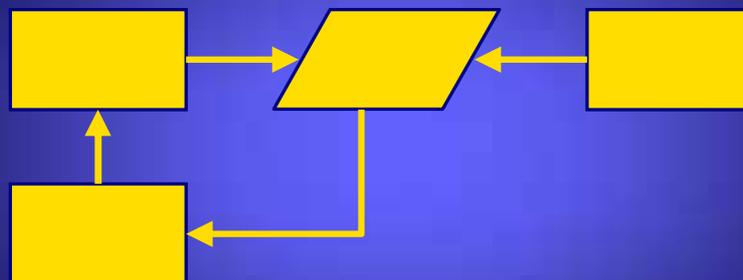


---

# Intelligentes Missionsmanagement für autonome mobile Systeme

---



Intelligentes Missionsmanagement  
für autonome mobile Systeme

D I S S E R T A T I O N

Torsten Pfützenreuter

### **Bibliografische Information Der Deutschen Bibliothek**

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

1. Aufl. - Göttingen : Cuvillier, 2005

Zugl.: (TU) Ilmenau, Univ., Diss., 2004

ISBN 3-86537-399-2

⊕ CUVILLIER VERLAG, Göttingen 2005

Nonnenstieg 8, 37075 Göttingen

Telefon: 0551-54724-0

Telefax: 0551-54724-21

[www.cuvillier.de](http://www.cuvillier.de)

Alle Rechte vorbehalten. Ohne ausdrückliche Genehmigung des Verlages ist es nicht gestattet, das Buch oder Teile daraus auf fotomechanischem Weg (Fotokopie, Mikrokopie) zu vervielfältigen.

1. Auflage, 2005

Gedruckt auf säurefreiem Papier

ISBN 3-86537-399-2

## Vorwort

Die vorliegende Dissertation ist das Ergebnis eines dreijährigen Kooperationsprojektes von Partnern aus Industrie und Forschung, dessen Ziel in der Entwicklung eines neuartigen autonomen Unterwasserfahrzeuges bestand. Die Arbeit wurde ermöglicht durch die finanzielle Unterstützung des Bundesministeriums für Bildung und Forschung (BmBF).

Mein besonderer Dank gilt dem Leiter des Fachgebietes Systemanalyse und Betreuer meiner Arbeit, Herrn Prof. Dr.-Ing. Jürgen Wernstedt. Er hat mich während unserer langjährigen gemeinsamen Tätigkeit stets unterstützt und die Erstellung der Dissertation durch Anregungen und kritische Hinweise zu Inhalt und methodischer Herangehensweise wesentlich geprägt.

Von meinen Kollegen im Fachgebiet sowie den Projektpartnern erhielt ich ebenfalls wertvolle Hinweise. Insbesondere die Herren Dr.-Ing. Thomas Rauschenbach und Dr.-Ing. Peter Bretschneider standen mir bei fachlichen Diskussionen und bei der Durchsicht des Manuskriptes hilfreich zur Seite.

Ohne die tatkräftige und geduldige Unterstützung zahlreicher weiterer Personen aus meinem privaten Umfeld wäre diese Arbeit nicht zustande gekommen, auch bei ihnen möchte ich mich herzlich bedanken.

Meiner Frau Nicole, meiner Familie sowie allen Freunden danke ich vor allem dafür, dass sie mich auch in kritischen Phasen der Arbeit erfolgreich aufgemuntert haben und mich nie den Bezug zur Realität verlieren ließen.

Ilmenau, den 28. Juni 2004

Torsten Pfitzenreuter

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
1.1	Autonome mobile Systeme - Aufgaben und Definitionen .....	3
1.2	Aufbau der Arbeit .....	5
<b>2</b>	<b>Stand der Technik und Ziele der Arbeit</b>	<b>7</b>
2.1	Einordnung bekannter Ansätze zum Missionsmanagement .....	7
2.1.1	Übersicht .....	7
2.1.2	Missionsmanagement ausgewählter autonomer mobiler Systeme .....	10
2.1.3	Bewertung .....	16
2.2	Softwarearchitektur .....	17
2.2.1	Das menschliche Handlungsmodell nach Rasmussen .....	17
2.2.2	Softwarearchitekturen autonomer mobiler Systeme .....	19
2.3	Rahmenbedingungen und Ziele der Arbeit .....	20
2.3.1	Rahmenbedingungen .....	20
2.3.2	Zielsetzung und Beitrag der Arbeit .....	22
2.4	Zusammenfassung .....	23
<b>3</b>	<b>Gesamtkonzeption des Missionsmanagements</b>	<b>25</b>
3.1	Aufbau des Missionsmanagements .....	25
3.1.1	Missionsplanung .....	27
3.1.2	Missionsüberwachung .....	28
3.1.3	Missionsumplanung .....	29
3.2	Softwaretechnische Aspekte .....	30
3.3	Zusammenfassung .....	30
<b>4</b>	<b>Missionsplanung</b>	<b>33</b>
4.1	Vorgehensweise bei der Missionsplanung .....	33
4.2	Gestaltung der Mensch-Maschine-Schnittstelle .....	34
4.2.1	Aspekte zum ergonomischen Entwurf von Bedienoberflächen .....	35
4.2.2	Gestaltung der Planungsoberfläche .....	35
4.3	Zusammenfassung .....	37
<b>5</b>	<b>Wissensbasierte Überwachung der Mission</b>	<b>39</b>
5.1	Aufbau und Arbeitsweise regelbasierter Systeme .....	39
5.1.1	Aufbau .....	39

5.1.2	Wissensrepräsentation .....	40
5.1.3	Wissensverarbeitung .....	41
5.1.4	Wissensermittlung .....	42
5.2	Konzeption des Missionsüberwachungssystems .....	44
5.2.1	Aufgaben der Missionsüberwachung .....	44
5.2.2	Aufbau der Missionsüberwachung.....	45
5.3	Zusammenfassung.....	49
<b>6</b>	<b>Strategien zur Missionsumplanung</b>	<b>51</b>
6.1	Ablauf der Missionsumplanung .....	51
6.2	Allgemeine Methoden zur Planmodifikation .....	51
6.2.1	Voraussetzungen .....	51
6.2.2	Einfügen von Planelementen.....	54
6.2.3	Löschen von Planelementen .....	55
6.2.4	Modifizieren von Planelementen .....	56
6.2.5	Planbereinigung.....	57
6.3	Spezielle Verfahren zur Planmodifikation.....	57
6.3.1	Missionsabbruch .....	58
6.3.2	Optimierung eines Planes .....	58
6.4	Geografische Planprüfung und -modifikation .....	64
6.4.1	Datenbasis .....	66
6.4.2	Prüfung des Missionsplanes .....	67
6.4.3	Modifikation des Missionsplanes .....	70
6.5	Zusammenfassung.....	73
<b>7</b>	<b>Anwendung und Ergebnisse</b>	<b>75</b>
7.1	Der mobile Kleinroboter MauSI .....	75
7.1.1	Aufbau und Steuerung .....	75
7.1.2	Softwarearchitektur .....	76
7.1.3	Intelligentes Missionsmanagement .....	80
7.2	Das autonome Unterwasserfahrzeug DeepC .....	86
7.2.1	Aufbau und Steuerung .....	87
7.2.2	Intelligentes Missionsmanagement .....	93
7.3	Zusammenfassung.....	107
<b>8</b>	<b>Zusammenfassung und Ausblick</b>	<b>109</b>
8.1	Zusammenfassung.....	109
8.2	Ausblick .....	111
	<b>Literaturverzeichnis</b>	<b>113</b>
	<b>Index</b>	<b>121</b>

