$ und gehe zu Schritt (2).

(5) Setze den Wert $Aktiv(v) = 0$ für alle $v \in VMI(t)$ mit $Basis(v) - Hemmung(v) \leq 0$.

Nach Hemmung der Verhaltensmuster wird die Menge der Exklusivpaare EV untersucht. Ist für zwei Verhaltensmuster a, b , die als Paar (a, b) in EV eingetragen sind, $Basis(a) - Hemmung(a) > 0$ und $Basis(b) - Hemmung(b) > 0$, so wird eines der beiden deaktiviert. Das Verhaltensmuster mit dem höheren verbleibenden Gewicht $w = Gewicht + (Basis - Hemmung) \cdot Klasse$ bleibt aktiv. Bei Gleichheit wird Verhaltensmuster a des Paares (a, b) gewählt.

Fusion und Generierung eines Steuervektors

Nachdem einige der Verhaltensmuster durch den direkten Wettbewerb deaktiviert worden sind, werden die verbleibenden aktiven Verhaltensmuster zur Generierung eines Steuervektors Sv_t herangezogen. Dabei sind zwei Aufgaben zu unterscheiden :

- die Erzeugung der Steuervorgaben für Längs- und Querführung des autonomen Fahrzeugs und
- das Ansteuern der Fahrzeugperipherie, d.h. in diesem Fall der Blinker.

Zur Erzeugung der Vorgaben für die Längsbeschleunigung und die Quergeschwindigkeit werden die jeweiligen Intensitätswerte der Verhaltensmuster gewichtet verknüpft. Dazu werden sowohl die a priori festgelegten Gewichtungen herangezogen, die an die Klasse des Verhaltensmusters geknüpft sind, als auch die aktuellen Attributwerte von 'Basis' und 'Gewicht'. Jeder Klasse ist eine Gewichtungsstufe zugeordnet. Der Wert des Attributs 'Basis' beschreibt den Grad der Nutzung der klassenspezifischen Gewichtung. Hat das Attribut 'Basis' den Wert Eins, so werden die Verhaltensmuster niedrigerer Klassen vernachlässigt. Betrachtet man eine Klasse als Präferenzniveau, so erhält man durch Multiplikation des Basiswerts mit der Klasse einen kontinuierlichen Präferenzwert für ein Verhaltensmuster. In einer strengen Präferenzinterpretation werden bei der Verknüpfung nur Werte der obersten, aktivierten Präferenzebene berücksichtigt. Diese Einschränkung wird hier aufgeweicht, so daß auch Verhaltensmuster mit geringerem Präferenzwert berücksichtigt werden, sofern das ihnen zugeordnete Gewicht hoch ist.

Bezeichnet M die Menge der Verhaltensmusterinstanzen, M_{Long} die Menge der Instanzen zur Bestimmung der Längsführung, M_{Lat} die zur Bestimmung der Querverführung, dann werden die Steuervorgaben nach der folgenden Berechnungsvorschrift ermittelt.

Zunächst wird für eine zu verknüpfende Menge M von Verhaltensmusterinstanzen der maximale Präferenzwert b_M unter Berücksichtigung der Hemmung bestimmt.

$$b_M = \max_{m \in M, \text{Aktiv}(m)=1} ((\text{Basis}(m) - \text{Hemmung}(m)) \cdot \text{Klasse}(m))$$

Im zweiten Schritt werden die Gewichtungen für die anschließende Verknüpfung bestimmt. Dabei werden die Gewichte der Verhaltensmusterinstanzen höchster Präferenz übernommen und die der anderen entsprechend der nachstehenden Formel reduziert.

$$w_m^M = \text{Gewicht}(m) + (\text{Basis}(m) - \text{Hemmung}(m)) \cdot \text{Klasse}(m) - b_M$$

Es werden nur die Verhaltensmusterinstanzen weiter berücksichtigt, deren Gewichtung nicht negativ ist. Die übrigen werden deaktiviert. Zur Ermittlung der Steuervorgaben für Längsbeschleunigung und Quergeschwindigkeit werden für die beiden Verhaltensmuster Mengen M_{Long} und M_{Lat} die normierten gewichteten Summen der Intensitätswerte der aktiven Verhaltensmuster gebildet.

$$\begin{aligned}
 s_{Long} &= \sum_{m \in M_{Long}, w_m \geq 0} w_m^{M_{Long}} & s_{Lat} &= \sum_{m \in M_{Lat}, w_m \geq 0} w_m^{M_{Lat}} \\
 ws_{Long} &= \sum_{m \in M_{Long}, w_m \geq 0} w_m^{M_{Long}} \cdot \text{Intensität}(m) & ws_{Lat} &= \sum_{m \in M_{Lat}, w_m \geq 0} w_m^{M_{Lat}} \cdot \text{Intensität}(m) \\
 a_K(Sv_t) &= \frac{ws_{Long}}{s_{Long}} & v_{KLat}(Sv_t) &= \frac{ws_{Lat}}{s_{Lat}}
 \end{aligned}$$

Die Steuervorgaben $v_K(Sv_t)$ und $y_{KLat}(Sv_t)$ werden durch Integration über die Zeit ermittelt. Zur Verdeutlichung des Fusionsprinzips zeigt Abbildung 28 das Beispiel einer Verknüpfung von drei Verhaltensmustern.

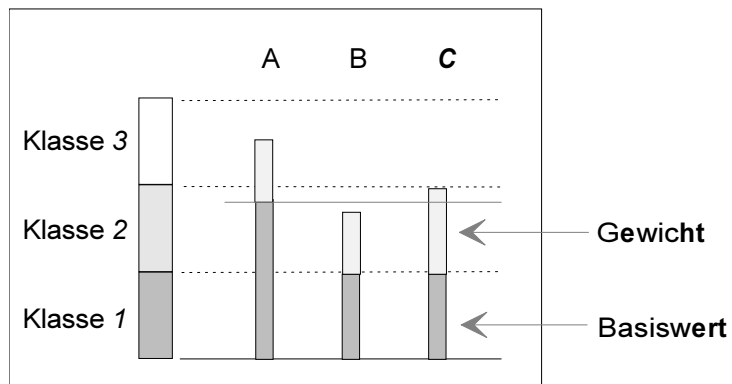


Abbildung 28 : Gewichtung und Verknüpfung von Verhaltensmustern

Beispiel:

Die Verhaltensmusterinstanzen A, B und C gehören zu unterschiedlichen Klassen der Menge der Verhaltensmuster zur Bestimmung der Längsführung. Die Intensitätswerte sind demnach Längsbeschleunigungsvorgaben des autonomen Fahrzeugs gleichzusetzen. Die Attributwerte der Verhaltensmusterinstanzen sind in der nachstehenden Tabelle aufgeführt:

Verhaltensmuster	Klasse	Gewicht	Basis	Hemmung	Intensität	Aktiv
A	3	0,7	0,80	0,2	0,54	1
B	2	0,7	0,60	0,1	-0,12	1
C	1	1,0	1,00	0,0	0,32	1

Tabelle 3 : Beispiel einer Verknüpfung von Verhaltensmustern

Die maximale Basis b berechnet sich zu

$$b = \max((0,8-0,2) \cdot 3, (0,6-0,1) \cdot 2, (1,0-0,0) \cdot 1) = 1,8$$

Damit ergeben sich die Gewichte der Verhaltensmusterinstanzen zu

$$w_A = 0,7 + (0,8-0,2) \cdot 3 - b = 0,7$$

$$w_B = 0,7 + (0,6-0,1) \cdot 2 - b = -0,1$$

$$w_C = 1,0 + (1,0-0,0) \cdot 1 - b = 0,2$$

Insgesamt führt die Kombination der Verhaltensmusterinstanzen zu der Längsbeschleunigungsvorgabe

$$\begin{aligned} a_k(Sv_t) &= \frac{w_A \cdot \text{Intensität}(A) + w_C \cdot \text{Intensität}(C)}{w_A + w_C} \\ &= \frac{0,7 \cdot 0,54 + 0,2 \cdot 0,32}{0,7 + 0,2} = \frac{0,442}{0,9} = 0,49\bar{1} \frac{m}{s^2} \end{aligned}$$

Aus der vorangehenden Geschwindigkeitsvorgabe und der aktuellen Beschleunigungsvorgabe läßt sich v_{KLong} unter Einbeziehung der Zykluszeit t_{zyklus} des Programms berechnen.

$$v_K(Sv_t) = v_K(Sv_{t-1}) + a_K(Sv_t) \cdot t_{zyklus}$$

Legt man eine Zykluszeit $T_{BC} = 40$ ms und $v_K(Sv_{t-1}) = 22$ m/s zugrunde, so ergibt sich

$$v'_{KLong} = 22 + 0,49\bar{1} \cdot 0,04 = 22,0196\bar{4} \frac{m}{s}$$

Neben der Bestimmung der Vorgaben zur Längs- und Querverführung des autonomen Fahrzeugs werden auch die Fahrtrichtungsanzeiger angesteuert. Dazu wird für jeden Blinker ein Zustand verwaltet, der durch die Verhaltensmuster geändert werden kann. Durch eine disjunktive Verknüpfung der Blinkerbefehle der Verhaltensmuster m für die gilt $\text{Basis}(m)\text{-Hemmung}(m) > 0$, wird zu jedem Zeitpunkt ein gemeinsamer Blinkerbefehl generiert. Um Instabilitäten im Grenzbereich zwischen Verhaltensmustern zu vermeiden, wird eine Konfidenzzeit t_k vereinbart. Während dieser Konfidenzzeit muß der Blinkerbefehl konstant sein, um einen Zustandswechsel zu akzeptieren.

In nachfolgender Formel bezeichnen $b_{Links}(t)$, $b_{Rechts}(t)$ die aus den Verhaltensmustern gewonnenen Blinkerbefehle zum Zeitpunkt t .

Die Zahl $n_k = (t_k \bmod t_{zyklus})$ bezeichnet die Anzahl der Konfidenzzyklen.

Die Blinkerzustände Z_{links} und Z_{rechts} sowie die daraus folgende Blinkeransteuerung ergeben sich dann wie folgt:

$$b_{Links}(t) = \max_{m \in M, Aktiv(m)=1} (Blinker_Links(m))$$

$$Z_{links}(t) = \begin{cases} 1 - \prod_{i=0}^{n_k} (1 - b_{Links}(t - i \cdot t_{zyklus})) & Z_{links}(t - t_{zyklus}) = 1 \\ \prod_{i=0}^{n_k} (b_{Links}(t - i \cdot t_{zyklus})) & Z_{links}(t - t_{zyklus}) = 0 \end{cases}$$

$$Blinker_{Links}(Sv_t) = Z_{links}(t)$$

Die Vorgaben für den rechten Blinker berechnen sich analog.

3.3.4 Verhaltensmuster für die Fahrzeugführung auf Autobahnen

Das oben vorgestellte Konzept zur autonomen Fahrzeugführung stellt einen allgemein gehaltenen Rahmen dar. Die Erfüllung der gestellten Fahraufgabe hängt hauptsächlich von der Wahl und der Realisierung der Verhaltensmuster ab, aus denen sich das System zusammensetzt. Auch wenn der hier vorgestellte Rahmen allgemeiner gefaßt ist, so ist das Ziel der vorliegenden Arbeit eingeschränkt auf die autonome Fahrzeugführung auf Autobahnen. Die dafür benötigten Verhaltensmuster werden im Folgenden näher erläutert. Dabei ist zu beachten, daß durch die nachfolgend grob beschriebenen Verhaltensmuster eine *subjektive* Modellierung des Fahrverhaltens gegeben ist, die keinen Anspruch auf Allgemeingültigkeit erhebt, sondern vielmehr eine exemplarische Fahrermodellierung darstellt.

Das realisierte Verhaltenssteuerungssystem für die Autobahnumgebung nimmt eine Einteilung der Verhaltensmuster in vier Prioritätsklassen vor, die sich aufgrund der verwendeten Menge von Verhaltensmustern als geeignet erwiesen hat. Den Notreaktionen wird die Klasse vier zugeordnet. Die übrigen Verhaltensmuster werden den Klassen eins bis drei zugeteilt. Die folgenden Aspekte werden beim Aufbau der Verhaltensmuster berücksichtigt:

- (1) *Unsicherheit*: Die Geschwindigkeitsmessung und Positionsmessung sind fehlerbehaftet. Dies muß bei Auslösung und Durchführung der Reaktion berücksichtigt werden.
- (2) *Zeitverzögerungen*: Durch die Übermittlung der Daten vom Sensorsystem zum Kontrollsystem und von dort zum Aktorsystem entstehen Zeitverzögerungen, die einbezogen werden müssen.
- (3) *Fahrzeugträgheit und Aktoren*: Die gewünschte Verzögerung ist nicht sofort erreicht. U. a. muß hier auch der aktuelle Zustand des autonomen Fahrzeugs mitberücksichtigt werden.

Alle Verhaltensmuster verwenden eine Korrektur der Eingangsdaten um Verzögerungen bei der Kommunikation zwischen den einzelnen Modulen zu kompensieren. Den Berechnungen wird beispielsweise nicht die aktuelle, sondern eine in die Zukunft projizierte Position eines Hindernisses H zugrundegelegt, die wie folgt bestimmt wird:

$$Distanz_{Long}^*(H) = Distanz_{Long}(H) + (v_H(H) - v^*) \cdot (t_{bremse} + t_{Sensoren})$$

$$Distanz_{Lat}^*(H) = Distanz_{Lat}(H) - v_{Lat}(Z_t) \cdot (t_{bremse} + t_{Sensoren})$$

Dabei bezeichnet v^* eine anhand der aktuellen Beschleunigung des autonomen Fahrzeugs korrigierte Geschwindigkeit für die Projektionszeitspanne.

$$v^* = v_{Long}(Z_t) + \frac{1}{2} a(Z_t) \cdot (t_{Sensoren} + t_{bremse})$$

Den Berechnungen liegt die Annahme zugrunde, daß laterale Position und longitudinale Geschwindigkeit des vorausfahrenden Fahrzeugs sowie die Beschleunigung des autonomen Fahrzeugs innerhalb der Projektionszeitspanne konstant sind.

Zur Behandlung der fehlerbehafteten Eingangsdaten wird ein diskreter zeitabschnittsbezogener Konfidenzwert definiert. $C_T(A,t)$ ein Konfidenzwert für die Gültigkeit von A innerhalb der letzten T Abarbeitungszyklen und $\lambda_H \in [0,1]$ ein Schwellwert. Für jeden Zustand (jede Phase) eines Verhaltensmuster wird ein Zeitzähler verwaltet, der mit Null initialisiert wird, sobald der Zustand erreicht wird. Liegt eine diskrete Zeitbetrachtung $t \in \{n \cdot t_{Zyklus} \mid n \in \mathbb{N}\}$ zugrunde, ist $n(t) = t/t_{Zyklus}$ und bezeichnet A(t) die Gültigkeit von A zum Zeitpunkt t, dann ist C_T ist wie folgt definiert:

$$C_T(A,t) = \begin{cases} \frac{\sum_{i=0}^{n(t)} A(t - i \cdot t_{Zyklus})}{n(t) + 1} & \text{für } n(t) < T - 1 \\ \frac{\sum_{i=0}^{T-1} A(t - i \cdot t_{Zyklus})}{T} & \text{für } n(t) \geq T - 1 \end{cases}$$

In den folgenden Abschnitten werden zunächst die Verhaltensmuster der drei Gruppen vorgestellt und einige davon detaillierter erläutert. Anschließend wird deren Zusammenwirken anhand von typischen Beispielen veranschaulicht.

3.3.4.1 Verhaltensmuster für Notreaktionen

Die Verhaltensmuster für Notreaktionen stellen eine Verhaltensebene dar, die nur dann genutzt werden soll, wenn eine unmittelbare Kollisionsgefahr besteht. Dazu kann es u. a. kommen, wenn in den untergeordneten Verhaltensebenen Fehler gemacht werden oder potentielle Hindernisse durch Probleme der Sensormodule erst

spät erkannt werden. Diese Verhaltensebene dient also auch der Kompensation von Modellierungsfehlern der übrigen Verhaltensebenen. Diese können insbesondere durch die gegenseitige Hemmung mehrerer Verhaltensmuster entstehen.

Aus Sicherheitsgründen wird deshalb eine Zusatzbedingung für die Erstellung des Inhibitionsnetzwerkes ergänzt:

$$\forall e \in E_{IN} : (e = (a, b, w) \wedge Klasse(b) = 4) \rightarrow Klasse(a) = 4$$

Damit wird verhindert, daß Notreaktionen durch Verhaltensmuster anderer Prioritätsklassen gehemmt werden.

Der Aufbau eines Verhaltensmusters für die Notreaktion entspricht (in gewisser Weise) einer *Vorwärtsregel* wie sie in der Regelsprache OPS 5 verwendet wird. Es besteht aus einer formulierten Bedingung und einer Aktion, eben der *Durchführung* der Notreaktion bzw. der Vorgabe von Steuerbefehlen. Betrachtet man die Bedingung im Sinne von Rasmussens Terminologie, so ist sie mit einem *Zeichen* gleichzusetzen, welches eine stereotype Handlung auslöst.

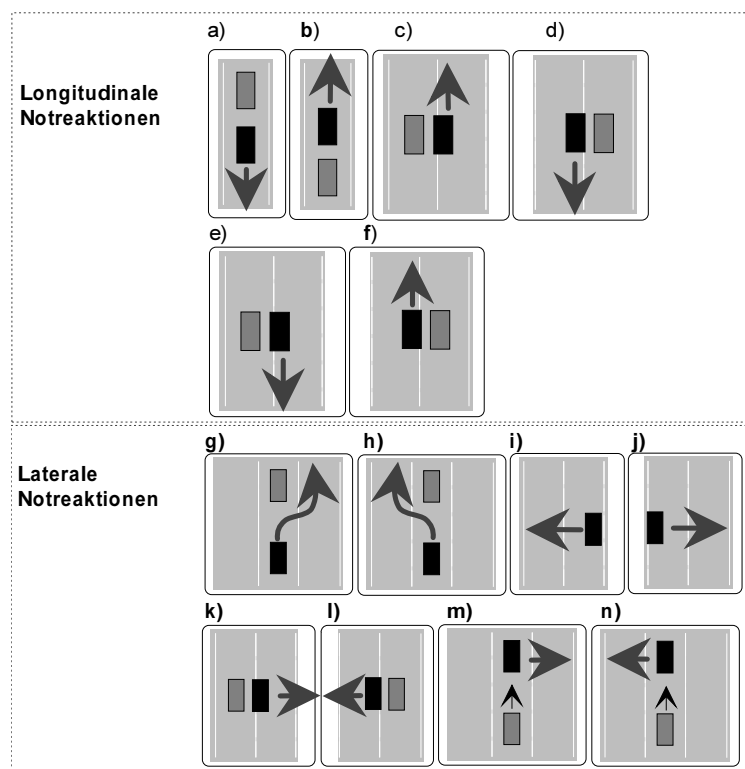


Abbildung 29 : Übersicht über die Notreaktionen

Abbildung 29 zeigt eine Übersicht über die Verhaltensmuster für Notreaktionen. Während die Verhaltensmuster i) und j) sich auf die Straße beziehen, stellen die übrigen Reaktionen auf Fahrzeuge dar. Vergleicht man die Reaktionen mit der in Abschnitt 3.2.1 beschriebenen direkten Gefahrenmodellierung, so stellt die Aufteilung der Reaktion in verschiedene Verhaltensmuster eine Dekomposition der

durch das Hindernis hervorgerufenen abstoßenden Kräfte dar. Die einzelnen Verhaltensweisen werden nachfolgend kurz skizziert. Anschließend werden einige Verhaltensmuster detaillierter beschrieben.

- a) *Notbremsung*: Kommandieren der maximalen Verzögerung zur Vermeidung einer Kollision mit einem langsamer vorausfahrenden Fahrzeug.
- b) *Notbeschleunigung*: Kommandieren der maximalen Beschleunigung zur Vermeidung einer Kollision mit einem schnelleren Fahrzeug, welches sich von hinten nähert.
- c), f) Beschleunigung wenn durch seitliche Annäherung an ein Hindernis eine Kollision droht.
- d), e) Verzögern wenn durch seitliche Annäherung an ein Hindernis eine Kollision droht.
- g), h) *Ausweichmanöver*: Kommandieren eines schnellstmöglichen Ausweichmanövers zur Vermeidung einer Kollision mit einem vorausfahrenden, langsameren Fahrzeug.
- i), j) *Positionskorrektur*: Kommandieren eines Lenkmanövers nach links (bzw. rechts) wenn ein Verlassen der Straße auf der rechten (bzw. linken) Seite droht.
- k), l) Seitliches Ausweichen nach rechts (bzw. links) bei Annäherung an ein Fahrzeug auf der linken (bzw. rechten) Seite.
- m), n) Kommandieren eines schnellstmöglichen Ausweichmanövers zur Vermeidung einer Kollision mit einem schnelleren Fahrzeug, welches sich von hinten nähert.

Da es bei der Durchführung weniger auf Komfort als auf Kollisionsvermeidung ankommt, sind Diskontinuitäten in den Steuervorgaben der nachfolgend beschriebenen Verhaltensmustern für Notreaktionen möglich. Dabei ist zu berücksichtigen, daß die Vorgaben an unterlagerte Längs- und Querregler gegeben werden, die eine „Glättung“ der Vorgaben vornehmen.

Notbremsung (Fall a)

Ähnlich der Zeitreservenberechnung der in Abschnitt 3.2.1 vorgestellten Potentialfelder wird hier zu jedem Zeitpunkt die Verzögerung $a_{\text{Kollision}}$ berechnet, die zur Kollisionsvermeidung nötig ist. Dazu enthält das Verhaltensmuster eine Referenz auf den Fahrzeugzustand Z und auf ein Hindernis H . Sei v_{diff} die Differenzgeschwindigkeit zu einem vorausfahrenden Fahrzeug und d dessen Distanz, dann berechnet sich $a_{\text{Kollision}}$ wie folgt:

$$a_{\text{Kollision}} = -\frac{\frac{1}{2} \cdot v_{\text{diff}}^2}{d}, \quad \text{für } v_{\text{diff}} \geq 0$$

Erreicht der Wert $a_{\text{Kollision}}$ den im Fahrzeugmodell festgelegten Grenzwert der maximalen Verzögerung \bar{a}_{Max} , dann muß das autonome Fahrzeug spätestens eine Vollbremsung einleiten.

Die Qualifikationsbedingung $Q_{\text{Notbremsung}}$ für ein Hindernis H wird durch die longitudinale und laterale Distanz, die Geschwindigkeit sowie durch die im Fahrermodell festgelegten Sicherheitsabstände bestimmt. Bezeichnet Z_t den aktuellen Fahrzeugzustand, dann berechnet sich $Q_{\text{Notbremsung}}$ wie folgt:

$$Q_{\text{Notbremsung}} = (\text{Distanz}_{\text{Long}}(H) > 0) \wedge (v_H(H) < v(Z_t)) \wedge (|\text{Distanz}_{\text{Lat}}(H)| < \frac{1}{2} \text{Breite}(H) + \frac{1}{2} b_{\text{fzg}} + d_{\text{Lat}})$$

Um die oben angegebene Formel für $a_{\text{Kollision}}$ zu benutzen, wird die Distanz durch eine weitere Projektion auf den Zeitpunkt, zu dem a_{Max}^- bei sofortiger Einleitung einer Notbremsung erreicht wird, korrigiert. Die Länge dieser Projektionszeitspanne t_{start} wird bestimmt durch den im Fahrzeugmodell festgelegten Wert da_{Max}^- , der festlegt, welche Verzögerungsänderung maximal erreicht wird⁴. Daraus ergibt sich:

$$t_{\text{start}} = \frac{a(Z_t) - a_{\text{Max}}^-}{da_{\text{Max}}^-}$$

Die Distanz wird daraufhin anhand eines Geschwindigkeitsmittelwerts v^{**} für diese Zeitspanne korrigiert.

$$v^{**} = v^* + \frac{1}{2}(a(Z_t) + a_{\text{Max}}^-) \cdot t_{\text{start}}$$

$$\text{Distanz}_{\text{Long}}^{**}(H) = \text{Distanz}_{\text{Long}}^* + (v_H(H) - v^{**}) \cdot t_{\text{start}}$$

Nach diesen Berechnungen wird nun die Ausgabe des Verhaltensmusters bestimmt. Dabei wird zunächst der projizierte Wert $\text{Distanz}_{\text{Lat}}^*$ zur Berechnung der Qualifikationsbedingung eingesetzt, um evtl. unnötige Notbremsungen zu verhindern. Ist die Bedingung erfüllt, so lassen sich zur Bestimmung der Ausgabe drei Fälle unterscheiden:

- (1) *Die projizierte Entfernung $\text{Distanz}_{\text{Long}}^{**}$ ist kleiner oder gleich Null*

In diesem Fall kann eine Kollision nicht durch das Verhaltensmuster Notbremsung verhindert werden, da sie eintritt, bevor die kommandierte Bremswirkung überhaupt erreicht ist. Trotzdem wird eine Notbremsung kommandiert, die aber nur bei Änderungen in der Umgebung erfolgreich sein kann. Da in diesem Fall höchste Kollisionsgefahr besteht, werden Gewichtswert und Basiswert des Verhaltensmusters auf Eins gesetzt und das Verhaltensmuster wird aktiviert. Der Intensitätswert entspricht der maximalen Verzögerung a_{Max}^- .

- (2) *Die benötigte Verzögerung ist größer oder gleich der maximalen Verzögerung*

Ist die projizierte Entfernung größer als Null, so wird die benötigte Verzögerung $a_{\text{Kollision}}$ durch Einsetzen dieses Entfernungswerts in die oben

⁴ Der hier festgelegte Wert für da_{Max}^- ist empirisch bestimmt. Er stellt eine obere Abschätzung der maximalen Verzögerungsänderung dar, die in Ermangelung präziser, echtzeitfähiger Modelle gewählt wird.

angeführte Berechnungsformel bestimmt. Ist die benötigte Verzögerung größer oder gleich der maximalen Verzögerung, dann wird die gleiche Ausgabe wie in (1) generiert.

(3) *Die benötigte Verzögerung ist kleiner als die maximale Verzögerung*

Da es erstens möglich ist, durch die zeitdiskrete Verarbeitung der Daten den genauen Startzeitpunkt der Notbremsung zu verpassen und zweitens Störungen durch fehlerbehaftete Messungen zu erwarten sind, wird ein Toleranzbereich für die Einleitung einer Notbremsung eingerichtet. Der Toleranzbereich wird durch einen Faktor $f_{\text{Toleranz}} \in [0,1]$ in Abhängigkeit von a_{Max}^- bestimmt⁵. Abhängig von $a_{\text{Kollision}}$ steigen Basiswert und Gewichtswert im Intervall $[f_{\text{Toleranz}} \cdot a_{\text{Max}}^-, a_{\text{Max}}^-]$ linear von Null auf Eins an. Der Intensitätswert entspricht dagegen stets a_{Max}^- .

Ausweichmanöver (Fall g,h)

Anders als die zuvor beschriebene Notbremsung berücksichtigt das hier beschriebene Ausweichmanöver drei Referenzen auf Objekte der Situationsbeschreibung. Neben einem Hindernis H und dem Fahrzeugzustand Z wird auch die Straßenbeschreibung S in die Reaktion mit einbezogen. Die Qualifikationsbedingung wird durch den Freiraum auf der linken bzw. rechten Seite neben dem Hindernis sowie durch die Position und Geschwindigkeit des Hindernisses bestimmt. Ist der Freiraum, der sich aus Spurbreite und Nachbarspurinformation zusammensetzt, groß genug um das Hindernis zu passieren, ist zudem das Hindernis in Fahrtrichtung und langsamer als das autonome Fahrzeug, so kann das Ausweichmanöver durchgeführt werden. Dadurch wird bereits eine eindeutige Präferenz des Verbleibens auf der Straße gegenüber der Kollision mit einem anderen Fahrzeug festgelegt. Diese ist dadurch begründet, daß über die Beschaffenheit der Umgebung neben der Straße nichts bekannt ist. Die mit einem Verlassen der Straße verbundene Gefahr ist also möglicherweise höher als die der Kollision.

Das Fahrzeugmodell gibt an, welche Spurwechselzeit im Extremfall erreichbar ist. Dieser Wert wird einer Zeitberechnung $t_{\text{Ausweichen}}$ für das Ausweichmanöver zugrundegelegt. Bei der Berechnung von $t_{\text{Ausweichen}}$ wird auch eine maximale zulässige Querbeschleunigung des autonomen Fahrzeugs berücksichtigt. Das Ausweichmanöver wird ausgelöst, wenn die berechnete Zeit bis zur Kollision (engl. Time To Collision; TTC) t_{tic} gleich der Zeit für das Ausweichmanöver ist.

⁵ In der Simulation hat sich die Wahl von $f_{\text{Toleranz}} = 0.85$ als geeignet erwiesen.

Zur Berücksichtigung der Zeitverzögerung wird auch hier eine projizierte Position (Distanz*_{Long}(H), Distanz*_{Lat}(H)) zugrunde gelegt, die u. a. die Latenzzeit zur Ansteuerung der Lenkung berücksichtigt. Die Zeit bis zur Kollision berechnet sich dann wie folgt:

$$t_{ttc} = \frac{Distanz_{Long}^*(H)}{v(Z_t) - v_H(H)}, \text{ für } v(Z_t) > v_H(H)$$

Dabei wird eine konstante Geschwindigkeit der Fahrzeuge angenommen. Wie schon bei der Notbremsung werden bei der Ausgabeberechnung drei Fälle unterschieden:

(1) *Die projizierte Entfernung Distanz*_{Long} ist kleiner oder gleich Null*

Die Kollision ist durch das Ausweichmanöver nicht zu vermeiden. Der Basiswert des Verhaltensmusters wird dennoch auf Eins gesetzt und das Verhaltensmuster wird aktiviert. Der Gewichtswert wird niedrig gesetzt um alternativen Verhaltensmustern eine Priorität zu geben. Nach Aktivierung des Verhaltensmusters folgt der Intensitätswert einem vorgegebenen Verlauf eines Ausweichmanövers⁶ der durch eine Funktion festgelegt ist, die den aktuellen Fahrzeugzustand berücksichtigt.

(2) $t_{ttc} \leq t_{Ausweichen}$

Der Basiswert des Verhaltensmusters wird auf Eins gesetzt und das Verhaltensmuster wird aktiviert. Der Gewichtswert steigt im Intervall $t_{ttc} = [0, t_{Ausweichen}]$ linear von einem zu wählenden Minimalwert $g_0 > 0$ auf Eins an. Wie in Fall (1) wird dadurch ein Deaktivieren des Verhaltensmusters durch ein zweites ermöglicht, welches die Kollision evtl. noch verhindern kann. Auch hier folgt der Intensitätswert einem vorgegebenen Verlauf eines Ausweichmanövers.

(3) $t_{ttc} > t_{Ausweichen}$

Zum Abfangen von Störungen wird eine Toleranzzeit $t_{Toleranz} > 0$ eingeführt. Bezeichnet $t_{Reserve} = t_{ttc} - t_{Ausweichen}$ die Zeitreserve bis zur Einleitung eines Ausweichmanövers, dann werden Gewichtswert und Basiswert des Verhaltensmusters für $t_{Reserve} < t_{Toleranz}$ wie folgt bestimmt:

$$Basis(Ausweichen) = Gewicht(Ausweichen) = \frac{t_{Reserve}}{t_{Toleranz}}$$

Innerhalb der Toleranzzeit wird das Verhaltensmuster aktiviert und der Intensitätswert gemäß der vorgegebenen Ausweichmanöverfunktion gewählt.

⁶ Dieser Verlauf wird anhand einer Funktion ermittelt, die bei Vorgabe einer Zeit und eines lateralen Versatzes, eine Trajektorie berechnet, die den aktuellen Fahrzeugzustand berücksichtigt.

Die Verhaltensmuster c), d), e), f), m) und n) sind sehr ähnlich zu dem hier exemplarisch beschriebenen Ausweichmanöver aufgebaut und werden nicht separat aufgeführt.

Straßenrand (Fall i,j)

Nähert sich das autonome Fahrzeug dem Straßenrand und droht die Straße zu verlassen, so wird ein Verhaltensmuster aktiviert, welches dies verhindert. Die Qualifikationsbedingung wird durch die Bewegungsrichtung und die Position des autonomen Fahrzeugs bestimmt. Bewegt sich das Fahrzeug auf den Straßenrand zu (oder hat es ihn bereits überschritten), so ist das Kriterium erfüllt⁷.

Zur Bestimmung der Ausgabe des Verhaltensmusters wird zunächst der verbleibende seitliche Abstand d_{Rand} berechnet. Um die Zeitverzögerungen durch Kommunikation und Aktortotzeiten einzubeziehen, wird der Abstand unter Berücksichtigung der Quergeschwindigkeit zeitlich voraus projiziert.

$$v_{Lat}^* = v_{Lat}(Z_t) + \frac{1}{2} a_{Lat}(Z_t) \cdot (t_{Sensor} + t_{Lenkung})$$

$$d_{Rand}^* = d_{Rand} \pm v_{Lat}^* \cdot (t_{Sensor} + t_{Lenkung})$$

Auf Basis dieses projizierten verbleibenden Abstands wird analog zur Notbremsung die benötigte Verzögerung der Quergeschwindigkeit aq_{Rand} berechnet. Das Fahrzeugmodell beinhaltet einen Grenzwert für die Querbeschleunigung $aq_{max}(v)$. Die Ausgabe des Verhaltensmusters basiert auf dem Vergleich der benötigten und der maximalen Querbeschleunigung. Ist der projizierte Abstand zum Straßenrand kleiner oder gleich Null, so werden Gewichts- und Basiswert auf Eins gesetzt. Ansonsten wird die nötige Querbeschleunigung aq_{Rand} bestimmt. Ist aq_{Rand} größer oder gleich $aq_{max}(v)$, so werden ebenfalls Gewichts- und Basiswert auf Eins gesetzt. Ist aq_{Rand} kleiner als $aq_{max}(v)$ aber größer als ein festgelegter Toleranzwert $T_{rand} = f \cdot aq_{max}(v)$, mit $f \in [0,1]$, dann werden Basiswert und Gewichtswert wie folgt bestimmt:

$$Basis(\text{Straßenrand}) = \text{Gewicht}(\text{Straßenrand}) = \frac{aq_{Rand} - T_{Rand}}{aq_{max}(v(Z_t)) - T_{Rand}}$$

Ist eines der oben aufgeführten Kriterien erfüllt, so wird das Verhaltensmuster aktiviert. Der Intensitätswert, der bei lateralen Verhaltensmustern eine Quergeschwindigkeitsvorgabe darstellt, berechnet sich aus der aktuellen

⁷ Die Qualifikationsbedingung ist damit immer dann erfüllt, wenn das Fahrzeug sich quer zur Fahrbahn bewegt. Dies ist in Realität (nahezu) immer gegeben, so daß das Kriterium für dieses Verhaltensmuster nicht notwendig ist.

Quergeschwindigkeit, der Zykluszeit des Verhaltensentscheidungsmoduls und der maximalen Querbeschleunigung.

3.3.4.2 Basisverhaltensmuster

Die Gruppe der Basisverhaltensmuster umfaßt die stark automatisierten Handlungen, zu denen es keiner großen kognitiven Beteiligung bedarf. Diesen Handlungen wird das Spurhalten und das Folgen eines vorausfahrenden Fahrzeugs zugeordnet (vgl. [Johannsen 90]). Die Eingaben werden als Signale betrachtet, und es wird ein Steuerungs- bzw. Regelungsverhalten gezeigt.

Im Gegensatz zu den Verhaltensmustern für Notreaktionen sind die Basisverhaltensmuster auf ein komfortables Fahrverhalten ausgerichtet. Diskontinuitäten in der Steuervorgabe sind daher nicht akzeptabel.

Das Verhaltensmuster zur Spurhaltung

Das Verhaltensmuster enthält eine Referenz auf eine Straßenbeschreibung Str und auf den Fahrzeugzustand Z . Wie schon in Abschnitt 3.2 beschrieben, läßt sich die Gefahr bei der Spurführungsaufgabe als eine „Gefahrenwanne“ darstellen, die ihr Minimum in der Fahrspurmitte hat und zum Fahrspurrand größer werdende Gefahrenwerte aufweist. Mit dieser Vorgabe läßt sich ein Gefahrenwert $g_{position}$ aus der Entfernung $d_{reserve}$ zum Fahrspurrand berechnen⁸:

$$d_{reserve} = \frac{1}{2} \cdot \text{Spurbreite}(\text{Spur}_{Aktuell}(Str_t)) - |Position_{AMV}(Str_t)| - \frac{1}{2} \cdot b_{fzg}$$

$$g_{position} = \begin{cases} \left(\frac{|Position_{AMV}(Str_t)|}{\frac{1}{2} \cdot (\text{Spurbreite}(\text{Spur}_{Aktuell}(Str_t)) - b_{fzg})} \right)^2 & d_{reserve} > 0 \\ 1 & d_{reserve} \leq 0 \end{cases}$$

Zur Bestimmung des Intensitätswerts wird zunächst eine gewünschte Querbeschleunigung a_q berechnet. Dieser Berechnung liegt der Gefahrenwert, die Abweichungstoleranz $y_{toleranz}$ und die im Fahrermodell vorgegebene maximale Querbeschleunigung zur Spurführung a_{quer} zugrunde. Zusätzlich wird der Fahrzeugzustand Z berücksichtigt und die Änderung der Querbeschleunigung begrenzt. Der Intensitätswert berechnet sich anschließend als $v_{Lat} + a_q \cdot t_{Zyklus}$.

Das Verhaltensmuster zum Folgen vorausfahrender Fahrzeuge

Das Verhaltensmuster enthält eine Referenz auf ein Hindernis H und den Fahrzeugzustand Z . An die Objektreferenz auf Hindernis H ist eine Projektionsrestriktion

⁸ Eine Ergänzung zu dieser statischen Gefahrenbetrachtung stellt ein dynamischer Gefahrenwert dar, der aus der Zeit bis zur Linienüberschreitung (engl. Time-to-Line Crossing; TLC) bestimmt wird. Dieser Gefahrenwert drückt die noch verfügbare Zeit für eine zulässige Fehlernachlässigung aus.

geknüpft, die nur Bindungen an Hindernisse zulässt, welche sich vor dem autonomen Fahrzeug befinden. Die *Signale* nach denen sich die Ausgabe des Verhaltensmusters hauptsächlich richtet sind die Relativgeschwindigkeit und Distanz des vorausfahrenden Fahrzeugs.

Das Verhaltensmuster deckt folgende Verhaltensweisen ab:

- (a) *Annäherung* an ein vorausfahrendes Fahrzeug und entsprechende Verzögerung,
- (b) *Einhalten eines Sicherheitsabstands* und
- (c) *Zurückfallen* bei Unterschreiten des Sicherheitsabstands.