# Abbildungsverzeichnis

2.1	Einteilung wissensbasierter Systeme .....	8
2.2	Systemarchitektur von AMOR .....	10
2.3	SAUSAGES: Link und Missionsplan bestehend aus Links .....	11
2.4	Missionsplan als Petri-Netz .....	14
2.5	Menschliches Handlungsmodell nach Rasmussen .....	18
2.6	Rational Behavior Model .....	19
2.7	Verwendete Softwarearchitektur .....	20
2.8	Klassisches missionsplanbasiertes Steuerungssystem .....	21
3.1	Aufbau und Integration des intelligenten Missionsmanagements .....	27
3.2	Missionsumplanung als Regelkreis .....	28
4.1	Ablauf der Missionsplanung .....	33
4.2	Graphenorientierte Missionsplanung .....	36
4.3	Geometriebasierte und geografiebasierte Missionsplanung .....	37
5.1	Aufbau eines regelbasierten Expertensystems .....	40
5.2	Ablauf der Inferenz .....	42
5.3	Phasen der Wissensermittlung .....	45
5.4	Aufbau der Missionsüberwachung .....	46
5.5	Klassenstruktur und Objektinstanzen des Arbeitsspeichers .....	47
6.1	Grundlegender Ablauf der Missionsumplanung .....	52
6.2	Aufbau eines sequentiellen Missionsplanes .....	52
6.3	Einfügen des Planelementes $E$ nach dem Element $i$ .....	55
6.4	Einfügen des Planelementes $E$ in das Element $i$ .....	56
6.5	Löschen des Planelementes $i$ .....	56
6.6	Modifizieren des Planelementes $i+1$ .....	56
6.7	Bereinigung eines Missionsplanes .....	57
6.8	Abbruch einer Mission bei der Ausführung des Planelementes $i$ .....	59
6.9	Erzeugen einer Manöverkonfiguration .....	63
6.10	Zweistufiges Konzept der Optimierung .....	64
6.11	Geografisches Koordinatensystem .....	65
6.12	Delaunay Triangulation mit Umkreisen .....	66
6.13	Datenstruktur von dreiecksvermaschten Netzen .....	66
6.14	Ablauf der geografischen Prüfung eines Missionsplanes .....	67
6.15	Kreisapproximation mit Vieleck .....	69
6.16	Approximation eines Kreisbogens .....	69
6.17	Baryzentrische Koordinaten im Dreieck .....	70
6.18	Graph mit positiven Kantenbewertungen .....	72
6.19	Distanzmatrix zum Graph aus Abbildung 6.18 .....	72

6.20	Ablauf des Dijkstra-Algorithmus für den Graph aus Abbildung 6.18 .....	73
6.21	Geografische Planmodifikation .....	73
7.1	Schematischer Aufbau des Roboters MauSI .....	76
7.2	Softwarearchitektur für die Steuerung von MauSI .....	77
7.3	Vereinfachtes Simulationsmodell des Kleinroboters MauSI .....	78
7.4	Modellierung des Bewegungsverhaltens des Kleinroboters MauSI .....	79
7.5	Bahnpunkte und Vektorpunkte .....	80
7.6	Bahnregelung mit Vektorpunkten .....	80
7.7	Grafischer Missionsplaneditor für MauSI-Roboter .....	82
7.8	Umplanung bei Ausfall der Nutzlast, Szenario I .....	84
7.9	Umplanung bei Ausfall der Nutzlast, Szenario II .....	84
7.10	Fahrtrajektorien bei Ausfall der Nutzlast .....	85
7.11	Umplanung bei Ausfall wesentlicher Fahrzeugmodule .....	86
7.12	Optimierung des Missionsplanes bei Ressourcenmangel .....	86
7.13	Fahrkurve bei Ausfall unkritischer Module .....	87
7.14	Fahrtrajektorie unter Einsatz der Optimierung .....	87
7.15	Aufbau des autonomen Unterwasserfahrzeuges DeepC .....	88
7.16	Softwarearchitektur des AUV DeepC (Auszug) .....	91
7.17	Geometriebasierte Planung für DeepC .....	94
7.18	DeepC: Missionsplan für die Tests des intelligenten Missionsmanagements ....	95
7.19	DeepC: Ablauf der Missionsumplanung .....	97
7.20	Digitale Seekarte der Eckernförder Bucht .....	98
7.21	Messpunkte der digitalen Navigationskarten des Superior Sees .....	99
7.22	Höhenmodell für die geografischen Umplanungsverfahren .....	100
7.23	Vorgehensweise bei der Planbereinigung .....	101
7.24	Beispiel für die Umplanung im Fall defekter Sensorik .....	102
7.25	Einfügen eines Event-Manövers in den Missionsplan .....	104
7.26	Koordinatenberechnung und Einfügevarianten von Event-Manövern .....	105
7.27	Missionsplan für den Test der geografischen Planungsverfahren .....	106
7.28	Beispiel für die geografische Planprüfung und -modifikation .....	107

## Tabellenverzeichnis

3.1	Verfügbare allgemeingültige Umplanungsbefehle.....	26
6.1	Eigenschaften von Planelementen.....	54
7.1	Manöverkatalog für MauSI .....	81
7.2	Getestete Umplanungssituationen für MauSI.....	82
7.3	DeepC: Wesentliche technische Daten.....	89
7.4	Manöverkatalog für DeepC, Erweiterungen gegenüber MauSI.....	92
7.5	Umplanungssituationen für DeepC (Auszug) .....	96
7.6	Daten des digitalen Höhenmodells .....	99
7.7	Ergebnisse und relativer Zeitbedarf der Missionsplanoptimierung .....	103

# Verzeichnis der wichtigsten Formelzeichen und Abkürzungen

## Formelzeichen

<b>a</b>	Aktionen, auszuführende Handlungen eines Planelementes
<b>akt</b>	Vektor der Aktivierungswerte eines Missionsplanes
$\alpha$	Orientierung eines mobilen Systems im Raum
$c$	Kosten, Aufwand an Ressourcen
<b>c</b>	Kostenvektor eines Planelementes
$c_{max}$	verfügbare Ressourcen eines mobilen Systems
<b>C</b>	Gesamtkosten einer Mission
$e_A$	Erfüllungsgrad am Anfang eines Planelementes
$e_E$	Erfüllungsgrad am Ende eines Planelementes
$e_{akt}$	aktueller Erfüllungsgrad eines Planelementes
$f_{max}$	maximaler Approximationsfehler
<b>f(x)</b>	Gütefunktional der Optimierung
$G$	Graph
<b>G</b>	dreidimensionale Position auf einer Geraden
<b>g(x)</b>	Funktional der Beschränkungen der Optimierung
$\gamma$	Öffnungswinkel eines Kreisbogens
$h$	Höhe, bezogen auf einen Rotationsellipsoid
$\lambda$	geografische Breite auf einem Rotationsellipsoid
<b>P</b>	zwei- oder dreidimensionale Position im Raum
<b>p</b>	Parametervektor eines Planelementes
$\varphi$	geografische Länge auf einem Rotationsellipsoid
$r$	Radius eines Kreises oder Kreisbogens
<b>u, v, w</b>	baryzentrische Koordinaten im Dreieck
$v$	Längsgeschwindigkeit eines mobilen Systems
$w$	Wert eines Planelementes
$W$	Gesamtwert einer Mission
$\omega$	Winkelgeschwindigkeit eines mobilen Systems um die Vertikalachse (Gierachse)
<b>x</b>	Zustandsvektor
$x_A$	Zustandsvektor am Anfang eines Planelementes
$x_E$	Zustandsvektor am Ende eines Planelementes
<b>x</b>	Vektor der Variablen der Optimierung

## Abkürzungen

AMS	Autonomes mobiles System
AUV	Autonomous Underwater Vehicle
CEP	Circular Error Probable
DOLOG	Doppler Velocity Log (DVL)
ENC	Electronic Navigational Chart
FPE	Füllplanelement
GPS	Global Positioning System
INS	Inertiales Navigationssystem
KI	Künstliche Intelligenz
PE	Planelement
ROV	Remotely Operated Vehicle
TIN	Triangulated Irregular Network
UAV	Unmanned Aerial Vehicle
WGS	World Geodetic System

# 1 Einführung

Eigenständiges und vorausschauendes Handeln erfordert Intelligenz. Dieser einfach formulierte Sachverhalt stellt bei der Realisierung eines autonom agierenden Systems eines der größten Probleme dar. Ansätze für ein autarkes Agieren von Robotern sind vielfältig zu finden, jedoch sind alle jeweils auf die besonderen Einsatzfälle oder -umgebungen zugeschnitten.

Dies trifft im engeren Sinne auch auf diese Arbeit zu, da sie auf ein spezielles Feld der autonomen mobilen Systeme ausgerichtet ist: Schwerpunkt der Arbeit bilden Roboter, die in (zum Teil für den Menschen unwirtlichen) Umgebungen Aufgaben lösen, ohne in Kontakt mit einem menschlichen Operator zu treten oder treten zu können. Das Mensch-Maschine-Interface beschränkt sich hier auf die Planung des Einsatzes, die Kontrolle nach dem Einsatz und - wenn eine Kommunikation existiert - die Möglichkeit zur manuellen Einflussnahme auf das System. Für den Einsatz in solchen Umgebungen erhält der Roboter üblicherweise eine Folge von Aufgaben, die er zu realisieren hat. Die Aufgaben können sowohl grundlegende Handlungen als auch komplexe Abläufe darstellen. Der Roboter ist somit in der Lage, über eine begrenzte Zeit selbständig agieren zu können.

Szenarien, in denen Roboter keinen Kontakt zu einem Operator haben, treten in der heutigen Zeit vielfältig in Erscheinung. Prominente Beispiele sind die Mars-Roboter der NASA, Spirit und Opportunity. Durch Einsatz der Steuerungssoftware Maestro in der Bodenstation wird der Aktivitätsplan für die Roboter mit Hilfe der von der Marsoberfläche übermittelten Informationen erstellt, simuliert und anschließend zur Abarbeitung an die Rover gesendet [BNP<sup>+</sup>03]. Aufgrund der engen Zeitfenster zur Kommunikation werden die zu lösenden Aufgaben an jedem Marstag für die folgenden knapp 25 Stunden im Voraus bestimmt [NAS04].

Ein weiteres, weitgehend unerforschtes Einsatzgebiet für autonome Roboter sind die Meere. 71 Prozent der Erdoberfläche sind mit Wasser bedeckt, davon zählen 80 Prozent zur Tiefsee und weisen damit Tiefen von mehr als 800 Meter auf. Diese Bereiche sind zum Beispiel weniger erforscht als die Mond- oder Marsoberfläche. Die Erkundung der Tiefsee wird heutzutage meist mit ferngesteuerten Unterwasserfahrzeugen (ROVs, Remotely Operated Vehicles) durchgeführt. Neben dem eigentlichen Tauchroboter sind dazu ein Mutterschiff mit der nötigen Energieversorgung sowie geschultes Personal zur Bedienung des Systems erforderlich. Bei typischen Erkundungsmissionen ist der finanzielle Aufwand für die sekundären Systeme (Mutterschiff, Personal) derartig hoch, dass zunehmend nach Alternativen gesucht wird.

Einen möglichen Ausweg stellt die Entwicklung von autonomen Unterwasserfahrzeugen (AUVs, Autonomous Underwater Vehicles) dar. Neben der finanziellen Seite sind die mangelnden Kommunikationsmöglichkeiten zwischen einem Schiff und dem tieftauchenden Roboter der zweite wesentliche Grund für die verstärkten Bemühungen um die Autonomie dieser mobilen Systeme. Die schlechten Ausbreitungseigenschaften von Funkwellen im Salzwasser der Meere führten zur Entwicklung akustischer Signalübertragungsverfahren, die aber die Erfordernisse an Reichweite und zur Verfügung stehender Bandbreite für ein teleoperier-

tes System bisher nicht erfüllen können.

Auch bei Flug- und Landrobotern ist eine zumindest zeitweilige autarke Arbeitsweise ohne Funkkontakt von Interesse. Dies beschränkt sich nicht nur auf militärische Bereiche wie zum Beispiel die Spionage, auch Routineaufgaben wie die Reinigung von Gebäuden werden durch autonome Systeme realisiert, ohne dass sie von einem Operator ständig überwacht werden. Im Unterschied zu den anderen Arbeitsumgebungen besitzen erdgebundene Roboter den Vorteil, bei unerwarteten Problemen einfach die Arbeit einstellen und auf Hilfe warten zu können. Ein fliegendes oder tauchendes System muss demgegenüber stets einen Übergang in einen sicheren Zustand ausführen.

All diese Systeme benötigen Methoden, um in unerwarteten Situationen angemessen reagieren zu können. Im einfachsten Fall bedeutet dies die Unterbrechung der Arbeit, bis eine Kommunikation mit der Leitzentrale möglich ist und ein neuer Handlungsplan übermittelt wird. Das andere Ende des Spektrums wird durch Systeme markiert, die innerhalb ihrer Arbeitsumgebung vollkommen autark auf derartige Situationen reagieren und sich aus einer Analyse heraus selbständig neue Arbeitsaufgaben definieren können.

Die erwähnten Mars-Sonden besitzen eine Hindernisvermeidungsstrategie, um sich auf dem Weg zu einem Zielpunkt keinem unnötigen Risiko auszusetzen. Weitergehende autonome Fähigkeiten sind nicht vorhanden. Das hängt unmittelbar mit der Gefahr zusammen, dass eine falsch getroffene Entscheidung des autonomen Systems in einer Vielzahl von Fällen zu einem Totalverlust führen kann. Im Gegensatz dazu ist bei erdgebundenen Robotern das Risikopotential verhältnismäßig gering, sodass häufig umfangreiche autonome Fähigkeiten anzutreffen sind.

Tieftauchende autonome Unterwasserfahrzeuge befinden sich, obwohl sie in einer verglichen mit den Mars-Rovern sehr geringen Entfernung zu ihrer Leitzentrale operieren, in einer ungleich höheren Gefahr. Durch die wenigen Informationen über die Verhältnisse in der Tiefsee wie die Struktur des Meeresbodens oder vorhandene Strömungen kann keine zuverlässige Prognose über den Verlauf einer Tauchmission in diese Region abgegeben werden. Die fast völlig fehlenden Kommunikationsmöglichkeiten verhindern ein Eingreifen des Operators in Situationen, die durch das Fahrzeug selbst nicht beherrscht werden. Aus diesem Grund sind die Missionspläne und die Softwarearchitektur für explorierende oder vermessende AUVs häufig sehr einfach aufgebaut. Damit soll ein höchstmögliches Maß an Sicherheit erreicht werden.

Motivation für die in der vorliegenden Arbeit beschriebenen Entwicklung eines intelligenten Missionsmanagements ist das in Kooperation zwischen unterschiedlichen Partnern entstehende autonome Unterwasserfahrzeug *DeepC* für Tiefsee-Missionen, das im Gegensatz zu den bereits existierenden tauchenden Systemen über erweiterte Methoden zum selbständigen Handeln verfügt. Neben einer sich optimal an die Umweltbedingungen anpassenden Hindernisvermeidung gehört auch das intelligente Management des Missionsplanes zu diesen Fähigkeiten. Die dafür entworfenen Verfahren sind nicht auf Unterwasserfahrzeuge beschränkt. Dies wird anhand der zweiten Anwendungsplattform, des mobilen Kleinroboters *MauSI*, gezeigt.

Die erweiterten autonomen Fähigkeiten sollen in eine bestehende, ausgereifte Softwarestruktur eingebettet werden. Dies bedeutet, dass viele verfügbare Ansätze für das intelligente Missionsmanagement nicht anwendbar sind, da sie häufig eine komplexe Anpassung der

Softwarearchitektur erfordern. Darauf wird im Kapitel 2 näher eingegangen.

In den weiteren Abschnitten dieses Kapitels sollen einige für das Verständnis der Arbeit notwendige, grundlegende Begriffe definiert und ein Überblick zu dem Inhalt der folgenden Kapitel gegeben werden.

## 1.1 Autonome mobile Systeme - Aufgaben und Definitionen

In den vorangegangenen Ausführungen wurde der Term *autonomes mobiles System* bereits mehrfach verwendet. Unter diesem Begriff soll in der vorliegenden Arbeit Folgendes verstanden werden:

### **Definition 1.1** *Autonomes mobiles System (AMS)*

Ein *autonomes mobiles System* soll in einer (bekannten oder unbekannt) Umgebung komplexe Aufgaben lösen, ohne ständigen Kontakt zu einem menschlichen Operator zu haben. Das System muss dafür die Fähigkeit besitzen, bei Bedarf ihm übertragene Probleme an die aktuelle Situation anzupassen.

Im Gegensatz zu anderen Definitionen wie zum Beispiel in [Edl97] steht hier die selbständige Erfüllung von umfangreichen Handlungsabläufen im Mittelpunkt. Aufgrund der eventuell nur eingeschränkt verfügbaren Verbindung zu einem Operator muss ein derartiges System in der Lage sein, autark auf seine Umwelt reagieren zu können.

Die zu realisierenden komplexen Aufgaben müssen dem System vor Beginn der Tätigkeit übergeben werden. Eine Menge derartiger Handlungen wird häufig als Missionsplan bezeichnet.

### **Definition 1.2** *Missionsplan, Planelement*

Ein *Missionsplan* besteht aus einer Folge von Planelementen. Jedes Planelement stellt dabei ein komplexes Handlungsschema für ein autonomes mobiles System dar und wird auch als Manöver bezeichnet.

Der Missionsplan beinhaltet eine Sequenz komplexer Aufgaben, die nacheinander zu lösen sind. Die Menge unterschiedlicher Planelemente, der so genannte Manöverkatalog, definiert schließlich die Bandbreite an Einsatzmöglichkeiten des autonomen mobilen Systems. Dieser Katalog ist an die konkrete Umwelt des Roboters angepasst, da beispielsweise ein erdgebundenes System andere Aufgaben als ein fliegendes oder tauchendes zu erfüllen hat.

Der Missionsplan wird mit Hilfe der Missionsplanung vor dem Beginn des Einsatzes erstellt. Dabei kann die Planung sowohl von einem Operator mittels einer Planungsoberfläche oder automatisch durch geeignete Verfahren erfolgen.

Wie bereits erwähnt ist das Ziel dieser Arbeit die Entwicklung von Verfahren zum intelligenten Missionsmanagement eines autonomen mobilen Systems. Um eine Einordnung unterschiedlicher Ansätze vornehmen zu können, sind zunächst damit zusammenhängende Begriffe zu klären.