Als Parameter für die Modellierung eines entsprechenden Verhaltens enthält das Fahrermodell die präferierten Verzögerungswerte $\bar{a}_{\text{annähern}}$ und \bar{a}_{folgen} . Diese bilden die Grundlage für eine gewünschte Relativgeschwindigkeit zwischen autonomem Fahrzeug und Hindernis in Abhängigkeit von der aktuellen Distanz und dem einzuhaltenden Sicherheitsabstand. Der Sicherheitsabstand s_d ist fahrerabhängig und berechnet sich aus einem fixen Anteil d_{fix} und einem geschwindigkeitsabhängigen Anteil d_{var} der in Sekunden angegeben wird:

$$s_d = d_{\text{fix}} + d_{\text{var}} \cdot (v(Z_t) - v_H(H))$$

Die subjektive Gefahr ist abhängig von der Abweichung von dieser gewünschten Relativgeschwindigkeit. Man kann sie als das vom Fahrer präferierte Gefahrenniveau betrachten. Auch hier ist wieder eine Parallele zu der in Abschnitt 3.2.2 beschriebenen Gefahrenmodellierung gegeben (vgl. Seite 43).

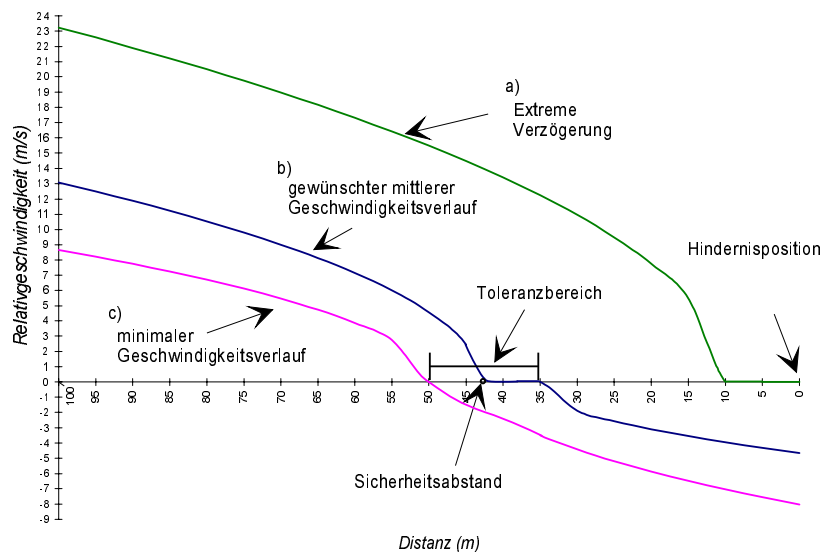


Abbildung 30 : Modellierung der Fahrerreaktion für das Folgen vorausfahrender Fahrzeuge

Abhängig von der gewünschten Relativgeschwindigkeit wird ein Bereich beschrieben, in dem das Verhaltensmuster aktiviert wird. Abbildung 30 zeigt eine mögliche Verhaltensmodellierung für das Folgen eines vorausfahrenden Fahrzeugs.

Das Verhaltensmuster ist immer dann aktiviert, wenn die Relativgeschwindigkeit des autonomen Fahrzeugs zwischen den in Abbildung 30 aufgetragenen Funktionen a) und c) liegt. Der Basiswert des Verhaltensmusters steigt zwischen den Funktionen c) und b) linear von Null auf Eins an und bleibt darüber auf Eins. Dadurch wird anderen, gleichpriorisierten Verhaltensmustern der Vorrang gelassen solange das gewünschte Verhalten nicht beeinträchtigt wird. Gewichtswert und Basiswert werden gleichgesetzt.

Bezeichnet $v_{\text{Folgen}}(d)$ die gewünschte Relativgeschwindigkeit für eine Distanz d^9 und $v_{\text{Schwelle}}(d)$ die Aktivierungsschwelle des Verhaltensmusters, die in Abbildung 30 als Begrenzungskurve c) dargestellt ist, dann wird der Intensitätswert in Abhängigkeit von der aktuellen Relativgeschwindigkeit $v_{\text{Diff}} = v(Z_t) - v_H(H)$ wie folgt bestimmt:

- *Fall I*: $v_{\text{Schwelle}}(\text{Distanz}_{\text{Long}}(H)) < v_{\text{Diff}} < v_{\text{Folgen}}(\text{Distanz}_{\text{Long}}(H))$

Ziel ist es, den Geschwindigkeitsverlauf an die Referenzkurve anzunähern.

Bezeichnet $\Delta v = v_{\text{Folgen}}(\text{Distanz}_{\text{Long}}(H)) - v_{\text{Diff}}$, dann ist dieser Wert über die Zeit zu reduzieren. Daraus ergibt sich für die zu setzende Sollgeschwindigkeit v_{Soll} :

$$v_{\text{Soll}} = v_H(H) + v_{\text{Folgen}}(\text{Distanz}_{\text{Long}}(H) - v_{\text{Diff}} \cdot t_{\text{Zyklus}}) - \Delta v \cdot (1 - k_1)$$

Dabei ist $k_1 \in [0,1]$ ein Konvergenzfaktor über den eingestellt wird, wie schnell sich die Geschwindigkeit der Referenzkurve anpassen soll. Obere Schranke für v_{Soll} ist die aktuelle Geschwindigkeit, d. h. zur Anpassung an die Referenzkurve wird *keine Beschleunigung* kommandiert.

- *Fall II*: $v_{\text{Diff}} \geq v_{\text{Folgen}}(\text{Distanz}_{\text{Long}}(H))$

Die gewünschte Sollgeschwindigkeit wird mittels eines Konvergenzfaktors k_2 analog zu Fall I bestimmt.

Der Intensitätswert ergibt sich dann unter Berücksichtigung der maximalen Verzögerung a_{max}^- wie folgt:

$$\text{Intensität}(\text{Folgen}) = \begin{cases} v_{\text{Soll}} - v(Z_t) & v_{\text{Soll}} - v(Z_t) < a_{\text{max}}^- \\ a_{\text{max}}^- & v_{\text{Soll}} - v(Z_t) \geq a_{\text{max}}^- \end{cases}$$

3.3.4.3 Verhaltensmuster für schematisierte Handlungen

Als schematisierte Handlungen werden Verhaltensmuster zum Überholen, Beachten von Verkehrszeichen und Richten nach allgemeinen Verkehrsregeln zusammengefaßt. Zusätzlich werden Verhaltensmuster integriert, die Heuristiken über Fahrverhalten anderer Verkehrsteilnehmer einbringen. Weitere Verhaltensmuster beinhalten keine Aktion, sondern dienen ausschließlich der Hemmung anderer

⁹ Um die Übersichtlichkeit zu wahren wird bei der Bezeichnung der Funktion v_{Folgen} auf die Angabe der übrigen Parameter verzichtet.

Verhaltensmuster. Die Grenzen zwischen Basisverhaltensmustern und schematisierten Handlungen sind fließend, so daß andere Zuordnungen als die hier vorgenommene denkbar sind. Allen gemeinsam ist eine Qualifikationsbedingung, nach deren Erfüllung eine Ausführung der Handlung folgt. Diese Qualifikationsbedingung ist mit einem Zeichen in Rasmussens Terminologie vergleichbar.

Abbildung 31 zeigt eine Übersicht über die Verhaltensmuster für schematisierte Handlungen. Diese Menge kann beliebig erweitert und verändert werden, um ein anderes Gesamtverhalten zu erhalten. Die einzelnen Verhaltensmuster und ihre Intentionen werden nachfolgend kurz skizziert. Anschließend werden einige ausgewählte Verhaltensmuster näher beschrieben.

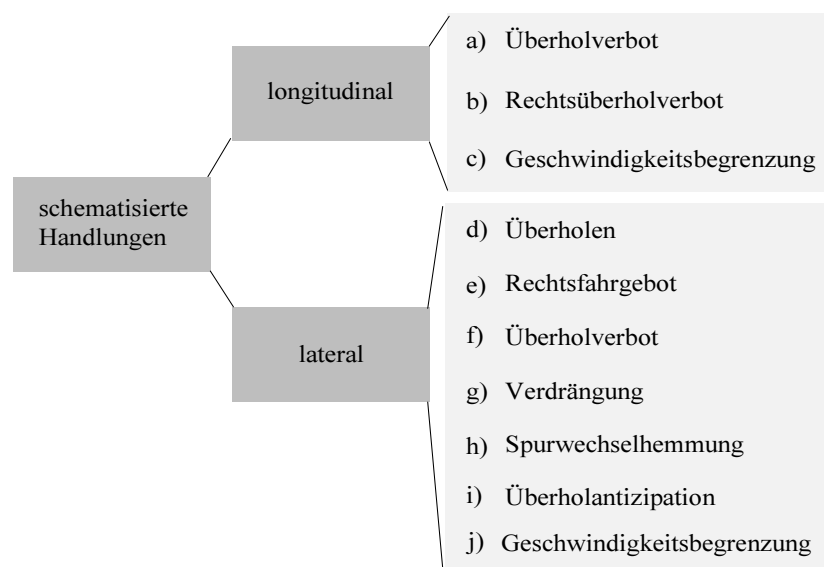


Abbildung 31 : Übersicht über die Verhaltensmuster für schematisierte Handlungen

- a) Ein auf der rechten Spur fahrendes Fahrzeug wird wie ein vorausfahrendes behandelt. Die Aktion entspricht der eines Basisverhaltensmusters zum Folgen eines Fahrzeugs. Dabei wird ein reduzierter Sicherheitsabstand (ggf. Null) vorgegeben und auf ein Einleiten der Handlung verzichtet, wenn die benötigte Verzögerung einen Schwellwert überschreitet.
- b) Das Rechtsüberholverbot entspricht dem Überholverbot mit veränderter Qualifikationsbedingung.
- c) Das Auftreten einer Geschwindigkeitsbegrenzung löst eine Anpassung der Eigengeschwindigkeit aus. Dieser Anpassung liegt eine fahrerabhängige Verzögerungsfunktion zugrunde.
- d) Das Überholen wird durch ein langsamer vorausfahrendes Fahrzeug ausgelöst. Vor dem Einleiten der eigentlichen Handlung, wird zur Ankündigung des Fahrmanövers der Blinker gesetzt.
- e) Das Rechtsfahrgebot beinhaltet eine Querführungsvorgabe zur Realisierung eines Spurwechsels nach rechts. Die Qualifikationsbedingung ist eine existierende rechte Spur. Zusätzlich wird eine Wartezeit bis zur Durchführung des Spurwechsels modelliert.

- f) Ein Überholverbot beinhaltet keine eigene Handlung sondern dient der Hemmung von Verhaltensmustern, die einen Spurwechsel nach links vorsehen.
- g) Nähert sich ein schnelleres Fahrzeug von hinten und ist eine rechte Spur vorhanden, so wird ein Spurwechsel nach rechts eingeleitet. Dieses Verhaltensmuster unterscheidet sich vom Rechtsfahrgebot durch eine höhere Gefahrezuordnung, so daß der Durchführung des Manövers eine höhere Priorität zugeordnet wird.
- h) Eine Spurwechselhemmung beinhaltet keine eigene Handlung. Zwei Verhaltensmuster zur Hemmung eines Spurwechsels nach links und rechts betrachten die Relativgeschwindigkeit von Fahrzeugen auf den angrenzenden Spuren und bestimmen eine von diesen ausgehende Gefahr.
- i) Nähert sich von hinten ein schnelleres Fahrzeug, welches sich auf der gleichen Spur befindet, so wird die Hypothese aufgestellt, daß dieses überholen wird. Anhand der Distanz und Annäherungsgeschwindigkeit wird ein Gefahrenwert berechnet um dadurch Verhaltensmuster zu hemmen, die einen Spurwechsel nach links vorsehen.

Als Beispiele für Verhaltensmuster für schematisierte Handlungen werden nachfolgend die Verhaltensmuster für das Überholen, für eine Geschwindigkeitsbeschränkung und für die Spurwechselhemmung näher beschrieben ohne eine formale Algorithmenbeschreibung anzugeben.

Überholen

Das Verhaltensmuster zum Überholen besitzt Referenzen auf ein Hindernis H , die Straße Str , den Fahrzeugzustand Z und die Benutzereingabe Be . Die Qualifikationsbedingung $Q_{\text{überholen}}$ ist erfüllt, wenn für das vorausfahrende Fahrzeug $C_T(v(H) < v_{\text{Soll}}(Be), t_0) > \lambda_Q$ gilt und sein seitlicher Versatz nicht ausreichend ist, um es unter Einhaltung des seitlichen Sicherheitsabstands d_{lat} zu passieren. Dabei bezeichnet t_0 die Verweilzeit im Startzustand und $\lambda_Q \in [0,1]$ einen Konfidenzschwellwert. Ist $Q_{\text{überholen}}$ erfüllt, so wird in Zustand (1) übergegangen.

Die Handlung ist in drei aufeinanderfolgende Phasen unterteilt, die durch Zustände eines Übergangnetzes repräsentiert werden:

- (1) *Akzeptanzphase*: Prüfen ob ein Überholmanöver möglich ist. In dieser Phase wird nur der Basiswert auf einen vorgegebenen Wert gesetzt, um zu prüfen wie stark das Verhaltensmuster gehemmt wird, d. h. ob Gefahren bestehen, die ein Überholen verhindern. Das Verhaltensmuster ist passiv.
- (2) *Ankündigungsphase*: Setzen des Blinkers zur Ankündigung der Überholabsicht. In dieser Phase wird dem Basiswert auf ein geschwindigkeits- und abstandsabhängiger Wert zugeordnet und der Blinker gesetzt. Das Verhaltensmuster bleibt passiv und es wird kein Intensitätswert berechnet.
- (3) *Durchführungsphase*: Steuervorgabe für einen Spurwechsel nach links. In dieser Phase wird das Verhaltensmuster aktiviert, der Basiswert auf Eins

gesetzt und der Gewichtswert abhängig von der Geschwindigkeit der Entfernung zum Hindernis bestimmt. Der Intensitätswert wird über eine Spurwechselfunktion ermittelt und der Blinker bleibt gesetzt.

Die Übergangsbedingung zwischen Zustand (1) und (2) setzt sich durch konjunktive Verknüpfung eines Zeitprädikats $Warten(t_1)$, eines Zustandsprädikat $Aktivierbar(\text{Überholen})$ und der Situationsprädikate $\text{Überholabstand}(Be,H,Z)$ und $\text{Überholgeschwindigkeit}(H,Be)$ zusammen. Das Zeitprädikat sorgt für die Einhaltung einer Verweilzeit in Zustand (1). Die anderen werden aufgrund der verrauschten Eingangsdaten über eine Konfidenz C_T betrachtet. Die Aktivierbarkeit ist gegeben wenn $\text{Basis}(\text{Überholen}) > \text{Hemmung}(\text{Überholen})$ für den vorangehenden Zyklus gilt. Übersteigt die Konfidenz $C_T(\text{Aktivierbar}(\text{Überholen}), t_1)$ einen Schwellwert λ_H , so ist dieser Teil der Übergangsbedingung erfüllt. Die Situationsprädikate prüfen, ob die Differenz zwischen Wunschgeschwindigkeit und der Geschwindigkeit des Hindernisses groß genug ist und ob eine „sinnvolle“ Überholentfernung erreicht ist. Die Übergangsbedingung zwischen Zustand (2) und (3) enthält ein Zeitprädikat $Warten(t_2)$, welches die gewünschte Vorankündigungszeit angibt, in der nur der Blinker gesetzt wird. Zusätzlich muß weiterhin $\text{Aktivierbar}(\text{Überholen})$ gelten. Neben den Übergängen zu Folgephasen ist auch ein Zurückgehen in eine vorangehende Phase möglich. Unterschreitet beispielsweise die Konfidenz für $\text{Aktivierbar}(\text{Überholen})$ einen Schwellwert λ_L so wird in Zustand (1) zurückgewechselt.

Spurwechselhemmung links

Ein Verhaltensmuster zur Spurwechselhemmung besitzt Referenzen auf eine Straßenbeschreibung, auf ein Hindernis und den Fahrzeugzustand Z . Die Qualifikationsbedingung für eine Spurwechselhemmung ist die Existenz eines Fahrzeugs auf der links angrenzenden Spur.

Da keine Aktion durchgeführt wird sind Intensitätswert und Gewichtswert stets Null. Für die Berechnung des Basiswerts wird angenommen, das Fahrzeug befände sich bereits auf der linken Spur. Daraus ergeben sich folgende mögliche Konsequenzen, aus denen ein Gefahrenwert abgeleitet wird:

- a) Die fiktive Position überlappt das andere Fahrzeug oder eine Kollision ist aufgrund der Geschwindigkeit und des Abstands unvermeidbar: In diesem Fall wird der Basiswert auf Eins gesetzt.
- b) Das Hindernis befindet sich vor dem autonomen Fahrzeug: In diesem Fall wird der Basiswert fahrerabhängig aus der nötigen Verzögerung und einem

einzuhaltenen Sicherheitsabstand abgeleitet (vgl. Basisverhaltensmuster zum Folgen eines Fahrzeugs).

- c) Das Hindernis befindet sich hinter dem autonomen Fahrzeug: In diesem Fall wird der Basiswert anhand der benötigten Verzögerung des Hindernisses berechnet. Dabei wird eine Toleranzschwelle für die Behinderung des anderen Verkehrsteilnehmers berücksichtigt. Ist diese Toleranzschwelle überschritten, so wird der Basiswert auf Eins gesetzt.

3.3.4.4 Integration der Fahrerintentionen

Neben den bisher beschriebenen Verhaltensmustern werden auch Verhaltensmuster zur Umsetzung von Fahrerkommandos integriert. Dazu gehören kommandierte Spurwechsel sowie die Einhaltung und Veränderung der Wunschgeschwindigkeit. Ähnlich zu der Integration von Fahrerintentionen in die Gefahrenkarte geschieht die Umsetzung von Fahrerkommandos in Sollvorgaben durch Fahrerkenntlinien (vgl. Abschnitt 3.2.3).

Da es sich bei der Einhaltung einer Wunschgeschwindigkeit um ein Verhalten ohne Vorbedingung handelt, wird ein Basisverhaltensmuster zur Repräsentation dieser Funktion eingeführt, das Referenzen auf den Fahrzeugzustand und auf die Benutzereingabe besitzt. Ein kommandierter Spurwechsel ist ereignisbezogen und daher eher auf Ebene der schematisierten Handlungen anzusiedeln.

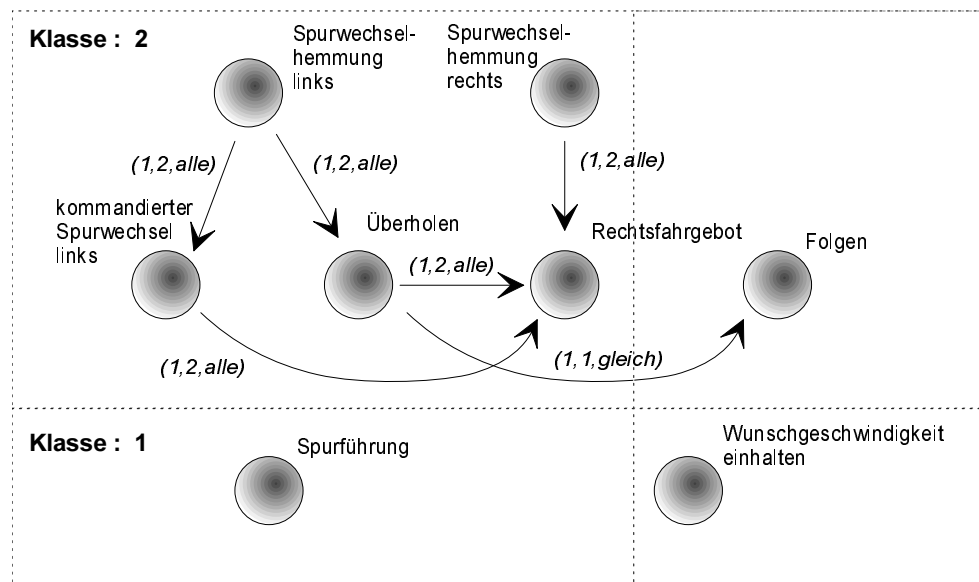


Abbildung 32 : Verhaltensmuster und Kantenschemata zum Aufbau eines Inhibitionsnetzes

3.3.4.5 Verknüpfung der Verhaltensmuster

Zum Abschluß der Beschreibung des Verhaltensmusterkonzepts wird die Verknüpfung der Verhaltensmuster in diesem Abschnitt anhand eines Ausschnitts aus dem Inhibitionsnetz beschrieben. Dieser Ausschnitt ist unvollständig und dient nur der Erklärung des Verknüpfungsprinzips. Abbildung 32 zeigt die Verhaltensmuster und die Kantenschemata zum Aufbau eines Inhibitionsnetzwerks.

Für diesen betrachteten Ausschnitt ist die Exklusivmenge $EV = \emptyset$. Die Verknüpfung wird am besten anhand eines Beispiels deutlich. Dazu wird ein Überholvorgang in mehreren Phasen betrachtet:

- *Annähern an ein vorausfahrendes Fahrzeug; ein Fahrzeug ist auf der linken Spur:*
Die Verhaltensmuster *Spurwechselhemmung links*, *Überholen*, *Folgen*, *Spurhalten* und *Wunschgeschwindigkeit einhalten* werden instanziiert und ihre Instanzen in diesem Beispiel mit I_{sh} , $I_{ü}$, I_f , I_s und I_w bezeichnet. In dieser Situation gelte $Basis(I_{sh})=1$, $Basis(I_{ü})=0.5$, $Basis(I_f)=0.1$, $Basis(I_s)=0.1$ und $Basis(I_w)=0.1$. Anhand der Kantenschemata werden die Kanten $e_1 = (I_{ü}, I_f, 1, 1)$ und $e_2 = (I_{sh}, I_{ü}, 1, 2)$ angelegt. Durch diese zwei einelementige Äquivalenzmengen gegeben. Die Hemmungen werden sukzessive bestimmt zu $Hemmung(I_{ü}) = (Basis(I_{sh}) - 0) \cdot 1 = 1$ über Kante e_2 und anschließend $Hemmung(I_f) = 1 \cdot \max(Basis(I_{ü}) - Hemmung(I_{ü}), 0) = 0$ über e_1 . Es bleiben I_f und I_w zur Bestimmung der longitudinalen Steuervorgabe und I_s zur lateralen Steuervorgabe. Die laterale Vorgabe wird von I_s übernommen während die Intensitäten von I_f und I_w gewichtet gemittelt werden.
- *Das überholende Fahrzeug verläßt den Sichtbereich:*
Die Verhaltensmusterinstanz I_{sh} und die Kante e_2 werden gelöscht. Es besteht dadurch keine Hemmung für $I_{ü}$ so daß dieses nach einigen Zyklen den Blinker setzt und etwas später ausschert. Während dieser Zeit wird I_f gehemmt, so daß die longitudinale Kontrolle I_w zugeteilt ist.
- *Das autonome Fahrzeug überquert die Mittellinie und passiert das Hindernis*
Die Verhaltensmuster *Spurhemmung rechts* (I_{sr}) und *Rechtsfahrgebot* (I_{rf}) werden instanziiert und die Kanten $e_3 = (I_{ü}, I_{rf}, 1, 2)$ und $e_4 = (I_{sr}, I_{rf}, 1, 2)$ werden in das Inhibitionsnetz eingefügt. I_{rf} wird durch I_{sr} und $I_{ü}$ gehemmt. Ist der seitliche Versatz des vorausfahrenden Fahrzeugs ausreichend, so wird $I_{ü}$ entfernt. Die Hemmung von I_{rf} bleibt so lange bestehen, bis die durch den Basiswert von I_{sh} repräsentierte Gefahr für einen Spurwechsel geringer wird und damit niedrigere Präferenz als die durch I_{rf} dargestellte Gebotseinhaltung bekommt.
Das autonome Fahrzeug wechselt schließlich auf die rechte Spur und der Überholvorgang ist abgeschlossen.

3.4 Aufbau einer Situationsbeschreibung aus Sensordaten

Beim Einsatz der Verhaltenssteuerung im realen Umfeld ergibt sich die Problematik der unzureichenden Qualität der Fahrzeugsensoren. Anders als in der perfekten Welt, wie sie in einer Simulationsumgebung generierbar ist, sind hier die Eingangsdaten mit Fehlern behaftet. Diese Fehler sind auf Unzulänglichkeit der Sensoren, unzureichende Modellierung der Umwelt und auf die zu geringe Kapazität der heute zur Verfügung stehenden Rechnertechnik zurückzuführen.

Dieser Abschnitt beschreibt die Kopplung der Verhaltenssteuerung mit realen Sensordaten, wie sie bei Fahrten mit dem Daimler-Benz Versuchsträger auftreten. Die Aufgabe besteht aus der Fusion und Glättung der eingehenden Sensordaten zur Erstellung und Verwaltung einer in sich konsistenten Situationsbeschreibung.

In einer vorangehenden Betrachtung zu diesem Themengebiet werden in Abschnitt 3.4.1 zunächst die Kernaufgaben und Lösungsmethoden vorgestellt, bevor in den folgenden Abschnitten näher auf die Architektur des hier vorgestellten Situationsverwaltungssystems eingegangen wird.

3.4.1 Aufgaben und Lösungsmethoden

Anhand der zunächst aufgeführten auftretenden Fehlerarten wird die Aufgabe der Erstellung einer Situationsbeschreibung verdeutlicht. Für die beiden Kernaufgaben der Meßwertglättung und Datenfusion werden bekannte Lösungsmethoden aufgeführt, die im nachfolgend diskutierten Ansatz integriert werden.

3.4.1.1 Auftretende Fehlerarten

Die auftretenden Fehler werden zunächst nach den Sensormodulen unterteilt, durch die sie produziert werden. Weiterhin lassen sich *Positiv-* und *Negativfehler* (*false positive*, *false negative*) unterscheiden. Während bei Positivfehlern Objekte erkannt werden, die in der Realität nicht existieren, bezeichnen Negativfehler das Ausbleiben der Erkennung eines existierenden Objekts durch ein Sensormodul.

Bei der Fehlerbetrachtung wird die Diskrepanz zwischen der mit Hilfe der benutzten Sensoren erkennbaren Welt und der realen Welt nicht berücksichtigt. Da die Hinderniserkennungsmodule nicht in der Lage sind, Personen oder Motorräder zu erkennen¹⁰, werden diese Negativfehler im Folgenden nicht aufgeführt. Das gleiche gilt für unmarkierte Straßenabschnitte, in denen der Straßenerkennungsmodule nicht funktionsfähig ist.

¹⁰ Das Modul Vision Bumper detektiert jegliche Erhebung vor dem autonomen Fahrzeug. Damit werden auch Motorräder und Personen erfaßt. Sie werden jedoch nicht als solche identifiziert. Den übrigen, modellbasierten Erkennungsverfahren sind nur Modelle für Fahrzeuge (PKW, LKW) vorgegeben.

Straßenerkennung

Den im Straßenerkennungsmodul auftretenden Fehlern kommt aufgrund der Definition des fahrzeuginternen Koordinatensystems (siehe Abschnitt 2.3) eine besondere Bedeutung zu. Auftretende Fehler in der Vermessung der Straßenkrümmung und der Spurbreite machen sich direkt bei der Koordinatenbestimmung der Hinderniserkennerdaten bemerkbar.

Aufgrund mehrerer Faktoren, u.a. der Auflösung der digitalisierten Videosignale, Beleuchtungsänderungen und Sichtverdeckungen, sind die Messungen der Fahrspurbreite und der damit zusammenhängenden lateralen Position des autonomen Fahrzeugs sowie der Straßenkrümmung und des Gierwinkels mit einem statistischen Fehler versehen. Die boole'schen Variablen, welche die Existenz einer weiteren linken oder rechten Spur anzeigen, weisen aus gleichen Gründen sowohl Positiv- als auch Negativfehler auf.

Hinderniserkennung

Neben den Umwelteinflüssen, insbesondere den Beleuchtungsbedingungen, ergeben sich bei der Hinderniserkennung auch Fehler, die durch die Erkennungsmethode hervorgerufen werden. In VITA II sind verschiedene Ansätze zur Hinderniserkennung realisiert (siehe Kapitel 2). Bei den *modellbasierten* Methoden (CarTrack, ODT) wird ein Fahrzeug anhand von typischen Merkmalen beschrieben. Um die Echtzeitbedingung einhalten zu können, sind die verwendeten Merkmale zur Erkennung eines Fahrzeugs auf ein Minimum reduziert. Daraus ergeben sich zwangsläufig methodisch bedingte Fehlerursachen. Die Menge der hinreichenden Merkmale zur Extraktion eines Fahrzeugs aus dem Bild ist sehr klein gewählt, so daß es aufgrund ähnlicher Strukturen im Bild zu Positivfehlern bei der Objektdetektion kommen kann. Ein Beispiel dafür sind querliegende Schatten auf der Fahrbahn, die einen starken Kontrast bilden und mit der Unterkante eines vorausfahrenden Fahrzeugs verwechselt werden können.

Während der Fahrt führen Vibrationen und Schwankungen des Fahrzeugs zu Verschiebungen der einem Objekt zugeordneten Punkte im Bild. Insbesondere durch Nickbewegungen, die durch Bodenunebenheiten, Beschleunigen und Abbremsen hervorgerufen werden, ergeben sich Schätzfehler für die Entfernung des Objekts. Ein Fehler in der Distanzbestimmung induziert einen verstärkten Fehler bei der Geschwindigkeitsschätzung, die sich aus der Ableitung der Distanz über die Zeit ergibt.

Bei den nicht modellbasierten Methoden VisionBumper und LOD ergeben sich u.a. Fehler durch Bodenwellen, Bordsteine oder Leitplanken, die eine Erhebung darstellen und somit bei entsprechender SchwellwertEinstellung als Hindernis gesehen werden können.

Eine weitere Fehlerquelle ist die unvollständige Erfassung des Bereichs um das Fahrzeug, aus der sich Negativfehler ergeben.

Verkehrszeichenerkennung

Verkehrszeichen werden anhand von Farbregionen und Mustervergleichen erkannt. Was die Verkehrszeichenklasse angeht, so ist das Erkennungsmodul sehr robust. Die große Ähnlichkeit der Piktogramme begründet jedoch insbesondere bei teilweisen Verdeckungen eines Verkehrsschilds einen Verwechslungsfehler (gleichzeitiger Positiv- und Negativfehler). Die Echtzeitanforderung ist für die Verkehrszeichenerkennung weniger restriktiv. Es ist ausreichend, wenn ein Verkehrszeichen innerhalb des Erkennungsbereichs ein- bis zweimal erkennbar ist. Damit ist jedoch eine Fehlerkorrektur seitens des Erkennungsmoduls eingeschränkt und mit höheren Geschwindigkeiten steigt die Wahrscheinlichkeit ein Verkehrszeichen aufgrund der großen Zykluszeit zu verpassen.

3.4.1.2 Die Fusionsaufgabe

Sensor Fusion bedeutet das Verstehen von Szenen unter Verwendung von Informationen unterschiedlichen Typs [Pau 89]. In unserem Fall werden zwar lediglich Kameras zur Aufnahme von Umgebungsinformationen genutzt, Unterschiede ergeben sich jedoch durch die Anbringung der Kameras, deren Öffnungswinkel sowie vor allem durch unterschiedliche Software zur Bildauswertung.

Eine weitere Definition wird in [Jörg 91] gegeben: „Jeder Vorgang innerhalb eines Integrationsprozesses, bei dem Sensorinformationen in eine Repräsentationsstruktur abgebildet werden, heißt *Multisensorfusion* oder kurz *Fusion*.“

Es stellt sich nun die generelle Frage nach den Gründen für die Multisensorfusion. Dazu lassen sich die folgenden vier Hauptaspekte der multisensoriellen Information hervorheben:

Redundanz Redundanz führt einerseits zu einer erhöhten Fehlertoleranz des Systems, da ein Sensorausfall nicht mehr zwingend zum Ausfall des ganzen Systems führt. Andererseits wird die Wahrscheinlichkeit erhöht, daß ein real vorhandenes Merkmal in der Szene auch erkannt wird.

- Komplementarität* Informationen mehrerer Sensoren können sich ergänzen und ermöglichen so die Detektion von Merkmalen, die ein einzelner Sensor nicht erkennen könnte. Desweiteren wird so die Auflösung möglicher Mehrdeutigkeiten eines Sensors ermöglicht.
- Rechtzeitigkeit* Wird die Informationsgewinnung auf mehrere Sensoren aufgeteilt, so kann dies u.a. durch Parallelisierung zu einer geringeren Zykluszeit (einzelner Sensoren und des Gesamtsystems) führen.
- Kosten/Nutzen* Die erhöhten Kosten für eine größere Zahl von Sensoren sind in Relation zu dem damit erreichten Informationsgewinn und den damit verbundenen Leistungsmerkmalen des Systems zu setzen.

Betrachtet man die Fusionsaufgabe in VITA II unter diesen Aspekten, so sind alle aufgeführten Punkte auch hier wiederzufinden. In dem zur Kollisionsvermeidung wichtigsten Bereich vor dem Fahrzeug ist Redundanz durch drei Sensormodule zur Hinderniserkennung gegeben. Komplementarität ist durch die verschiedenen, sich ergänzenden Sichtbereiche der Hinderniserkennungsmodule gegeben. Durch die Aufteilung der Fahrzeugumgebung in Sichtbereiche verschiedener Bildverarbeitungsmodule wird eine Parallelverarbeitung ermöglicht und somit die oben aufgeführte Rechtzeitigkeit der Informationsgewinnung gewährleistet. Die Kosten/Nutzen-Frage stellt sich in diesem Fall nur bedingt, da bei dem vorgegebenen Ziel der autonomen Fahrzeugführung auf keines der Sensormodule verzichtet werden kann, sofern es der Abdeckung eines sonst nicht einseharen Bereichs um das Fahrzeug dient. Die Redundanz durch überlappende Sichtbereiche vor dem Fahrzeug ist durch die dadurch erreichte größere Zuverlässigkeit begründet, da gerade Fahrzeuge in diesem Bereich ein Kollisionsrisiko darstellen.

Zur Integration der Informationen können drei Hauptvorgehensweisen unterschieden werden. Nach der *Averaging*-Strategie werden die Informationsquellen nach Zuverlässigkeit des Sensors gewichtet und anschließend durch Durchschnittsbildung verknüpft. Dagegen wählt die *Deciding*-Strategie für jedes Merkmal die geeignetste Informationsquelle aus.

Neben diesen beiden Strategien ist die *Guiding*-Strategie zu erwähnen, die aufgrund von Umweltwissen gezielt die Messung eines Sensors anstößt. Nähere Erläuterungen zum Thema Multisensorfusion sind [Jörg 91] zu entnehmen.

Die erste Aufgabe der Fusion ist es, eine durchgehende Rundumsicht um das autonome Fahrzeug zu erhalten. Ein überholendes Fahrzeug wird beispielsweise zunächst von einem der hinteren Sensoren erfaßt und verfolgt, bis es dessen Sichtbereich verläßt. Im Idealfall ist der Übergang zum Sichtbereich des lateralen Sensors lückenlos, und das Fahrzeug kann von diesem übernommen und weiterverfolgt werden. Die Aufgabe besteht dann darin, ein in den Sichtbereich eines Sensors eintretendes Objekt als ein bereits bekanntes zu identifizieren. Existiert eine Lücke zwischen den Sichtbereichen, so erweitert sich die Aufgabenstellung insofern, als diese Lücke durch Extrapolation zurückliegender Meßwerte überbrückt werden muß.

Die zweite Aufgabe der Fusion ist die Kombination von Informationen verschiedener Sensoren, die überlappende Bereiche abdecken. Die Fusionsmethode muß Meßwerte kombinieren, die teilweise widersprüchlich sind, und muß das Korrespondenzproblem lösen. Dieses ergibt sich daraus, daß die Sensormodule jeweils mehrere Objekte in der gleichen Szene erkennen können. Ziel dieser Informationsverknüpfung ist einerseits die Verbesserung der Robustheit der Erkennung und andererseits die Erweiterung der Informationsmenge.

3.4.1.3 Glättung von Sensordaten

Genau betrachtet umfaßt die oben aufgeführte Definition der Fusion auch die Glättung der Sensordaten [Jörg 91], da es sich dabei um die Fusion der Information eines einzelnen Sensors über die Zeit handelt. Trotzdem erscheint eine isolierte Betrachtung der Meßwertglättung aufgrund ihrer Bedeutung sinnvoll.

Die einfachste Methode zur Glättung ist eine Mittelung der letzten Meßwerte einer Meßreihe. Wählt man ihre Zahl sehr groß, so erhält man eine starke Dämpfung des Meßrauschens, eine Veränderung des Meßwerts wird dann allerdings auch erst sehr spät registriert.

Eine geeignetere Methode der Meßwertfilterung erhält man durch die Integration eines Systemmodells in das Filter. Dabei wird nicht mehr eine isolierte Größe, sondern ein Systemzustand betrachtet. Zur optimalen Filterung der Sensordaten ist also zunächst ein diskretes Systemmodell und ein diskretes Meßmodell zu bestimmen.

Eine bekannte Methode zur Schätzung des zeitlich veränderlichen Systemzustands aus verrauschten Meßdaten stellt das Kalman-Filter [Schick 89, Brammer & Siffling 85] dar.

Um ein zeitdiskretes, rückgekoppeltes System zu definieren, werden zunächst folgende Vereinbarungen über die Schreibweise getroffen:

$x(t)$	Systemzustandsvektor zum Zeitpunkt t
$A(t)$	Systemmatrix zur Repräsentation der Abhängigkeiten der Zustandsgrößen untereinander
$w(t)$	Vektor zur Repräsentation des Systemrauschens, d.h. der erwarteten möglichen Störung der Zustandsgrößen
$z(t)$	Meßvektor zum Zeitpunkt t
$H(t)$	Meßmatrix
$v(t)$	Vektor zur Repräsentation des Meßrauschens

Ein System wird damit wie folgt beschrieben:

$$\begin{aligned}x(t+1) &= A(t) \cdot x(t) + w(t) \\z(t) &= H(t) \cdot x(t) + v(t)\end{aligned}$$

Das Kalman-Filter führt ein Modell des rauschfreien beobachteten Systems mit. Anhand dieses Modells wird der jeweilige Folgezustand prädiziert und diese Prädiktion zur Korrektur der nachfolgenden Messung verwendet.

Der Kalman-Filter-Algorithmus läßt sich in die Prädiktionsphase und die Korrekturphase unterteilen.

In der Prädiktionsphase wird der Zustandsvektor x^* für den Zeitpunkt $t+1$ aus dem vorliegenden Zustandsvektor x zum Zeitpunkt t sowie der Systemmatrix vorhergesagt. Gleichzeitig verwaltet das Filter eine Fehlerkovarianzmatrix $P(t)$ des Systemzustands. Diese wird ebenso für den Zeitpunkt $t+1$ prädiziert (P^*). Dabei wird eine Kovarianzmatrix $Q(t)$ des Systemrauschens in die Prädiktion einbezogen. Folgende Formeln werden verwendet:

$$\begin{aligned}P^*(t+1) &= A(t) \cdot P(t) \cdot A^T(t) + Q(t) \\x^*(t+1) &= A(t) \cdot x(t)\end{aligned}$$

Liegt ein Meßvektor vor, so wird in der Korrekturphase zunächst die Kalman-Matrix (Kalman-Gain) $K(t)$ bestimmt. Diese wird dann zur Korrektur des prädizierten Zustands und der Fehlerkovarianzmatrix herangezogen. Bezeichnet $R(t)$ die Kovarianzmatrix des Meßrauschens, dann ergeben sich folgende Gleichungen:

$$\begin{aligned}K(t) &= P^*(t) \cdot H^T(t) \cdot \left(H(t) \cdot P^*(t) \cdot H^T(t) + R(t) \right)^{-1} \\P(t) &= (I - K(t) \cdot H(t)) \cdot P^*(t) \\x(t) &= x^*(t) + K(t) \cdot (z(t) - H(t) \cdot x^*(t))\end{aligned}$$

Die Kalman-Matrix stellt zu jedem Zeitpunkt die optimale Kombination der systembedingten Vorkenntnisse mit der aktuellen Messung dar. Nähere Informationen zu Kalman-Filtern und deren Einsatz sind [Heichel 95, Hagen 93] zu entnehmen.

3.4.2 Architektur des Situationsverwaltungssystems

Das Ziel der Situationserstellung ist die Transformation von fehlerbehafteter, unvollständiger und inkonsistenter Information, wie sie vom Sensorsystem geliefert wird, in eine räumlich-zeitlich konsistente und vollständige Situationsbeschreibung. Die Aufgabenstellung unterteilt sich in drei Teilgebiete:

- *Verarbeitung der Straßeninformation*

Die hier gestellte Aufgabe ist die Glättung von Meßwerten um eine robuste Information über den Straßenverlauf, die eigene Fahrspur und angrenzende Fahrspuren zu erhalten.

- *Verarbeitung der Verkehrszeicheninformation*

Die von dem Verkehrszeichenerkennungsmodul gelieferten Informationen stellen jeweils eine Momentaufnahme der aktuell sichtbaren Verkehrszeichen dar. Für das Fahrverhalten ist jedoch das Wissen über die Gültigkeit einer Verkehrsregel von Bedeutung. Die Aufgabe ist, anhand der Sensorinformationen zu jeder Zeit anzugeben, welches Verkehrszeichen gerade gültig ist.

- *Verarbeitung der Hindernisinformation*

Die Verarbeitung der Hindernisinformation stellt die Hauptaufgabe dar. Dabei gilt es, eine räumlich-zeitlich konsistente Hindernisbeschreibung zu generieren. Teilaufgaben sind

- die Erkennung und Unterdrückung von Fehldetektionen der Sensormodule,
- Fusion der Hinderniserkenner in überlappenden Sichtbereichen,
- Extrapolation von Fahrzeugen, die kurzfristig nicht detektiert werden oder die sich in uneinsehbaren Bereichen befinden und
- die Glättung der Meßwerte.

Betrachtet man die oben angeführte Beschreibung eines Filters, so läßt sich die Aufgabe auch als wissensbasiertes Filtern der durch das Sensorsystem gelieferten Umweltinformation interpretieren. Eingabe des Filters ist die durch die Sensoren erfaßte Umgebung. An die Stelle der Systemmatrix aus obiger Beschreibung eines Kalman-Filters tritt hier eine Modellierung von Zusammenhängen und Systemzuständen der Komponenten der Situation. Die einfache Meßmatrix des Filters wird in diesem Fall durch eine Abbildung der Sensorinformationen auf die Komponenten der Situation ersetzt, und das erwartete Meßrauschen der Sensormodule a priori statistisch ermittelt. Diese Information wird zur Gewichtung der Sensorinformation bei der Bestimmung der Korrektur des Systemzustandes verwendet. Der jeweils

aktuelle Systemzustand, also die Situation $SitExtern_t$, dient als Eingabe für die Verhaltenssteuerung.

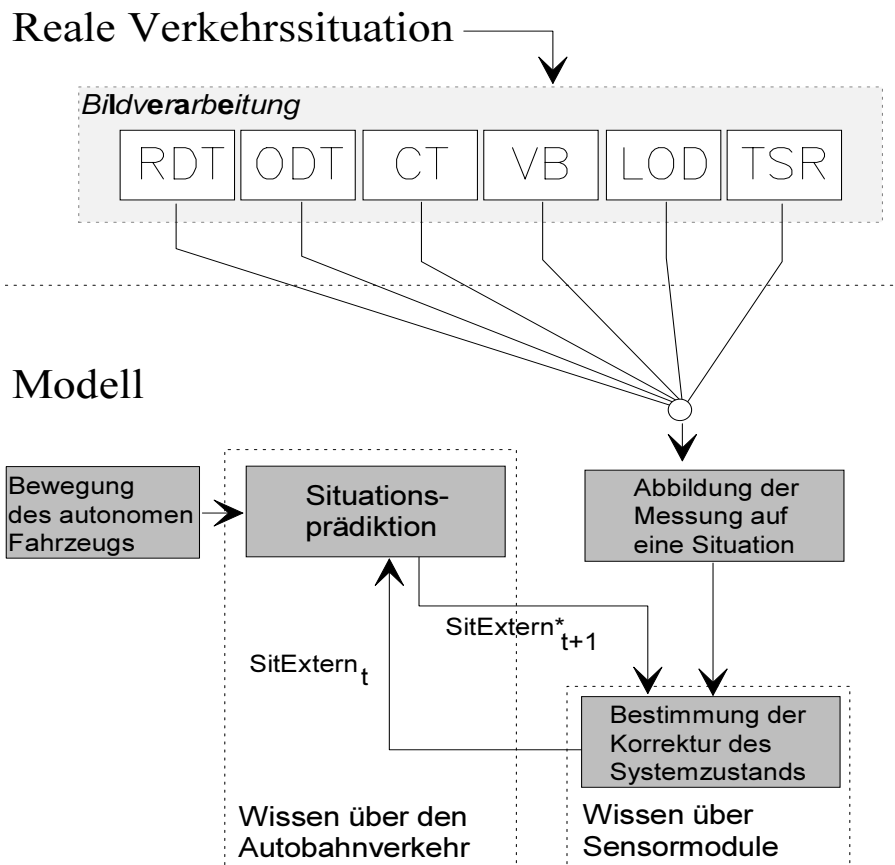


Abbildung 33 : Funktionsschaubild des SA-Moduls

Das System ist ähnlich einer Blackboard-Architektur aufgebaut (vgl. [Shapiro 87]). Die verwalteten Daten werden zentral in einem Szenenspeicher abgelegt, wo sie einer Menge von Prozessen zur Verfügung stehen. Einer dieser Prozesse stellt aus den aktuell im Szenenspeicher abgelegten Daten eine Situationsbeschreibung zusammen und übergibt diese an die Verhaltenssteuerung.

Das Situationsverwaltungssystem besteht demnach aus zwei Komponenten, einer Menge von verwalteten Objekten und einer Menge von Verwaltungsprozessen. Ein verwaltetes Objekt stellt eine Annahme über Position und Zustand eines Hindernisses, über den Verlauf einer Straße oder die Gültigkeit eines Verkehrszeichens aus der realen Welt dar. Aufgabe der Verwaltungsprozesse ist die Aufnahme, das Löschen und das Verändern von Objekten des Szenenspeichers.

Das im System enthaltene Wissen läßt sich unterteilen in

- *Objektwissen*, welches die Eigenschaften der Komponenten einer Verkehrssituation betrachtet und
- *Kontrollwissen*, welches die Vorgehensweise zum Aufbau einer Situation beschreibt.

Die erforderliche Echtzeitfähigkeit des Systems drückt sich durch ein strenges Effizienzkriterium der Wissensrepräsentation aus. Der hier verfolgte Ansatz beschränkt sich daher auf lokales, objektbezogenes Wissen. In Abschnitt 3.4.5 wird dann eine Ergänzung des Systems durch ein explizites schemabasiertes Fehlermodell vorgestellt, das auch Relationen zwischen Objekten berücksichtigt.

Eine weitere Form expliziten Wissens ist die Speicherung von Abhängigkeiten von Verkehrszeichen in einer Netzwerkstruktur. Diese wird in Abschnitt 3.4.4.2 näher beschrieben.

Aufgrund der Abhängigkeit des Systems von der Qualität der Sensorinformationen verfügt es über ein empirisch gewonnenes Modell der verwendeten Sensormodule, in dem die erwartete Qualität der Messungen, die Zykluszeiten sowie die abgedeckten Sichtbereiche festgehalten werden¹¹. Mit Hilfe dieses Wissens über die Sensormodule wird die Interpretation der Sensorinformationen und deren Verarbeitung durch Filter verbessert.

3.4.3 Konzeptrahmen als Repräsentationsstruktur

Zur Repräsentation des Objektwissens werden Konzeptrahmen (Frames) verwendet. Dieses Konzept für strukturierte Wissensrepräsentation geht auf Minsky [Minsky 75, Minsky 77] zurück. Minsky schlägt damit vor, daß ein Computer jeder neuen Szene mit einem Konzeptrahmen oder einer Hierarchie von Konzeptrahmen begegnet, die ihm helfen, die Situation zu antizipieren. Zu dieser Vorgehensweise finden sich auch Parallelen in der kognitiven Psychologie (vgl. [Neisser 76]). Präzise Details zu seinem Ansatz liefert Minsky jedoch nicht, so daß es weder einen festen Formalismus noch eine einheitliche Terminologie zu Konzeptrahmen gibt. Beschreibungen von Konzeptrahmen sind [Rich 88, Richter 89, Bibel 94] zu entnehmen. Eine besonders für die vorliegende Anwendung wichtige Eigenschaft der Konzeptrahmen ist es, daß die angegebenen Merkmale des modellierten Konzepts nicht vollständig zu sein brauchen. Weitere Gründe für die Nutzung der Konzeptrahmen als Wissensrepräsen-

¹¹ Der Erstellung des Modells für Sensormodule liegen statistische Untersuchungen im realen Straßenverkehr sowie in gestellten Testszenarien zugrunde. Diese Untersuchungen entstanden mit Unterstützung einer Diplomarbeit (FH) und eines Praktikums [Wieczorek 96, Schiele & Schneider 95]. Detaillierte Analyseergebnisse sind diesen Arbeiten zu entnehmen und werden hier nur soweit erläutert, wie sie zum Gesamtverständnis notwendig sind.

tationsform sind die dadurch gegebene Möglichkeit der Datenabstraktion und die Modularisierung.

Jedem Bestandteil der betrachteten Verkehrssituation wird ein Konzeptrahmen zugeordnet. Die Repräsentation des Objektwissens setzt sich demnach aus drei verschiedenen Typen von Konzeptrahmen zusammen:

- Konzeptrahmen für andere Verkehrsteilnehmer (Hindernisse),
- Konzeptrahmen für die Straße und
- Konzeptrahmen für geltende Verkehrszeichen.

Die Konzeptrahmen erlauben, u.a. durch Verwaltung von Default-Werten und Dämonen, einen einfachen, geordneten Zugriff mehrerer Administrationsprozesse auf das gleiche Objekt.

In der Terminologie der Konzeptrahmen bezeichnet ein *Slot* (Schlitz) eine strukturierte Variable. Die Komponenten dieser Variablen werden als *Facetten* bezeichnet. Zu den Facetten eines Slots gehören *Defaultwerte*, *Slotbedingungen* und sogenannte *Dämonen*. Defaultwerte stellen Arbeitshypothesen dar, die gewählt werden, falls ein Slot nicht mit einem Wert belegt ist. Slotbedingungen dienen zur Einschränkung des Wertebereichs eines Slots. Eine besondere Art von Facetten stellen die Dämonen dar. Es handelt sich dabei um angehängte Prozeduren, die beispielsweise zur Berechnung von Slotwerten, zur Fehlermeldung oder zur Aktualisierung abhängiger Slots genutzt werden.

Je ein Konzeptrahmen beschreibt ein Hindernis, eine Straße oder ein gültiges Verkehrszeichen. Die drei Konzeptrahmen werden im Folgenden genauer beschrieben.

3.4.3.1 Straßenkonzeptrahmen

Da der Konzeptrahmen eine Straße im Sinne der in Abschnitt 3.1 definierten Diskurswelt beschreiben soll, entsprechen die Slots den Komponenten der Straße. Der sich daraus ergebende Konzeptrahmen ist in Abbildung 34 schematisch dargestellt.

Während die Slots „Position_{AMV}“ und „Krümmung“ Zahlenwerte aufnehmen, werden die ersten drei Slots wiederum mit Konzeptrahmen gefüllt, die je eine Fahrspur beschreiben. Jedem Slot sind die Facetten Wert, Default und Wertebereich zugeordnet. Die Facette Default beschreibt den angenommenen Wert eines Slots für den Fall, daß keine Messung vorliegt. In diesem Fall wird von einer geraden Straße ausgegangen und die laterale Position des Fahrzeugs auf die Spurmitte gesetzt.

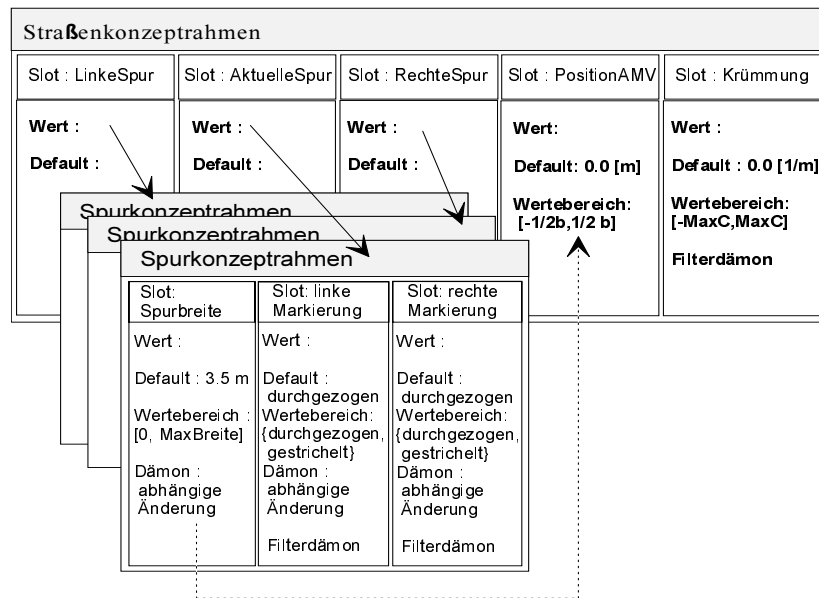


Abbildung 34 : Straßenkonzeptrahmen

Für die Slots „LinkeFahrspur“ und „RechteFahrspur“ wird eine Default-Instanz eines Spurkonzeptrahmens eingetragen, der eine Spur der Breite Null Meter beschreibt. Dagegen wird für den Slot „AktuelleSpur“ eine Default-Instanz mit der Spurbreite 3.50 Meter und durchgezogenen Markierungen festgelegt. Der Wertebereich für den Slot „Position_{AMV}“ ist durch die Spurbreite der aktuellen Spur begrenzt. Diese Abhängigkeit wird durch einen Dämon verwaltet, der dem Spurkonzeptrahmen zugeordnet ist. Bei jeder Änderung der Spurbreite der aktuellen Spur wird der Wertebereich mit Hilfe des Dämons angepaßt. Weitere Dämonen regeln die Übereinstimmung der Markierungen von angrenzenden Fahrspuren. Die Filterdämonen, die den Slots „Krümmung“, „linke Markierung“ und „rechte Markierung“ sowie „LinkeFahrspur“ und „RechteFahrspur“ zugeordnet sind, werden durch Eingabe eines neuen Meßwerts aktiviert und generieren den sich daraus ergebenden neuen Wert des Slots. Auf diese Dämonen wird im Rahmen der Beschreibung des Straßenverwaltungsprozesses noch näher eingegangen.

3.4.3.2 Verkehrszeichenkonzeptrahmen

Ein Konzeptrahmen für geltende Verkehrszeichen umfaßt die Komponenten eines Verkehrszeichens aus der Diskurswelt. Die hier modellierten Abhängigkeiten beschränken sich auf die Aufhebung von Streckenverboten durch nachfolgende Verkehrszeichen.

Ein Verkehrszeichenkonzeptrahmen besteht aus zwei Slots. Der erste Slot, der mit „VzKode“ benannt ist, beinhaltet den Code des Verkehrszeichens nach StVO. Der zweite Slot, „Zusatz“, beinhaltet Zusatzinformationen wie beispielsweise die Geschwindigkeitsangabe bei dem Verkehrszeichentyp Geschwindigkeits-

beschränkung. Die Wertebereiche der Slots entsprechen den in Anhang B beschriebenen Verkehrszeichenkodes und Zusatzinformationen.

3.4.3.3 Hinderniskonzeptrahmen

Ein Hinderniskonzeptrahmen beschreibt die äußere Form, die Position und den dynamischen Zustand eines Hindernisses. Darüber hinaus wird in dieser Repräsentationsform auch Information aufgenommen, die zur Verwaltung des Hindernisses im Szenenspeicher benötigt wird.

Die Slots, aus denen sich ein Hinderniskonzeptrahmen zusammensetzt, lassen sich demnach in zwei Gruppen unterteilen:

a) Form, Position und Zustand

Distanz	Der Wertebereich ist unbegrenzt. Der Defaultwert kann eine sehr große Distanz sein.
Geschwindigkeit	Der Wertebereich ist entsprechend des Autobahnszenarios auf positive Zahlen kleiner 70 (70 m/s = 252 km/h) festgelegt. Als Defaultwert wird die aktuelle Geschwindigkeit des autonomen Fahrzeugs verwendet.
Seitlicher Versatz	Der seitliche Versatz ist auf zwei Fahrspuren nach links und rechts begrenzt. Als Defaultwert wird ein Versatz von Null Metern angenommen.
Fahrzeugtyp	Der Wertebereich besteht aus den Typen PKW und LKW. Als Defaultwert wird ein PKW angenommen. Bei Eintrag in diesen Slot wird der Defaultwert für die Fahrzeuglänge, Breite und Höhe über einen Dämon aktualisiert.
Länge	Der Wertebereich für die Fahrzeuglänge ist auf positive Zahlen festgelegt. Als Default-Werte werden abhängig von dem Fahrzeugtyp fünf Meter für PKW und 10 Meter für LKW angenommen.
Breite	Der Wertebereich des Slots Fahrzeugbreite ist auf positive Zahlen beschränkt. Als Default-Werte werden 1,80 m für PKW und 2,20 m für LKW angenommen.
Höhe	Der Wertebereich für die Fahrzeughöhe ist ebenfalls auf positive Zahlen beschränkt. Als Default-Werte werden 1,50 m für PKW und 3 m für LKW angenommen.

b) Hindernisverwaltung

Vertrauen	Der Wertebereich liegt zwischen Null und Eins. Defaultwert für das Vertrauen in die Existenz eines Hindernisses ist Eins.
Blindzeit	Die Blindzeit bezeichnet die Zeitspanne von der letzten Messung für das Hindernis bis zum aktuellen Zeitpunkt. Der Wertebereich sind die positiven Zahlen. Defaultwert ist Null.
bestätigende Messungen	Eine Liste von Hindernismessungen verschiedener Hinderniserkennung, die das verwaltete Hindernis bestätigen. Der Wertebereich besteht aus Listen mit Elementen der Form (<i>Distanz, Geschwindigkeit, Versatz, Typ, Länge, Breite, Höhe, Sensor</i>). Defaultwert ist die leere Liste. Wird ein neues Element in die Liste eingefügt, so sorgt ein Dämon für das Rücksetzen des Slotwerts der Blindzeit auf Null.

Filter	Dieser Slot beinhaltet ein Filter zur Meßwertglättung mit allen von diesem verwendeten internen Daten. Die Verwendung des Filters ist als Dämon realisiert, der die Eingabe von bestätigenden Messungen entgegennimmt, daraus den dynamischen Zustand des Hindernisses bestimmt und die entsprechenden Slots aktualisiert. Auf diesen Dämon und die Filterstruktur wird später genauer eingegangen.
Status	Der Wertebereich ist durch die Menge {Kandidat, Bestätigt, Zurückgewiesen} gegeben. Defaultwert ist Kandidat.
Aufnehmen	Der Wertebereich ist zwischen Null und Eins. Defaultwert ist Null. Die Veränderung des Werts wird über einen Dämon geregelt, der die neuen Meßwerte aufnimmt und sowohl diesen Slotwert als auch den des Slots 'Status' verändern kann. Der Wert des Slots bewegt sich zwischen Null und Eins, solange das verfolgte Hindernis nicht ausreichende Bestätigungen durch Messungen erhalten hat.
Verlieren	Der Wertebereich ist zwischen Null und Eins. Defaultwert ist Null. Die Veränderung des Werts wird über einen Dämon geregelt, der die neuen Meßwerte aufnimmt und sowohl diesen Slotwert als auch den des Slots 'Status' verändern kann. Der Wert des Slots bewegt sich zwischen Eins und Null solange das verfolgte Hindernis den Status „Bestätigt“ hat und eine Konfidenz in seine Existenz besteht.

3.4.4 Verwaltung des Szenenspeichers

Das Kontrollwissen ist durch den Blackboard-Manager sowie durch die Prozesse festgelegt, die auf den gemeinsamen Szenenspeicher zugreifen. Da für die Straßen-erkennung und Verkehrszeichenerkennung nur jeweils ein Erkennungsmodul zur Verfügung steht, ist das Hauptaugenmerk auf die Verknüpfung der verschiedenen Informationsquellen zur Hinderniserkennung zu lenken. Die folgenden sechs Administrationsprozesse haben Zugriff auf den Szenenspeicher :

<i>Straßenverwaltung</i>	Glätten der Information des Straßenerkenners und Transformation der Sensordaten in die im Szenenspeicher verwendete Struktur
<i>Verkehrszeichenverwaltung</i>	Verwaltung von Verkehrszeichenrahmen für aktuell geltende Verkehrszeichen im Szenenspeicher
<i>„Matching“</i>	Zuordnen neuer Messungen zu Hindernisobjekten im Szenenspeicher
<i>„Set & Forget“</i>	Aufnehmen und Löschen von Hindernisobjekten
<i>„Merge & Split“</i>	Zusammenfassen und Auseinanderbrechen verwalteter Hindernisobjekte des Szenenspeichers
<i>„Filter & Extrapolation“</i>	Glätten der kombinierten Meßwerte der Hinderniserkennung und Extrapolation der Werte, falls keine Messung vorliegt

Die Administrationsprozesse werden innerhalb eines Abarbeitungszyklus sequentiell in festgelegter Reihenfolge aufgerufen¹². Am Ende des Zyklus wird die Straßen-

¹² Wird paralleler, konkurrierender Zugriff mehrerer Administrationsprozesse auf den Szenenspeicher zugelassen, so ist ein Semaphorkonzept einzuführen, um die Datenkonsistenz zu gewährleisten.

information, die Verkehrszeicheninformation und die Hindernisse, die den Status 'Bestätigt' haben, an die Verhaltenssteuerung weitergegeben.

Abbildung 35 veranschaulicht den Aufbau des Situationsverwaltungssystems. Die einzelnen Prozesse werden in den folgenden Abschnitten näher beschrieben. Die schemabasierte Situationsanalyse wird als Erweiterung des Systems in Abschnitt 3.4.5 erläutert.

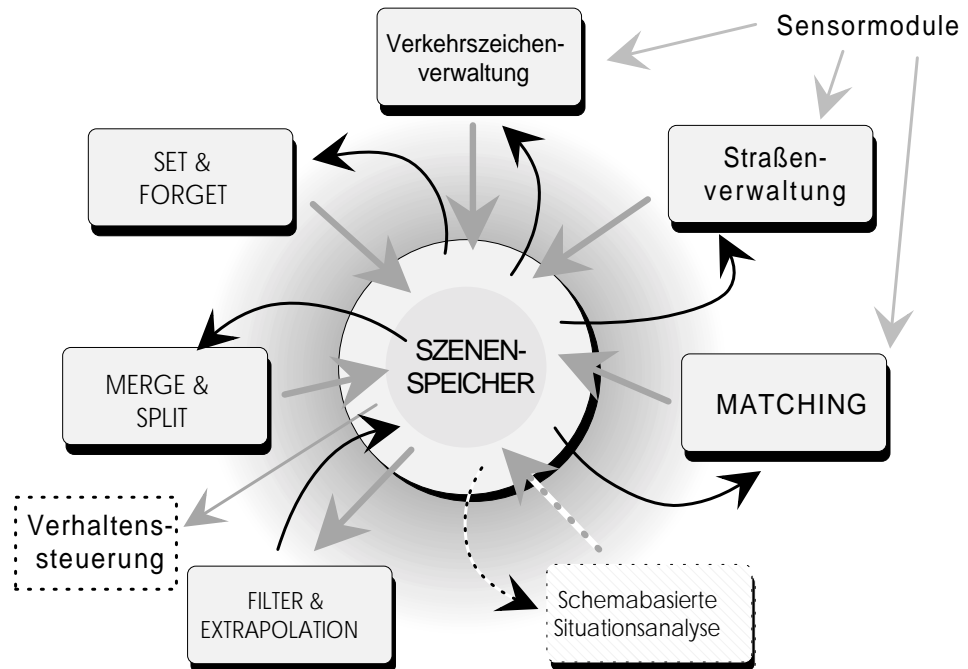


Abbildung 35: Aufbau des Situationsverwaltungssystems

3.4.4.1 Der Straßenverwaltungsprozeß

Aufgabe des Straßenverwaltungsprozesses ist die Abbildung der Information des Straßenerkenners von VITA II auf einen intern verwalteten Straßenkonzeptrahmen. Da das Versuchsfahrzeug nur über einen einzigen Straßenerkennungsalgorithmus verfügt, kommt hier lediglich der zeitliche Aspekt der Fusion, die Glättung, zum Tragen. Aufgrund der bereits vorgenommenen Filterung der Meßwerte im Straßenerkennungsmodule werden Krümmung, Straßenbreite und Position des autonomen Fahrzeugs keiner weiteren Glättung unterzogen.

Die Informationen über existierende linke und rechte Fahrspuren sowie Informationen über die Art der Spurmarkierung hingegen werden durch ein Hysterese-Filter geglättet.

Bezeichnet $P_A(t): \mathfrak{R} \rightarrow \{TRUE, FALSE\}$ den Wert eines Slots A zum Zeitpunkt t, dann ist ein Hysterese-Filter für das Attribut A durch die nachfolgenden Formeln beschrieben. Der Einfachheit halber wird der Wertebereich eines Slotinhalts auf die boole'schen Werte TRUE und FALSE abgebildet.

$$Z(t) = \frac{\sum_{i=0}^n P_A^M(t - i \cdot dt_{SA})}{n + 1}$$

$$P_A(t) = (P_A(t - dt_{SA}) \wedge Z(t) > s_H) \vee (Z(t) > s_S)$$

Dabei bezeichnet $P_A^M(t)$ die dem Slot zugehörige Messung zum Zeitpunkt t , $Z(t)$ die Zusicherung für Slot A zum Zeitpunkt t und s_H und s_S Schwellwerte für das Halten und das Setzen eines Slotwerts. Für die Schwellwerte gilt $0 < s_H < s_S < 1$. Die Zeit wird dabei gemäß der Zykluszeit dt_{SA} des SA-Moduls diskretisiert. Bei zeitkontinuierlicher Betrachtung ergibt sich eine Hysterese für die Bestimmung der Slotwerte. Die Anzahl n der gespeicherten Meßwerte ist dabei abhängig von der Güte des Sensors zu wählen.

Ein solches Hysterese-Filter ist mehreren Slots des Straßenkonzeptrahmens als Dämon zugeordnet, die einen zweiwertigen Wertebereich besitzen. Der Dämon nimmt die aktuelle Messung für den Slot entgegen und bestimmt nach oben aufgeführter Methode den Slotwert.

3.4.4.2 Der Verkehrszeichenverwaltungsprozeß

Aufgabe der Verkehrszeichenverwaltung ist die Bereitstellung der Information über die geltenden Verkehrszeichen. Das Verkehrszeichenerkennungsmodul liefert Informationen über ein erkanntes Verkehrszeichen in der aktuell betrachteten Szene. Diese bestehen aus einem Verkehrszeichenkode gemäß der Straßenverkehrsordnung (vgl. [HAV 87]) und einer Zusatzinformation zur genaueren Spezifikation der Information des Verkehrszeichens (siehe Anhang B).

Bei Erkennung eines Verkehrszeichens wird eine Instanz eines Verkehrszeichenkonzeptrahmens gebildet, sofern dieses Verkehrszeichen nicht bereits früher erkannt worden ist und noch Gültigkeit besitzt. Alle bereits existierenden Verkehrszeichenkonzeptrahmen werden auf ihre weitere Gültigkeit überprüft.

Um die Überprüfung der Gültigkeit durchzuführen, ist das Wissen über Abhängigkeiten zwischen Verkehrszeichen in einem Netzwerk abgelegt, in dem die semantischen Relationen zwischen Verkehrszeichen gespeichert sind. Die Gültigkeit von Verkehrszeichen ist entweder auf eine bestimmte Weglänge begrenzt, bzw. nur im Zusammenhang mit einer Umweltkonstellation (Baustelle, Kurve etc.) gültig, oder sie wird explizit durch ein weiteres Verkehrszeichen aufgehoben. Da die Erfassung der Umgebung unzureichend ist, um ersteres zu modellieren, beschränkt sich die Verwaltung der Verkehrszeichen auf explizite Aufhebungen der Verkehrszeichen untereinander.

Das zur Repräsentation der Verkehrszeicheninformation verwendete Abhängigkeitsnetzwerk ist wie folgt aufgebaut.

Definition 3.4.1 Verkehrszeichenknoten

Die Menge der Verkehrszeichenknoten VzN ist definiert als eine Menge von Tupeln der Form (Zeichenkode, Typ, Instanzreferenz). Dabei bezeichnet Zeichenkode einen dreistelligen Zahlenkode zur Identifikation von Verkehrszeichen gemäß [HAV 89]. Das zweite Element des Tupels, $Typ \in \{\text{Verbot, Aufhebung}\}$, bestimmt, ob es sich um ein Streckenverbotszeichen oder ein reines Aufhebungszeichen handelt. Ist ein Verkehrszeichen mit diesem Kode aktuell gültig, so zeigt das dritte Element des Tupels auf die Instanz eines Verkehrszeichenkonzeptrahmens. Anderenfalls wird eine leere Referenz eingefügt.

Definition 3.4.2 Abhängigkeitskanten

Die Menge der Abhängigkeitskanten AK ist definiert als

$$AK = \{(v_1, v_2) \mid v_1, v_2 \in VzN, v_1 \text{ hebt } v_2 \text{ auf}\}$$

Definition 3.4.3 Abhängigkeitsnetzwerk

Das Abhängigkeitsnetzwerk ist definiert als eine Menge VzN von n Verkehrszeichenknoten und eine Menge AK von m Abhängigkeitskanten. Zusätzlich ist als Randbedingung für die Mengen VzN und AK gefordert, daß es keine isolierten Verkehrszeichenknoten gibt:

$$\forall v \in VzN: \exists v^* \in VzN: (v, v^*) \in AK \vee (v^*, v) \in AK.$$

In diesem Abhängigkeitsnetz sind diejenigen Verkehrszeichen abgelegt, die durch das Verkehrszeichenerkennungsmodul des Fahrzeugs erkannt und später verarbeitet werden können. Im wesentlichen handelt es sich dabei um das Überholverbot, Geschwindigkeitsbegrenzungen und deren Aufhebungsschilder. Isolierte Knoten werden ausgeschlossen, um eine unbefristete Gültigkeit von Verkehrszeichen zu verhindern.

Beschreibt TS_{SA} die Menge der aktuell im Szenenspeicher verwalteten Verkehrszeichenkonzeptrahmen und TS_{Neu} die Menge der aktuell durch das Verkehrszeichenerkennungssystem gelieferten Verkehrszeichen, dann läßt sich der Algorithmus zur Verkehrszeichenverwaltung wie folgt beschreiben:

- (1) Der Prozeß wartet darauf, daß neue Verkehrszeichen erkannt werden.
- (2) Aus der Menge TS_{Neu} werden alle redundanten und in ihrer Bedeutung widersprüchlichen Verkehrszeichen gelöscht. Für den Fall mehrerer erkannter Geschwindigkeitsbeschränkungen wird beispielsweise die restriktivste ausgewählt.
- (3) Das erste Element $ts \in TS_{Neu}$ wird ausgewählt und in TS_{Neu} gelöscht.
- (4) Bezeichne k den Verkehrszeichenknoten, dessen Kode dem von ts entspricht, dann bestimmt die Menge $D_k = \{i \mid i \in VzN, (k,i) \in AK\}$ die durch das neue Zeichen aufgehobenen Streckenverbote. Die von den Knoten $i \in D_k$ referenzierten Verkehrszeichenkonzeptrahmen werden aus TS_{SA} gelöscht.
- (5) Für das Verkehrszeichen ts wird ein Konzeptrahmen angelegt und in TS_{SA} eingetragen, falls es sich um ein Streckenverbot handelt.
- (6) Ist die Menge TS_{Neu} leer, so wird in (1), ansonsten in (3) fortgefahren.

3.4.4.3 Der „Matching“-Prozeß

Der Matching-Prozeß ist zuständig für die Zuordnung neuer Messungen zu bereits im Szenenspeicher verwalteten Hinderniskonzeptrahmen. Dazu wird ein Ähnlichkeitswert für den Vergleich des gemessenen Hindernisses mit dem durch einen Konzeptrahmen repräsentierten Hindernis im Szenenspeicher bestimmt.

Es bezeichnet S eine Menge von Sensormodulen¹³, O_s eine Menge von Hindernissen, die durch ein Sensormodul $s \in S$ erkannt wird, A eine Menge von Attributen, $v: A \rightarrow \mathfrak{R}$ den Wert eines Attributs und $r: S \times A \times O_s \rightarrow [0,1]$ eine Zuverlässigkeitsfunktion. Diese Funktion r wird empirisch aus den Messungen der Sensormodule gewonnen und beschreibt die Zuverlässigkeit der Messung eines Attributs durch ein Sensormodul in Abhängigkeit von der Position des Hindernisses. Weiterhin sei $s: O \rightarrow S$ eine Zuordnung eines Objekts zu dem Sensormodul, das es erkannt hat.

Ein Hindernis $o \in O = \bigcup_{s \in S} O_s$ wird durch n Attribute $a_{1,o}, \dots, a_{n,o}$ beschrieben.

Neben der Zuverlässigkeit der Attributwerte wird auch eine Relevanz der Attribute zur Bestimmung des Distanzmaßes berücksichtigt. Dazu sind jedem Attribut A ein Gewicht w_A und ein Toleranzfaktor e_A zugeordnet. Bezeichne d_A den Betrag der Differenz von zwei zu vergleichenden Attributwerten, so ist die Gewichtungsfunktion f wie folgt definiert:

$$f(w_A, e_A, d_A) = w_A \cdot d_A^{e_A}$$

Mit Hilfe der Gewichtungsfunktion f wird eine Distanzfunktion $d: O \times O \rightarrow \mathfrak{R}^+_o$ bestimmt, die zum Vergleich zweier Messungen auch die Sensormodule mit einbezieht:

$$d(o_1, o_2) = \frac{\sum_{i=1}^n f(w_{a_i}, e_{a_i}, |v(a_{i,o_1}) - v(a_{i,o_2})|) \cdot r(s(o_1), a_i, o_1) \cdot r(s(o_2), a_i, o_2)}{\sum_{i=1}^n w_{a_i} \cdot r(s(o_1), a_i, o_1) \cdot r(s(o_2), a_i, o_2)}$$

Damit d der Definition eines Distanzmaßes entspricht müssen einige Voraussetzungen erfüllt sein (vgl. [Weß 91]). Für $d(o_1, o_1)$ gilt wegen $f(w, e, 0) = 0$ stets $d(o_1, o_1) = 0$ (*Reflexivität*). Die Symmetrieeigenschaft ergibt sich aus der Symmetrie der Multiplikation. Analog zur Reflexivität gilt auch $x=y \Rightarrow d(x, y) = 0$. Für die Gegenrichtung ist der Definitionsbereich für d einzuschränken auf $\text{Def}_d = \{(o_1, o_2) \mid o_1 \in O, o_2 \in O, \exists i, 0 \leq i \leq n: (w_{a_i} \cdot r(s(o_1), a_i, o_1) \cdot r(s(o_2), a_i, o_2) > 0)\}$.

Bei Vorliegen von Messungen ist dies bei den verwendeten Sensormodulen und Attributen gegeben. Diese Einschränkung genügt jedoch nicht, um aus $d(x, y) = 0$ die

¹³ Für diese Betrachtung stellt der Szenenspeicher selbst ebenfalls ein Sensormodul im Sinne einer Datenquelle dar.

Identität von x und y folgen zu lassen, da es (theoretisch) möglich ist, daß zwei Sensormodule ein Objekt exakt gleich messen. Man spricht in diesem Fall von einer Semi-Distanz (vgl. [Weß 91]).

Die Ähnlichkeitsfunktion $sim: O_A \times O_B \rightarrow [0,1]$ leitet sich aus der oben definierten Distanz d wie folgt ab:

$$sim(o_1, o_2) = 1 - \left(\frac{d(o_1, o_2)}{1 + d(o_1, o_2)} \right)^k \quad k \in \mathbb{N}^+$$

Mit Hilfe der Parameter k , w_a und e_a läßt sich das Distanzmaß den Anforderungen der Fusion anpassen. Abbildung 36 zeigt eine Ähnlichkeitsfunktion mit den Parametern $k = 4$, $w_{Distanz} = 1$, $w_{Offset} = 2$, $e_{Distanz} = 2$ und $e_{Offset} = 3$. Diese Werte sind empirisch ermittelt und stellen ein bewährtes Maß für die in der Simulation eingesetzte Matching-Funktion dar. Der Einfachheit halber liegen der Grafik konstante Zuverlässigkeitswerte für die Messungen zugrunde.

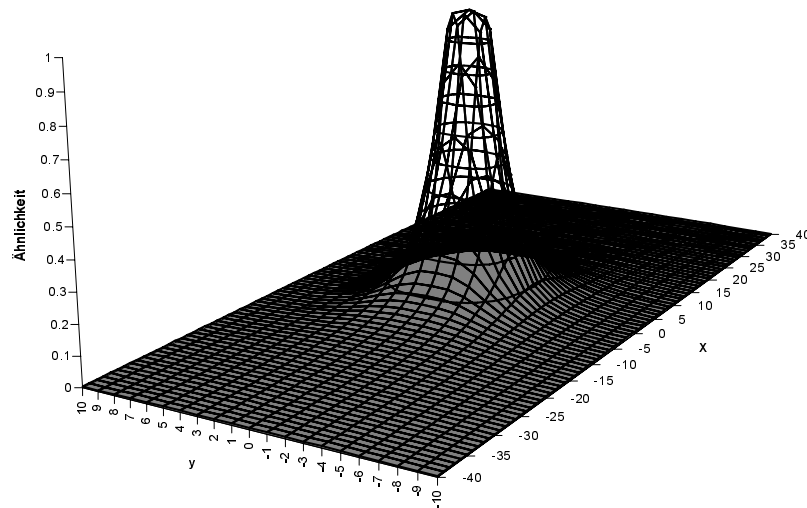


Abbildung 36 : Einfache Ähnlichkeitsfunktion für Hindernispositionen

Eine zweite Konfiguration hat sich insbesondere beim Einsatz im Fahrzeug als robuster erwiesen. Hier wird im Vergleich zu obigem Beispiel der Parameter $e_{Distanz}$ auf 1 gesetzt. Damit wird die Ähnlichkeit zwischen Hindernissen vergrößert, deren Position sich hauptsächlich in Längsrichtung unterscheidet. Das bringt vor allem bei der Verfolgung eines vorausfahrenden Fahrzeugs Vorteile mit sich. Insbesondere wird mit diesem Aufbau der Ähnlichkeitsfunktion der Tatsache Rechnung getragen, daß sich voreinander fahrende Fahrzeuge verdecken und somit nur mit geringer Wahrscheinlichkeit beide erkannt werden.