**Definition 1.3** *Missionsmanagement*

Das *Missionsmanagement* umfasst diejenigen Verfahren, die sich mit der Planung, Realisierung und Modifikation des Missionsplanes eines autonomen mobilen Systems beschäftigen.

Der Schwerpunkt der zu entwickelnden Verfahren liegt dabei auf der Modifikation des Missionsplanes während seiner Realisierung im mobilen System. Der Begriff der Umplanung ist in diesem Zusammenhang weiter gefasst als im üblichen Sprachgebrauch autonomer mobiler Systeme. Dort wird mit der Umplanung die Umgehung vorgefundener Hindernisse und auch die Neuplanung eines Weges zum gewünschten Endpunkt der Mission verstanden. Diese in der vorliegenden Arbeit als geografische Planmodifikation bezeichneten Methoden sind nachfolgend lediglich als ein Bestandteil der Missionsumplanung zu sehen.

Um eine an das zweifellos sehr gute autonome Verhalten des Menschen angelehnte Handlungsweise zu erreichen, sollen intelligente Methoden für das Missionsmanagement eingesetzt werden. Allgemein wird der Begriff *Intelligenz* als die Fähigkeit des Menschen erklärt, zu verstehen, zu lernen und sich in konkreten Situationen erfolgreich zu verhalten. Angewandt auf das Missionsmanagement des mobilen Systems kann folgende Definition vorgenommen werden.

**Definition 1.4** *Intelligentes Missionsmanagement*

Das *intelligente Missionsmanagement* hat die Aufgabe, einen Roboter dazu zu befähigen, auf unerwartete oder neue Situationen mit Korrekturen des Missionsplanes zu reagieren und die Situation erfolgreich zu bewältigen. Dabei können lernfähige Verfahren eingesetzt werden, um aus dem Erlebten Rückschlüsse auf verbesserte Handlungsweisen abzuleiten.

Verfahren zur Abbildung intelligenten Verhaltens sind dem Forschungsgebiet der künstlichen Intelligenz zuzuordnen. Sie beschäftigt sich im Allgemeinen mit der Simulation kognitiver Prozesse des Menschen durch Computerprogramme.

Für intelligentes Verhalten muss ein gewisser Umfang an Wissen vorhanden sein. Da der Begriff Wissen für die weiteren Ausführungen eine wesentliche Bedeutung besitzt, soll er analog zu [Hau00] folgendermaßen definiert werden.

**Definition 1.5** *Wissen*

Unter *Wissen* werden alle als wahr oder falsch angenommenen Aussagen über die repräsentierte Welt zusammengefasst.

Wissen kann damit als Verallgemeinerung der Terme Daten (Sammlung von computerverarbeitbaren Fakten) und Informationen (zweckgebundene Datensammlungen) verstanden werden. Diese Begriffsbestimmung bildet die Basis für die Abbildung und Nutzung des Wissens in einem Computerprogramm, die beide in den nachfolgenden Kapiteln eine große Rolle spielen werden.

Zu beachten ist hier, dass Wissen nicht gleichzeitig Wahrheit bedeutet. Es ist vielmehr als der Besitz von objektiv bzw. subjektiv als wahr zu klassifizierenden Erfahrungen zu sehen.

Subjektiv als wahr empfundene Informationen, die das Produkt der Wahrnehmung sind, müssen nicht zwangsläufig als objektiv wahr akzeptiert werden. Bereits Platon sprach in seinem Dialog Theaitetos von der „Wahrheit“ des Protagoras, die sich in Endkonsequenz lediglich als Wahrnehmung oder „wahre Überzeugung“ herausstellte.

Für ein autonomes mobiles System bedeutet diese Feststellung, dass das durch Sensorinformationen entstehende Abbild der realen Welt niemals eine wahrhaftige Widerspiegelung dessen ist. Das Abbild sollte vielmehr als *wahrscheinlicher Zustand* der Umwelt betrachtet werden. Infolgedessen sind die zu entwerfenden Verfahren eher darauf ausgerichtet, einen sicheren Systemzustand zu erhalten oder zu erreichen, als mit Hilfe von Wahrgenommenem alle erdenklichen Möglichkeiten in einer Situation auszunutzen.

## 1.2 Aufbau der Arbeit

Die vorliegende Arbeit gliedert sich in folgende Abschnitte. *Kapitel 2* stellt den Einstieg in die Thematik dar. Zu Beginn soll der Versuch unternommen werden, eine Einordnung unterschiedlicher Ansätze für das Missionsmanagement autonomer Systeme in die entsprechenden Wissenschaftsdisziplinen vorzunehmen. Beispiele für den Einsatz dieser Methoden runden diesen Teil ab. Die eingesetzte Softwarearchitektur und die sich daraus ergebenden Schnittstellen zwischen den einzelnen Modulen spielen für die Auswahl der anwendbaren Verfahren eine große Rolle. Aus diesem Grund wird in einem eigenen Abschnitt näher auf typische hierarchische Strukturen und ihre Rückführung auf ein Handlungsmodell des Menschen eingegangen. Die konkreten Rahmenbedingungen für das intelligente Missionsmanagement, die für die vorliegende Arbeit gelten, sowie die daraus resultierenden Zielstellungen beschließen diesen Abschnitt.

Das neuartige Konzept des intelligenten Missionsmanagements für autonome mobile Systeme wird in *Kapitel 3* vorgestellt. Dabei wird auf die benötigten Verfahren und die zu entwickelnden Teilmodule eingegangen. Auch Aspekte der Programmierung, die durch den gewählten Ansatz zu beachten sind, finden Berücksichtigung in diesem Kapitel.

Die Grundlagen für den Entwurf der Missionsplanung sind im *Kapitel 4* zu finden. Dabei bilden die Analyse der Planungsaufgabe sowie die Gestaltung der Bedienoberfläche wesentliche Schwerpunkte. Zwei unterschiedliche Konzepte für die Generierung von Missionsplänen, die struktur- und die geometriebasierte Planung, werden anhand von entsprechenden Darstellungen präsentiert.

*Kapitel 5* gibt einen detaillierten Einblick in die wissensbasierte Überwachung der Missionsdurchführung des autonomen mobilen Systems, der intelligenten Komponente des Missionsmanagements. Dabei werden die verwendeten Methoden wissensbasierter Systeme sowie Aufbau und Inhalt der Wissensbasis vorgestellt. Besonderes Augenmerk liegt auf der Erweiterbarkeit des Systems mit neuen Regeln - auch mit Hilfe automatischer Lernverfahren - sowie der Anpassungsfähigkeit an unterschiedliche Einsatzszenarien.

In *Kapitel 6* werden die Verfahren zur Adaption von Missionsplänen beschrieben. Nach der Erörterung der allgemeinen Zusammenhänge, die bei der Planmodifikation zu beachten sind, werden die grundlegenden Umplanungsbefehle präsentiert. Darauf folgt die Darstellung der Optimierung eines Missionsplanes unter beschränkten Ressourcen und der geografischen Planmodifikation als Schwerpunkte dieses Abschnitts.

Die konkrete Implementierung des intelligenten Missionsmanagements für die Prototypen *DeepC* und *MauSI* wird im *Kapitel 7* diskutiert. Nach der Vorstellung der Systeme sowie der Software- und Missionsplanstrukturen sind Anmerkungen zur Realisierung des Missionsmanagements und Testergebnisse unterschiedlicher Umplanungssituationen zu finden.

*Kapitel 8* bildet schließlich den Abschluss der Arbeit und wertet die gefundene Lösung des Problems. Auf weitere Problemstellungen, die im Rahmen dieser Arbeit nicht mehr untersucht werden konnten, wird ebenfalls hingewiesen.

Jedes dieser Kapitel wird von einem eigenen Rahmen zur Einführung und Zusammenfassung der wesentlichen Gedanken umschlossen. Damit soll eine Erleichterung des Zugangs zu der entsprechenden Thematik für den Leser erreicht werden.

## 2 Stand der Technik und Ziele der Arbeit

In diesem Kapitel wird zu Beginn ein Überblick zu bekannten Ansätzen für ein intelligentes Missionsmanagement gegeben. Dazu gehört die Einordnung der unterschiedlichen Methoden in die entsprechenden Wissenschaftsgebiete sowie die Diskussion typischer Vertreter. Daran schließen sich die Vorstellung von Softwarearchitekturen für autonome mobile Systeme sowie die Darlegung der konkreten Rahmenbedingungen und der sich daraus ergebenden neuartigen Ziele der vorliegenden Arbeit an.