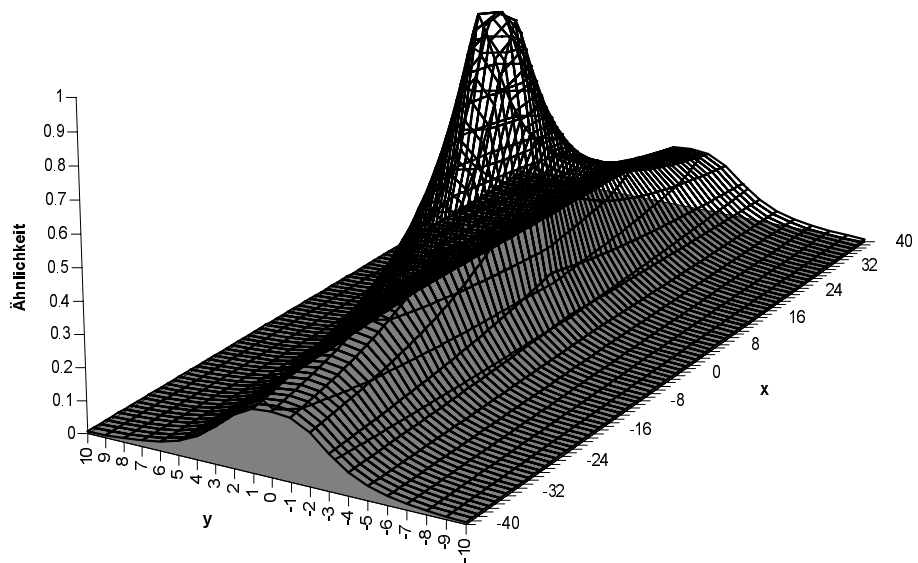


Abbildung 37 : Ähnlichkeitsfunktion für Verfolgungsfahrten

Ist die größte Ähnlichkeit eines neuen Hindernisses mit einem vorhandenen Hindernis im Szenenspeicher kleiner als ein Akzeptanzschwellwert s_M , dann wird diese neue Messung keinem vorhandenen zugeordnet, sondern es wird ein neuer Hinderniskonzeptrahmen instanziiert. Anderenfalls wird es als bestätigende Messung in dem Hinderniskonzeptrahmen mit der größten Ähnlichkeit eingetragen.

3.4.4.4 Der „Set & Forget“- Prozeß

Aufgabe des Set & Forget-Prozesses ist es, zwischen existierenden, relevanten Hindernissen einerseits und Fehldetektionen sowie irrelevanten Hindernissen andererseits zu unterscheiden.

Die Unterscheidung zwischen relevanten und irrelevanten Hindernissen wird durch die vom System gegebene Ressourcenbegrenzung nötig. Um eine Echtzeitverarbeitung auf der Zielhardware zu gewährleisten, ist die Anzahl der verwalteten Hinderniskonzeptrahmen in diesem Kontext auf 15 begrenzt.

Sobald ein neues Hindernis von einem Sensor modul erkannt wird, instanziiert der oben beschriebene Matching-Prozeß einen neuen Konzeptrahmen und trägt diesen in den Szenenspeicher ein. Der Status des Konzeptrahmens wird gleichzeitig auf den Wert „Kandidat“ gesetzt. Bevor dieser Status jedoch nicht den Wert „Bestätigt“ hat, wird das verwaltete Hindernis nicht an die Verhaltensentscheidung weitergeleitet. Es bleibt für diese somit unsichtbar.

Der Set & Forget Prozeß bearbeitet für jeden Hinderniskonzeptrahmen die Slots ‘Aufnehmen’ und ‘Verlieren’. Der Wert des Slots ‘Aufnehmen’ eines Hindernis-

konzeptrahmens H zum Zeitpunkt t ergibt sich aus den Sensormodulen, die bisher zur Erkennung beigetragen haben. Zu jedem Sensormodul wird anhand der Häufigkeit möglicher Fehldetektionen eine Anzahl von benötigten Bestätigungen einer Messung festgelegt. Der Beitrag einer Messung eines Sensormoduls s zum Wert des ‘Aufnehmen’ Slots ergibt sich aus dem Kehrwert der Zahl der benötigten Bestätigungen n_s . Findet innerhalb der Zykluszeit des Sensormoduls dt_s keine weitere Bestätigung statt, so wird sein Beitrag gelöscht. Bezeichne $d_{c_s}(t)$ die Detektion eines Objekts durch Sensormodul s zum Zeitpunkt t und S die Menge der verwendeten Sensoren, dann läßt sich der Wert des ‘Aufnehmen’-Slots zum Zeitpunkt t , $u(t)$, wie folgt bestimmen:

$$b_s(t) = \sum_{s=0}^{n_s-1} dt_c(t - s \cdot dt_s) \quad c(o_s, t) = \frac{b_s(t)}{n_s} \quad u(t) = \min\left(\sum_{s \in S} c(o_s, t), 1\right)$$

Erreicht der Wert des ‘Aufnehmen’-Slots den Wert Eins, so wechselt der Status des Hinderniskonzeptrahmens auf ‘Bestätigt’ und es wird „gesehen“. Erreicht er jedoch wieder Null, so wird der Hinderniskonzeptrahmen aus dem Szenenspeicher gelöscht. Die Akzeptanz-Latenzzeit, die durch die benötigte Bestätigung von Meßwerten hervorgerufen wird, ist gering zu halten, d. h. im Bereich weniger Videozyklen, da ansonsten die Reaktionszeit des autonomen Fahrzeugs intolerabel vergrößert wird. Auf der anderen Seite ist es Aufgabe des Set & Forget-Prozesses, lange Zeit unbestätigte, extrapolierte Hinderniskonzeptrahmen aus dem Szenenspeicher zu entfernen, d. h. sozusagen zu „vergessen“. Ähnliches gilt für Szenarien, in denen mehr Fahrzeuge verwaltet werden müssen als eine Echtzeitverarbeitung zuläßt. Hier müssen diejenigen Hindernisse vernachlässigt werden, die zur Zeit die geringste Gefahr für das autonome Fahrzeug darstellen. Als Gefahrenkriterium wird hier die Entfernung des Hindernisses gewählt und im Fall von zu vielen Objekten diejenigen mit der größten Entfernung gelöscht.

Betrachtet man nur den ersten Fall, so ist die Zeitspanne, in der unbestätigte, extrapolierte Hinderniskonzeptrahmen weitergeführt werden, von zwei Faktoren abhängig:

- (1) von der Zeit und Zuverlässigkeit der Messungen, bevor diese ausblieben und
- (2) von der Position des Hindernisses, insbesondere der Zuordnung zum Sichtbereich eines Sensors.

Zur Behandlung extrapolierter Hinderniskonzeptrahmen wird der ‘Vergessen’ Slot verwendet. Jeder neuen Messung wird (durch das jeweilige Sensormodul) ein Grad der Zusicherung $g_M \in [0,1]$ zugeordnet. Dieser wird mit dem bisherigen Wert des ‘Vergessen’-Slots, welcher die Zusicherung für den Hinderniskonzeptrahmen selbst

darstellt, gewichtet verknüpft. Liegt keine Messung vor, so gilt $g_M = 0$. Der Wert des ‘Vergessen’-Slots wird über die Funktion $f: T \rightarrow [0,1]$ bestimmt:

$$f(t) = \alpha \cdot f(t - dt_{SA}) + \beta \cdot g_M(t)$$

Die Koeffizienten α und β werden jedesmal abhängig von den Sensormodulen gewählt, in deren Sichtbereich sich das gemessene (oder extrapolierte) Hindernis befindet. Dabei gilt $\alpha, \beta \geq 0$ und $\alpha + \beta = 1$.

Befindet sich ein extrapoliertes Hinderniskonzeptrahmen im Szenenspeicher, dessen zugeordnetes Hindernis sich in dem Sichtbereich eines Sensors befindet, so bedeutet dies, daß es von diesem nicht, oder an anderer Position erkannt wird. Liegt die Wahrscheinlichkeit für eine korrekte Detektion eines Hindernisses an dieser Position durch einen Sensor sehr hoch, so ist die extrapolierte Position weniger plausibel als bei einer niedrigen Detektionswahrscheinlichkeit. Insbesondere ist die Wahrscheinlichkeit für eine korrekte Detektion eines Hindernisses in ungedeckten Bereichen gleich Null.

In nachfolgender Betrachtung bezeichnet $P_S(o)$ die Detektionswahrscheinlichkeit eines Hindernisses o durch das Sensormodul S . Der Koeffizient β entspricht der Detektionswahrscheinlichkeit des Hindernisses. Es sind bei der Bestimmung des Koeffizienten β zwei Sonderfälle zu beachten:

- Das Hindernis befindet sich im Sichtbereich mehrerer Sensormodule.
- Das Hindernis befindet sich außerhalb der abgedeckten Sichtbereiche.

Im ersten Fall müssen die Detektionswahrscheinlichkeiten der Sensormodule mit überlappenden Sichtbereichen, $S_U \subseteq S$, verknüpft werden. Bei *unabhängigen* Wahrscheinlichkeiten bietet sich dazu die folgende Verknüpfung an:

$$P(o) = 1 - \left(\prod_{s \in S_U} (1 - P_S(o)) \right)$$

Zwar handelt es sich bei den verschiedenen Sensormodulen um methodisch unterschiedliche Verfahren, sie verwenden jedoch den gleichen Sensortyp und unterliegen damit einer Vielzahl von Einflüssen, die auf diesen zurückzuführen sind. Die völlige Unabhängigkeit der Detektionswahrscheinlichkeiten ist demnach nicht gegeben. Berücksichtigt man bei der Verknüpfung, daß die Detektionswahrscheinlichkeit bei Hinzunahme eines weiteren Sensormoduls nicht geringer wird, so ergibt sich eine Begrenzung der Wahrscheinlichkeit durch

$$P(o) \geq \max_{s \in S_U} (P_S(o)).$$

Je nach dem Grad der Abhängigkeit der Detektionswahrscheinlichkeiten liegt die Gesamtwahrscheinlichkeit also zwischen den beiden oben bestimmten Werten.

$$\max_{s \in \mathcal{S}_U} (P_s(o)) \leq \beta \leq 1 - \left(\prod_{s \in \mathcal{S}_U} (1 - P_s(o)) \right)$$

Die hier gewählte Verknüpfungsmethode berechnet β über den Mittelwert aus den beiden oben aufgeführten Wahrscheinlichkeiten. Der zweite Koeffizient α ergibt sich als Komplement von β zu $\alpha = 1 - \beta$.

Befindet sich ein Hindernis außerhalb der abgedeckten Sichtbereiche, so ist die Detektionswahrscheinlichkeit Null. D. h. nach obiger Berechnung würde β zu Null und damit bliebe der Wert des ‘Vergessen’-Slots unverändert und das Hindernis würde nicht vergessen. Aufgrund der Dynamik von Verkehrsszenen einerseits sowie der Ressourcenbegrenzung andererseits, ist es sinnvoll, einen konstanten Faktor k_f einzuführen, der β in diesem Fall zugewiesen wird.

Unterschreitet der Wert des ‘Vergessen’-Slots einen Schwellwert $s_f > 0$, so wird der Hinderniskonzeptrahmen aus dem Szenenspeicher gelöscht.

3.4.4.5 Der „Merge & Split“-Prozeß

Da die Meßwerte zeitweise stark verrauscht sind, kann es vorkommen, daß zwei Fahrzeuge als ein einziges interpretiert werden, bzw. ein reales Fahrzeug durch Abweichungen der Meßwerte verschiedener Sensormodule als zwei oder mehrere Hindernisse erkannt wird. Dies geschieht insbesondere bei Merkmalsverwechslungen durch ein Hinderniserkennungsmodule.

Der Merge & Split - Prozeß untersucht, ob sich Hindernisse im Szenenspeicher überlappen. Ist dies der Fall, so werden diese in ihre bestätigenden Messungen aufgespalten und erneut kombiniert, wobei entweder zwei klar getrennte Objekte herauskommen, oder ein einziger Hinderniskonzeptrahmen, dem alle Messungen zugeordnet werden. Diese werden neu in den Szenenspeicher eingefügt, ohne jedoch die bisherigen Hinderniskonzeptrahmen zu löschen.

Findet der Prozeß einen Hinderniskonzeptrahmen, dessen bestätigende Messungen sich in zwei klar getrennte Hindernisse gruppieren lassen, so werden zwei neue Hinderniskonzeptrahmen erzeugt und als ‘Kandidat’ in den Szenenspeicher eingefügt.

3.4.4.6 Der „Filtering & Extrapolation“-Prozeß

Bevor die Hinderniskonzeptrahmen einem Filter übergeben werden können, müssen die aktuellen Messungen und die vorangehenden Werte fusioniert werden. Dazu können verschiedene Methoden angewandt werden. Die einfachste Methode ist die Auswahl derjenigen Messung, die von dem Sensormodul stammt, welches a priori als das zuverlässigste eingestuft wurde.

Eine zweite Methode wird im Folgenden beschrieben. Dazu beschreibe $A^S = (a_0^S, \dots, a_n^S)$ einen vollständigen Attributvektor einer Hindernisbeschreibung, die durch ein Sensormodul s erzeugt wird und $R^S = (r_0^S, \dots, r_n^S)$ einen zugeordneten Zuverlässigkeitsvektor. Ist Attribut a_i^S nicht unter den von Sensor s gemessenen Merkmalen, dann wird r_i^S auf Null gesetzt.

Die Fusionsaufgabe ist offensichtlich einfach für den Fall, daß die verknüpften Sensormodule disjunkte Attributmengen liefern. In diesem Fall werden die gelieferten Attributmengen lediglich vereinigt. Das CT-Modul liefert beispielsweise eine Klassifikation des Hindernisses nach PKW und LKW. Da diese Information von keinem weiteren Sensormodul geliefert wird, wird der zugehörige Slot des Hinderniskonzeptrahmens auf den vom CT-Modul gelieferten Wert gesetzt.

Allgemein läßt sich die Fusion als eine gewichtete Summe der Attributwerte mit ihren zugeordneten Zuverlässigkeitswerten bestimmen:

$$a_i = \frac{\sum_{j=1}^{|S|} (a^{S_j} \cdot r_i^{S_j})}{\sum_{j=1}^{|S|} r_i^{S_j}}.$$

Beide Methoden sind realisiert und können alternativ eingesetzt werden. In jedem Fall ist das Ergebnis der Fusion ein einzelnes Hindernis, welches die aktuellen sowie die vorausgehenden Messungen verschiedener Sensormodule repräsentiert.

Der nächste Schritt ist die Filterung der Hindernisinformationen. Jedem Hinderniskonzeptrahmen wird ein Filter zugeordnet. Ist die Liste der aktuellen bestätigenden Messungen für einen Hinderniskonzeptrahmen leer, so wird die Korrekturphase des Filters übergangen und das Filter wird nur zur Extrapolation genutzt. Ansonsten wird das Fusionsergebnis als Korrektur des Filters verwendet. Zur Filterung der Meßgrößen Abstand und Geschwindigkeit wird ein einfaches, zweidimensionales Kalman-Filter verwendet. Die laterale Position des Hindernisses wird mit Hilfe von Mittelwertbildung über die letzten n zurückliegenden Werte bestimmt. Wird das Hindernis extrapoliert, so wird die Position mit Hilfe eines Korrekturglieds an die Mitte der Fahrspur herangeführt, auf der sich das Hindernis befindet¹⁴.

¹⁴ Eine Tendenzbestimmung für eine laterale Bewegung des Hindernisses ist aufgrund der Qualität der Meßwerte nur sehr eingeschränkt möglich, so daß hier darauf verzichtet wird.

3.4.5 Schemabasierte Situationsanalyse

Den bisher beschriebenen Methoden zum Aufbau einer Situationsbeschreibung liegt kein explizites Hintergrundwissen, d. h. kein Modell der Verkehrssituationen zugrunde. Sie beschränken sich auf die Betrachtung einzelner Objekte der Umgebung, ohne deren Zusammenwirken zu betrachten. Neben diesen Methoden lassen sich einige Meß- und Extrapolationsfehler durch *Modellwissen* über Straßenszenen und Szenenfolgen detektieren und korrigieren. Dieser Abschnitt beschreibt eine Methode zur echtzeitfähigen Integration von Wissen in das Situationsverwaltungssystem.

Zur Repräsentation von Wissen über Straßenszenen sind bereits einige Systeme mit großer Ausdrucksfähigkeit entwickelt worden, an denen sich Teile der nachfolgend beschriebenen Methode orientieren [Kollnig & Nagel 93, Krüger 91].

Die zusätzliche Problematik der hier zu lösenden Aufgabe besteht in der Echtzeitanforderung. Dadurch wird eine umfassende Modellierung der Domäne und eine große Inferenztiefe ausgeschlossen¹⁵.

Aus diesem Grund wird hier ein explizites *schemabasiertes Fehlermodell* zur Analyse der Verkehrssituation verwendet¹⁶. Das Ziel dieses Fehlermodells ist es, bekannte Wahrnehmungs- und Extrapolationsfehler schematisch zu speichern, in Verkehrsszenen (und -szenenfolgen) wiederzuerkennen und daraufhin die Fehler zu korrigieren. Das Fehlermodell besteht aus einer Menge von Schemata, die typische Fehlerfälle abstrakt beschreiben. Ein Fehlerschema besteht aus zwei Komponenten

- einer abstrakten *Szenenbeschreibung* und
- einer *Korrekturbeschreibung*.

Die dazu ausgewählten Fehlerfälle sind auf solche beschränkt, die eine *Fehlreaktion* des autonomen Fahrzeugs hervorrufen. Jedem Fehlerfall ist entsprechend einer subjektiven Gefährlichkeit der durch den Fehler hervorgerufenen Fehlreaktion und der Eintrittswahrscheinlichkeit der Fehlersituation ein Gefahrenwert zugeordnet.

3.4.5.1 Anforderungen an ein Fehlermodell

Das Fehlermodell dient der Verbesserung der Robustheit der Szenenbeschreibung.

Um dies zu erreichen, sind die folgenden Aspekte der Verkehrsszenen bei der Repräsentation eines Fehlermodells zu beachten:

³ Diese Einschränkung bezieht sich auf die im Versuchsfahrzeug verwendete Hardware.

¹⁶ Der hier beschriebene Ansatz entstand in mit Unterstützung einer Studienarbeit [Auerswald 96]. Einige Definitionstexte sind in beiden Arbeiten daher identisch. Ausführlichere Beschreibungen einzelner Teilgebiete sind ggf. dort nachzuschlagen.

- *Echtzeitfähigkeit*
Die Beschreibungen der Fehlersituationen und Beobachtungsfolgen müssen hinreichend sein, gleichzeitig ist aber eine Eingrenzung der Beschreibung auf notwendige Bestandteile aus Effizienzgründen anzustreben.
- *Repräsentation von zeitlichen Verläufen*
Während eine Situationsbeschreibung lediglich eine Momentaufnahme darstellt, enthält eine Beobachtungsfolge auch zeitliche Abläufe. In vielen Fällen läßt sich das Vorliegen eines Fehlerfalles nur aus einem bestimmten Ablauf heraus erkennen.
- *Repräsentation von Unschärfe*
Die Daten der Bildverarbeitungsmodule sind verrauscht und teilweise mit systematischen Fehlern behaftet. Die Fehlerrepräsentation muß diesen Eigenschaften der Eingangsdaten Rechnung tragen.

3.4.5.2 Beispiele kritischer Verkehrsszenen

Die nachfolgend aufgeführten Beispiele (siehe Abbildung 38) dienen der Motivation der hier beschriebenen Methode und stellen eine kleine Auswahl möglicher Fehlerfälle dar.

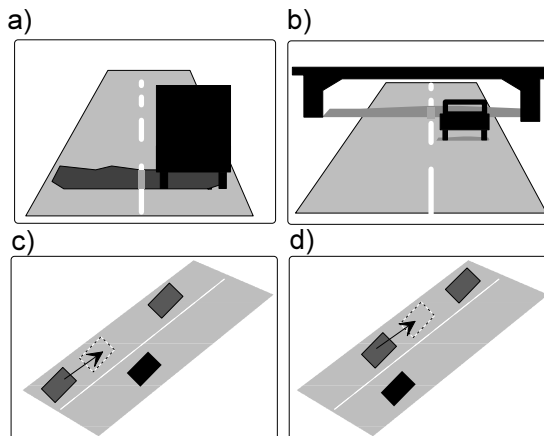


Abbildung 38 : Beispiele für Fehlerfälle

Die Fehlerfälle (a) bis (d) basieren auf Erfahrungen aus Versuchsfahrten mit dem autonomen Fahrzeug VITA II.

- (a) *Fehler:* Der Schatten eines Hindernisses wird selbst als ein solches erkannt. Das autonome Fahrzeug reagiert durch Bremsen oder führt ein mögliches Überholmanöver nicht durch.
Korrektur: Durch Betrachtung zeitlicher Verläufe wird geprüft, ob ein anderes Hindernis das „imaginäre“ Hindernis durchdrungen hat. In diesem Fall wird es verworfen, d. h. aus dem Szenenspeicher gelöscht.

- (b) *Fehler:* Ein Brückenschatten wird als stehendes Objekt erkannt. Vorfahrende Fahrzeuge werden beim Eintauchen in diesen Schatten vom Hinderniserkennungsmodul verloren. Die Reaktion ist ein scharfes Abbremsen oder ein Ausweichmanöver.

Korrektur: analog zu Fall (a)

- (c) *Fehler:* Ein Fahrzeug fährt in einen toten Winkel¹⁷ und wird extrapoliert. Ein weiteres, langsames Fahrzeug fährt voraus. Die Extrapolation berücksichtigt nicht, daß zwischen den Fahrzeugen ein Sicherheitsabstand eingehalten werden sollte und geht von gleichbleibender Geschwindigkeit aus, so daß es zu einer Verschmelzung der beiden Hindernisse kommt. Dadurch nimmt das autonome Fahrzeug irrtümlich an, es befände sich kein Hindernis links von ihm. Das Risiko eines Spurwechsels wird falsch eingeschätzt, so daß es in einer Überholsituation zu einem Unfall kommen kann.

Korrektur: Nach Erkennung der Situation durch Verknüpfung der zeitlichen Verläufe der Positionen und Sichtbarkeit der Hindernisse wird eine wissensbasierte Extrapolation durchgeführt, die von einer Anpassung der Geschwindigkeit des hinteren Fahrzeugs auf die des vorfahrenden Fahrzeugs zur Einhaltung eines Sicherheitsabstands ausgeht.

- (d) *Fehler:* Ein extrapoliertes Fahrzeug befindet sich im toten Winkel. Das extrapolierte und ein vorfahrendes Fahrzeug hatten bisher etwa die Geschwindigkeit des autonomen Fahrzeugs. Das vorfahrende Fahrzeug beschleunigt, das zweite Fahrzeug zieht nach und wird bei Eintreten in den Sichtbereich der Sensoren wieder detektiert, jedoch als neues Hindernis aufgenommen während das extrapolierte Fahrzeug im toten Winkel verbleibt. Dadurch wird ein Spurwechsel des autonomen Fahrzeugs verhindert.

Korrektur: Nach Erkennung des Fehlerfalles wird das extrapolierte Hindernis aus dem Szenenspeicher gelöscht.

3.4.5.3 Erkennung von dynamischen Szenen

Der schemabasierte Ansatz beruht auf einer expliziten Repräsentation von Unschärfe und zeitlichen Verläufen. Ein Schema, welches eine Beschreibung eines verallgemeinerten Fehlerfalls darstellt, wird durch Szenenelemente sowie zeitliche und örtliche Relationen zwischen diesen beschrieben. Im Folgenden wird zunächst auf die unscharfen Beschreibungspredikate eingegangen und anschließend die

¹⁷ Neben dem Fahrzeug existieren Bereiche, die von keinem Sensormodul abgedeckt werden, die hier als tote Winkel bezeichnet sind.

intervallbasierte zeitliche Verknüpfung unscharfer Begriffe beschrieben. Abschließend wird die Szenenauswertung kurz skizziert.

Unscharfe Beschreibungsprädikate

Die verwendeten Beschreibungsprädikate lassen sich in statische und dynamische Prädikate unterteilen. Die Unschärfe eines Prädikats P wird durch einen Konfidenzwert $C(P)$ ausgedrückt. Für statische Prädikate ist dieser Konfidenzwert zeitpunktunabhängig und wird über eine Abwägung der positiven und negativen Hinweise bestimmt. Dagegen wird für dynamische Prädikate ein zeitpunktgebundener Konfidenzwert $C(P,t) \in [0,1]$ benötigt.

Hängt die Gültigkeit eines Beschreibungsprädikats von der Messung einer physikalischen Größe m ab, so läßt sich eine Unschärfe mit Hilfe einer Quasi-Partitionierung des Wertebereichs von M modellieren (vgl. [Kollnig & Nagel 93]). Sei W der Wertebereich der Meßgröße m . Eine unscharfe Menge $A \subseteq W$ wird durch die Zugehörigkeitsfunktion $\mu_A : W \rightarrow [0,1]$ definiert. Der Konfidenzwert eines über m definierten Prädikats P_A zum Zeitpunkt t wird dann durch $C(P_A,t) = \mu_A(m(t))$ bestimmt.

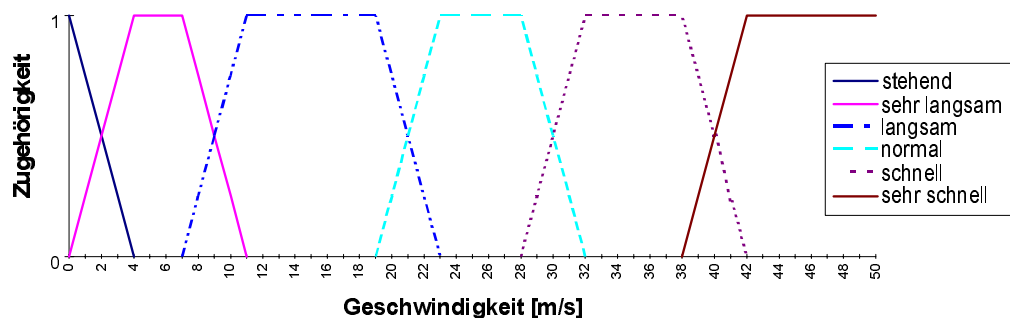


Abbildung 39: Veranschaulichung unscharfer Begriffe für die Geschwindigkeit

Abbildung 39 zeigt die Partitionierung des Wertebereichs des Meßwerts 'Geschwindigkeit' durch die Prädikate 'stehend', 'sehr langsam', 'langsam', 'normal', 'schnell' und 'sehr schnell'.

Zeitliche Verknüpfung unscharfer Begriffe

Zur Repräsentation von zeitlichen Abläufen sind verschiedene Ansätze bekannt, die Nökel in zwei Kategorien unterteilt [Nökel 91]:

1. Der *zustandsorientierte* Ansatz beschreibt Abläufe durch Folgen von Zuständen, die durchlaufen werden (vgl. [Krüger 91]).

2. Der *meßgrößenorientierte* Ansatz betrachtet Episodenfolgen für jede einzelne Meßgröße und verknüpft die Episoden über intervallbasierte Zeitrelationen (vgl. [Allen 84]).

Der hier verfolgte Ansatz läßt sich der zweiten Kategorie zuordnen. Anstelle der Meßgrößen werden hier Ereignisse betrachtet. Diese werden durch mehrstellige Prädikate beschrieben, deren Argumente Referenzen auf Objekte der Szenenbeschreibung darstellen.

An dieser Stelle ergibt sich nun die Problematik der Zusammenführung unscharfer Prädikate mit einer intervallbasierter Zeitlogik. Eine Lösungsmöglichkeit bietet das im Folgenden beschriebene Konzept für eine *unscharfe Zeitlogik* auf Basis der Allen'schen Zeitrelationen (siehe Anhang C). Ein *Elementarereignis* ist dann beschrieben durch ein unscharfes Prädikat (bzw. eine n-stellige Relation) und ein unscharfes Zeitintervall. Zunächst wird ein unscharfes Intervall definiert:

Definition 3.4.4 : Unscharfes Intervall

Ein unscharfes Intervall I ist ein geordnetes Paar (A,E) , wobei A und E scharfe Intervalle $[a1;a2]$ bzw. $[e1;e2]$ sind, die in der Zeitrelation $A \{o,m,<\}E$ gemäß der Allen'schen Notation für Intervallrelationen stehen.

Der nächste Schritt ist die Übertragung der Allen'schen Intervalllogik auf unscharfe Intervalle. Die Allen'schen Intervallrelationen lassen sich durch Konjunktionen der Relationen zwischen ihren Anfangs- und Endpunkten beschreiben. Betrachtet man Anfangs- und Endintervall als *unscharfen Zeitpunkt*, so benötigt man eine auf Intervalle erweiterte Definition der Punktrelationen $\{<,>,\}=\}$, um alle Intervallrelationen darstellen zu können.

Am Beispiel der 'meets' - Relation wird die Übertragung auf unscharfe Intervalle verdeutlicht. Dazu bezeichnen $I_1=(A_1,E_1)$, $I_2=(A_2,E_2)$ unscharfe Intervalle für zwei Ereignisse, zwischen denen eine 'meets'-Relation bestehen soll. In der Punktbeschreibung scharfer Intervalle ist die 'meets'-Relation durch Gleichheit des Endpunkts von I_1 und des Anfangspunkts von I_2 gegeben. Für unscharfe Intervalle heißt das, daß zunächst $E_1 \cap A_2$ nicht leer sein darf. Für einen möglichen gemeinsamen Endpunkt der unscharfen Intervalle ist nur das Intervall $S = E_1 \cap A_2$ plausibel. Steigt der Konfidenzwert des zweiten Ereignisses in S im gleichen Maße an, wie die Konfidenz des ersten abnimmt, dann soll die Konfidenz für das Vorliegen der 'meets'-Relation am größten sein. Die Konfidenz ist also umso höher, je näher die Summe der Konfidenzen bei Eins ist. Bezeichnen $\mu_1(t)$, $\mu_2(t)$ die Zugehörigkeitsfunktionen der Ereignisse, die den Intervallen I_1 , I_2 zugeordnet sind, dann ergibt sich folgende Formel:

$$C(I_1 \text{ meets } I_2) = C(E_1 = A_2) = 1 - \frac{1}{\|E_1 - A_2\|} \cdot \int_{t \in E_1 \cap A_2} |1 - \mu_1(t) - \mu_2(t)| dt$$

Dabei bezeichnet $\|E_1 \cap A_2\|$ die Länge des Intervalls $E_1 \cap A_2$. Überlappen sich E_1 und A_2 nicht, so ist der Konfidenzwert Null.

In ähnlicher Weise wie die oben hergeleitete Konfidenz für die Relation ‘=’ läßt sich diese auch für ‘>’ und ‘<’ definieren. Dabei wird zwischen Intervallende und -anfang differenziert, so daß man Konfidenzen für die Relationen $\langle_{AA}, \rangle_{AA}, =_{AA}, \langle_{AE}, \rangle_{AE}, =_{AE}, \langle_{EA}, \rangle_{EA}, =_{EA}, \langle_{EE}, \rangle_{EE}$ und $=_{EE}$ erhält. Die Definitionen dafür sind in Anhang C aufgeführt. Mit Hilfe dieser Konfidenzwerte lassen sich nun Konfidenzwerte für die Allen’schen Intervallrelationen auf unscharfen Intervallen I_1, I_2 definieren:

$C(I_1 < I_2)$	=	$C(E_1 \langle_{EA} A_2)$
$C(I_1 \text{ m } I_2)$	=	$C(E_1 =_{EA} A_2)$
$C(I_1 \text{ o } I_2)$	=	$\min(C(A_1 \langle_{AA} A_2), C(E_1 \rangle_{EA} A_2), C(E_1 \langle_{EE} E_2))$
$C(I_1 \text{ s } I_2)$	=	$\min(C(A_1 =_{AA} A_2), C(E_1 \langle_{EE} E_2))$
$C(I_1 \text{ d } I_2)$	=	$\min(C(A_1 \rangle_{AA} A_2), C(E_1 \langle_{EE} E_2))$
$C(I_1 \text{ f } I_2)$	=	$\min(C(A_1 \rangle_{AA} A_2), C(E_1 =_{EE} E_2))$
$C(I_1 = I_2)$	=	$\min(C(A_1 =_{AA} A_2), C(E_1 =_{EE} E_2))$
$C(I_1 \text{ fi } I_2)$	=	$\min(C(A_1 \langle_{AA} A_2), C(E_1 =_{EE} E_2))$
$C(I_1 \text{ di } I_2)$	=	$\min(C(A_1 \langle_{AA} A_2), C(E_1 \rangle_{EE} E_2))$
$C(I_1 \text{ si } I_2)$	=	$\min(C(A_1 =_{AA} A_2), C(E_1 \rangle_{EE} E_2))$
$C(I_1 \text{ oi } I_2)$	=	$\min(C(A_1 \rangle_{AA} A_2), C(A_1 \langle_{AE} E_2), C(E_1 \rangle_{EE} E_2))$
$C(I_1 \text{ mi } I_2)$	=	$C(A_1 =_{AE} E_2)$
$C(I_1 > I_2)$	=	$C(A_1 \rangle_{EA} E_2)$

Auswertung in Echtzeit

Bevor auf den Erkennungsalgorithmus eingegangen wird, soll an dieser Stelle eine formale Definition einer Szenenbeschreibung gegeben werden:

Definition 3.4.5 : Szenenbeschreibung

Eine Szenenbeschreibung ist gegeben durch ein 4-Tupel (O, R, E, T) .

- O bezeichnet eine Menge von Objektreferenzen auf Objekte des Szenenspeichers. Eine Objektreferenz beinhaltet neben einem Zeiger auf einen Objektrepräsentanten eine Typrestriktion und eine Projektionsrestriktion.
- $R \subset \{f: \text{Pot}(O) \rightarrow [0, 1]\}$ ist eine Menge von statischen Prädikaten.
- $E \subset \{f: \text{Pot}(O) \times \mathfrak{R} \rightarrow [0, 1]\}$ ist eine Menge von dynamischen Prädikaten.
- $T \subseteq E \times E \times (\text{Pot}(Al) \setminus \emptyset)$ ist eine Menge von Intervall-Constraints zwischen den dynamischen Prädikaten. Dabei ist $Al = \{\langle, m, s, s, d, =, fi, di, si, oi, mi, \rangle\}$ die Menge der Allen’schen Intervallrelationen.

Die Echtzeitfähigkeit des Systems erfordert eine rückwärtsverkettete Auswertung der Szene. In Anlehnung an Krüger wird die Abbildung von Objektreferenzen der Szenenbeschreibung eines Schemas auf Elemente des Szenenspeichers als

Projektion bezeichnet. Zur Einschränkung der Projektionskomplexität werden zu den Objektreferenzen der Szenenbeschreibung der Schemata Projektionsrestriktionen eingeführt. Eine Projektion wird nur durchgeführt, wenn die Konfidenz der Projektionsrestriktion, d. h. eines n -stelligen unscharfen Prädikats, einen vorgegebenen Schwellwert überschreitet.

Das Matching von Elementarereignissen geschieht mit Hilfe eines Automaten für das Verfolgen des Ausführungszustands dieses Ereignisses. Dazu werden zwei Konfidenzschwellwerte λ_H und λ_L eingeführt, mit $0 < \lambda_L < \lambda_H < 1$. Sei $E = (P, I)$ ein Elementarereignis bestehend aus Prädikat P und unscharfem Intervall $I = (A, E)$. Ist $C(P) < \lambda_L$, so wird das Ereignis verworfen. Ist $C(P) > \lambda_H$, so wird das Ereignis als stattfindend erkannt. In den Zeitintervallen A und E gilt $\lambda_L < C(P) < \lambda_H$.

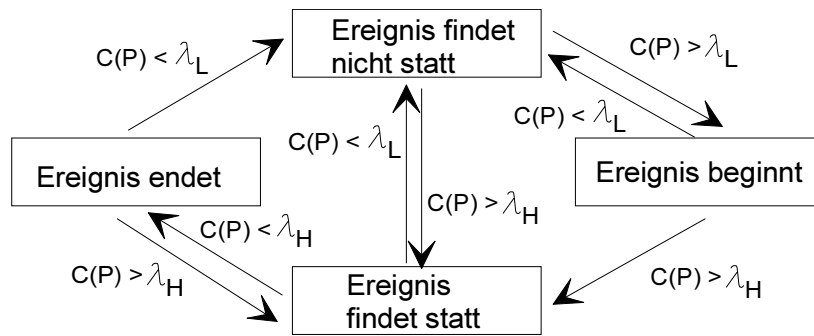


Abbildung 40 : Automat zur Erkennung von Ereignissen

Der in Abbildung 40 dargestellte Automat dient der inkrementellen Erkennung eines Ereignisses. Wie man in der Darstellung erkennt, sind die Übergänge zu Anfangs- und Endphase temporär. Befindet sich ein Ereignis im Anfangszustand, und seine Konfidenz unterschreitet λ_L wieder, dann wird es verworfen. Wird jedoch der obere Schwellwert überschritten, so wird das Ereignis als stattfindend betrachtet und nach seiner Beendigung weiter verwaltet (sofern die Information benötigt wird). Zur inkrementellen Aktualisierung der Zeitrelationen in Szenenbeschreibungen werden die Ausführungszustände und ihre aktuellen Änderungen als Eingabe für einen Matching-Algorithmus verwendet, der zu jedem instanziierten Schema einen momentanen Zustand verwaltet. Der Zustand enthält für jedes Intervall-Constraint die beschreibenden Endpunkte, die Endpunkt-Relationen und deren Konfidenzwerte. Ein Bewertungsalgorithmus bestimmt die Konfidenzwerte der Constraint-Klausen, die sich aus dem Maximum der in einer Klausel erlaubten Elementarrelationen ergeben.

Beispiel eines Fehlerschemas

Die nebenstehende Spezifikation der Szenenbeschreibung eines Fehlerschemas beschreibt den in Abschnitt 3.4.5.2 dargestellten Fehlerfall d).

Zeile 3 gibt die Zahl der anschließend spezifizierten Objektreferenzen an. Die Zeilen 4 bis 6 geben den jeweils zugehörigen Typ und die Projektionsrestriktion an. Die

1	Scene_d
	References: 3
	Obstacle always
5	Obstacle always
	Obstacle always
	StaticRelations: 0
	Episodes: 5
	e1: agens_rel is_unvisible 1
10	e2: agens_rel is_visible 2
	e3: object_rel same_lane_close_behind 1 2
	e4: agens_rel exists 3
	e5: object_rel same_lane_close_behind 3 2
	Constraints: 4
15	e3 {o} e1
	e2 {o,fi,di} e1
	e4 {s,=,si} e5
	e1 {o,fi,di} e4

Zeilen 9 bis 13 beschreiben Elementarereignisse durch Angabe eines symbolischen Namens, eines Relationentyps, eines Relationennamens und Referenzen auf die Objekte. Die einzuhaltenden zeitlichen Intervallrelationen sind in Zeile 15 bis 18 angegeben.