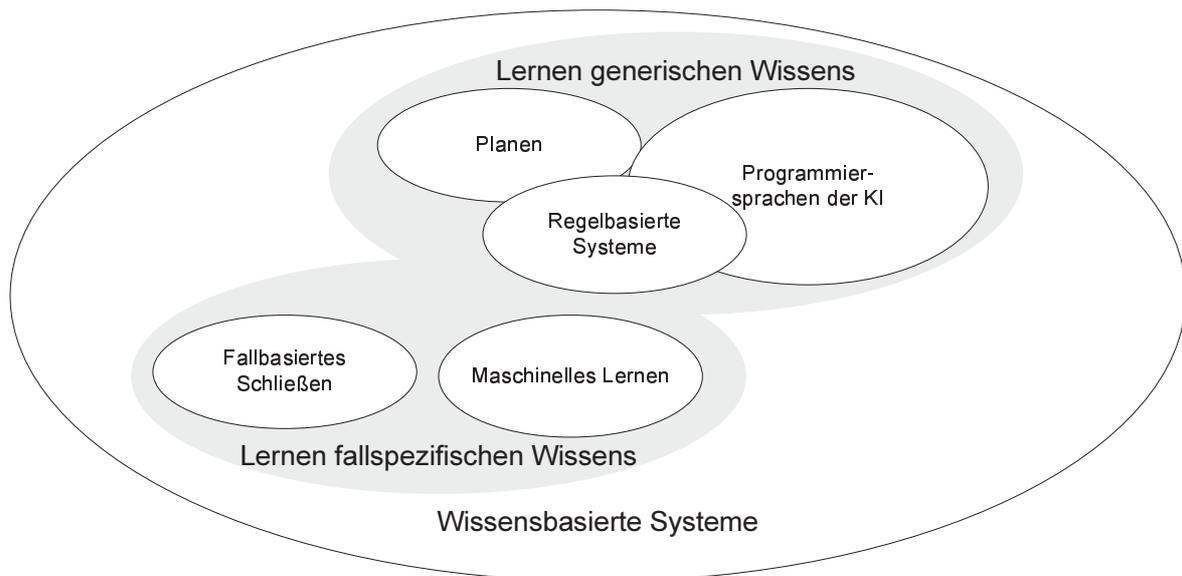
### 2.1 Einordnung bekannter Ansätze zum Missionsmanagement

#### 2.1.1 Übersicht

Entsprechend der Definition 1.1 müssen autonome mobile Systeme Eigenschaften intelligenten Verhaltens aufweisen, um selbständig in einer realen Umgebung Aufgaben erfüllen zu können. Die Grundlagen für intelligentes Verhalten von Maschinen wurden, wie bereits in Kapitel 1 erwähnt, mit der Wissenschaftsdisziplin der künstlichen Intelligenz (KI) geschaffen.

Ein bekanntes Denkmodell und dabei gleichzeitig ein klassischer Prüfstein für künstliches intelligentes Verhalten ist der so genannte Turing-Test [Tur95]. Inhalt des Tests ist ein Vergleich der Fähigkeiten von Mensch und Maschine durch einen menschlichen Richter. Der Richter steht über ein Terminal mit beiden Kontrahenten in Kontakt und soll anhand der geführten Dialoge entscheiden, welcher der Kommunikationspartner der Rechner ist. Inhalt und Ablauf des Dialoges sind dabei im Voraus nicht bekannt. Ist der Richter nicht imstande, eine Entscheidung zu treffen, oder entscheidet er sich falsch, kann der Computer (genauer das eingesetzte Computerprogramm) als intelligent bezeichnet werden.

In den 60er und 70er Jahren standen intelligente Maschinen im Mittelpunkt vielfältiger Forschungsaktivitäten. Getrieben von ersten vielversprechenden Ergebnissen kannte die Hochstimmung zu dieser Zeit keine Grenzen, sodass mutige Zukunftsvisionen eine vollständige Nachbildung des Menschen versprachen. Im so genannten Gordon-Helmer-Report wurden Anfang der 60er Jahre durch die amerikanische Denkfabrik RAND (Research and Development) Prognosen für die Entwicklung der Wissenschaft innerhalb der nächsten 50 Jahre entwickelt [Hel66]. Eine Befragung führender Forscher lieferte die Basis der getroffenen Aussagen. Danach sollten unter anderem bis 1995 eine vollautomatische Fremdsprachenübersetzung mit korrekter Grammatik und Roboter im Haushalt und zur Müllbeseitigung verfügbar sein. Viele der im Bericht vorgestellten Prognosen sind eingetreten, wenn auch nicht immer zum richtigen Zeitpunkt. Gerade die Vorhersagen zur künstlichen Intelligenz sind jedoch - auch durch die Euphorie der Anfangsjahre - weit von der heutigen Realität entfernt. Die Idee einer selbständig denkenden Maschine, die den Turing-Test bestehen könnte, wich der Nutzung der entwickelten Methoden für praktikable Anwendungen. Heutzutage sind intelligente Systeme vielfältig anzutreffen, dabei aber stets auf ein eng umrissenes Aufgabengebiet



**Abb. 2.1:** Einteilung wissensbasierter Systeme aus dem Blickwinkel des intelligenten Missionsmanagements

begrenzt.

Bei der Analyse des aktuellen Wissensstandes auf dem Gebiet des intelligenten Missionsmanagements wurde sich auf Verfahren der KI konzentriert. Mit Hilfe dieser Verfahren sind viele erfolgreiche Missionen autonomer mobiler Systeme absolviert worden, sodass sie als ein, wenn nicht sogar der wesentliche Ansatz für das intelligente Missionsmanagement zu sehen sind.

Ein für diese Arbeit wichtiger Teilbereich der KI beschäftigt sich mit der Erfassung, Hinterlegung und Verarbeitung von Wissen, das von Menschen gewonnen oder aus erlebten Situationen eines autonomen Systems erzeugt wurde. Die resultierenden *wissensbasierten Systeme* gaben diesem Wissenschaftszweig den Namen [BKI00]:

**Definition 2.1** *Wissensbasiertes System*

Ein *wissensbasiertes System* simuliert intelligentes Wissen und Handeln in einem bestimmten Bereich. Dazu muss das Wissen abgebildet und verarbeitet werden.

Ein wesentliches Merkmal wissensbasierter Systeme ist die vorhandene Trennung von Wissensrepräsentation und -verarbeitung, deren Aufbau in Kapitel 5 vorgestellt wird. Beide Elemente können unterschiedlich gestaltet sein; daraus ergeben sich die verschiedenen Methoden wissensbasierter Systeme. Für den Einsatz als intelligentes Missionsmanagement in autonomen mobilen Systemen eignen sich vor allem die in der Abbildung 2.1 dargestellten Verfahren, die im Folgenden näher beleuchtet werden sollen.

*Programmiersprachen der künstlichen Intelligenz* wie LISP oder PROLOG sind aufgrund der integrierten Symbolverarbeitung direkt in der Lage, Schlussfolgerungen aus vorhandenem Wissen zu ziehen. Sie bilden oft die Basis für die Erstellung weiterer Arten wissensbasierter

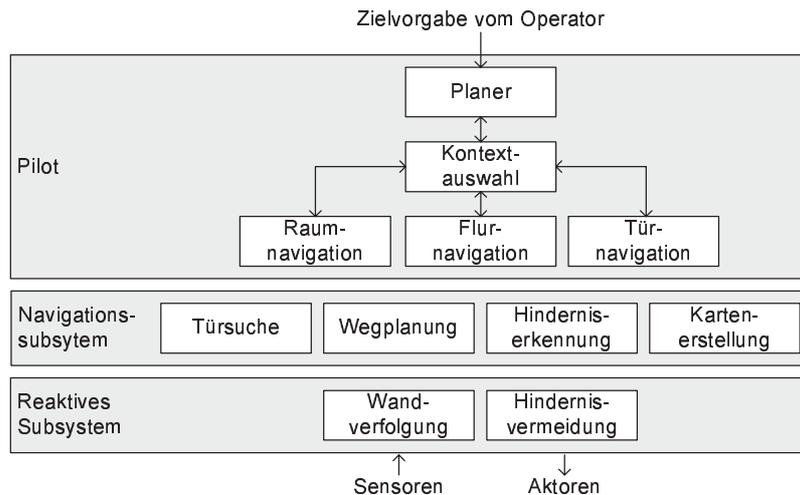
Systeme, wie zum Beispiel für Planungs- oder regelbasierte Systeme. Die enorme Flexibilität dieser Sprachen erlaubt einerseits die Implementierung unterschiedlicher intelligenter Methoden, erfordert aber andererseits einen beträchtlichen Erstellungs- und Wartungsaufwand. In diese Klasse fallen Ansätze für das intelligente Missionsmanagement wie der des autonomen Unterwasserfahrzeugs Phoenix oder SAUSAGES, die im Abschnitt 2.1.2.2 näher beschrieben werden.