3.4.5.4 Korrektur erkannter Fehler

Ein Fehlerschema $F = (S_F, K_F)$ setzt sich aus einer Szenenbeschreibung S_F und einer Korrekturbeschreibung K_F zusammen. Nach erfolgreicher Erkennung der Szenenbeschreibung S_F muß nun der Fehler korrigiert werden. Diese Korrektur kann nur durch Manipulation der Objekte des Szenenspeichers geschehen. Die Korrekturbeschreibung stellt also eine Aktionsbeschreibung dar. Dabei sind die folgenden Manipulationen des Szenenspeichers möglich:

- Hinzufügen eines Konzeptrahmens
- Löschen eines Konzeptrahmens
- Verändern von Konzeptrahmen

Zwei verschiedene Aktionstypen werden unterschieden: *einmalige* Aktionen und *fortdauernde* Aktionen. Während manche Fehler wie der eingangs beschriebene Fehlerfall d) durch einmalige Aktionen behoben werden können, wird für die Behebung anderer Fehler eine Folge von Korrekturen benötigt.

Die einer solchen Korrektur zugrundeliegende Idee ist die *wissensbasierte Extrapolation*. Wird ein Fahrzeug nicht mehr erkannt, so wird seine Position unter Berücksichtigung der Geschwindigkeit bestimmt. Der Extrapolation liegt die

Annahme einer konstanten Geschwindigkeit¹⁸ zugrunde. Der in Abschnitt 3.4.5.2 beschriebene Fehlerfall c) ist ein gutes Beispiel für die Anwendbarkeit der wissensbasierten Extrapolation. Wird die Fehlersituation erkannt, so kann jetzt der Extrapolation des Hindernisses die Annahme zugrundegelegt werden, daß dieses abbremst und einen Abstand zum vorausfahrenden Fahrzeug einhält. Damit erhält man folgende Definition für eine Aktionsbeschreibung:

Definition 3.4.6 : Aktionsbeschreibung

Eine Aktionsbeschreibung ist eine Anweisung zur Manipulation des Szenenspeichers. Sie besteht aus drei möglichen Arten von Einzelaktionen:

- **Löschen** : Löscht ein referenziertes Objekt aus dem Szenenspeicher.
- **Setzen** : Erzeugt ein neues Objekt im Szenenspeicher. Der Einzelaktion ist eine Funktion $f : Pot(O) \rightarrow O$ zugeordnet, die zur Initialisierung des neuen Objekts anhand der vorliegenden Szene dient.
- **Ändern** : Verändert ein referenziertes Objekt. Der Einzelaktion ist eine Funktion $f : O \times Pot(O) \rightarrow O$ zugeordnet, die der Berechnung der Veränderungen am referenzierten Objekt dient.

Zur Veranschaulichung der Funktionsweise der Korrektur läßt sich das oben aufgeführte Beispiel für Fehlerfall d) um die nebenstehende Aktionsbeschreibung erweitern. Fehlerfall c) erfordert eine Veränderung des extrapolierten

...	Actions: 1
20	remove 1

Hindernisses. Dies wird durch eine Elementaraktion des Typs ‘Ändern’ erreicht, der eine Funktion $f_{\text{abstandhalten}} : O \times O \rightarrow O$ zugeordnet ist. Diese Funktion beinhaltet eine Annahme über den einzuhaltenden Abstand und berechnet aus dem Abstand und den Geschwindigkeiten der referenzierten Objekte eine Verzögerung des hinteren Fahrzeugs. Die daraus abgeleitete Geschwindigkeit und Position für den nächsten Zyklus werden dem referenzierten Objekt zugewiesen.

3.5 Simulation

Die Verifikation der Algorithmen zur autonomen Fahrzeugführung stellt aufgrund des damit verbundenen Kostenaufwands, des Zeitaufwands, aber vor allem der Gefahren im realen Straßenverkehr ein nicht unwesentliches Problem dar. Zur Entwicklung eines solchen Systems wird daher eine Simulationsumgebung verwendet. In dieser

¹⁸ Die Einbeziehung der Beschleunigung in die Extrapolation ist denkbar. Dann jedoch wird wiederum diese als konstant angenommen.

Simulationsumgebung werden die für das gestellte Problem relevanten Vorgänge auf dem Rechner nachgebildet.

Da die im Rahmen dieser Arbeit entwickelte Simulationsumgebung eine wesentliche Grundlage für die Erstellung und Erprobung des Kontrollsystems von VITA II darstellt, soll ihr Aufbau, ihre Eigenschaften und die damit verbundenen Möglichkeiten in diesem Abschnitt kurz skizziert werden.

Die hier erstellte und verwendete Simulation bildet das Versuchsfahrzeug VITA II mit seinen optischen Sensoren, seinem Kontrollsystem und der Fahrzeugdynamik nach. Durch die Modellierung der Umgebung, in der sich das Fahrzeug bewegt, bestehend aus einem Straßennetz, Verkehrszeichen und anderen Verkehrsteilnehmern, können in der Simulation beliebige Verkehrssituationen generiert werden. Es ist somit möglich, die Reaktionen des autonomen Fahrzeugs vor dem Einsatz im realen Straßenverkehr in der Simulation zu überprüfen.

Die Aufnahme von realen Verkehrsszenen per Datenrecorder und das anschließende Einspielen der aufgenommenen Daten zur Überprüfung der Funktionalität des Kontrollsystems außerhalb des Straßenverkehrs ist in dieser Simulation ebenfalls integriert. Eine Verifikation der autonomen Fahrzeugführung selbst ist jedoch durch diese Methode nur bedingt durchführbar, da man einen geschlossenen Regelkreis benötigt, um ein rückgekoppeltes System zu testen.

Im Folgenden wird zunächst der Aufbau des Simulationssystems und anschließend die Realisierung des VITA II spezifischen Teils der Simulation beschrieben.

3.5.1 Der Aufbau des Simulationssystems

Das Simulationssystem besteht auf oberster Organisationsebene aus einem Manager für die *graphische Benutzeroberfläche* und einer Repräsentation einer *virtuellen Welt*. Über den Manager werden die Benutzereingaben sowie die Visualisierung von Informationen der simulierten Welt auf einer fensterbasierten graphischen Oberfläche organisiert. Die virtuelle Welt beinhaltet statische Informationen, die a priori festgelegt werden, sowie dynamische Informationen über Agenten, die sich nach verschiedenen Strategien in der Welt bewegen.

Das in Abbildung 41 dargestellte Datenflußdiagramm veranschaulicht die Struktur der Simulation. Den eigentlichen Kern bildet der Prozeß „Agenten bewegen“. Jeder in der virtuellen Welt enthaltene Agent stellt ein Simulationssystem eines autonomen mobilen Systems dar. Durch die Integration mehrerer Simulationen von verschiedenen Agenten in einer gemeinsamen Welt wird es ermöglicht, die Kontrollstrategie eines autonomen mobilen Systems dahingehend zu untersuchen, wie sie auf Agenten reagiert, die mit der gleichen Strategie ihre Verhaltensentscheidungen treffen.

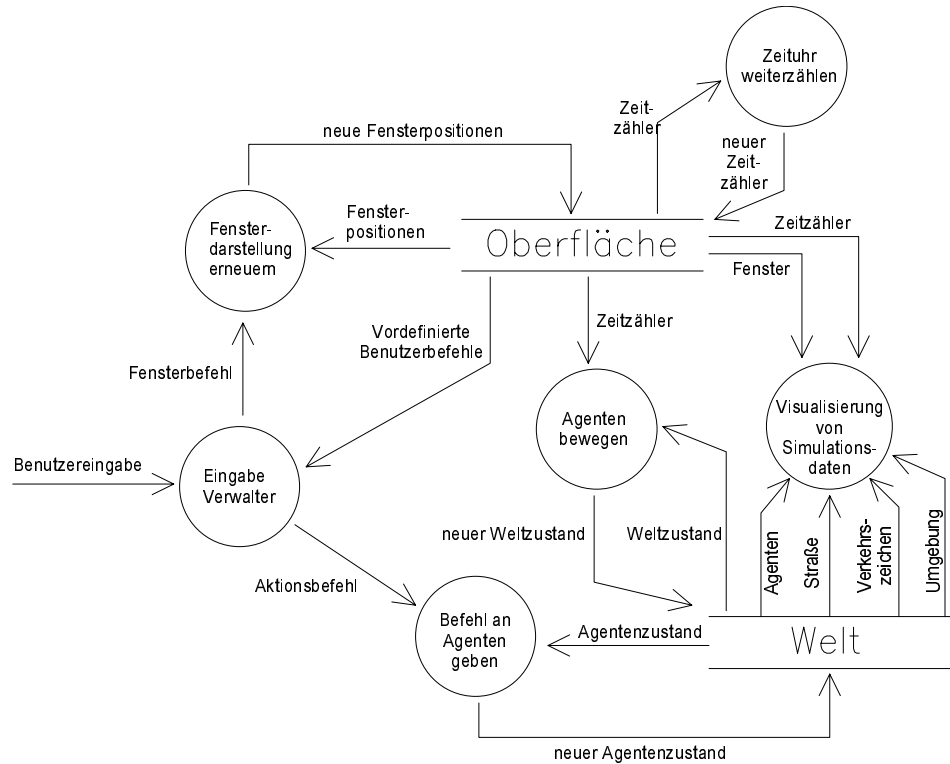


Abbildung 41 : Datenflußdiagramm der Simulation

3.5.1.1 Die Ein-/Ausgabeverwaltung

Die Ein-/Ausgabeverwaltung gibt den äußeren Rahmen für die Kommunikation zwischen dem Benutzer und dem Simulationssystem vor. Gleichzeitig werden von hier aus alle Simulationsprozesse verwaltet.

Die Ein-/Ausgabeverwaltung ist als ein *E/A-Objekt* realisiert, welches hauptsächlich die Nachrichten „Initialisierung“, „Eingabebehandlung“ und „Aktionsschritt“ kennt. Dem E/A-Objekt ist eine Menge von Simulationsobjekten und eine Menge von Fensterobjekten zugeordnet. Die Fensterobjekte sind ihrerseits Simulationsobjekten zugeordnet, die diese als Ausgabebereich nutzen. Einem Simulationsobjekt ist ferner ein Datenbereich, eine Eingabemethode sowie eine Ausführungs- und Darstellungsmethode zugeordnet.

Die folgenden Methoden sind den an das E/A-Objekt gerichteten Nachrichten zugeordnet:

Initialisierung Die zu verwaltenden Simulationsobjekte und Fensterobjekte werden festgelegt, und deren Anfangskonfiguration wird bestimmt. Desweiteren werden die zulässigen Benutzereingaben und deren Behandlung festgelegt.

- Eingabe-
behandlung
- Durch Eingaben kann der Benutzer die Konfiguration der Fensteroberfläche manipulieren, Fenster vergrößern, ikonisieren, verschieben, schließen und auswählen. Ein Fenstermanager nimmt die Benutzereingabe entgegen und führt die gewünschten Aktionen durch.
- Eingaben können jedoch auch an ein Simulationsobjekt gerichtet sein. Beispielsweise kann durch eine Benutzereingabe die Wunschgeschwindigkeit eines simulierten autonomen Fahrzeugs erhöht werden.
- Erfolgt eine Eingabe, die nicht an den Fenstermanager gerichtet ist, so wird diese von der Eingabebehandlung des E/A-Objekts an die Eingabebehandlung der Simulationsobjekte weitergeleitet.
- Aktionsschritt
- Jedem der dem E/A-Objekt zugeordneten Simulationsobjekte wird die Nachricht „Aktionsschritt“ gesendet. Dient ein Simulationsobjekt beispielsweise der Visualisierung eines Datenbereichs, so wird bei einem Aktionsschritt dieser Datenbereich ausgelesen. Mittels der dem Simulationsobjekt zugeordneten Darstellungsmethode werden die Daten dann im zugeordneten Fenster dargestellt.

Weiterhin verwaltet das E/A-Objekt einen Zeitzähler, mit dessen Hilfe die Zykluszeiten der simulierten Prozesse nachempfunden werden.

Mit Hilfe dieser allgemein gehaltenen Ein-/Ausgabeverwaltung läßt sich die Simulationsumgebung flexibel konfigurieren. Der eigentliche Simulationskern besteht aus der virtuellen Welt. Um diese an den durch die Ein-/Ausgabeverwaltung gegebenen Rahmen anzupassen, wird ein Simulationsobjekt mit Zugriff auf den Datenbereich der virtuellen Welt definiert. Diesem ist die Aktionsmethode „Agenten bewegen“ und eine geeignete Methode zur Behandlung von Benutzereingaben zur Interaktion mit Agenten zugeordnet.

Beliebige Ausschnitte der virtuellen Welt lassen sich somit mit Hilfe von Simulationsobjekten während der Simulationsläufe graphisch und textlich darstellen und erleichtern damit die Aufgabe der Verifikation der Algorithmen zur autonomen Fahrzeugführung.

3.5.1.2 Die virtuelle Welt

Um den realen Straßenverkehr zu simulieren, wird eine virtuelle Welt beschrieben, in der die wesentlichen, zur Algorithmenverifikation benötigten Komponenten der realen Verkehrsszenarien enthalten sind. Es wird hier zwischen einem statischen und einem dynamischen Teil der virtuellen Welt unterschieden.

- Dem *statischen* Teil der virtuellen Welt gehören die *Straßenverläufe* sowie die Positionen von *Verkehrszeichen* an.
- Den *dynamischen* Teil der virtuellen Welt bilden die modellierten Verkehrsteilnehmer, die sogenannten *Agenten*, die sich während eines Simulationslaufs in der virtuellen Welt bewegen und aufeinander reagieren.

Da das Einsatzgebiet des autonomen Fahrzeugs VITA II auf die Autobahn begrenzt ist, werden in der Simulation auch nur Autobahnen berücksichtigt. Eine Erweiterung auf Landstraßen und Innenstadtszenarien ist jedoch möglich.

Statische Welt

Zur Modellierung des Straßennetzes sind die Straßen segmentweise beschrieben. Diese Segmente werden durch vordefinierte Verbindungstypen miteinander verknüpft. Damit sind Verzweigungen und Kreuzungen verschiedenen Typs modellierbar.

Ein *virtuelles Straßensegment* wird beschrieben durch eine Länge, eine vertikale und horizontale Krümmung und eine Menge von Fahrspuren, angegeben durch ihre Breite und Begrenzungslinien zwischen den Fahrspuren.

Eine *Straßensegmentverknüpfung* ist durch die eingehenden und ausgehenden Straßensegmente und durch die zwischen diesen festgelegten Winkeln beschrieben.

An Verknüpfungsstellen werden die Werte der Straßenkrümmung während der Simulation geglättet, so daß keine Krümmungssprünge entstehen.

Zum statischen Teil der virtuellen Welt gehören neben der Straße auch die Verkehrszeichen. Diese werden durch eine in der Straßenverkehrsordnung definierte Kodierung zuzüglich einer Ergänzungsinformation beschrieben (vgl. [HAV 87]). Ihre Position wird anhand einer Referenz auf das Straßensegment angegeben. Dabei ist zu bemerken, daß die Straßensegmente gerichtet sind und somit eine Angabe der Position durch eine Entfernung vom Segmentanfang gegeben werden kann.

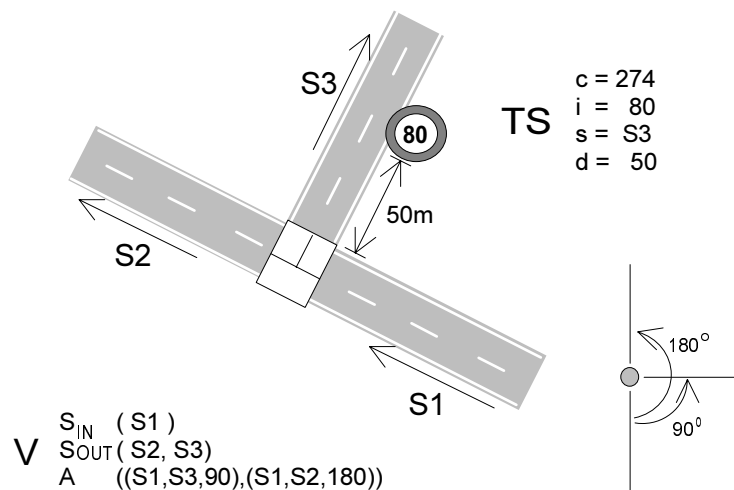


Abbildung 42: Straßenmodellierung in der virtuellen Welt

Abbildung 42 veranschaulicht die Modellierung eines Straßennetzes anhand eines Ausschnitts.

Ein weiterer Bestandteil ist das herrschende *Wetter*. Da dieses für die Bildverarbeitung eine große Rolle spielt, wird es in die Simulation mit einbezogen. Als fixe Parameter für einen Simulationslauf werden die Helligkeit und die Bewölkung qualitativ angegeben (z.B. dunkel, bedeckt, hell, sehr hell etc.).

Dynamische Welt

Den dynamischen Teil der virtuellen Welt machen die Agenten aus. Ein Agent ist durch seine äußere Form, seinen dynamischen Zustand und seine Position in der virtuellen Welt beschrieben. Zu Beginn eines Simulationslaufs wird die Position und der Startzustand der Agenten in der virtuellen Welt festgelegt. Während des Simulationslaufs ändern die Agenten ihre Position und ihren Zustand durch Reaktion auf die sie umgebende Situation anhand einer vorgegebenen Kontrollstrategie. Die Aktionen der Agenten sind von einem simulierten Sensorsystem, einem Kontrollsystem und einem Aktorsystem abhängig.

3.5.1.3 Die Simulation von Agenten

Die Agentensimulation stellt den Kern der Simulationsumgebung dar. Jeder Agent simuliert ein Fahrzeug, welches sich in der virtuellen Welt bewegt. Er besteht aus einer Sensorsimulation, einer Aktorsimulation und einem Kontrollsystem. Durch die Sensorsimulation wird festgelegt, wie der Agent seine Umgebung wahrnimmt. Das

Kontrollsystem bestimmt sein Verhalten und die Aktorsimulation legt die Dynamik des Agenten fest.

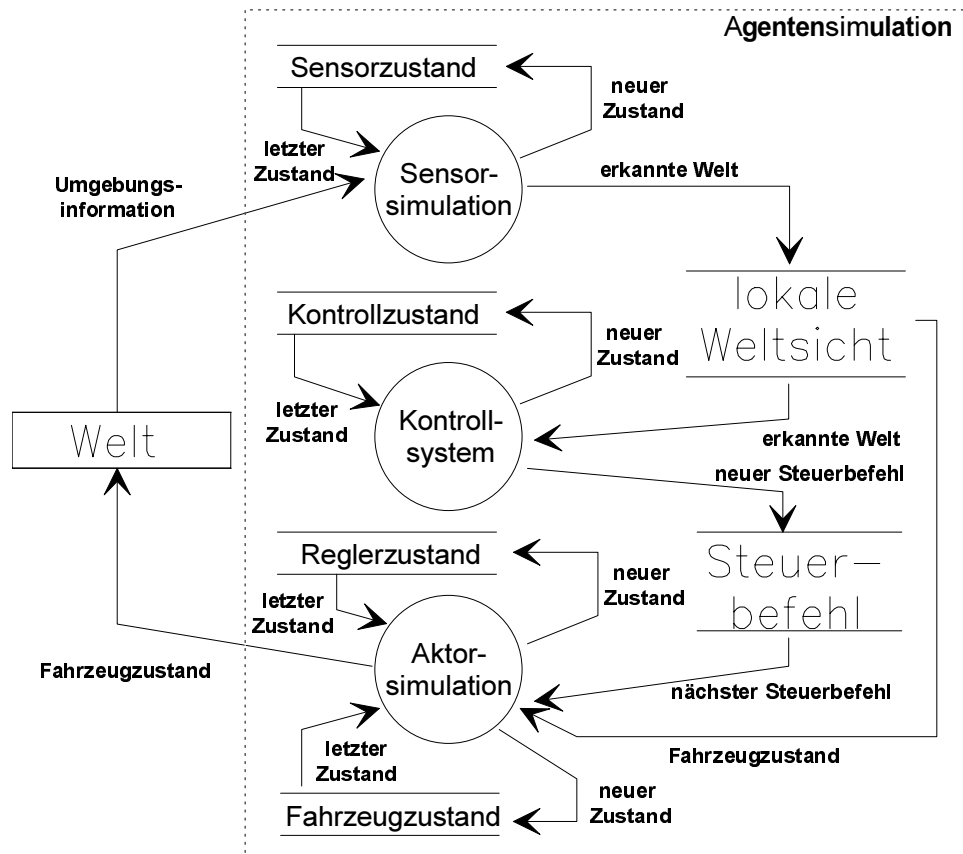


Abbildung 43 : Datenflußdiagramm für die Agentensimulation

Sensorsimulation

Die Sensorsimulation ist eine stochastische Simulation der Sensormodule und Erkennungsmethoden. Sie legt fest, welchen Teil der virtuellen Welt der Agent wahrnehmen kann. Ist ein Agent beispielsweise nicht mit einem logischen Sensor zur Hinderniserkennung ausgerüstet, so wird er nicht auf andere Agenten reagieren. Weiterhin bildet die Sensorsimulation die typischen Fehler des jeweiligen Sensors nach. Jedem Agenten ist eine Menge verschiedener, simulierter Sensoren zugeordnet, über die er seine Umgebungsinformation bezieht.

Ein simulierter logischer Sensor stellt eine Abbildung mit den folgenden Eigenschaften dar:

- Eine Abbildung der Welt auf einen räumlichen Ausschnitt.
- Eine Abbildung der Weltinformationen auf Teilaspekte.
- Eine Abbildung der präzisen Information auf eine sensorspezifisch veränderte Information. Darunter fallen simulierte Meßungenauigkeiten des Sensors.

- d) Erzeugung positiver und negativer Detektionsfehler, d.h. Melden von nicht vorhandenen Objekten und Auslassen existierender Objekte.
- e) Verzögerung der Ausgabe der simulierten Meßwerte um sensorspezifische Latenz- und Zykluszeiten.

Aktorsimulation

Im Gegensatz zu der stochastischen Sensorsimulation liegt der Aktorsimulation ein physikalisches Modell zugrunde. Man spricht in diesem Fall von einer deterministischen Simulation. Hier wird auf vorangehende Arbeiten im Bereich der Regelungstechnik zurückgegriffen. In die Simulation fließen ein Längsdynamikmodell und ein Querdynamikmodell des simulierten Fahrzeugs ein.

Die Modellierungstiefe ist von der für den Verwendungszweck benötigten Genauigkeit und der zur Verfügung stehenden Rechenleistung abhängig.

Um das Längsdynamikmodell zu vereinfachen, kann es auf eine Punktmasse reduziert werden. Alle auf das Fahrzeug wirkenden Kräfte greifen dann am Schwerpunkt des Fahrzeugs an. Bei der Querdynamiksimulation lassen sich ein Einspur- und ein Zweispurmodell unterscheiden. Nähere Informationen zu Modellen der Fahrzeuglängs- und -querdynamik sind [Bösch 91] zu entnehmen.

Kontrollsysteme für Agenten

Das Kontrollsystem selbst wird nicht simuliert. Es werden verschiedene Kontrollsysteme für Agenten verwendet, die unterschiedliche Verhaltensweisen von Agenten erzeugen. Dabei werden die drei folgenden Typen von Kontrollsystemen unterschieden:

<i>Agendagesteuerte Systeme</i>	Das Kontrollsystem beschränkt sich auf striktes Befolgen von Befehlen, die aus einer Agenda stammen. Es beachtet dabei keine anderen Agenten in seiner Umgebung.
<i>Agendagesteuerte primitiv reaktive Systeme</i>	Das Kontrollsystem erhält Fahrbefehle aus einer Agenda, verfügt darüber hinaus über einfache Funktionen zum Abstandhalten zu anderen Agenten.
<i>Autonome Systeme</i>	An diese Stelle treten das im Rahmen dieser Arbeit vorgestellte Kontrollsystem und Varianten davon.

Das Agendasystem

Die Agenda stellt eine Menge von Ausführungsanweisungen an die Agenten dar. Jede Ausführungsanweisung ist mit einem Zeitstempel versehen. Ausgehend von einer Startkonfiguration werden den Agenten die jeweiligen Ausführungsanweisungen geschickt, wenn die Simulationszeit dem Zeitstempel entspricht. Es ergibt sich somit ein *vorprogrammiertes* Verhalten der Agenten.

Die Handlungsanweisungen der Agenda sind sehr grob gehalten. Sie entsprechen einer in Abschnitt 3.1 definierten Benutzereingabe *Be*. Die dadurch ausgelösten Reaktionen der Agenten sind abhängig vom Kontrollsystem des Agenten.

3.5.2 Die Simulation von VITA II

Das Fahrzeug VITA II wird in der Simulation als ein Agent in der virtuellen Welt modelliert. Dazu wird eine Nachbildung der dynamischen Datenbasis als Schnittstelle zwischen dem Sensorsystem, dem Kontrollsystem und dem Aktorsystem des Agenten benutzt. Die Schnittstellen des Kontrollsystems sind somit identisch zu denen im realen Fahrzeug, so daß eine Portierung der Software problemlos ist. Der Aufbau der Sensorsimulation, der Aktorsimulation und der Integration des Kontrollsystems wird nachfolgend beschrieben.

3.5.2.1 Simulation des Sensorsystems von VITA II

Die Simulation des Sensorsystems umfaßt stochastische Simulationen für das Straßenerkennungsmodul RT, das Verkehrszeichenerkennungssystem TSR und die Hinderniserkennungsmodule ODT, CT, VB und LOD.

Hauptelemente der Simulation sind Zufallsereignisse und ein simuliertes Meßrauschen, mit dem die exakten Daten der virtuellen Welt überlagert werden.

Die Simulation des Straßenerkennungsmoduls und des Verkehrszeichenerkennungsmoduls ist sehr grob gehalten. Der Simulation der einzelnen Hinderniserkennungsmodule geht eine Analyse voraus. Anhand des Wissens über interne Abläufe in den Sensormodulen und über physikalische Grenzen der Sensoren wird ein Sensormodell erstellt. Durch eine Vielzahl von Messungen auf Testfahrten mit dem realen Fahrzeug werden statistische Sensorkenngrößen ermittelt, die auch typische Fehler und deren Auftretenshäufigkeiten enthalten.

Zufallsereignisse

Zur Generierung von Zufallsereignissen wird ein Zufallsgenerator verwendet. Dieser bestimmt zusammen mit einem vorgegebenden, empirisch ermittelten Wahrschein-

lichkeitswert, wann das Ereignis eintritt. Die Eintrittswahrscheinlichkeit des Zufallsereignisses selbst entspricht dann dem vorgegebenen Wahrscheinlichkeitswert.

Meßrauschen

Ein einfaches Meßrauschen wird durch Vorgabe einer Meßwertabweichung erzeugt. Auch hier wird ein Zufallsgenerator verwendet. Dieser erzeugt eine Zufallszahl $z \in [0,1]$. Bezeichnet m einen Eingabewert und Δm eine prozentuale Meßwertabweichung, dann läßt sich ein einfaches Rauschen durch $m^r = m \cdot (1 + (1 - 2z) \cdot \Delta m / 100)$ erzeugen.

Simulation von RT

Die Simulation von RT beschränkt sich im wesentlichen auf ein Belegen der Straßenbreite und $Position_{AMV}$ mit einem einfachen Rauschen und der Störung der Informationen über die Existenz angrenzender Fahrspuren. Für linke und rechte Spur wird ein Zufallsereignis 'Fehlmessung' eingeführt. Tritt dieses Ereignis ein, so wird eine existierende Fahrspur nicht gemeldet und umgekehrt. Für eine grobe Überprüfung des Fahrzeugverhaltens ist diese einfache Simulation ausreichend.

Simulation von TSR

Die Simulation des Verkehrszeichenerkennungssystems beschränkt sich auf die Prüfung, welche Verkehrszeichen der virtuellen Welt sich im Sichtbereich befinden. Die zugehörige Information wird in der dynamischen Datenbank abgelegt. Dieses Vorgehen ist mit der Art und Verwendung der Information von TSR begründet. Die gelieferten Werte sind diskret, d. h. ein Verrauschen der Daten könnte nur zu simulierten Verwechslungen von Verkehrszeichen führen. Da die Information weder über Zusammenhänge in der Szene noch über eine zeitliche Betrachtung zu verifizieren ist¹⁹, muß stets von der Korrektheit ausgegangen werden.

Simulation der Hinderniserkennungsmodule

Zu jedem simulierten Sensormodul sind Angaben über dessen Sichtbereiche gespeichert. Hauptbestandteil der Sensorsimulation ist eine Erfahrungstabelle, in der statistisch ermittelte Kenngrößen enthalten sind. Für jedes der Hindernis-

¹⁹ Die Zykluszeit von TSR erlaubt im Geschwindigkeitsbereich von ca. 100 km/h lediglich eine einmalige Erkennung eines Verkehrszeichens.

erkenntnismodule ODT, CT und VB ist eine Erfahrungstabelle zusammengestellt, die u. a. distanzabhängige Kenngrößen enthält, die größtenteils in Form von Wahrscheinlichkeiten angegeben werden.

Allen simulierten Sensormodulen liegt eine einfache Struktur zugrunde, die in Abbildung 44 als Datenflußdiagramm angegeben ist. Es wird eine Liste von Referenzen auf Agenten der virtuellen Welt verwaltet, die aktuell erkannt werden. Zu jeder der Referenzen ist ein simulierter Meßvektor gespeichert, der die Werte enthält, die vom simulierten Sensormodul zu einem Agenten erzeugt werden. Die Aktionen Löschen und Hinzufügen werden durch Zufallsvariablen beeinflusst, die anhand der Wahrscheinlichkeiten aus einer sensorspezifischen Erfahrungstabelle gebildet werden. Die Einfügeaktion berücksichtigt zusätzlich Verdeckungen von Agenten in der virtuellen Welt sowie die Sichtbereiche des Sensormoduls.

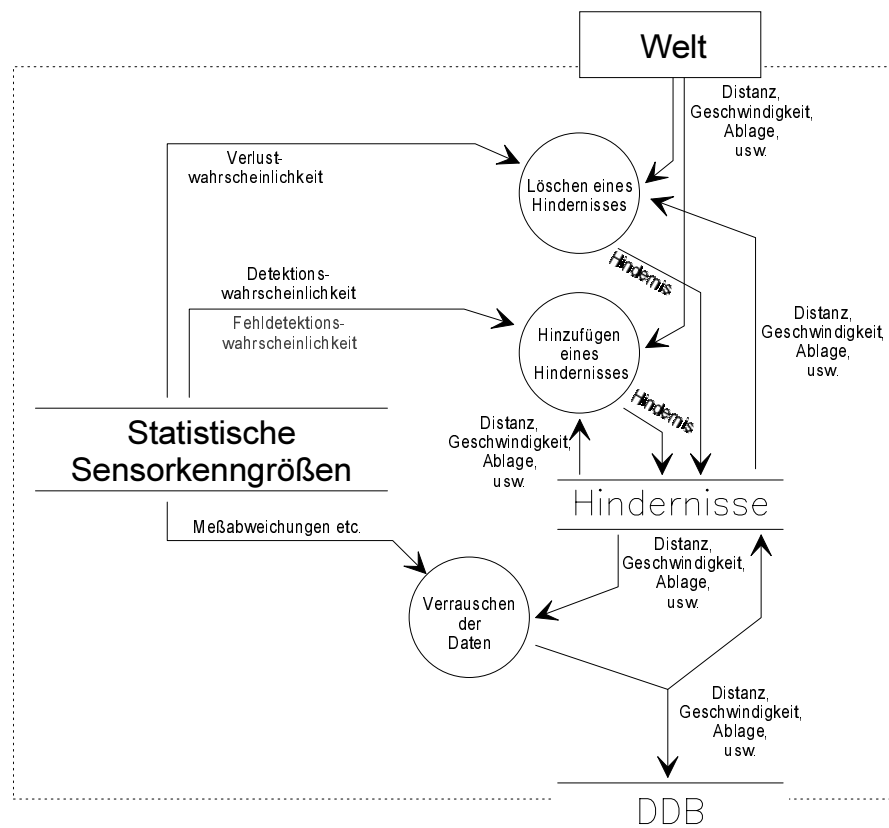


Abbildung 44 : Datenflußschema der Sensorsimulation

Die folgenden statistisch ermittelten Kenngrößen sind in allen simulierten Sensormodulen enthalten:

- Detektionswahrscheinlichkeit* Wahrscheinlichkeit der Erkennung eines existierenden Hindernisses.
- Fehldetektionswahrscheinlichkeit* Wahrscheinlichkeit der Erkennung (Meldung) eines Hindernisses, welches in der Realität nicht existiert.

<i>Verlustwahrscheinlichkeit</i>	Wahrscheinlichkeit des Verlierens eines durch den Hinderniserkennung über die Zeit verfolgten Objekts im Bild.
<i>Meßwertabweichung</i>	Durchschnittliche Abweichung der gemessenen von der realen Distanz des Hindernisses.

Darüber hinaus werden Kenngrößen verwaltet, die bestimmte Eigenschaften des Erkennungsprinzips widerspiegeln. Bei dem Modul ODT wird daher eine *Merkmalswahrscheinlichkeit* und eine *Merkmalswechselwahrscheinlichkeit* berücksichtigt. Typische Fehlerquellen des Moduls sind Merkmalsverwechslungen. Die gesuchte Schattenkante unter dem Fahrzeug wird beispielsweise in manchen Fällen mit der Stoßstange, der Fensterkante oder auch dem Nummernschild verwechselt. Daraus ergeben sich falsche Entfernungsschätzungen. Die Merkmalswahrscheinlichkeit gibt an, mit welcher Wahrscheinlichkeit bei Detektion eines Hindernisses eines dieser Merkmale erkannt wird. Jedem Merkmal ist eine relative Abweichung der Distanzmessung zugeordnet, die durch die falsche Merkmalsinterpretation hervorgerufen wird. Die Merkmalswechselwahrscheinlichkeit bezieht sich auf Hindernisse, die über die Zeit im Bild verfolgt werden. Über sie wird ein Neuaufsetzen des Hinderniserkenners simuliert.

Für das CT Modul werden ähnliche Wahrscheinlichkeiten für die Klassifikation des Hindernisses als PKW oder LKW ermittelt und in der Simulation benutzt.

Alle Kenngrößen sind distanzabhängig und in der Tabelle in Form von Entfernungsstützstellen angegeben. Werte zwischen den Stützstellen werden durch Interpolation ermittelt.

3.5.2.2 *Integration des Kontrollsystems von VITA II*

Über eine simulierte dynamische Datenbank (DDB) werden dem Kontrollsystem die gleichen Schnittstellen zur Verfügung gestellt, die auch im VITA II Fahrzeug existieren.

Darüber hinaus werden ‘Sonden’ eingefügt, die es ermöglichen, die Inhalte des Szenenspeichers der Situationsverwaltung sowie die verwendeten Datenbereiche der Verhaltensmusterinstanzen während eines Simulationslaufs zu betrachten.

Die Benutzerbefehle, die in VITA II über Tempomat und Blinkerhebel mitgeteilt werden, sind in der Simulation über Tastatur oder Joystick eingebbar.

3.5.2.3 *Simulation des Aktorsystems von VITA II*

Die Aktorsimulation besteht aus einem mathematischen Modell des Fahrzeugs²⁰, in dem die physikalischen Zusammenhänge der Fahrzeugdynamik definiert sind, sowie aus den Programmen zur unterlagerten Längs- und Querregelung des VITA II Fahrzeugs. Damit stehen dieselben Schnittstellen wie im realen Fahrzeug zur Verfügung, und es kann von einer adäquaten Simulation der Fahrzeugreaktion auf vorgegebene Steuerbefehle ausgegangen werden.

3.5.3 Die Analysehilfsmittel der Simulationsumgebung

3.5.3.1 *Generierung von Eingangsdaten*

Zur Generierung von Eingangsdaten stehen im wesentlichen drei Methoden zur Verfügung:

- (a) Die direkte Steuerung von zwei anderen Fahrzeugen durch den Benutzer mittels Tastatur oder Joystick.
- (b) Die Festlegung einer Agenda, die eine Liste von Benutzereingaben an bis zu 15 Fahrzeuge enthält, die zu festgelegten Zeitpunkten an die entsprechenden Fahrzeuge gesendet werden. Die anderen Fahrzeuge bewegen sich im gleichen Szenario und können mit verschiedenen Sensor- und Aktorsimulationen sowie unterschiedlichen Kontrollsystemen (u. a. dem des VITA II) ausgestattet sein.
- (c) Das Einspielen von Meßwerten, die während der Fahrt mit VITA II aufgenommen werden. Dafür ist ein Datenrecorder an die DDB angeschlossen, der die aktuellen DDB-Daten in bestimmten Abständen (40/80 ms) abspeichert. Für diese Messungen ist eine Schnittstelle vorgesehen, über die aufgezeichnete Daten direkt in die simulierte DDB eingespielt werden. Diese Option wird vor allem zum Testen des Systems zum Aufbau einer Situationsbeschreibung genutzt.

Die Agenda bietet eine gute Möglichkeit zur Erstellung von Demonstrationen und ist gleichzeitig für längere automatische Tests geeignet. Wird eine Agenda verwendet, ist zusätzlich eine Steuerung von Fahrzeugen per Tastatur oder Joystick möglich.

²⁰ Auf die Modelle selbst wird an dieser Stelle nicht eingegangen, da sie nicht Bestandteil der Arbeit sind. Es wurde auf existierende Dynamiksimulationsprogramme des Forschungsbereichs F1M/I zurückgegriffen. Der prinzipielle Aufbau solcher Modelle ist [Bösch 91] zu entnehmen.