*Regelbasierte Systeme*, häufig auch als Produktionssysteme bezeichnet, besitzen als Wissensverarbeitungseinheit ein hocheffizientes Inferenzsystem, das die hinterlegten Regelwerke auf abzuarbeitende Regeln prüft und realisiert. Jede Regel liegt gut verständlich als generisches Wissen in einer Wenn-Dann-Form vor. Durch das Vorhandensein des Inferenzsystems, das den kompletten Arbeitsablauf steuert, ist ein Einsatz für die unterschiedlichsten Problemfälle in kürzester Zeit möglich. Voraussetzung dafür ist, dass das Wissen in Form von Regeln existiert. Ist das nicht der Fall, muss durch den aufwendigen Prozess der Wissensermittlung erst ein Regelwerk generiert werden. Die meisten vorhandenen regelbasierten Systeme nutzen deterministische Regeln, die mit klassischer Logik bearbeitet werden können. Darüber hinaus existieren auch Systeme zur Verarbeitung unsicheren Wissens [Orc98], [SKLC96]. Ein bekanntes Beispiel für die Anwendung vager Informationen ist das medizinische Expertensystem MYCIN [BS85] zur Diagnose und Therapie von Infektionskrankheiten. Auf dem Gebiet der mobilen Robotik wurden und werden regelbasierte Systeme aufgrund ihres universellen Charakters für unterschiedlichste Aufgaben eingesetzt. In [BPC94] wird ein Führungssystem für einen autonomen mobilen Roboter vorgestellt, das alle höheren Planungs- und Führungsaufgaben mit Hilfe eines regelbasierten Systems löst. Abschnitt 2.1.2.1 geht näher auf dieses System ein. Andere Anwendungsgebiete sind zum Beispiel die Bildauswertung und die daraus folgende Bewegungsplanung eines mobilen Systems [LA03].

Handlungsabläufe autonomer mobiler Systeme können oft auch durch klassisches *Planen* erzeugt werden. Ausgangspunkt ist dabei ein Anfangs- und ein Zielzustand des Roboters. Planungssysteme wie zum Beispiel STRIPS generieren aus einem Pool verfügbarer Aktionen eine Kette von Handlungen, die den Anfangs- in den Zielzustand überführen. Komplexere Zusammenhänge sowie bedingte Aktionen sind mit einem solchen klassischen Planer nicht realisierbar. In den letzten Jahren wurden unterschiedliche Ansätze untersucht, die eine Erweiterung des einem Planungssystem zugrunde liegenden Situationskalküls vornehmen [Bee02]. Damit sind auch verschiedene Programmstrukturen wie die bedingte Ausführung von Aktionen oder rekursive Prozeduren realisierbar. Ein typisches Beispiel für ein derart verbessertes System ist GOLOG [LRL<sup>+</sup>97]. Es wird im Abschnitt 2.1.2.4 vorgestellt.

*Maschinelles Lernen* versucht, den Prozess menschlichen Lernens zu modellieren. Es existieren unterschiedlichste Konzepte, so zum Beispiel das Lernen von Entscheidungsbäumen oder die automatische Wissensfindung in Datenbeständen. Bisher werden Verfahren des maschinellen Lernens für das Missionsmanagement in autonomen mobilen Systemen kaum eingesetzt. Ein Grund dafür dürfte die Menge an Trainingsbeispielen sein, die für das Lernen benötigt wird.

*Fallbasiertes Schließen* erzeugt Wissen durch Sammeln von Fällen mit spezifischen Erfahrungen. Die Lösung neuer Probleme basiert auf der Auswahl des relevantesten Falls aus der Wissensbasis und der Anwendung der gespeicherten Lösung in angepasster Form. Ein solches System kann auch als erinnerungsbasiertes System bezeichnet werden (im Gegensatz zum regelbasierten System mit generischem Wissen). Wie auch beim maschinellen Lernen



**Abb. 2.2:** Systemarchitektur von AMOR

besteht ein hoher Bedarf an Erlebtem, um eine sinnvoll nutzbare Falldatenbank zu erzeugen. Im Abschnitt 2.1.2.3 wird ein Missionsplanungssystem auf der Grundlage fallbasierten Schließens vorgestellt.

## 2.1.2 Missionsmanagement ausgewählter autonomer mobiler Systeme

Nachfolgend werden exemplarisch Steuerungsarchitekturen mobiler Systeme vorgestellt, die sich in die im vorangehenden Abschnitt entwickelte Struktur für intelligentes Missionsmanagement einordnen lassen. Den Schwerpunkt der Darstellung bildet dabei die Anpassungsfähigkeit der geplanten Mission während ihrer Realisierung.

### 2.1.2.1 Regelbasierte Systeme

In [BP94] wird das Navigationssystem AMOR für autonome mobile Roboter als eine hybride Architektur vorgestellt, die verhaltensbasierte Ansätze für lokale Aktivitäten mit einem regelbasierten Expertensystem für die Planung und Kontrolle der Mission verknüpft (Abbildung 2.2). Unter Vernachlässigung der Schnittstellenmodule für die Hardware des Roboters sind im Wesentlichen drei Schichten in dem hierarchisch strukturierten System vorzufinden.

Die als Pilot bezeichnete regelbasierte Schicht beinhaltet die Module für die möglichen Navigationskontexte des Roboters Raum-, Flur- und Türnavigation, die je nach aktueller Situation über die Kontextauswahl aktiviert werden. Der Planer generiert aus einer vorgegebenen, globalen Zielstellung des Operators eine Folge von Aufgaben, die sequentiell zu lösen sind. Die Zielstellung umfasst dabei die Angabe der zu erreichenden Position, sodass die Planungsaufgabe als abstrakte Wegplanung aufgefasst werden kann.

Entsprechend des ausgewählten Navigationskontextes arbeiten einzelne Module des Navigationssubsystems konkurrierend und werden über das Subsystem Pilot koordiniert. Das reaktive Subsystem implementiert verschiedene Verhaltensmuster, die für die sichere Navigation erforderlich sind.

In Programmauszug 2.1 ist eine Regel des Moduls Türnavigation dargestellt. Hier wird deutlich, dass eine primär prozedurale Funktionsweise des regelbasierten Systems entwickelt

**Programmauszug 2.1:** Regel für das Passieren einer Tür in AMOR (Pseudocode)

```

(defrule DOOR::controlloop "Door traversing"
  (phase navigate)
=>
  (while (or (MissionImpossible) (not (DoorTraversed)))
    (FindDoor)
    (GetDoorPosition)
    (MoveTowardsDoor)
    (if (DoorLost) then
      (LookOtherWay)
    )
  )
)

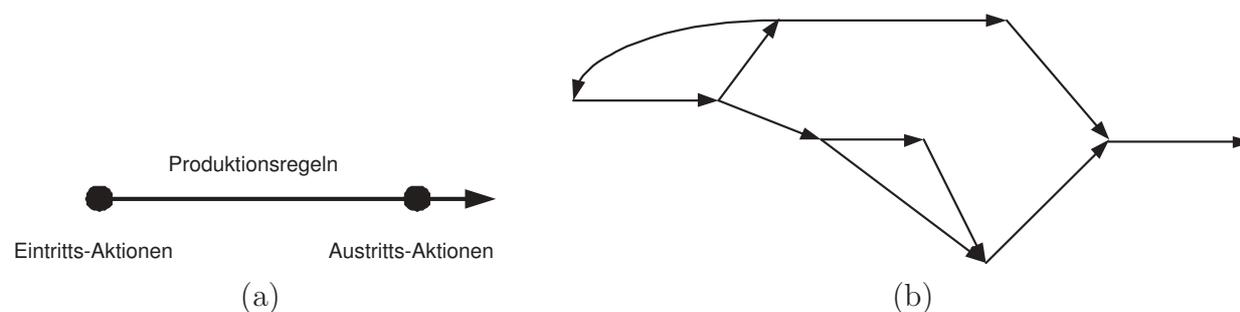
```

wurde. Auch die weiteren Module arbeiten auf diese Art. Die Vorteile, die regelbasierte Systeme gegenüber klassischen prozeduralen Sprachen bieten, wurden in AMOR nicht genutzt. Die sich aus der Zielstellung des Operators ergebende Folge abzuarbeitender Aktionen kann vom mobilen System nicht selbständig an die aktuelle Situation angepasst werden. So wird im Fall unerwarteter Ereignisse, wie zum Beispiel dem Auftauchen einer geschlossenen Tür, lediglich eine Meldung an den Operator abgesetzt und auf neue Anweisungen gewartet.

## 2.1.2.2 Programmiersprachen der künstlichen Intelligenz

*SAUSAGES*. Das „System for AUtonomous Specification, Acquisition, Generation and Execution of Schemata“ (SAUSAGES) wurde an der Carnegie Mellon Universität (USA) entwickelt [Gow97]. Ausgangspunkt waren die Probleme, die bei der Überführung von unter Laborbedingungen getesteten Robotern in die natürliche Umgebung auftraten. Die vorgefundene unstrukturierte Welt erschwerte die Erkennung derjenigen Ereignisse, die zur Weiterschaltung der Aufgaben innerhalb eines Planes führten.

Deshalb wurde ein neues Softwaresystem zur Ausführung komplexer Missionen erstellt. Wesentliche Erkenntnis aus der vergangenen Forschung war, dass ein Plan benötigt wird, der das Verhalten der unterlagerten (low-level) wahrnehmenden und agierenden Module (Sensorik und Aktorik) durch Nutzung von übergeordnetem (high-level) Wissen (Ziele und Absichten der Mission) beeinflusst. Das übergeordnete Wissen liegt in Form eines Missionsplanes vor. Dieser Plan besteht aus diskreten semantischen Einheiten, die auch als Links bezeichnet werden. Die Mission kann damit auch als gerichteter Graph bestehend aus einzelnen Links dargestellt werden (Abbildung 2.3).



**Abb. 2.3:** SAUSAGES: Link (a) und Missionsplan bestehend aus Links (b)

Jeder Link stellt ein Element im Missionsplan dar, das gekennzeichnet ist durch:

- Eintrittsaktionen, die beim Start des Links ausgeführt werden;
- Austrittsaktionen, die beim Verlassen des Links ausgeführt werden;
- ein Satz von Regeln für
  - die Ermittlung des Endes des Links (Austrittsbedingung) und
  - die Ermittlung der fehlerhaften Ausführung des Links (Abbruchbedingung);
- den Verweis auf den nächsten Link im Plan - bedingte Verweise, die je nach angegebener Bedingung zu unterschiedlichen Links führen, sind ebenfalls möglich;
- eine Liste von Sub-Links, um einen hierarchischen Plan zu generieren.

Der Aufbau der Links ermöglicht eine hohe Flexibilität bei der Zusammenstellung der Missionspläne. So kann durch gewissenhafte Planung eine Vielzahl unterschiedlicher Plantypen generiert werden. Nachteil der Vorgehensweise ist der enorme Aufwand, der bei der Bestimmung der unterschiedlichen Regeln und Aktionen betrieben werden muss. Für eine sinnvolle Nutzung der Planungssprache sind unter anderem eine ausreichende Einarbeitungszeit in die Logik sowie längere Vorbereitungszeiten der einzelnen Missionen notwendig. Außerdem muss die gesamte Softwarearchitektur des autonomen mobilen Systems für den Einsatz eines derartigen Systems ausgelegt sein, da die Planrepräsentation in LISP vorliegt und damit auch die Planrealisierung in dieser Sprache erstellt werden muss (Programmauszug 2.2).

---

**Programmauszug 2.2:** *Beispiel für einen SAUSAGES-Missionsplan*

---

```
(defplan the-plan ()
  (sequence
    (link user-wait (message "Waiting for start mission") (plan-id 0)
      (position '(500.0 250.0)))

    (link xcountry
      (points (vector '(500.0 250.0)
                     '(500.0 1000.0)
                     '(500.0 1900.0)))
      (initial-speed 3.0)
      (plan-id 1))

    (link user-wait (message "Done")
      (position '(500.0 1900.0))
      (plan-id 2))
  )
)
```

---

*Missionsmanagement des AUV Phoenix.* Die Missionsführung des an der Naval Postgraduate School Monterey (USA) entwickelten autonomen Unterwasserfahrzeugs Phoenix ist ein weiteres Beispiel für den Einsatz von Programmiersprachen der künstlichen Intelligenz. Der Missionsplan liegt hier in Form von Regeln und Fakten in der Programmiersprache PROLOG vor [BHMM97]. Aus den Fakten und Regeln ermittelt die Missionsführung Befehle für die unterlagerte Steuerung des Roboters. In Programmauszug 2.3 ist ein Teil eines solchen

Missionsplanes dargestellt. Daran lässt sich erkennen, dass für ein auszuführendes Manöver - hier das Anfahren eines Wegpunktes - ein Satz von Prädikaten erforderlich ist. Ein entscheidender Nachteil dieses Systems wird dabei ersichtlich: Der Entwurf eines Missionsplanes, in dem für jedes Manöver ein alternatives Ende bei Nichterreichen des Manöverzieles vorgegeben werden muss, gestaltet sich für einen Operator nicht einfach. Durch den Einsatz einer Planungsoberfläche wurde der Prozess erheblich vereinfacht, da diese aus den Vorgaben des Operators automatisch den notwendigen PROLOG-Code generiert [Leo96].

---

**Programmauszug 2.3:** *Beispiel für einen Missionsplan in Prolog (Auszug)*

---

```
%      Go to next waypoint

execute_phase(6) :- nl, printsc('PHASE 6 STARTED. '),
                   ood('waypoint -2 5 2',X),X==1,
                   printsc('waypoint -2 5 2 !'),
                   ood('start_timer 120',X),X==1,
                   repeat, phase_completed(6).

phase_completed(6) :- ood('ask_waypt_reached',X),X==1,
                    printsc('WAYPOINT REACHED. '),
                    asserta(complete(6)).

phase_completed(6) :- ood('ask_time_out',X),X==1,
                    printsc('PHASE 6 ABORTED DUE TO TIME OUT. '),
                    asserta(abort(6)).

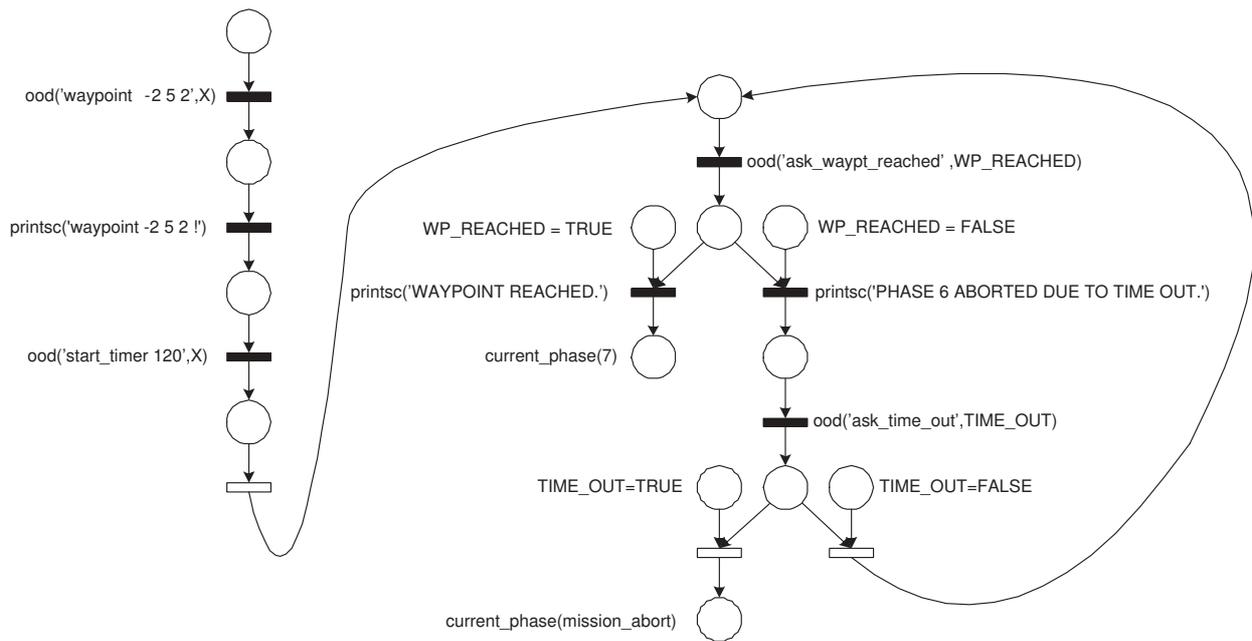
next_phase(6) :- complete(6),
                retract(current_phase(6)),
                asserta(current_phase(7)).

next_phase(6) :- abort(6),
                retract(current_phase(6)),
                asserta(current_phase(mission_abort)).
```

---

In [OPSS98] wird CORAL vorgestellt, ein auf Petri-Netzen basierendes System zur Missionsplanung und -ausführung. Es implementiert eine ähnliche Funktionalität wie die dargestellte PROLOG-basierte Missionsführung. Zur Demonstration seiner Leistungsfähigkeit wurde CORAL auf das gleiche AUV portiert und in typische Missionen getestet [HMO<sup>+</sup>96]. Dabei wurde deutlich, dass das Petri-Netz die Zusammenhänge zwischen den einzelnen Abschnitten der Mission besser veranschaulicht und sich einfacher während der Vorbereitung einer Mission erstellen lässt. Abbildung 2.4 zeigt den zu Programmauszug 2.3 korrespondierenden Teil eines Missionsplanes. Ein entscheidender Nachteil des Petri-Netzes ist jedoch, dass der Umfang des Planes enorm anwächst. Ab einer bestimmten Größe ist die Überschaubarkeit kaum noch gewährleistet.