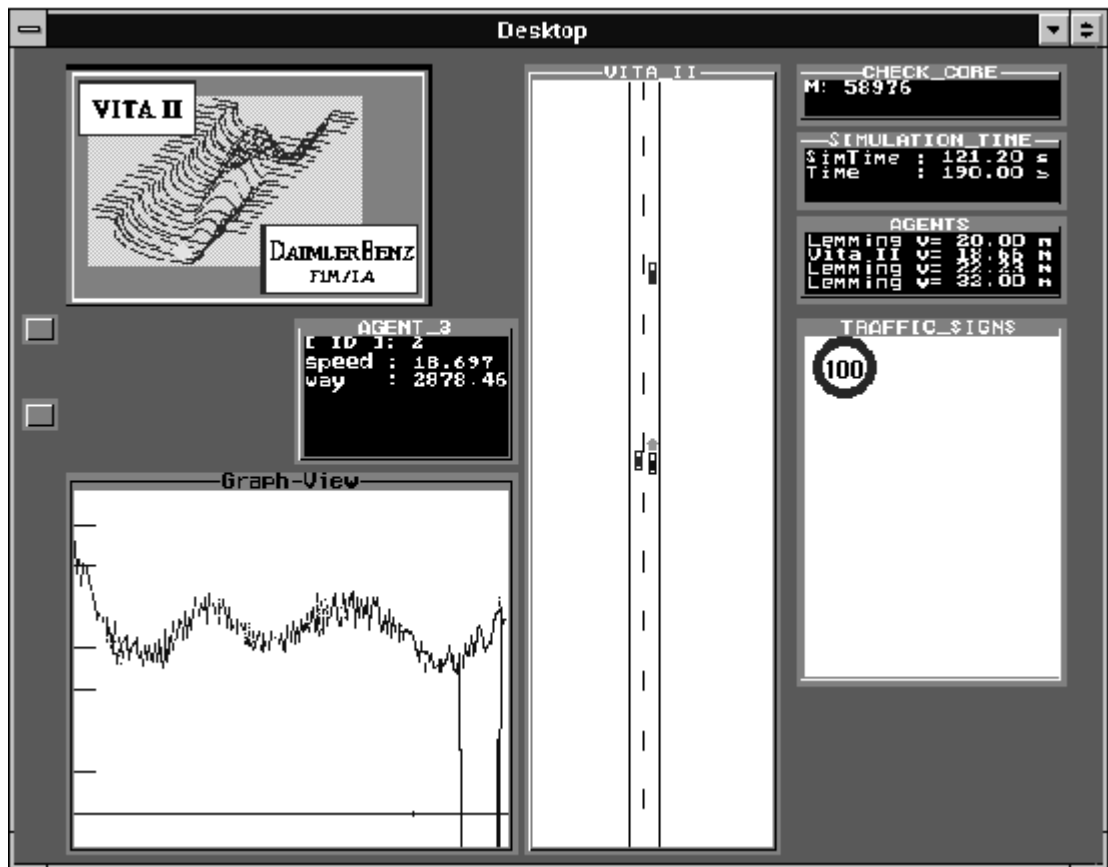


Abbildung 45 : Benutzeroberfläche der Simulation

3.5.3.2 Darstellung und Auswertung

Die Simulationsumgebung stellt eine Vielzahl von graphischen Darstellungen der simulierten Daten zur Verfügung. Dazu zählen:

- Textliche Darstellung von Daten während der Simulation.
- Animationen des Verkehrsszenarios aus der Vogelperspektive während eines Simulationslaufs. Die Darstellung ist skalierbar, d. h. die Beobachtungshöhe und damit die Größe des Szenenausschnitts sind einstellbar. Als Eingangsdaten können die exakten Daten der virtuellen Welt, die verrauschten Daten der Sensorsimulationen und die aufbereiteten Daten der Situationsbeschreibung dienen. Steuerbefehle des Verhaltensentscheidungsmoduls werden anhand von Pfeilen eingezeichnet.
- Graphische Darstellung von Wertverläufen. Die darzustellenden Werte sind frei wählbar. Eine kleine Zahl von Werten ist vorbereitet, für weitere Werte sind Zeiger auf die entsprechenden Objekte und Datenbereiche einzurichten.

Für die nachträgliche Auswertung von Testläufen steht ein Datenrecorder zur Verfügung, der den DDB-Recorder von VITA II simulieren kann. Darüber hinaus ist auch der Zugriff auf interne Daten des Situationsverwaltungssystems sowie der Verhaltenssteuerung möglich. Die Daten werden in einer Form gespeichert, die durch kommerzielle Systeme wie Microsoft EXCEL lesbar sind, so daß sie mit deren Hilfe ausgewertet werden können.

4 Ergebnisse in Simulation und Realität

In diesem Kapitel werden die Ergebnisse der Verfahren in der Praxis beschrieben. Dabei wird unterschieden zwischen den Ergebnissen in der Simulationsumgebung und den während Testfahrten mit dem Versuchsträger VITA II gewonnenen Erkenntnissen.

4.1 Ergebnisse im realen Straßenverkehr

Mit Hilfe der Algorithmen zur Fahrzeugsteuerung sind mit dem VITA II Versuchsfahrzeug ca. 10.000 km automatische Fahrt auf deutschen und französischen Autobahnen sowie autobahnähnlichen Landstraßen absolviert worden.

Aufgrund von technischen Randbedingungen seitens des Versuchsfahrzeugs liegen diesen Test- und Demonstrationsfahrten verschiedene Einstellungen des Kontrollsystems zugrunde:

- (1) Im *vollautonomen* Modus ist eine Rundumsicht durch die Fahrzeugsensoren gegeben und alle implementierten Verhaltensmuster sind verfügbar. In diesem Modus wird die Durchführung eines autonomen Überholmanövers zugelassen.
- (2) Im *teilautonomen* Modus werden diejenigen Verhaltensmuster entfernt, die neben der Spurhaltung weitere selbständige Eingriffe auf die Querführung des Fahrzeugs durchführen. Ein Spurwechsel ist in diesem Modus nur durch Setzen des Blinkers durch den Fahrer möglich. Dieser wird jedoch nicht durchgeführt falls es die Verkehrssituation nicht erlaubt (z. B. bei Fahrzeugen links oder rechts in sicherheitskritischen Abständen).
- (3) Im *assistierenden* Modus nimmt das Kontrollsystem nur noch die Aufgaben der Spurhaltung und Abstandhaltung wahr. Der Fahrer kann weiterhin durch Setzen des Blinkers ein automatisches Überholmanöver durchführen, muß dessen Durchführbarkeit im Hinblick auf andere Verkehrsteilnehmer jedoch selbst überwachen.

Der Unterscheidung der drei Modi liegen Verfügbarkeit und Qualität der Sensormodule zugrunde. Durch die ständige Weiterentwicklung der einzelnen optischen Sensoren und der zugehörigen Bildverarbeitung bezüglich ihrer Qualität sind Anpassungen insbesondere in dem Modul zum Aufbau einer Situationsbeschreibung nötig. Aufgrund des modularen Aufbaus des Kontrollsystems sind Einzelfunktionen leicht separat überprüfbar. Notreaktionen in Querrichtung werden aus Sicherheitsgründen nicht durchgeführt, da die Spurbreiten- und Ablagenschätzung des Straßenerkenners bei schnellen

Querbewegungen und den damit verbundenen Wankbewegungen des Fahrzeugs zu große Qualitätsverluste aufweisen.

4.1.1 Laufzeitbetrachtungen

Das Kontrollsystem besteht aus zwei Modulen, denen auf der transputerbasierten Rechnerplattform des VITA II Versuchsfahrzeugs zwei Prozesse zugeordnet sind, die parallel auf zwei Transputern des Typs T805 ablaufen und miteinander kommunizieren. Dem Modul zum Aufbau einer Situationsbeschreibung ist der Prozeß SA (engl. Situation Assessment) zugeordnet und dem Modul zur Verhaltenssteuerung der Prozeß BC (engl. Behaviour Control).

Die in VITA II gestellte Echtzeitanforderung schreibt eine Zykluszeit des Kontrollsystems von 40 bis 80 Millisekunden vor. Beide Prozesse erreichen unter folgenden Randbedingungen eine Zykluszeit von 40 Millisekunden:

- Es laufen keine weiteren Prozesse auf dem Transputer, der SA bzw. BC zugeordnet ist.
- Es laufen keine zusätzlichen virtuellen Kommunikationskanäle anderer Prozesse über die Verbindung zwischen den Transputern für SA und BC.
- Die Anzahl der verwalteten Hindernisse in SA ist auf 15 beschränkt¹.
- Die Anzahl der verwalteten Hindernisse in BC ist im vollautonomen Modus auf 8 beschränkt. In anderen Modi ist die Begrenzung von der Zahl der verwendeten Verhaltensmuster abhängig. Maximal ist eine Zahl von zehn Hindernissen vorgesehen.

Die Prozesse SA und BC sowie die zugehörigen Transputer sind in Form einer *Pipeline* angeordnet, so daß eine Gesamtzykluszeit von 40 Millisekunden zuzüglich einer Latenzzeit von weiteren 40 Millisekunden für das Kontrollsystem erreicht wird.

4.1.2 Ergebnisse aus Messfahrten

Aufgrund von Erfahrungen und Auswertungen der Sensormodule sind Einschränkungen für die Sensorfusion und die Durchführung autonomer Fahrmanöver zu treffen. Bestehende Randbedingungen und ihre Folgen werden nachfolgend aufgeführt. Anschließend werden einige durchgeführte Fahrmanöver exemplarisch dargestellt.

4.1.2.1 Randbedingungen bei autonomen Fahrten mit VITA II

Die nachfolgend aufgeführten Randbedingungen sind beim Einsatz der entwickelten Methoden und bei der Betrachtung erzielter Ergebnisse zu berücksichtigen:

¹ Der Prozeß 'schemabasierte Situationsanalyse' (vgl. Abschnitt 3.4) wird dabei noch nicht berücksichtigt.

- Vision Bumper (VB) liefert zu erkannten Fahrzeugen keine Geschwindigkeit und keine Angabe über den seitlichen Versatz gegenüber dem autonomen Fahrzeug. Ein autonomes Überholen wird damit ausgeschlossen und die Einsatzmöglichkeit ist auf die automatische Abstandhaltung beschränkt. **Fehler! Verweisquelle konnte nicht gefunden werden.** zeigt eine Abstandsmessung von Vision Bumper bei manueller Fahrt. Dabei wechselt VITA II die Spur.

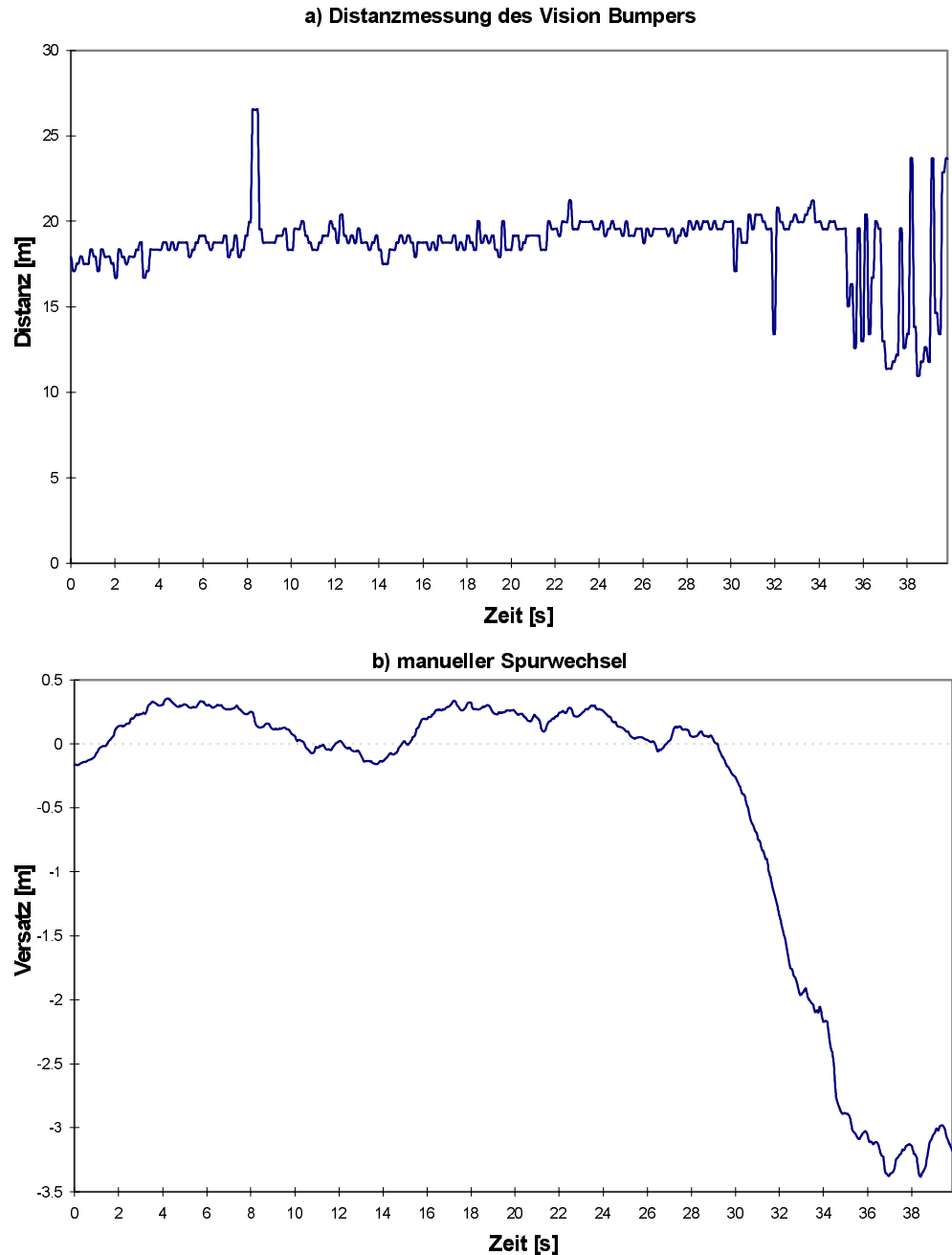


Abbildung 46 : Messung von Vision Bumper

- LOD liefert wie VB keine Geschwindigkeit und gibt nur eine seitliche Distanz zu einem Fahrzeug an, welches sich neben VITA II befindet. In SA werden diese Messungen daher nur zur Bestätigung von extrapolierten Objekten verwendet. Wird ein bisher nicht verwaltetes Fahrzeug initial von LOD erkannt, so wird ihm die Eigengeschwindigkeit von VITA II zugeordnet. Wird es nicht bestätigt, so wird es nach einer geringen Zahl von Zyklen wieder aus dem Szenenspeicher entfernt.
- Alle Bildverarbeitungsmodule sind in ihrer Qualität sehr wetterabhängig. Eine Meßdaten-Qualität die ausreicht, um auch Überholmanöver autonom durchführen zu können wird daher nicht immer erreicht.
- Fahrzeuge werden durchschnittlich erst in einer Entfernung von 50 bis 70 Metern erkannt. In höheren Geschwindigkeitsbereichen und bei großen Differenzgeschwindigkeiten führt das zu einem abrupteren, stärkeren Abbremsen des Fahrzeugs. Auf Autobahnen führt diese verkürzte 'Sichtweite' aufgrund der hohen Differenzgeschwindigkeiten von überholenden Fahrzeugen zusätzlich zu einer zu geringen Abschätzbarkeit der Durchführbarkeit eines Überholmanövers.

4.1.2.2 Exemplarische Darstellung von Fahrmanövern

Im Folgenden werden Fahrmanöver aufgeführt, die in dieser und ähnlicher Form separat und kombiniert während Test- und Versuchsfahrten durchgeführt wurden.

Automatischer Spurwechsel

Ein automatischer Spurwechsel geht entweder von der autonomen Entscheidung der Verhaltenssteuerung aus oder wird durch den Fahrer kommandiert. Durch Setzen des Blinkers kommandiert der Fahrer einen Spurwechsel der vom Kontrollsystem durchgeführt wird, sofern es die Verkehrssituation erlaubt.

Abbildung 47 zeigt den Verlauf von Soll- und Istwert der lateralen Position von VITA II während zwei aufeinanderfolgenden Spurwechseln. Zusätzlich wird der vom Straßenerkennungsmodul gelieferte Ablagenwert von der Spurmitte sowie der verwaltete Spurwechselzähler und die Straßenbreite dargestellt.

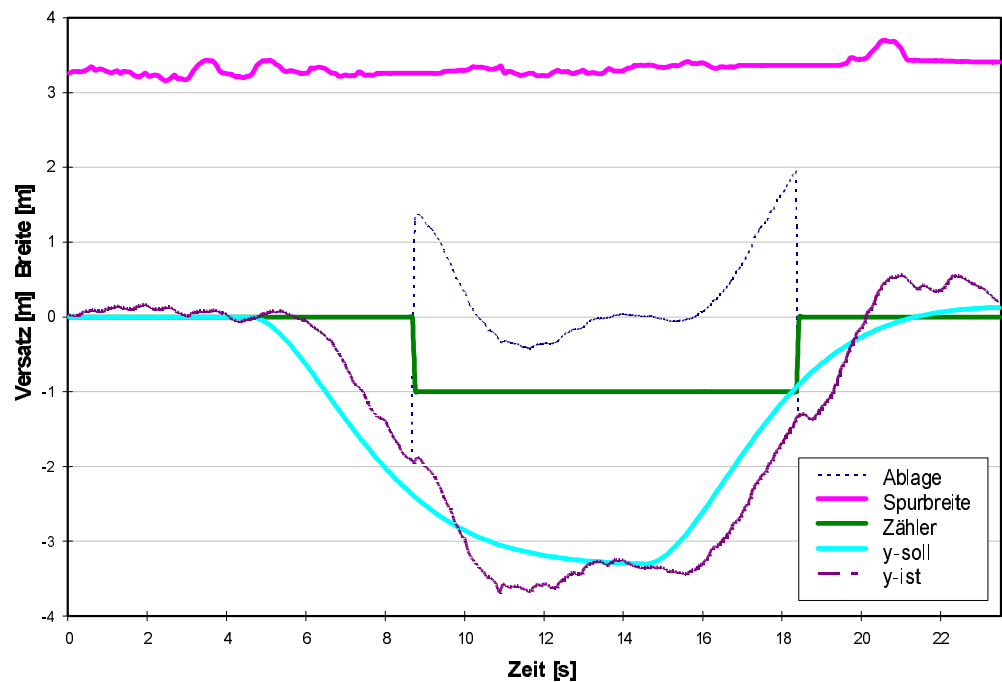


Abbildung 47 : Automatischer Spurwechsel des VITA II Fahrzeugs

Verfolgungsfahrt

Am Beispiel einer Verfolgungsfahrt wird nachfolgend die Fusion der Meßwerte verschiedener Hinderniserkennungsmodule veranschaulicht. Dazu ist zu bemerken, daß in der zugrundeliegenden Hardwarekonfiguration nur ein alternatives Ablaufen von VisionBumper und CarTrack möglich ist². Die Fusion beschränkt sich daher auf CarTrack und ODT. Außerdem liefert CT keine verlässliche Schätzung der Relativgeschwindigkeit, so daß hier nur die Resultate von ODT zugrundegelegt werden.

Die Verschiebung der in Abbildung 48 dargestellten fusionierten Entfernungswerte ergibt sich aus unterschiedlichen Referenzpunkten für die Entfernungsangabe. Während ODT und CT die Entfernung zur Fahrzeugbegrenzung messen, wird diese in der Situationsbeschreibung bezüglich des Fahrzeugmittelpunkts angegeben.

² Das Modul Behaviour Control (BC) hat die Möglichkeit die Auswahl zwischen VB und CT während der Laufzeit zu ändern. Eine angekündigte Version der Module soll eine parallele Ausführung ermöglichen.

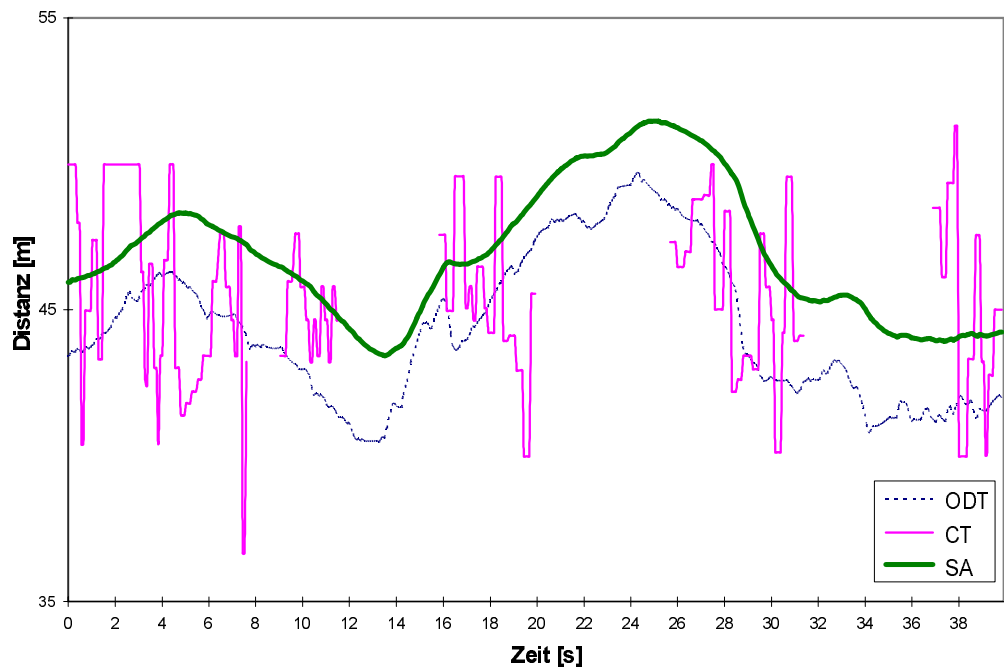


Abbildung 48 : Fusion der Entfernungsmessung von CT und ODT

Mit Hilfe der Entfernungsmessungen und der Geschwindigkeitsmessung von ODT wird die Geschwindigkeit des vorausfahrenden Fahrzeugs geschätzt.

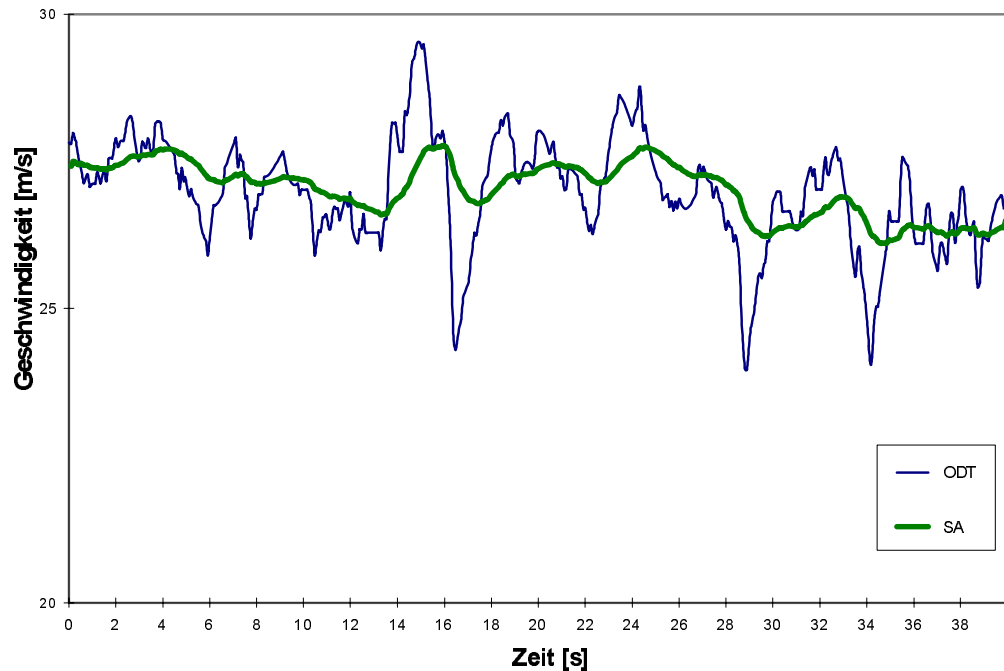


Abbildung 49 : Schätzung der Hindernisgeschwindigkeit aus Messungen

Ein Hindernis wird mit der so gewonnenen Information über Distanz und Geschwindigkeit in der Situationsbeschreibung des SA Moduls verwaltet. Die aufbereiteten Informationen über Straße, Verkehrszeichen und Hindernisse werden (alle 40 ms) an das Modul BC geschickt.

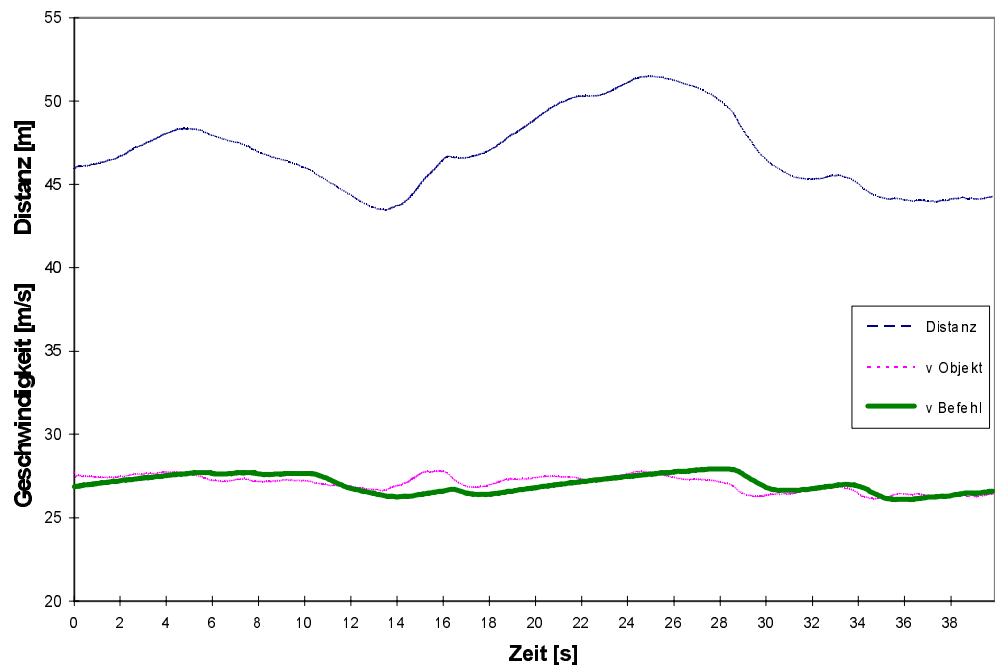


Abbildung 50 : Automatische Abstandhaltung

Abbildung 50 zeigt die Fahrzeugreaktion auf ein vorausfahrendes Fahrzeug, die durch die Verhaltenssteuerung hervorgerufen wird. Der gezeigten Reaktion liegt eine durch den Fahrer eingestellte Wunschgeschwindigkeit von 30 m/s zugrunde.

Abbremsen auf Sicherheitsabstand

Abbildung 51 stellt die Reaktion auf ein langsames, vorausfahrendes Fahrzeug dar. Hier wird das oben erwähnte Problem der späten Erkennung von Hindernissen deutlich. Das vorausfahrende Fahrzeug fährt mit einer Geschwindigkeit von ca. 22 m/s. Zum Abbau der hohen Differenzgeschwindigkeit wird der Sicherheitsabstand kurzzeitig unterschritten. Dabei wird die Eigengeschwindigkeit soweit reduziert, daß die Geschwindigkeit des vorausfahrenden Fahrzeugs unterschritten wird und sich der Abstand wieder vergrößert. Anschließend wird leicht beschleunigt um dem Fahrzeug schließlich im Sicherheitsabstand zu folgen.

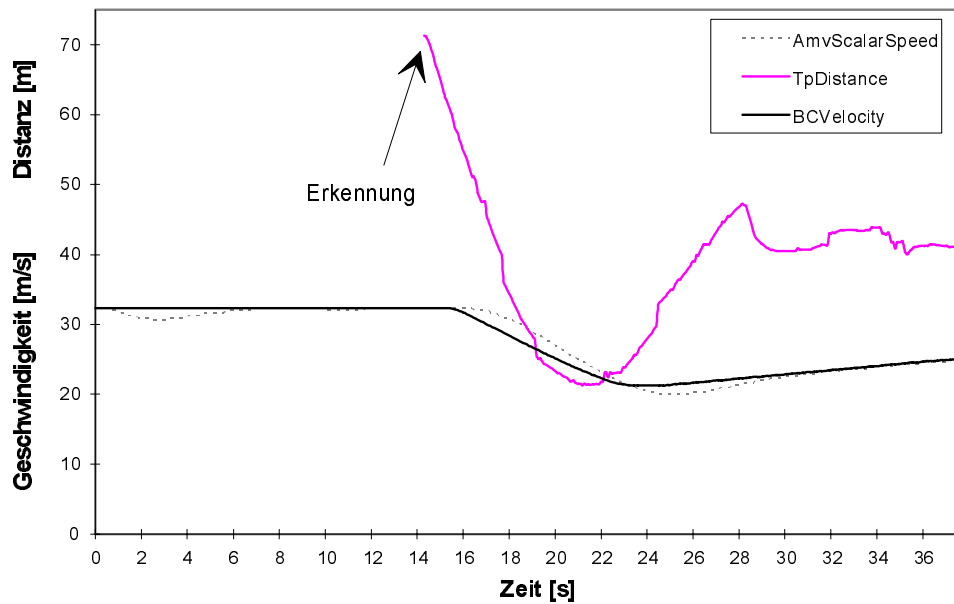


Abbildung 51 : Abbremsen auf ein langsames Fahrzeug

4.2 Simulationsergebnisse

Die Simulation dient dem realitätsnahen Testen des Kontrollsystems. Die Genauigkeit der Modellierung des Sensor- und Aktorsystems wird dabei je nach der zu testenden Eigenschaft gewählt. Beispielsweise lassen sich auf diese Weise die in Realität alternativ ablaufenden Hinderniserkennungsmodule VB und CT gleichzeitig simulieren.

Die Ergebnisse der Softwaremodule SA und BC in der Simulation werden nachfolgend anhand eines Überholvorgangs und eines Szenarios zur Demonstration von Notreaktionen veranschaulicht.

4.2.1 Beispiel eines Überholvorgangs

Am Beispiel einer simulierten Fahrt des autonomen Fahrzeugs wird in diesem Abschnitt das Zusammenwirken von Verhaltensmustern veranschaulicht. In der Ausgangssituation befindet sich ein Fahrzeug auf der linken Spur direkt neben dem autonomen Fahrzeug und ein zweites, langsames Fahrzeug fährt in 90 Metern Entfernung voraus. Die Reaktion des autonomen Fahrzeugs wird nun im wesentlichen durch die vier Verhaltensmuster *Überholen*, *Spurhemmung links*, *Folgen* und *Spurhalten* bestimmt. Abbildung 52 zeigt die Distanzverläufe der beiden simulierten Hindernisse während des Simulationslaufs. Beide fahren mit konstanter Geschwindigkeit.

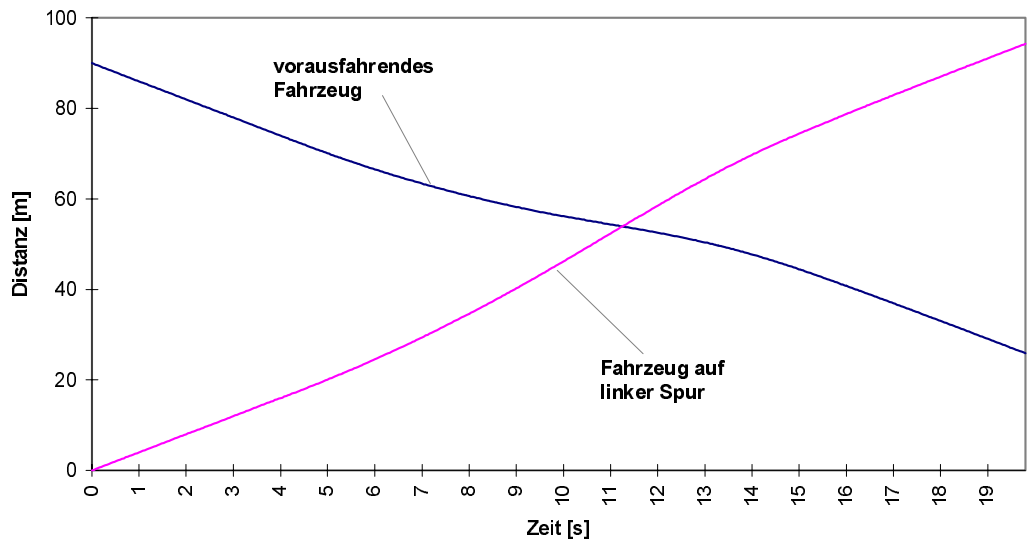


Abbildung 52 : Distanzverläufe der Hindernisse eines Simulationslaufs

Die Positionen und Geschwindigkeiten der Hindernisse liegen der Bestimmung der Basiswerte der Verhaltensmuster zugrunde. Abbildung 53 veranschaulicht die schematisierte Handlung des Überholens. Die Hemmung entspricht (in diesem Beispiel) dem Basiswert des Verhaltensmusters *Spurhemmung links*. Nach Erreichen eines ausreichenden seitlichen Abstands zum vorausfahrenden Fahrzeug wird die Qualifikationsbedingung nicht mehr erfüllt und die Aktivität wird eingestellt.

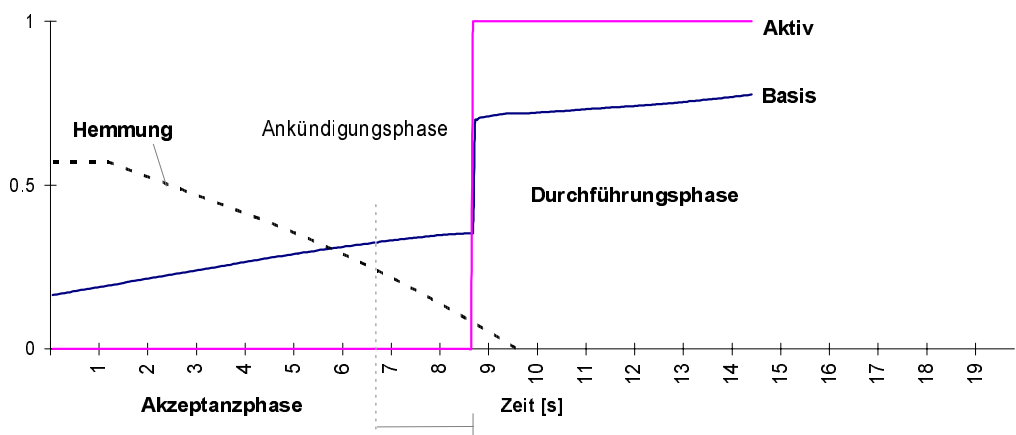


Abbildung 53 : Aktivierung des Überhol-Verhaltensmusters

Aufgrund der Hemmung des Überhol-Verhaltensmusters reagiert das autonome Fahrzeug auf das vorausfahrende Hindernis durch abbremsten. Diese Aktion ist durch das Basisverhaltensmuster zum Folgen eines Fahrzeugs beschrieben.

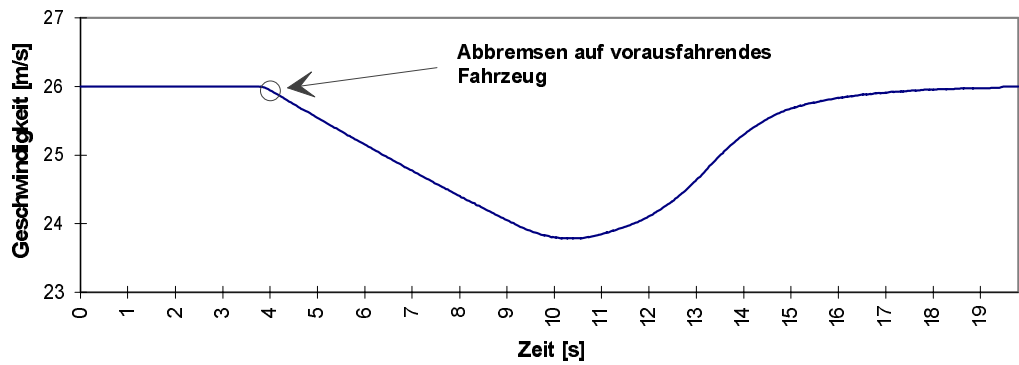


Abbildung 54 : Geschwindigkeitsverlauf für das autonome Fahrzeug

Bis zur Aktivierung des Überhol-Verhaltensmusters wird abgebremst, bevor diese Aktion gehemmt wird um zu Überholen. Abbildung 54 zeigt die zugehörige Geschwindigkeitskurve.

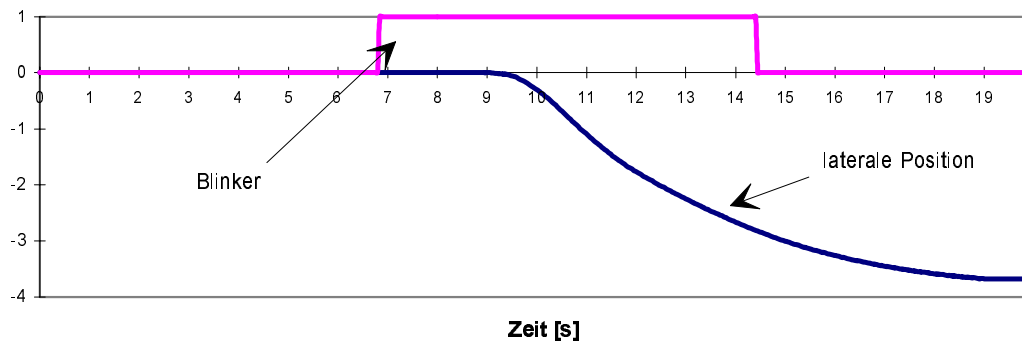


Abbildung 55 : Simulierter Überholvorgang

Das Überhol-Verhaltensmuster tätigt zwei Aktionen: das Setzen des Blinkers und die Vorgabe einer Quergeschwindigkeit. Während seiner Aktivierung überlagert die Intensität des Überhol-Verhaltensmusters die des Basisverhaltensmusters zum Spurhalten. Nach der anschließenden Deaktivierung übernimmt wieder das Basisverhaltensmuster die Kontrolle und führt das autonome Fahrzeug auf die Mitte der aktuellen Spur. Abbildung 55 zeigt den Verlauf der lateralen Position und die Aktivierung des Blinkers.

4.2.2 Szenario für Notreaktionen

Die Notreaktionen sind für Situationen gedacht, in denen Basisverhaltensmuster und schematisierte Handlungen keine adäquate Lösung bieten. Sie werden insbesondere dann ausgeführt, wenn Hindernisse sehr spät erkannt werden oder andere Verkehrsteilnehmer unerwartete, risikoreiche Manöver durchführen.

Das hier dargestellte Szenario zeigt den fiktiven Fall einer Annäherung an ein Stauende. Die stehenden Fahrzeuge werden erst unmittelbar vor einer Kollision erkannt, so daß durch eine Kombination aus Ausweichmanöver und Vollbremsung reagiert werden muß.

Dem dargestellten Fall liegt ein simulierter logischer Sensor zugrunde, der Hindernisse der gesamten Umgebung erfasst und mit einem geringen Rauschen (maximal 10 % Abweichung vom Realwert) versehen ist. Die Fahrzeugbewegung entsteht durch die Dynamiksimulation des VITA II Fahrzeugs. Abbildung 56 zeigt die Positionen von statischen Hindernissen und die Trajektorie des autonomen Fahrzeugs.

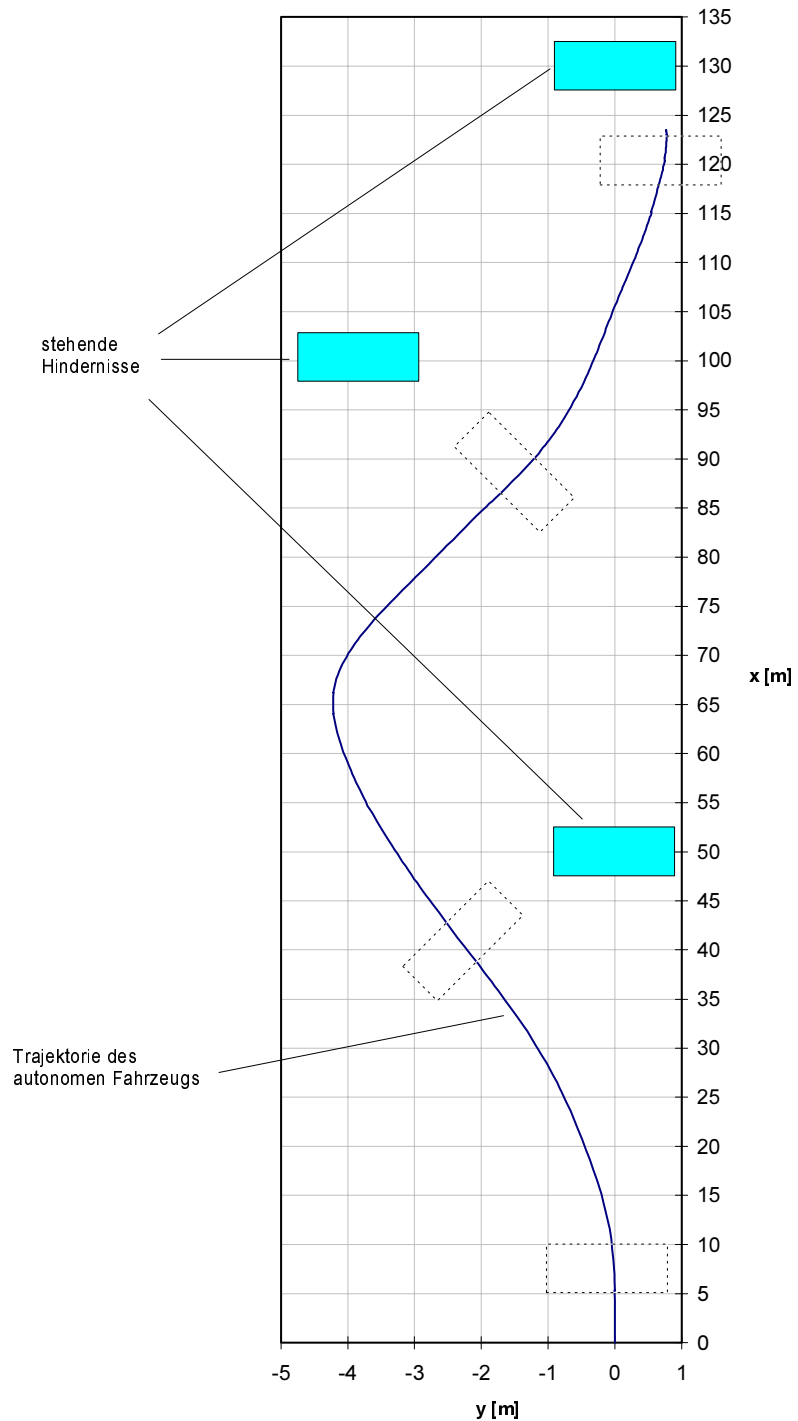


Abbildung 56 : Notreaktionsszenario

Gerade bei Notreaktionen ist das Wissen über die Fahrzeugdynamik wichtig für die Berechnung von Steuervorgaben. Abbildung 57 zeigt den kommandierten Soll-Wert sowie den erreichten Ist-Wert der lateralen Position für das oben aufgeführte Szenario.

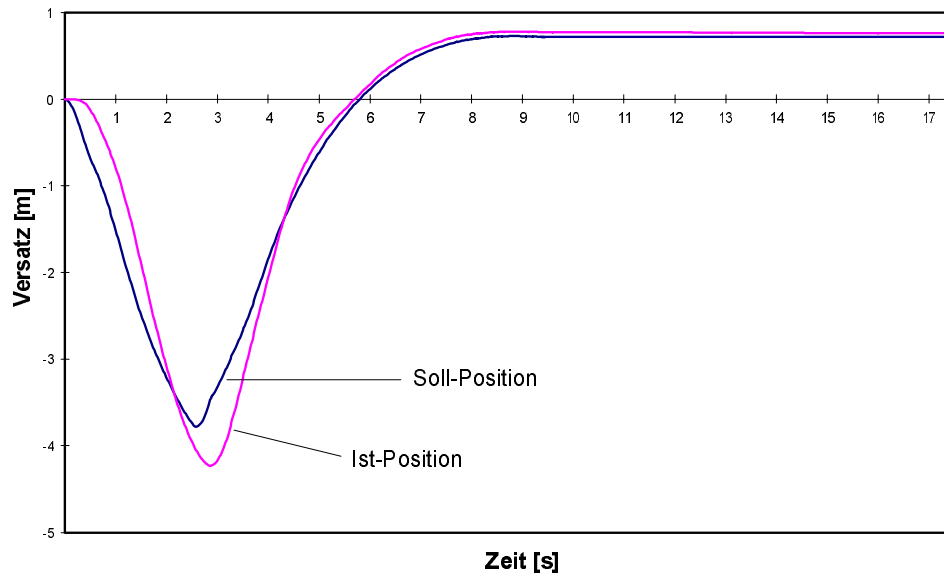


Abbildung 57 : Steuervorgabe für die laterale Position

Gleichzeitig wird das Verhaltensmuster Notbremsung aktiviert und kommandiert eine Vollbremsung bis zum Stillstand. Ab dem Zeitpunkt des Stillstands wird auch keine laterale Positionsänderung mehr kommandiert.

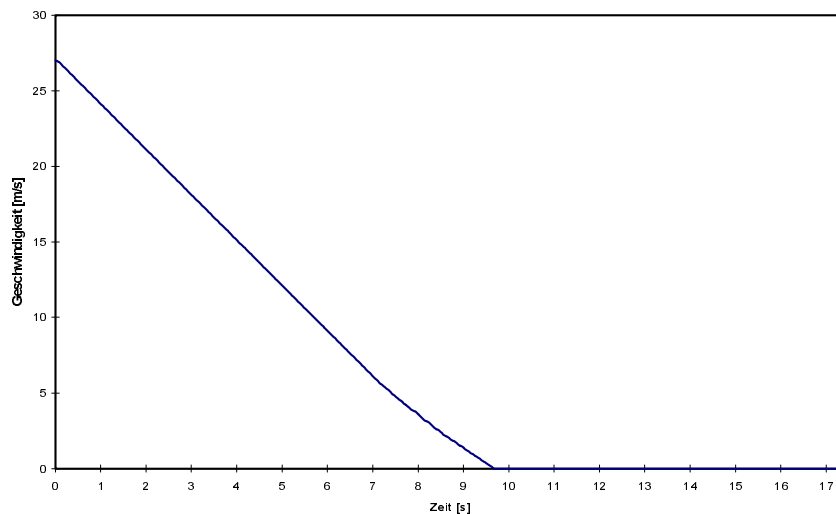


Abbildung 58 : Notbremsung bis zum Stillstand

Zu den Notreaktionen bleibt abschließend noch einmal festzuhalten, daß sie auch unter Voraussetzung perfekter Sensoren nicht für alle Situationen eine Lösung bieten. Setzt man beispielsweise die Position des dritten Fahrzeugs von 130 auf 120 Meter so ist eine Kollision unvermeidbar.

5 Zusammenfassung und Ausblick

5.1 Zusammenfassende Kritik

Das Thema „Autonomes Fahren“ führt seit den Ergebnissen der PROMETHEUS-Forschung immer wieder zu kontroversen Diskussionen. Aus den verschiedenen Interpretationen des Begriffs ergibt sich eine Vielzahl von Aufgabenstellungen. Die vorliegende Arbeit beschäftigt sich in diesem Zusammenhang mit der Verhaltenssteuerung eines autonomen Fahrzeugs auf Autobahnen.

Auf Wissen über Eigenschaften von Objekten der Umgebung, wie der Straße, der Fahrzeuge und der Verkehrszeichen, sowie über Abhängigkeiten zwischen diesen und über allgemeine Verhaltensregeln kann hier nicht verzichtet werden. Vor dem Hintergrund der Echtzeitanforderung eines solchen Systems, das in der Lage sein muß direkt mit seiner Umgebung zu interagieren, stellt sich die Frage nach der Repräsentation des verwendeten Wissens.

Charakteristisch für das hierzu entwickelte Konzept ist die Verknüpfung von Verhaltensmustern, die schematische Beschreibungen von Verhaltensweisen darstellen. Das Wissen über ein Objekt wird in dieser Repräsentation durch eine *Handlung* repräsentiert. Die Ableitung des Fahrverhaltens geschieht mittels einer Gefahrenrepräsentation. Jedem Objekt der Szene wird eine subjektive Gefahr zugeordnet und die Verhaltensmuster dienen der Reduktion dieser Gefahr. Insbesondere bestimmt jedes Verhaltensmuster einen subjektiven Gefahrenwert für einen Aspekt der Situation.

Das Verhaltensmusterkonzept berücksichtigt die wichtige Eigenschaft der Kontinuität der Bewegungsabläufe in der Verkehrswelt. Die Verknüpfung der Verhaltensmuster geschieht daher nicht auf qualitativer Ebene. Alle Verhaltensmuster bestimmen parallel eine Steuervorgabe für den nächsten Zeitschritt. Diese Steuervorgaben werden mit Hilfe einer dynamischen Priorisierung der Verhaltensmuster und eines Inhibitionsnetzwerks, welches eine Hemmung von Verhaltensweisen untereinander realisiert, fusioniert.

Das Konzept integriert reaktive Komponenten zur Kollisionsvermeidung in Form von Notreaktionen, die der weitestgehenden Absicherung des Konzepts gegen Modellierungsfehler dienen.

Das Verhaltensmusterkonzept wird im Hauptteil dieser Arbeit vorgestellt und ist für ein Autobahnumfeld im VITA II Versuchsfahrzeug realisiert. Auch wenn eine Vielzahl von

Testfahrten bislang zu keiner kritischen Situation führte, muß festgehalten werden, daß derzeit kein technisches System denkbar ist, welches eine annähernd so geringe Ausfallhäufigkeit wie der Mensch aufweist. Eine Begründung dafür ist nicht allein bei den Sensoren zu suchen. Auch die hier vorgestellten Notreaktionen sind zwar in den meisten Fällen ausreichend, führen aber nur bedingt zur Lösung von Konfliktsituationen, die u. a. auch durch das Verhalten anderer Verkehrsteilnehmer hervorgerufen werden.

Die eingangs aufgestellten Anforderungen an die Verhaltenssteuerung sind durch die Realisierung in VITA II vor allem im Punkt Echtzeitfähigkeit erfüllt. Auch die Erklärbarkeit der generierten Fahrmanöver ist durch die Modellierbarkeit eines Fahrers gegeben. An dieser Stelle ist jedoch das Designerproblem zu erwähnen. Die Zuordnung von subjektiven Gefahren zu Verhaltensmustern und deren Abstimmung im Sinne einer gemeinsamen, plausiblen Handlung bleibt dem Ersteller einer Anwendung überlassen. Die hier zusammengestellten Verhaltensmuster sind auf die fehlerbehafteten Eingangsdaten eingestellt und berücksichtigen daher nur grobe Fahrmanöver (z. B. Spurwechsel). Die Verhaltensmuster und ihre subjektive Gefahreuzuordnung stellen die Repräsentation von Erfahrung eines „Experten“ dar. Die direkte Akquisition dieser Gefahrenwerte ist sehr schwierig, so daß eine Adaption der Gefahrenwerte und Hemmungen in simulierten Verkehrsszenarien folgen muß. Die so gewonnenen Verhaltensmuster sind plausibel und funktionieren auch in ihrer Kombination in einem Großteil der Situationen. Betrachtet man das Drei-Ebenen Modell von Rasmussen, so sind die bisher verwendeten Verhaltensmuster auf den Ebenen der *Fähigkeiten* und des *regelbasierten Verhaltens* angesiedelt.

Tritt eine Situation ein, die nicht auf diesen Ebenen gelöst wird, so findet der Mensch dies heraus und kann eine Lösung durch „Nachdenken“ finden. Diese Verhaltensweise ordnet Rasmussen dem wissensbasierten Verhalten zu (vgl. [Rasmussen 86]).

An dieser Stelle besteht ein enormes Entwicklungspotential für die Erweiterung des Verhaltensmusteransatzes, der aus Gründen der Echtzeitfähigkeit diese Ebene bisher ausklammert.

Für den Einsatz der Verhaltenssteuerung in VITA II wird eine Aufbereitung der Sensorinformationen nötig. Die hier vorgestellte Methode verwendet einen Szenenspeicher, in dem Kandidaten für Objekte der Szene gespeichert sind. Auf diesem Szenenspeicher arbeiten mehrere Verwaltungsprozesse, welche die Aufnahme von neuen Objekten, das Löschen von Objekten und die Fusion und Glättung der Messungen behandeln.

Dem Basissystem zum Aufbau einer Situationsbeschreibung liegt kein Modell der Verkehrssituation zugrunde. Hauptaufgaben sind die Sensorfusion sowie die Unterdrückung von Fehldetektionen. Ergänzt wird das System durch ein explizites

schemabasiertes Fehlermodell. Dieses repräsentiert Wissen durch eine abstrakte Szenenbeschreibung und eine Korrekturbeschreibung für bekannte Fehlerfälle. In vielen Fällen läßt sich das Vorliegen eines Fehlerfalls nur aus einem bestimmten zeitlichen Ablauf heraus erkennen. Aus diesem Grund ist in diesen Szenenbeschreibungen eine explizite Repräsentation von zeitlichen Verläufen enthalten.

Die entwickelte Methode der Verwaltung und des Aufbaus von Situationsbeschreibungen ist Grundlage für die Verhaltenssteuerung des autonomen Fahrzeugs im realen Straßenverkehr.

Die hier vorgestellten Konzepte und Methoden sind stark am Einsatz im realen Straßenverkehr und insbesondere an den Gegebenheiten und Einschränkungen des Versuchsfahrzeugs VITA II orientiert. Insgesamt dienen sie einem reaktiven Fahrzeugverhalten in einem eingeschränkten Einsatzgebiet.

Die folgenden Weiterentwicklungen sind daher vorstellbar

- *Die Erweiterung des Einsatzgebietes auf Innenstadtverkehr und Landstraßen.*

Um eine solche Erweiterung zu realisieren müssen in jedem Fall Methoden hinzugefügt werden, die eine Situationsanalyse auf wissensbasierter Ebene ermöglichen, da insbesondere im Stadtverkehr eine Vielzahl von Informationen verarbeitet werden muß und es sich nicht mehr nur um „Routine“-Handlungen wie auf der Autobahn handelt.

- *Die Integration einer Planung.*

Bisher wird eine Vorausschau nur bedingt realisiert. Eine deutliche Verbesserung insbesondere im Hinblick auf Kollisionsvermeidung ist durch die Planung und die Abschätzung von Handlungsabsichten anderer Verkehrsteilnehmer zu erreichen.

- *Die Integration eines Navigationssystems.*

Das im Hauptteil beschriebene Kontrollsystem erlaubt als externe Vorgaben lediglich die Eingriffe des Fahrers über Tempomat und Blinkerhebel. Darauf aufbauend kann ein kommerziell verfügbares Navigationshilfesystem für globale Planungsaufgaben eingesetzt werden um beispielsweise auf relevante Ausfahrten zu reagieren und sich rechtzeitig rechts einzuordnen.

- *Die Untersuchung der Interaktion mehrerer autonomer Fahrzeuge*

Für den - zugegebenermaßen weit entfernten - Einsatz vollautonomer Systeme ist es wichtig, deren gegenseitige Beeinflussung zu untersuchen. Die hier vorgestellte Simulationsumgebung ist darauf ausgelegt und erste Tests zur Untersuchung des Verhaltens zweier autonomer Agenten mit gleichem Kontrollsystem wurden durchgeführt. Diese führten jedoch aus Zeitgründen bislang noch zu keinem aussagekräftigen Ergebnis.

Abschließend bleibt festzuhalten, daß im Rahmen dieser Arbeit ein umfassendes Konzept zur autonomen Fahrzeugführung entwickelt und realisiert worden ist, und auch im realen Straßenverkehr erfolgreich eingesetzt wird. Bis ein menschlicher Fahrer durch ein technisches System abgelöst werden kann, ist sicherlich noch ein weiter Weg zu gehen.

5.2 Vergleich zu existierenden Verfahren

Neben den Forschungsthemen innerhalb des PROMETHEUS Programms werden in Europa von vielen Automobilherstellern, Universitäten und Forschungsinstituten Themen bearbeitet, die sich mit der autonomen Fahrzeugführung befassen. Auch in den USA werden Teilthemen im Rahmen des AHS Programms verstärkt vorangetrieben. In Japan widmet man sich ähnlichen Themen und hat bereits Teilsysteme auf den japanischen Markt gebracht.

In den meisten Fällen handelt es sich jedoch mehr um isolierte Teilsysteme als um ein umfassendes autonomes System, wie es das VITA II Fahrzeug der Daimler-Benz AG darstellt. Dem Verfasser ist neben dem VITA II Versuchsträger nur ein weiteres Fahrzeug bekannt, welches derzeit in der Lage ist, im realen Autobahnverkehr autonom zu fahren. Dabei handelt es sich um das Versuchsfahrzeug VaMoRs-P der Universität der Bundeswehr München. Ein Vergleich zu anderen Verfahren zur Verhaltenssteuerung autonomer Fahrzeuge ist daher schwierig. Im folgenden Abschnitt werden Unterschiede und Gemeinsamkeiten zwischen dem hier vorgestellten Verfahren zur Verhaltenssteuerung und dem Kontrollsystem des VaMoRs-P aufgezeigt.

Anschließend werden andere Ansätze aufgeführt, die sich mit ähnlichen Aufgabenstellungen befassen.

5.2.1 Das Kontrollsystem von VaMoRs-P

Das VaMoRs-P Versuchsfahrzeug der Universität der Bundeswehr München ist wie VITA II Resultat der Aktivitäten innerhalb des PROMETHEUS-Projekts Collision Avoidance [Dickmanns et al. 94]. Beide Systeme basieren sowohl auf dem gleichen zugrundeliegenden Fahrzeugtyp als auch auf der gleichen verwendeten Hardware- und Systemarchitektur. Unterschiede bestehen in der Anzahl der Sensoren und in der Realisierung des Kontrollsystems. Während VITA II den vorderen Sichtbereich mit mehreren redundanten Hinderniserkennern abdeckt und ein zusätzliches Modul zur Überwachung des Seitenbereichs verwendet, beschränkt sich VaMoRs-P auf ein Hinderniserkennungssystem für den vorderen und hinteren Bereich. Das Kontrollsystem des Fahrzeugs untergliedert sich in das Modul *Vehicle Control* und das Modul *Behaviour Decision* [Kujawski 93, Kujawski 95]. Diese Module werden im Folgenden kurz zusammengefaßt und ein Vergleich zu dem hier vorgestellten

Verhaltenssteuerungssystem angestellt. Das Kontrollsystem verfügt über eine explizite Repräsentation der Verkehrssituation.

5.2.1.1 Zusammenfassung

Die Aufgabe des Vehicle Control Moduls ist die Ansteuerung von Lenkung, Gas und Bremse. Es stellt Routinen zur Abstandhaltung auf ein ausgewähltes Objekt sowie zum Wechsel der Fahrspur zur Verfügung. Das Modul Vehicle Control soll an dieser Stelle nicht näher betrachtet werden.

Die Aufgabe des Moduls Behaviour Decision ist die Auswertung der aktuellen Verkehrsszene, das Treffen einer Entscheidung über das Verhalten und die Übermittlung der entsprechenden Kommandos an Vehicle Control. Das Modul unterscheidet vier verschiedene Modi:

- Spurhaltung/Verbleiben in der aktuellen Spur,
- Spurwechsel nach links oder rechts,
- Abbruch eines Spurwechsels und
- Überholen (als zusammengesetztes Manöver).

Der Verhaltensentscheidung liegt eine Bewertung der möglichen Fahrmanöver zugrunde. Dazu werden die zugehörigen Trajektorien des autonomen Fahrzeugs und der anderen Verkehrsteilnehmer extrapoliert (Vorausschau 7.5 s). Das Bewertungskriterium basiert auf der minimalen Entfernung des autonomen Fahrzeugs zu einem anderen Fahrzeug an, die bei Durchführung eines Manövers erreicht wird.

Nach Ermittlung und Bewertung möglicher Fahrmanöver wird mit Unterstützung einer Wissensbasis eine Entscheidung gefällt. Als Repräsentationsform der Wissensbasis wird ein Zustandsübergangssystem (engl. state transition system; STS) verwendet.

Jedem Zustand sind ein oder mehrere Operationen zugeordnet, die bei Eintritt in den Zustand ausgeführt werden. Ein Zustandsübergang findet statt, wenn eine Übergangsbedingung erfüllt ist. Zur Lösung von Konflikten, die durch gleichzeitiges Erfüllen mehrerer Übergangsbedingungen entstehen, wird den Transitionen eine Priorität zugeordnet.

Übergangsbedingungen bestehen aus boole'schen Ausdrücken. Dazu gehören u. a. Prädikate, die durch die Trajektorienbewertung entstehen („Spurwechsel links ist möglich“, „Spurwechsel rechts ist möglich“, „Spurwechsel links abbrechen“, usw.) sowie weitere abgeleitete und berechnete Prädikate („Spurwechsel links erwünscht“, „Spurwechsel rechts erwünscht“, usw.).

Das Modul Behaviour Decision beansprucht eine Zykluszeit von bis zu 300 ms auf zwei Transputern des Typs T800. Diese Zykluszeit ist ausreichend, da die direkte Ansteuerung des Fahrzeugs über das in Echtzeit ablaufende Modul Vehicle Control läuft.

5.2.1.2 Vergleich

Aufgrund der nahezu gleichen Voraussetzungen und Schnittstellen zu Sensormodulen und Fahrzeugaktoren läßt sich ein Vergleich der Verfahren anhand der durchführbaren Fahrzeugaktionen anstellen. Die von VaMoRs-P durchführbaren Fahrmanöver sind auch durch den Verhaltensmusteransatz realisiert.

Das Fahrmanöver 'Spurhalten' wird durch ein Basisverhaltensmuster realisiert, während Spurwechsel links und rechts durch schematisierte Handlungen (Überholen, Rechtsfahrgebot) realisiert sind. Auch das Überholmanöver wird durch Zusammenwirken mehrerer Verhaltensmuster realisiert. Ein Spurwechselabbruch ist jederzeit durch Hemmung und Überlagerung durch ein konkurrierendes Verhaltensmuster möglich.

An dieser Stelle wird auch der Hauptunterschied zwischen den beiden Verfahren deutlich. Während das Modul Behaviour Decision Entscheidungen über elementare Fahrmanöver fällt, d. h. eine *diskrete* Entscheidung liefert, sind die Steuervorgaben des Verhaltensmusterkonzepts *kontinuierlich*. Der Vorteil einer Beschränkung auf elementare Fahrmanöver ist eine bessere Optimierbarkeit einzelner Aktionen. Der Nachteil ist eine geringere Anpassungsfähigkeit an komplexere dynamische Verkehrsszenarien, wie sie durch eine kontinuierliche Steuerung möglich ist.

5.2.2 Verwandte Arbeiten

Ähnliche Themengebiete unterteilen sich in hauptsächlich in drei Gruppen. Während sich die ersten durch eine weniger operationale Zielsetzung unterscheiden und hauptsächlich off-line konzipiert und eingesetzt werden, entstammen die anderen einem unterschiedlichen Einsatzgebiet.

Bei ersteren handelt es sich um Verfahren zur Analyse der umgebenden Verkehrssituation mit Hilfe verschiedener klassischer Wissensrepräsentationen. Als Vertreter sind hier die Arbeiten von Mock-Hecker zur wissensbasierten Erkennung von Plankonflikten (vgl. [Mock-Hecker 94]) und die Arbeiten von Krüger und Nagel (vgl. [Krüger 91, Kollnig & Nagel 93]) zu erwähnen.

Charakteristisch für Mock-Heckers Ansatz zur Plankonflikterkennung sind Verhaltensmodelle zur qualitativen Beschreibung der Pläne in der Verkehrswelt (Überholen etc.) und Bewegungsmodelle zur quantitativen Bestimmung der Bewegungsabläufe der zugehörigen Einzelaktionen.

Krüger stellt eine Modellierung von Verkehrssituationen anhand von Situationsgraphen vor. Dynamische Szenen werden über eine Folge von Zuständen beschrieben. Jeder Knoten des Graphen besteht aus einem Zustandsschema und einem Aktionsschema, das eine Aktion beschreibt, welche das System im vorliegenden Zustand durchführen soll.

Kollnig und Nagel befassen sich mit der Beschreibung von dynamischen Verkehrsgeschehen auf der Basis von Verbalphrasen, die aus Bildsequenzen generiert werden.

Weitere verwandte Arbeiten sind im Bereich der autonomen mobilen Roboter zu finden. Insbesondere sind hier noch einmal die Arbeiten von Brooks zu erwähnen (vgl. [Brooks 86]), der eine Architektur eines Robotersteuerungssystems vorstellt, die sich aus aufgabenerfüllenden Verhaltensweisen zusammensetzt. In dieser Architektur werden Sensoren und Aktoren durch eine mit ihnen zusammenhängende Verhaltensweise eng miteinander verknüpft, so daß eine schnelle Reaktion auf Eingangsdaten möglich wird.

Zu einem weiteren Ansatz zur Steuerung von Robotern, der auf dynamischen Feldern beruht (vgl. [Schöner & Dose 92]), wurde eine Erweiterung für den Einsatz im Autobahnumfeld vorgestellt (vgl. [Werner & Engels 95]). Information wird hier als Aktivierung in einem dynamischen Feld interpretiert und die Steuervorgaben werden aus den Attraktoren eines generierten Ausgabefeldes bestimmt.

Die hier erwähnten Arbeiten stellen nur einen kleinen Ausschnitt der Tätigkeiten dar, die eine gewisse Verwandtschaft mit dem Thema der vorliegenden Arbeit aufweisen.

5.3 Autonome Fahrzeuge im Straßenverkehr von morgen

Betrachtet man den aktuellen Stand der Forschung auf dem Gebiet der autonomen Fahrzeuge, so stellt sich die Frage, in welcher Form sich autonome Systeme im Straßenverkehr der Zukunft wiederfinden werden. Aus den Betrachtungen der Probleme, die mit einem vollständig autonomen Fahrzeug verbunden sind, geht hervor, daß sicherlich in naher Zukunft kein solches System als kommerzielles Produkt erhältlich sein wird. Liegt die Kommerzialisierung des fahrerlosen Automobils auch noch in ferner Zukunft, so können Vorstufen davon bereits in bekannter Einsatzumgebung (z. B. Baustellen, Fabrikhalle) mit kalkulierbaren Randbedingungen zum Einsatz kommen. Eine weitere Einsatzmöglichkeit existiert für informierende, warnende und teilautomatisierte Systeme, die als sogenannte „Spin-Offs“ aus Teilthemen des autonomen Fahrzeugs abgeleitet werden. Hier sind folgende Anwendungen denkbar:

- Ein *Verkehrszeichenerkennungssystem* liefert dem Fahrer Informationen über die aktuell geltenden Geschwindigkeitsbeschränkungen (Erinnerung an das letzte gültige Verkehrszeichen). Eine Warnung bei Überschreiten der Begrenzung ist möglich. In Kombination mit der Serientempomatfunktion ist sogar ein automatisches Eingreifen zur Einhaltung der Geschwindigkeitsbeschränkung denkbar.
- Ein *Hinderniserkennungssystem* liefert Informationen über vorausfahrende Fahrzeuge. Auch hier ist in Kombination mit der Tempomatfunktion ein

automatisches Eingreifen des Rechners in Gas und Bremse zur Abstandsregelung (*AICC - Autonomous Intelligent Cruise Control*) vorstellbar. Ein ähnliches System kann zur *Überwachung der toten Winkel* eingesetzt werden und den Fahrer durch geeignete Information oder Warnung unterstützen.

- Ein *Straßenerkennungssystem* kann durch die Information über die Position des Fahrzeugs zu einer Stabilisierung der Steuerung des Fahrzeugs führen und somit den Fahrer von der Feinregelaufgabe befreien [Franke et. al 94].

Diese Themen werden zur Zeit von verschiedenen Forschungsgruppen bearbeitet. Für diese Teilbereiche sind in den nächsten Jahren Produkte zu erwarten. Vor dem Hintergrund eines immer größer werdenden Verkehrsaufkommens beschäftigen sich mehrere Forschungsgruppen mit der Durchsatzoptimierung. Diese basiert auf einer elektronischen Längsführung (in manchen Fällen auch auf einer Querführung) mehrerer Fahrzeuge (hauptsächlich LKW). Durch eine präzise elektronische Längsregelung können Abstände von bis zu fünf Metern verwirklicht werden. Während dies in den USA und auch in Japan mit Hilfe separater, mit unterstützender Infrastruktur ausgerüsteter Fahrspuren konzipiert wird, ist solch ein infrastruktureller Aufwand in Europa kaum denkbar. Hiersche führt dazu in einer kritischen Analyse insbesondere bauliche und betriebliche Aspekte solcher automatisierter Autobahnen an, hebt aber auch die verkehrlichen Aspekte wie das Verlassen und Auffahren sowie sicherheitstechnische Aspekte hervor [Hiersche 90]. Ein anderes Konzept des „Platooning“¹ verfolgt eine Forschungsgruppe der Daimler-Benz AG [Franke et al. 95]. Dieses Konzept beschreibt eine Form der Zugbildung von LKW's im gemischten Verkehr, die mit dem heutigen Straßenverkehr kompatibel ist, d. h. keine zusätzliche Infrastruktur benötigt. Das erste Fahrzeug wird manuell gefahren, während die folgenden Fahrzeuge automatisch geführt werden.

Neben der technologischen Reife solcher teilautomatisierter Systeme sind vor deren Einführung noch einige andere Aspekte zu berücksichtigen. Auch wenn das technische System selbst den Anforderungen entspricht, bleibt die Problematik einer geeigneten *Mensch-Maschine Schnittstelle*. Insbesondere bei teilautomatisierten Systemen muß sichergestellt sein, daß es dem Fahrer keine Probleme bereitet, zwischen dem autonomen und dem konventionellen Fahrmodus umzuschalten, d. h. die Kontrolle über das Fahrzeug abzugeben und wieder zu übernehmen. Erste Untersuchungen dazu sind in Form einer Simulatorstudie von der Daimler-Benz Forschung durchgeführt worden [Hahn 93].

¹ Platooning (engl.) = Zugbildung, hier: automatische Führung von Konvois mehrerer Fahrzeuge

Ein weiterer wichtiger Punkt ist die *Kundenakzeptanz*, die vom Standpunkt der Automobilhersteller vor allem im Zusammenhang mit der Preisfindung für diese Systeme zu sehen ist. Dazu kommen versicherungstechnische und juristische Fragestellungen (u. a. die Produkthaftung des Herstellers), die sich insbesondere bei eingreifenden Systemen, aber auch bei informierenden und warnenden Systemen ergeben.

Die bisherigen Ausführungen beschränken sich auf die (technologische) Machbarkeit und Randbedingungen der Einführung von autonomen und teilautonomen Systemen im Straßenverkehr. Die Auswirkungen auf den Straßenverkehr werden dabei nicht berücksichtigt. Ziel der Einführung technischer Systeme zur Unterstützung des Fahrers ist vor allem die Erhöhung des Fahrkomforts sowie die Unfallvermeidung. Der Erreichbarkeit dieses Ziels widerspricht das von Wilde aufgestellte „Gesetz der Erhaltung der Unfallzahl“, das wie folgt formuliert ist: *„Die Zahl der Unfälle in einem bestimmten Land hängt ausschließlich von der Unfallzahl ab, die die Bevölkerung bereit ist zu tolerieren, und nicht von den Maßnahmen in anderen Bereichen in diesem Kontrollsystem, jedenfalls nicht auf längere Dauer“* (vgl. [Huguenin 88]). Diese These wird zwar kritisiert, kann aber aufgrund ihrer allgemeinen Formulierung weder falsifiziert noch bewiesen werden. Unbestritten bleibt die mögliche Erhöhung des Fahrkomforts durch autonome Teilsysteme. Die dadurch erreichte Entlastung des Fahrers darf jedoch nicht dazu führen, daß dieser in Notsituationen einer umso höheren Belastung ausgesetzt wird (vgl. [Panik 87]). Bei genauer Betrachtung wird klar, daß Probleme nicht vom technischen System selbst ausgehen², sondern vielmehr vom Zusammenwirken von Mensch und Maschine. In seinen Ausführungen über die Zukunft des Verkehrs sieht Vester das Problem einer Automatisierung darin, daß sich der Mensch auf das technische System verläßt und damit seine objektive Sicherheit durch ein gesteigertes Sicherheitsgefühl stärker reduziert, als diese durch die technische Unterstützung angehoben wird [Vester 95].

Zusammenfassend bleibt eine Aussicht auf einen Straßenverkehr der nahen und fernen Zukunft, der eine wachsende Zahl von automatisierten und autonomen Teilsystemen in Fahrzeugen mit sich bringen wird. Vor dem Hintergrund wachsenden Verkehrsaufkommens und dem Bedürfnis des Menschen nach Mobilität werden neben der Automatisierung weitere Wege zur Lösung dieses Konflikts und der damit zusammenhängenden Probleme beschrritten werden müssen.

² Dabei wird davon ausgegangen, daß eine sichere Realisierung der spezifizierten Aufgaben des technischen Systems vorhanden ist.

ANHANG

A Transputer

Die Echtzeitanforderungen der Prozesse in einem autonomen Fahrzeug erfordern eine Hardwareplattform, die eine große Rechenleistung zur Verfügung stellt. Da sich die Anwendung leicht in mehrere, parallel ausführbare Module unterteilen läßt, bietet sich der Transputer als Zielhardware an.

Aus Transputern lassen sich leicht Systeme mit verschiedenen Verbindungsstrukturen aufbauen, so daß die Architektur des Rechnersystems gut der Problemstellung angepaßt werden kann. Im Folgenden wird eine kurze Beschreibung der Transputer gegeben. Einleitend wird in die Parallelverarbeitung eingeführt und die Eingliederung der Transputer in Flynns Taxonomie der Rechnerarchitekturen gegeben, bevor schließlich auf die Eigenschaften von Transputern eingegangen wird.

A.1 Parallelverarbeitung

Die Ausführung zweier Anweisungen wird als *parallel* bezeichnet, wenn es einen Zeitpunkt gibt, zu dem beide Anwendungen gleichzeitig auf verschiedenen Prozessoren bearbeitet werden [Umland & Vollmar 92]. Durch die Parallelisierung von Algorithmen wird eine Verringerung der Rechenzeit erreicht, die mit der Zahl der parallel betriebenen Prozessoren steigen sollte. Die Effizienz der Parallelisierung ist jedoch stark vom jeweiligen Algorithmus abhängig.

Zur Unterscheidung von Rechnertypen wird in vielen Fällen die Taxonomie von Flynn verwendet [Hwang & Briggs 89]. Flynn unterscheidet zwischen einfachen und mehrfachen (multiplen) Daten- und Befehlsströmen. Anhand dieser Unterscheidung ergeben sich die folgenden vier grundlegenden Rechnerarchitekturen:

- Single Instruction Stream, Single Data Stream (SISD),
- Multiple Instruction Stream, Single Data Stream (MISD),
- Single Instruction Stream, Multiple Data Stream (SIMD) und
- Multiple Instruction Stream, Multiple Data Stream (MIMD).

Zur Gruppe der SISD-Rechner gehören die herkömmlichen sequentiellen Rechner mit von-Neumann-Architektur. Eine Einsatzmöglichkeit der sonst sehr seltenen vertretenen MISD-Rechnerkategorie ist durch fehlertolerante Systeme gegeben. In einem SIMD-Rechner arbeiten mehrere Prozessoren gleichzeitig auf verschiedenen Daten den gleichen Befehl ab. Diese Rechnerarchitekturen werden auch als Array-Rechner bezeichnet. MIMD-Rechner werden auch als asynchrone Parallelrechner bezeichnet. In dieser

Architektur arbeiten verschiedene Prozessoren unterschiedliche Programme auf unterschiedlichen Datenbereichen ab. Man untergliedert die MIMD-Rechner nach der Art ihrer Kopplung. Besitzen die Prozessoren einen gemeinsamen, globalen Speicher, so spricht man von einer starken Kopplung. Im Gegensatz dazu ist eine schwache Kopplung gegeben, wenn die Prozessoren über lokale Speicher verfügen und über Datenleitungen verbunden sind. Zu dieser Kategorie gehören auch die Transputer.

A.2 Aufbau eines Transputers

Bei einem Transputer handelt es sich um einen Mikroprozessor der mit besonderen Erweiterungen zur Unterstützung von Parallelverarbeitung versehen ist. Der Aufbau eines Transputers wird im Folgenden anhand eines Inmos T800 beschrieben.

Zur bidirektionalen Kommunikation mit anderen Transputern ist der T800 mit vier Links ausgerüstet. Es handelt sich dabei um eigenständige Kommunikationseinheiten, die parallel zum Prozessor arbeiten und eine Datenübertragungsleistung von 20 Mbit/s leisten. Die Datenübertragung wird nach einem byteweisen Handshake-Verfahren abgewickelt. Neben diesen Komponenten verfügt der Transputer über effiziente Hardware zur Prozeßumschaltung und unterstützt damit nebenläufige Prozesse, die sich den Prozessor in einem Zeitscheibenverfahren teilen. Bild 1 zeigt den Aufbau eines Transputers des Typs T800.

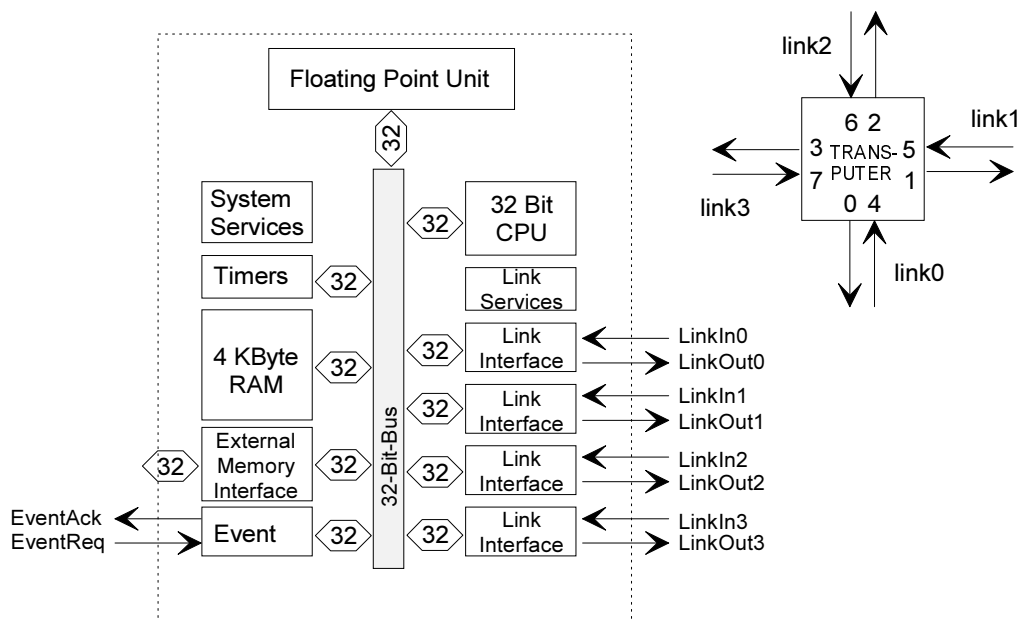


Bild 1: Blockschaltbild eines Transputers T800

Bei den im Versuchsfahrzeug VITA II verwendeten Transputern handelt es sich um die Transputertypen T805 und T222. Der T805 ist eine Weiterentwicklung des T800, die sich im wesentlichen bei der Behandlung externer Events von ihrem Vorgänger

unterscheidet. Der T222-Transputer besitzt eine geringere Wortbreite (16 Bit) und keine Gleitkommaeinheit.

Betrachtet man einen Transputer als einen Baustein, so lassen sich durch Verbindung der Transputerlinks verschiedene Topologien eines Multiprozessorsystems konfigurieren. Diese Verbindungen können durch feste Leitungen oder über einen Kreuzschienenverteiler realisiert werden, durch den alle Links miteinander verknüpft werden können. Inmos stellt dafür einen programmierbaren Kreuzschienenverteiler mit 32 Ein- und Ausgängen zur Verfügung.

A.3 Programmieren von Transputern

Die Standardprogrammiersprache für Transputer ist OCCAM. Sie wurde zusammen mit den Transputern von Inmos entwickelt. Die Sprache ist angelehnt an Hoares CSP Konzept (communicating sequential processes). Nebenläufige Programme werden aus sequentiellen Prozessen zusammengesetzt, die miteinander kommunizieren.

Mittlerweile sind auch Erweiterungen der Programmiersprache C erhältlich, mit denen Transputernetzwerke programmierbar sind. Der Umfang der Programmiersprache selbst erweitert sich dadurch nur unwesentlich. Allerdings stellt die Konzeption von Programmen, die aus mehreren nebenläufigen Prozessen bestehen, höhere Anforderungen an den Programmierer. Auch die Abbildung eines Programms auf eine Hardwarekonfiguration gestaltet sich bei größeren Prozessornetzen recht schwierig.

Um komplexe Systeme beherrschen zu können, wurde für Transputer ein Werkzeug namens TRAPPER [Schäfers et al. 92] entwickelt. Die TRAPPER Entwicklungsumgebung beinhaltet einen graphischen Editor für Software- und Hardwaregraphen des komplexen Systems. Bild 2 zeigt einen mit TRAPPER erzeugten Softwaregraphen. Jedes Kästchen stellt einen Prozess dar, jede Verbindungslinie einen Link zwischen zwei Prozessen. Zum Laden des ausführbaren Programmcodes, sowie zur Ein-/Ausgabe wird die Verbindung zu einem Wirtsrechner (host) verwendet. In VITA II wird dazu eine SUN Workstation unter UNIX benutzt.

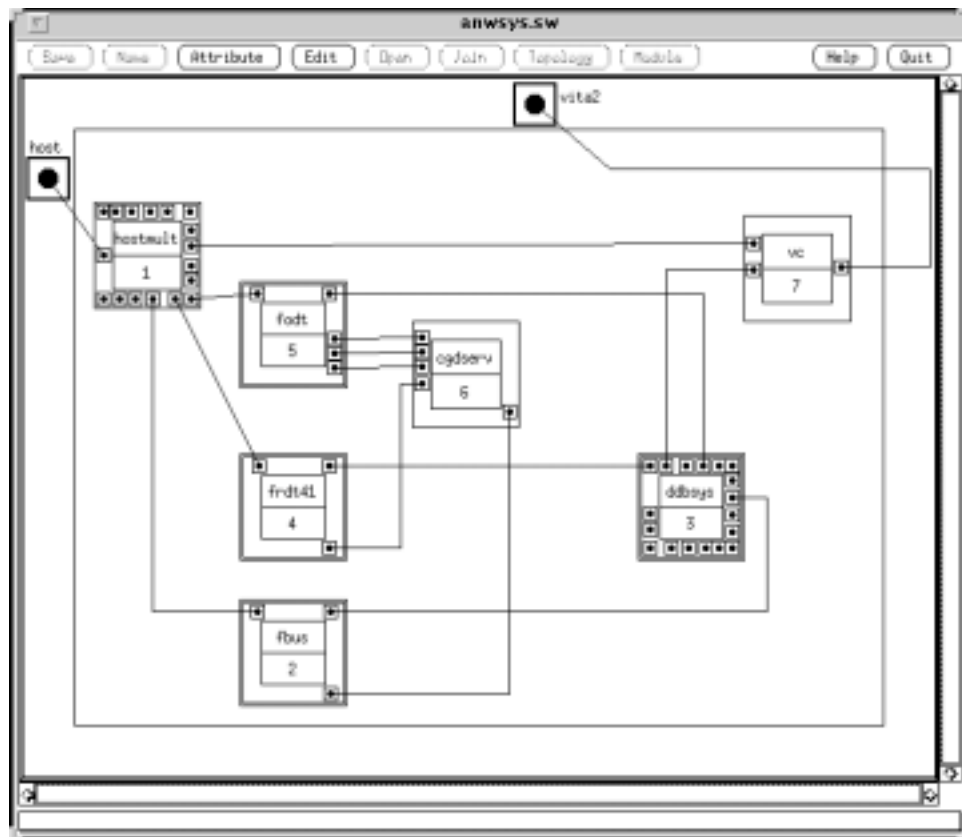


Bild 2 : Beispiel eines Softwaregraphen

Weiterhin unterstützt es den Programmierer bei der Abbildung der Software- auf die Hardwarestrukturen und stellt Visualisierungsmöglichkeiten für das Systemverhalten zur Verfügung. Für die Erstellung und Handhabung des VITA II Rechnersystems mit ca. 70 Prozessoren und über 200 parallelen Prozessen erwies sich TRAPPER als hervorragende Unterstützung.

B Verkehrszeichenkodierung

Die nachfolgende Tabelle zeigt die im SA-Modul verarbeiteten Verkehrszeichen mit ihren Codes und Zusatzinformationen. Das in VITA II integrierte Modul zur Verkehrszeichen-erkennung ist in der Lage, wesentlich mehr als die unten aufgeführten Zeichen zu erkennen. Diese Zeichen werden jedoch vom Kontrollsystem nicht berücksichtigt.











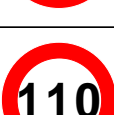
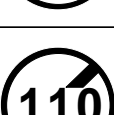


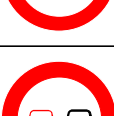



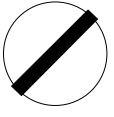
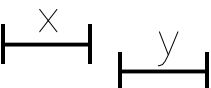
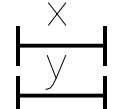
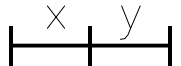
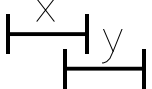
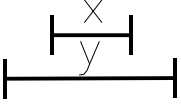
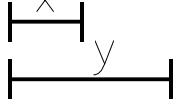
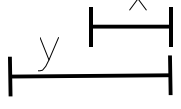
	VzKode: 274 Zusatz : 50		VzKode: 274 Zusatz : 50
	VzKode: 274 Zusatz : 60		VzKode: 274 Zusatz : 60
	VzKode: 274 Zusatz : 80		VzKode: 278 Zusatz : 80
	VzKode: 274 Zusatz : 90		VzKode: 278 Zusatz : 90
	VzKode: 274 Zusatz : 100		VzKode: 278 Zusatz : 100
	VzKode: 274 Zusatz : 110		VzKode: 278 Zusatz : 110
	VzKode: 274 Zusatz : 130		VzKode: 278 Zusatz : 130
	VzKode: 276 Zusatz :-		VzKode: 280 Zusatz :-
	VzKode: 277 Zusatz :-		VzKode: 281 Zusatz :-
			VzKode: 282 Zusatz :-

Bild 3 : Verarbeitete Verkehrszeichen

C Zeitrepräsentation

C.1 Allen'sche Zeitrelationen

Eine Repräsentation von Zeit wird von J. F. Allen eingeführt [Allen 84]. Diese enthält die axiomatische Behandlung von Zeitintervallen, die auch als Allen's Zeitlogik bezeichnet wird. Allen verwendet 13 Relationen um Beziehungen zwischen Intervallen darzustellen. Diese Relationen sind in nachfolgender Tabelle aufgeführt (entnommen aus [Richter 89]).

Relation	Symbol	Inverses	Veranschaulichung
x vor y	<	>	
x gleich y	=	=	
x an y	m	mi	
x überlagert y	o	oi	
x während y	d	di	
x startet y	s	si	
x beendet y	f	fi	

Die Symbole entstammen Abkürzungen der Relationen aus dem Englischen: meets, overlaps, during, starts und finishes. Bezeichnen $A(x)$, $E(x)$ den Anfangs- und Endpunkt eines Intervalls x , so lassen sich die Intervallrelationen durch Punktrelationen darstellen, wie folgende Tabelle zeigt. Die inversen Relationen erhält man durch Vertauschen von x und y .

Intervallbeschreibung	Punktbeschreibung
$x < y$	$E(x) < A(y)$
$x = y$	$\hat{A}(x) = A(y) \wedge E(x) = E(y)$
$x m y$	$E(x) = A(y)$
$x o y$	$A(x) < A(y) \wedge A(y) < E(x) \wedge E(x) < E(y)$
$x d y$	$A(y) < A(x) \wedge E(x) < E(y)$
$x s y$	$A(x) = A(y) \wedge E(x) < E(y)$
$x f y$	$A(y) < A(x) \wedge E(x) = E(y)$

C.2 Relationen für Intervallenden unscharfer Intervalle

Die nachfolgend beschriebenen unscharfen Relationen sind vergleichbar mit den Punktrelationen $<$, $>$ und $=$. Durch Übertragung dieser Punktrelationen auf Relationen für Intervallenden unscharfer Intervalle wird eine Anwendung der Allen'schen Intervallrelationen auf diese unscharfe Intervalle ermöglicht.

Im Folgenden werden Konfidenzfunktionen für Intervallenden der unscharfen Intervalle $I_1 = (A_1, E_1)$ und $I_2 = (A_2, E_2)$ definiert.

Anfangsintervall-Anfangsintervall

$$C(A_1 <_{AA} A_2) = \begin{cases} 1 & \text{für } A_1 \{<\} A_2 \\ 0 & \text{für } A_1 \{>\} A_2 \\ \frac{1}{2} \left(1 + \frac{1}{\|A_1 \cap A_2\|} \cdot \int_{t \in A_1 \cap A_2} (\mu_{A_1}(t) - \mu_{A_2}(t)) dt \right) & \text{sonst} \end{cases}$$

$$C(A_1 =_{AA} A_2) = \begin{cases} 0 & \text{für } A_1 \cap A_2 = \emptyset \\ 1 - \frac{1}{\|A_1 \cap A_2\|} \cdot \int_{t \in A_1 \cap A_2} |\mu_{A_1}(t) - \mu_{A_2}(t)| dt & \text{sonst} \end{cases}$$

$$C(A_1 >_{AA} A_2) = \begin{cases} 0 & \text{für } A_1 \{<\} A_2 \\ 1 & \text{für } A_1 \{>\} A_2 \\ \frac{1}{2} \left(1 - \frac{1}{\|A_1 \cap A_2\|} \cdot \int_{t \in A_1 \cap A_2} (\mu_{A_1}(t) - \mu_{A_2}(t)) dt \right) & \text{sonst} \end{cases}$$

Anfangsintervall-Endintervall

$$C(A_1 <_{AE} E_2) = \begin{cases} 1 & \text{für } A_1 \{<\} E_2 \\ 0 & \text{für } A_1 \{>\} E_2 \\ \frac{1}{2} \left(1 + \frac{1}{\|A_1 \cap E_2\|} \cdot \int_{t \in A_1 \cap E_2} (\mu_{A_1}(t) + \mu_{E_2}(t) - 1) dt \right) & \text{sonst} \end{cases}$$

$$C(A_1 =_{AE} E_2) = \begin{cases} 0 & \text{für } A_1 \cap E_2 = \emptyset \\ 1 - \frac{1}{\|A_1 \cap E_2\|} \cdot \int_{t \in A_1 \cap E_2} |\mu_{A_1}(t) - \mu_{E_2}(t) - 1| dt & \text{sonst} \end{cases}$$

$$C(A_1 >_{AE} E_2) = \begin{cases} 0 & \text{für } A_1 \{<\} E_2 \\ 1 & \text{für } A_1 \{>\} E_2 \\ \frac{1}{2} \left(1 - \frac{1}{\|A_1 \cap E_2\|} \cdot \int_{t \in A_1 \cap E_2} (\mu_{A_1}(t) + \mu_{E_2}(t) - 1) dt \right) & \text{sonst} \end{cases}$$

Endintervall-Anfangsintervall

$$C(E_1 <_{EA} A_2) = \begin{cases} 1 & \text{für } E_1 \{<\} A_2 \\ 0 & \text{für } E_1 \{>\} A_2 \\ \frac{1}{2} \left(1 + \frac{1}{\|E_1 \cap A_2\|} \cdot \int_{t \in E_1 \cap A_2} (1 - \mu_{E_1}(t) - \mu_{A_2}(t)) dt \right) & \text{sonst} \end{cases}$$

$$C(E_1 =_{EA} A_2) = \begin{cases} 0 & \text{für } E_1 \cap A_2 = \emptyset \\ 1 - \frac{1}{\|E_1 \cap A_2\|} \cdot \int_{t \in E_1 \cap A_2} |1 - \mu_{E_1}(t) - \mu_{A_2}(t)| dt & \text{sonst} \end{cases}$$

$$C(E_1 >_{EA} A_2) = \begin{cases} 0 & \text{für } E_1 \{<\} A_2 \\ 1 & \text{für } E_1 \{>\} A_2 \\ \frac{1}{2} \left(1 - \frac{1}{\|E_1 \cap A_2\|} \cdot \int_{t \in E_1 \cap A_2} (1 - \mu_{E_1}(t) - \mu_{A_2}(t)) dt \right) & \text{sonst} \end{cases}$$

Endintervall-Endintervall

$$C(E_1 <_{EE} E_2) = \begin{cases} 1 & \text{für } E_1 \{<\} E_2 \\ 0 & \text{für } E_1 \{>\} E_2 \\ \frac{1}{2} \left(1 + \frac{1}{\|E_1 \cap E_2\|} \cdot \int_{t \in E_1 \cap E_2} (\mu_{E_2}(t) - \mu_{E_1}(t)) dt \right) & \text{sonst} \end{cases}$$

$$C(E_1 =_{EE} E_2) = \begin{cases} 0 & \text{für } E_1 \cap E_2 = \emptyset \\ 1 - \frac{1}{\|E_1 \cap E_2\|} \cdot \int_{t \in E_1 \cap E_2} |\mu_{E_2}(t) - \mu_{E_1}(t)| dt & \text{sonst} \end{cases}$$

$$C(E_1 >_{EE} E_2) = \begin{cases} 0 & \text{für } E_1 \{<\} E_2 \\ 1 & \text{für } E_1 \{>\} E_2 \\ \frac{1}{2} \left(1 - \frac{1}{\|E_1 \cap E_2\|} \cdot \int_{t \in E_1 \cap E_2} (\mu_{E_2}(t) - \mu_{E_1}(t)) dt \right) & \text{sonst} \end{cases}$$

D Literatur

- [Allen 84] J. F. Allen, „Towards a General Theory of Action and Time”, *Artificial Intelligence* 23, S. 123-154, 1984
- [Arkin 87] R. C. Arkin, „Motor Schema Based Navigation for a Mobile Robot: An Approach to Programming by Behaviour”, in: *International Conference on Robotics and Automation, IEEE*, April 1987, pp. 264-271
- [Auerswald 96] M. Auerswald, „Analyse von Verkehrssituationen mit Hilfe eines expliziten schemabasierten Fehlermodells”, Projektarbeit, Universität Kaiserslautern, Fachbereich Informatik, durchgeführt bei der Daimler-Benz AG, F1M/IA, Esslingen, 1996
- [Behringer 94] R. Behringer, „Road Recognition from Multifocal Vision”, in: *Proceedings of the Intelligent Vehicles Symposium, Paris, 1994*
- [Behringer 95] R. Behringer, „Detection of Discontinuities of Road Curvature Change by GLR”, in: *Proceedings of the Intelligent Vehicles Symposium, Detroit 1995*
- [Besserer et al. 93] B. Besserer, S. Estable, B. Ulmer, D. Reichardt, „Shape Classification for Traffic Sign Recognition”, *1st IFAC International Workshop on Intelligent Autonomous Vehicles, Southampton, 1993*
- [Beste 95] D. Beste, „Prometheus auf der Autobahn”, in: *Spektrum der Wissenschaft, Dossier: Verkehr und Auto, Spektrum der Wissenschaft Verlagsgesellschaft mbH, August 1995*
- [Bibel 94] W. Bibel, „Wissensrepräsentation und Inferenz”, Vieweg Verlag, 1994
- [Bohrer et al. 95] S. Bohrer, T. Zielke, V. Freiburg, „An Integrated Obstacle Detection Framework for Intelligent Cruise Control on Motorways”, in: *Proceedings of the Intelligent Vehicles Symposium, Detroit, 1995*
- [Bösch 91] P. Bösch, „Der Fahrer als Regler”, Dissertation, Fakultät für Maschinenbau, Technische Universität Wien, 1991
- [Brammer & Siffling 85] K. Brammer, G. Siffling, „Kalman-Bucy-Filter - Deterministische Beobachtung und stochastische Filterung”, 4. verbesserte Auflage, Oldenbourg Verlag GmbH, München, 1994
- [Brauckmann et al. 94] M.E. Brauckmann, C. Goerick, J. Groß, T. Zielke, „Towards All Around Automatic Visual Obstacle Sensing for Cars”, in: *Proceedings of the Intelligent Vehicles Symposium, Paris, 1994*
- [Brooks 86] R.A. Brooks, „A Robust Layered Control System for a Mobile Robot”, *IEEE Transactions on Robotics and Automation*, Vol. RA-2, No. 1, 1986, pp. 14-23
- [Chattergy 85] R. Chattergy, „Some Heuristics for the Navigation of a Robot”, in: *The International Journal of Robotics Research*, Volume 4, No. 1, Spring 1985, pp. 59-66
- [Crowley 85] J. L. Crowley, „Navigation for an Intelligent Mobile Robot”, in: *IEEE Journal of Robotics and Automation*, Volume RA-1, No. 1, March 1985, pp. 31-41
- [Dickmanns 93a] D. Dickmanns, „The Dynamic Database User's Guide”, PROMETHEUS PRO-ART Abschlußbericht, Informatik-Forschungsgruppe, UniBw München, Germany, 1994
- [Dickmanns 93b] D. Dickmanns, „Elemente einer Wissensbank für Bildverarbeitung: Bildmeßsystem KRONOS, Objektrepräsentation, Dynamische Datenbasis DDB”, PROMETHEUS PRO-ART Abschlußbericht, Informatik-Forschungsgruppe, UniBw München, Germany, 1994

- [Dickmanns et al. 94] E. D. Dickmanns, R. Behringer, D. Dickmanns, T. Hildebrandt, F. Thomanek, J. Schielen, „The Seeing Passenger Car ‘VaMoRs-P’ ”, in: Proceedings of the Intelligent Vehicles Symposium, Paris, 1994
- [Estable et al. 94] S. Estable, J. Schick, F. Stein, R. Janssen, R. Ott, W. Ritter, Y.-J. Zheng, „A Real-Time Traffic Sign Recognition System”, in: Proceedings of the Intelligent Vehicles Symposium, Paris, 1994
- [Franke 89] U. Franke, „PROMETHEUS - Wissensbasierte Systeme eröffnen neue Perspektiven im Straßenverkehr”, 3. Internationaler GI-Kongreß, München, Oktober 1989, Informatik Fachberichte 227, Springer Verlag, 1987, S. 363-376
- [Franke et al. 94] U. Franke, S. Mehring, A. Suissa, S. Hahn, „The Daimler-Benz Steering Assistant - a Spin-off from Autonomous Driving”, in: Proceedings of the Intelligent Vehicles Symposium, Paris, 1994
- [Franke et al. 95] U. Franke, F. Böttiger, Z. Zomotor, D. Seeberger, „Truck Platooning in Mixed Traffic”, in: Proceedings of the Intelligent Vehicles Symposium, Detroit 1995
- [Gachelin 94] O. Gachelin, „Implantation sur Transputers d’un systeme de guidage pour vehicule autonome et intégration des panneaux de signalisation routière”, Mémoire en vue de l’obtention du titre d’ingénieur, C.U.S.T Université Blaise Pascal, Clermont Ferrand 2, ets: Daimler-Benz AG, 1994
- [Hagen 93] M. Hagen, „Echtzeit-Bildverarbeitung von Verkehrsszenen zur Positionsbestimmung von Objekten mittels Bewegungstereo”, Diplomarbeit, Berufsakademie Stuttgart, Fachbereich Technische Informatik, 1993
- [Hahn 93] S. Hahn, „Switching between Autonomous and Conventional Car Driving - A Simulator Study”, in: Proceedings of the Intelligent Vehicles Symposium, Tokyo, 1993
- [HAV 87] „Hinweise für das Anbringen von Verkehrszeichen und Verkehrseinrichtungen”, Kirschbaum Verlag, 1987
- [Heichel 95] M. Heichel, „Zeitschnittverfahren zur Hinderniserkennung in Straßenfahrzeugen”, Diplomarbeit, Universität Karlsruhe, Institut für Algorithmen und Kognitive Systeme der Fakultät für Informatik, durchgeführt in der Forschungsabteilung F1M/IA der Daimler-Benz AG in Esslingen, September 1995
- [Herzberg 89] J. Herzberg, „Planen - Einführung in die Planerstellungsmethoden der Künstlichen Intelligenz”, Reihe Informatik, Band 65, Hrsg. K. H. Böhling, U. Kulisch, H. Maurer, B.I. Wissenschaftsverlag, Mannheim/Wien/Zürich, 1989
- [Hiersche 90] E. -U. Hiersche, „Automatische Fahrzeugführung auf Autobahnen ?”, in: Automobil-Industrie 2/1990, S. 199-204
- [HighTechReport 94] DaimlerBenz High Tech Report 4/94 special PROMETHEUS, Daimler-Benz AG, Stuttgart, 1994
- [Huguenin 88] R. D. Huguenin, „Fahrerverhalten im Straßenverkehr”, Reihe: Faktor Mensch im Verkehr 37, Herausgeber: Forschungsgemeinschaft „Der Mensch im Verkehr“ e.V., Köln, Rot-Gelb-Grün Verlag, Braunschweig, 1988
- [Hwang & Briggs 89] K. Hwang, F. A. Briggs, „Computer Architecture and Parallel Processing”, International Edition, 5.Auflage, McGraw-Hill, 1989, ISBN 0-07-031556-6
- [Johannsen 90] G. Johannsen, „Fahrzeugführung”, in: Enzyklopädie der Psychologie, Themenbereich D, Praxisgeb. Serie 3. Wirtschafts-, Organisations- u. Arbeitspsychologie; Band 2 Ingenieurpsychologie, Verlag für Psychologie, 1990, ISBN 3-8017-0519-6

- [Jörg 91] K. W. Jörg,, „Echtzeitfähige Multisensor-Integration für autonome mobile Roboter“, Reihe Informatik, Band 91, B.I. Wissenschaftsverlag, Hrsg. K.H. Böhling, U. Kulisch, H.Maurer, 1991
- [Kämmler 93] A. Kämmler, „Kommunikationsstruktur von Parallelprozessen zur autonomen Fahrzeugführung“, Diplomarbeit, Institut für Informatik, Universität Stuttgart, 1993
- [Khatib 85] O. Khatib, „Real-Time Obstacle Avoidance for Manipulators and Mobile Robots“, in: Proceedings of the Conference on Robotics and Automation, 85CH2152-7, IEEE, New York, March 1985
- [Knieriemen 91] T. Knieriemen, „Autonome mobile Roboter - Sensordateninterpretation und Weltmodellierung zur Navigation in unbekannter Umgebung“, BI-Wissenschaftsverlag, Reihe Informatik, Band 80, ISBN 3-411-15031-9, 1991
- [Kollnig & Nagel 93] H. Kollnig, H.-H. Nagel, „Ermittlung von begrifflichen Beschreibungen von Geschehen in Straßenverkehrsszenen mit Hilfe unscharfer Mengen“, Informatik Forschung und Entwicklung 8, Springer-Verlag, Heidelberg, 1993
- [Krogh 83] B. H. Krogh, „Feedback Obstacle Avoidance Control“, in: Proceedings of the 21st Allerton Conference on Communication, Control and Computing, Urbane Il., October 1983
- [Krogh 84] B. H. Krogh, „A Generalized Potential Field Approach to Obstacle Avoidance Control“, in: Robotics Research: The Next Five Years and Beyond, SME Conference Proceedings, Bethlehem, PA, August 1984
- [Krogh & Thorpe 86] B. H. Krogh, C. H. Thorpe, „Integrated Path Planning and Dynamic Steering Control for Autonomous Vehicles“, in: Proceedings of the International Conference on Robotics and Automation, IEEE, 1986
- [Krüger 91] W. Krüger, „Situationsgraphen zur Modellierung von Verkehrssituationen“ FhG-Berichte 1/1991, München, 1991, S. 35-41
- [Kujawski 93] C. Kujawski, „Der Modul BEHAVIOUR DECISION“, in: PROMETHEUS PRO-ART Abschlußbericht, Informatik-Forschungsgruppe, UniBw München, Germany, 1993
- [Kujawski 95] C. Kujawski, „Deciding the Behaviour of an Autonomous Mobile Road Vehicle“, in: 2nd IFAC Conference on Intelligent Autonomous Vehicles (IAV) 1995, Helsinki University of Technology, Espoo, Finland, June 1995
- [Lozano-Pérez & Wesley 79] T. Lozano-Pérez, M.A. Wesley, „An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles“, Communications of the ACM, Vol. 22, No.10, 1979
- [Manigel 93] J. Manigel, „Autonome Fahrzeugführung durch Rechnersehen“, Dissertation, Fakultät für Maschinenbau und Elektrotechnik der Technischen Universität Carolo-Wilhelmina zu Braunschweig, Januar 1993
- [Mezger 88] K. Mezger, „Automatische Führung von Binnenschiffen durch Bordradar-Bildauswertung“, VDI Fortschrittsberichte, Reihe 8: Meß-, Steuerungs- und Regelungstechnik, Nr. 170, VDI Verlag GmbH, Düsseldorf, 1988
- [Minsky 75] M. Minsky, „A Framework for Representing Knowledge“, in: The Psychology of Computer Vision, P. Winston (Hrsg.), McGraw-Hill, New York, 1975
- [Minsky 77] M. Minsky, „Frame System Theory“, in: „Thinking“, P.N. Johnson-Laird & P.C. Wason (Eds.) , Cambridge, University Press, 1977, pp. 355-376

- [Mock-Hecker 94] R. Mock-Hecker, „Wissensbasierte Erkennung kritischer Verkehrssituationen - Erkennung von Plankonflikten“, Reihe 12: Verkehrstechnik/Fahrzeugtechnik, Nr. 209, Fortschrittsberichte VDI, VDI Verlag, Düsseldorf, 1994
- [Neisser 76] U. Neisser, „Cognition and Reality - Principles and Implications of Cognitive Psychology“, W. H. Freeman and Company, San Francisco, 1976
- [Nökel 91] K. Nökel, „Temporally distributed Symptoms in Technical Diagnosis“, Lecture Notes in AI 517, Springer Verlag ISBN 3-540-54316-3, 1991
- [Niegel 95] W. Niegel, „Methodical Structuring of Knowledge in an Intelligent Driving System“, in: 2nd IFAC Conference on Intelligent Autonomous Vehicles (IAV) 1995, Helsinki University of Technology, Espoo, Finland, June 1995
- [Onken 93] R. Onken, „Rechnersehen zur Erhöhung der Verkehrssicherheit durch Fahrerunterstützung auf Autobahnen - Monitor und Warnfunktionen“, PRO-ART Schlußbericht, Universität der Bundeswehr München, Institut für Systemdynamik und Flugmechanik, Neubiberg, Juni 1993
- [Panik 87] F. Panik, „Automobiltechnik als Korrektiv menschlichen Unvermögens“, in: FHG-Berichte 4-87, S. 25-29
- [Pau 89] L.F. Pau, „Knowledge Representation and Spacial Data Structures for Sensor Fusion“, Proceedings of the Intelligent Autonomous Systems, 1989
- [Piaget 71] J. Piaget, „Biology and Knowledge“, The University of Chicago Press, Chicago and London, 1971, ISBN: 0-226-66775-8
- [Priese et al. 94] L. Priese, J. Klieber, R. Lakmann, V. Rehrmann, R. Schian, „New Results on Traffic Sign Recognition“, in: Proceedings of the Intelligent Vehicles Symposium, Paris, 1994
- [Puttkamer 90] E. von Puttkamer, „Autonome mobile Roboter“, Vorlesungsskriptum, AG Prozeßrechenstechnik, Fachbereich Informatik, Universität Kaiserslautern, 1990
- [Puttkamer et al. 91] E. von Puttkamer, I. Tjutjunikow, R. Trieb, „Local Obstacle Avoidance and Acceleration Based Motion Control for an Autonomous Mobile Robot“, IARP - International Advanced Robotics Programme, 2nd Workshop on Sensor Fusion and Environmental Modelling, Oxford, UK, 2-5 September 1991, Session 7B
- [Puttkamer & Trieb 91] E. von Puttkamer, R. Trieb, „Modellierung und Hierarchie der Steuerung des autonomen mobilen Roboters MOBOT-III“, Autonome Mobile Systeme, 7-Fachgespräche, Karlsruhe, Dezember 1991
- [Rasmussen 83] J. Rasmussen, „Skills, Rules and Knowledge; Signals, Signs and Symbols, and Other Distinctions in Human Performance Models“, in: IEEE Transactions On Systems, Man and Cybernetics, Volume SMC-13, No. 3, May/June 1983, pp. 257-266
- [Rasmussen 86] J. Rasmussen, „Information Processing and Human-Machine Interaction“, Elsevier Science Publishing Co., North-Holland, Amsterdam, 1986
- [Reichardt 92] D. Reichardt, „Ähnlichkeitsbasierte Verkehrszeichenerkennung“, Diplomarbeit, Universität Kaiserslautern, Fachbereich Informatik, durchgeführt in der Forschungsabteilung FIM/IA der Daimler-Benz AG in Esslingen, Februar 1992
- [Reichardt & Schick 94] D. Reichardt, J. Schick, „Collision Avoidance in Dynamic Environments Applied to Autonomous Vehicle Guidance on the Motorway“, in: Proceedings of the Intelligent Vehicles Symposium, Paris, 1994
- [Reichardt 95] D. Reichardt, „A Real-Time Approach to Traffic Situation Representation from Image Processing Data“, in: Proceedings of the Intelligent Vehicles Symposium, Detroit, 1995

- [Rich 88] E. Rich, „KI-Einführung und Anwendungen“, McGraw-Hill Book Company GmbH, Hamburg, 1988, ISBN 3-89028-031-5
- [Richter 89] M. M. Richter, „Prinzipien der Künstlichen Intelligenz“, B.G. Teubner Stuttgart, 1989
- [Sandewall et al. 90] E. Sandewall, G. Adorni, H.-H. Nagel, M. Thonnat, „PROMETHEUS Session“, in: Proceedings of the 9th European Conference on Artificial Intelligence, ECAI90, Stockholm, August 1990, S. 779-780
- [Sandler & Gilles 95] M. Sandler, E. D. Gilles, „Integrated Navigation on Inland and Coastal Waterways“, 3rd International Workshop on High Precision Navigation, Stuttgart, April 3-6 1995
- [Sauerwein 94] H. Sauerwein, „Konzeption und Integration von Prozessen zur autonomen Fahrzeugführung“, Diplomarbeit, Fachhochschule Rheinland-Pfalz, Abteilung Trier, Fachbereich Angewandte Informatik, November 1994
- [Schäfers et al. 92] L. Schäfers, C. Scheidler, O. Krämer-Fuhrmann, „A Graphical Programming Environment for Parallel Embedded Systems“, IFIP Conference, April 1992
- [Schick 89] J. Schick, „Schätzung von Bewegungsgrößen mit Kalman-Filtern“, Institut für Systemdynamik und Flugmechanik, Universität der Bundeswehr, München, 1989
- [Schiehlen & Dickmanns 94] J. Schielen, E.D. Dickmanns, „A Camera Platform for Intelligent Vehicles“, in: Proceedings of the Intelligent Vehicles Symposium, Paris, 1994
- [Schiele & Schneider 96] A. Schiele, M. Schneider, „Untersuchung des optischen Hinderniserkenners ‘Vision Bumper’“, Fachbericht, Berufsakademie Stuttgart Fachrichtung Elektrotechnik Automatisierungstechnik, Mercedes-Benz AG, 1996
- [Schöner & Dose 92] G. Schöner, M. Dose, „A dynamical systems approach to task-level system integration used to plan and control autonomous vehicle motion“, in: Robotics and Autonomous Systems 10, Elsevier Science Publishers, 1992
- [Shapiro 87] S.C.Shapiro, „Encyclopedia of Artificial Intelligence“, Wiley-Interscience Publications, 1987
- [StatJB 91] Statistisches Jahrbuch 1991. Herausgegeben vom Statistischen Bundesamt
- [Thomanek & Dickmanns 92] F. Thomanek, E.D. Dickmanns, „Obstacle Detection, Tracking and Stat Estimation for Autonomous Road Vehicle Guidance“, IROS 92, Raleigh NC, 7.-10.July 92
- [Thomanek et al. 94] F. Thomanek, E.D. Dickmanns, D. Dickmanns, „Multiple Object Recognition and Scene Interpretation for Autonomous Road Vehicle Guidance“, in: Proceedings of the Intelligent Vehicles Symposium, Paris, 1994
- [Thorpe 84] C. E. Thorpe, „Path Relaxation: Path Planning for a mobile Robot“, in: Proceedings of the Conference of the American Association for Artificial Intelligence, Austin, Texas, August 1984
- [Tilove 90] R. B. Tilove, „Local Obstacle Avoidance for Mobile Robots Based on the Method of Artificial Potentials“, in: Proceedings of the Conference on Robotics and Automation, IEEE, 1990
- [Ulmer 92] B. Ulmer, „VITA - An Autonomous Road Vehicle (ARV) for Collision Avoidance in Traffic“, in: Proceedings of the Intelligent Vehicles Symposium, Detroit, 1992
- [Ulmer 94a] B. Ulmer, „VITA II - Active Collision Avoidance in Real Traffic“, in: Proceedings of the Intelligent Vehicles Symposium, Paris, 1994

- [Ulmer 94b] B. Ulmer, „Autonomous Automated Driving in Real Traffic”, in: Proceedings of the first world congress on applications of transport telematics and intelligent vehicle-highway systems, Paris, 1994
- [Umland & Vollmar 92] T. Umland, R. Vollmar, „Transputerpraktikum”, Verlag B.G. Teubner, Stuttgart, 1992
- [Vester 95] F. Vester, „Crashtest Mobilität - Die Zukunft des Verkehrs”, Wilhelm Heyne Verlag GmbH & Co. KG, München, 1995, ISBN 3-453-08875
- [Wieczorek 96] R. Wieczorek, „Analyse und Simulation von Verfahren zur Hinderniserkennung in Straßenszenen”, Diplomarbeit, Fachbereich Technische Informatik der Fachhochschule Esslingen, Durchgeführt bei der Daimler-Benz AG, Esslingen, 1996
- [Weß 91] S. Weß, „PATDEX/2 - ein System zum adaptiven fallfokussierenden Lernen in technischen Diagnosesituationen”, SEKI Working Paper SWP-91-01 (SFB), Universität Kaiserslautern, Fachbereich Informatik, 1991
- [Zetsche 95] D. Zetsche, „Konzepte gegen Stau und Smog”, in: Spektrum der Wissenschaft, Spezial 4 : Schlüsseltechnologien, Spektrum der Wissenschaft Verlagsgesellschaft mbH, Heidelberg, Oktober 1995
- [Zheng et al 94] Y.-J. Zheng, W. Ritter, R. Janssen, „An Adaptive System for Traffic Sign Recognition”, in: Proceedings of the Intelligent Vehicles Symposium, Paris 1994

E Index

—A—

Agenda 127
 Agent 124
 Aktorsimulation 126
 Aktorsystem 8; 10; 13
 Anwendungsrechner 14; 16; 17; 24

—B—

Basisverhaltensmuster 57
 Benutzereingabe 34
 Blackboard 100

—D—

Daimler-Benz AG 12
 DDB 16; 24; 25
 Diskurswelt 32
 Drei-Ebenen Modell 54

—E—

Echtzeit 11; 15; 17; 37
 elektrisches Feld 48

—F—

Fächerkamera 21
 Fahraufgabe 52
 Fahrerintentionen 86
 Fahrzeugdynamik 51
 Fahrzeugführung 72
 Fahrzeugrechner 14; 16; 17
 Fahrzeugzustand 34
false negative 88
false positive 88

—G—

Gefahr 27
 Gefahrengebirge 35
Gefahrenkarte 28
 geschlossenen Regelkreis 120
 Glättung 92

—H—

Handlungsbeschreibungen 62
 Handlungsschema 55
 Hemmung 67
 Hinderniserkennung 15; 20; 21
 Hysterese-Filter 102

—I—

Inhibitionsnetzwerk 65

—K—

Kalman-Filter 93
 Kollisionsvermeidung 8; 13
Kontrollsystem 8; 10; 13; 14; 52
 Konzeptrahmen 96
 KRONOS 18; 20
 kybernetisches System 55

—N—

n-Äquivalenzmenge 65
Notreaktionen 57

—P—

Parallelisierung i
 Parallelverarbeitung i
Plattformrechner 14; 17
 Potentialfeldansatz 36
 Potentialfeldmethode 37; 39
 PROMETHEUS 2

—R—

Regelung 10; 14; 17
 Risiko 53
Risikokompensationstheorien 27; 28
 Roboter 7

—S—

schemabasiertes Fehlermodell 111
Schematisierte Handlungen 57
 Sensor 8
 Sensor Fusion 90
 Sensorsimulation 125
Sensorsystem 7
 Simulation 119
 Situation 35
 Situationsbeschreibung 88
 Steuervektor 35
 Stimulation 66
 Straße 33
Subsumptionsarchitektur 52

—T—

Time To Collision 77
 Transputer 15; i
 TRAPPER 15

—U—

Unschärfe Beschreibungsprädikate 114
 Unschärfes Intervall 115

—V—

Verhaltensmodelle 52
 Verhaltensmuster 61
 Verhaltensmusterkonzept 52
 Verhaltenssteuerung 27; 36
 Verkehrszeichen 34
 Verkehrszeichenerkennung 15; 22; 23
 Versuchsträger 12; 88; 133
 virtuelle Welt 123
 VITA II 12; 14; 20; 133
 von-Neumann-Architektur i

—W—

Wissensrepräsentation 58

—Z—

Zeitliche Verknüpfung 114