In beiden Ansätzen sind Methoden zur Planmodifikation nicht vorgesehen, obwohl eine Erweiterung ihrer Methoden um adaptive Fähigkeiten prinzipiell möglich ist. Deshalb muss der Missionsplan neben der eigentlichen Abfolge der Aktionen auch die Alternativen bei Auftreten von Problemen (technischer Natur oder durch Umwelteinflüsse) enthalten. Dadurch verkompliziert sich die off-line durchzuführende Missionsplanung erheblich. Vom Operator werden umfangreiche Kenntnisse der Arbeitsweise des Systems sowie eine genaue Analyse der Umweltbedingungen verlangt, um eine erfolgreiche Missionsausführung zu gewährleisten.



**Abb. 2.4:** Missionsplan als Petri-Netz

### 2.1.2.3 Planungssysteme auf der Basis fallbasierten Schließens

Mittels fallbasiertem Schließen soll Erfahrungswissen von Menschen oder Robotern gespeichert werden, um in späteren Situationen darauf zurückgreifen zu können. Dazu werden erlebte Situationen in Form von Fällen in einer Datenbank hinterlegt. Die Suche und der Zugriff auf Fallbeispiele erfolgt über eine Indizierung. Der vergebene Index muss ein Ähnlichkeitsmaß widerspiegeln, damit nach einem der aktuellen Situation ähnlichen Fall gesucht werden kann. In den letzten Jahren wurde verschiedene Planungssysteme auf der Basis fallbasierten Schließens entworfen, unter anderem auch für autonome mobile Roboter.

Das in [VG94] vorgestellte System nutzt fallbasiertes Schließen für die Erstellung von Missionsplänen und die Generierung fahrbarer Routen für ein autonomes Unterwasserfahrzeug. Ausgangspunkt ist die Definition globaler Missionsziele wie zum Beispiel „Vermesse das Temperaturprofil des Gebietes X“. Als Beschränkungen können Prioritäten und verfügbare Ressourcen übergeben werden. Der Missionsplaner analysiert die vorhandenen Informationen und generiert daraus den Missionsplan, indem die Plandatenbank nach ähnlichen Fällen durchsucht wird. War die Suche erfolgreich, geschieht anschließend die Anpassung des Falls an die Missionsziele (zum Beispiel an andere geografische Positionen) und die Planrealisierung.

Ein weiteres Beispiel für den Einsatz fallbasierten Schließens ist der als „Mission Expert“ bezeichnete Assistent für das grafische Missionsplanungssystem MissionLab [ACE99]. MissionLab dient der Spezifikation, Simulation und Kontrolle von Missionen für unterschiedliche Typen mobiler Roboter. Der „Mission Expert“-Assistent vereinfacht den Planungsprozess, da der Operator zu einem grob spezifizierenden neuen Einsatz eine Auswahl an erfolgreich absolvierten Missionen erhält. Diese werden anhand ihrer Ähnlichkeit zum vorliegenden Fall selektiert und an diesen angepasst.

Fallbasiertes Schließen ist in den bekannten Einsatzfällen nur zum Entwurf eines Missionsplanes genutzt worden. Die Modifikation eines existierenden Planes aufgrund veränderter Rahmenbedingungen ist bisher noch nicht untersucht worden. Nachteil einer Planerstellung auf Basis bekannter, ähnlicher Situationen ist, dass dafür eine Vielzahl von Beispielen in der zugehörigen Datenbank hinterlegt sein müssen. Doch gerade für eingetretene unbekanntere Situationen, für die noch keine ähnlichen Fallbeispiele vorliegen, ist eine Anpassung des Missionsplanes notwendig. Hier können Systeme auf der Basis fallbasierten Schließens allein keine Lösung liefern. Eine zu untersuchende Möglichkeit wäre die Kombination mit einem weiteren Planungssystem, das nicht auf derartige Vorinformationen angewiesen ist.

#### 2.1.2.4 Missionsmanagement durch Planen

Klassische Planungssysteme wie STRIPS generieren eine Folge von Aktionen, den Plan, um einen gegebenen Anfangszustand in den gewünschten Endzustand zu überführen. Der Plan wird mittels Vorwärtssuche (ausgehend von den Startbedingungen) oder durch Rückwärtssuche (beginnend beim Zielzustand) durch Aneinanderreihung der benötigten Aktivitäten erzeugt. Es entsteht ein vollständiger, total geordneter Plan, der im Anschluss realisiert werden kann.

Ein entscheidender Nachteil der klassischen Planer ist, dass bedingte Aktionen (zum Beispiel in Abhängigkeit von Sensorinformationen aus der Umwelt) nicht verwendet werden können. Deshalb wurden neue Sprachen wie GOLOG und ihre Derivate ConGOLOG oder pGOLOG entwickelt, die eine Beschreibung komplexer Zusammenhänge erlauben [Gro02]. Die praktische Einsatzfähigkeit von GOLOG wurde unter anderem im Deutschen Museum Bonn demonstriert, indem ein Roboter als Museumsführer Besucher mit verschiedenen Exponaten vertraut machte [BCF<sup>+</sup>98].

Die Planspezifikation erfolgt in GOLOG in Form von Prolog-Klauseln (Programmauszug 2.4). Im gezeigten Beispiel soll ein Roboter Kaffee an bestimmte Personen (entspricht dem Zielzustand) ausliefern. Mit Hilfe komplexer Aktionen wie `deliverCoffee(T)` oder `goto(L,T)` sowie der zugehörigen Voraussetzungen und Effekte kann der Plan für das Erreichen des angestrebten Endzustandes generiert und realisiert werden. Anhand des Beispiels wird deutlich, dass die Erstellung eines solchen Missionsplanes genaue Kenntnis über die Arbeitsweise des zugrunde liegenden PROLOG-Interpreters und der Programmierung der GOLOG-spezifischen Routinen erfordert. Das System bietet eine sehr hohe Flexibilität im Hinblick auf die Missionsplanung, benötigt aber andererseits einen großen Aufwand für die Einarbeitung sowie die Erstellung der Missionen. Zusätzlich ist eine Implementierung des gesamten Missionsmanagements in der Sprache des Interpreters notwendig.

Ein weiterer Nachteil klassischer Planer ist, dass ein gültiger Plan erst nach Abschluss des Algorithmus vorliegt. Deshalb wurden Verfahren entwickelt, die jederzeit eine nächste auszuführende Aktion liefern können. Diese als Anytime-Planer bezeichneten Verfahren finden in kürzester Zeit eine - nicht unbedingt optimale - Lösung, die mit zunehmender Rechenzeit weiter verbessert wird. Gerade für autonome mobile Systeme ist diese Eigenschaft besonders wichtig. So kann es bei klassischen Planern passieren, dass in einer unerwarteten Situation die notwendigen Vorbedingungen für die nächste Aktion nicht mehr erfüllt sind und eine komplette Neuplanung durchgeführt werden muss. Während dieser Zeit kann das System keine Aktivität ausführen. Anytime-Planer gewährleisten hingegen die kontinuierliche Arbeitsweise des Systems.

**Programmauszug 2.4:** *Beispiel für einen Missionsplan in GOLOG (Auszug)*


---

```

% Sequential, Temporal Golog Program for a Coffee Delivery Robot
% GOLOG Procedures
proc (deliverCoffee(T),
  ?(some(t, now(t) & t $<=T)) :
  (? (all(p, all(t1, all(t2, wantsCoffee(p, t1, t2) => hasCoffee(p))))))
  #
  pi(rloc, ?(robotLocation(rloc)) :
    if(rloc = cm, /* THEN */ deliverOneCoffee(T),
      /* ELSE */ goto(cm,T) : pi(t, ?(now(t)) :
        deliverOneCoffee(t))) :
    pi(t, ?(now(t)) : deliverCoffee(t))).
proc (goto(L,T),
  pi(rloc, ?(robotLocation(rloc)) : pi(deltat, ?(travelTime(rloc, L, deltat)) :
  goBetween(rloc, L, deltat, T))).
...

% Preconditions for Primitive Actions.
poss (pickupCoffee(T), S) :- not holdingCoffee(S),
  robotLocation(cm, S).
poss (giveCoffee(Person, T), S) :- holdingCoffee(S),
  robotLocation(office(Person), S).
...

% Successor State Axioms.
robotLocation(Loc, do(A, S)) :- A = endGo(Loc1, Loc, T) ;
  (robotLocation(Loc, S),
  not A = endGo(Loc2, Loc3, T)).
going(Loc1, Loc2, do(A, S)) :- A = startGo(Loc1, Loc2, T) ;
  (going(Loc1, Loc2, S),
  not A = endGo(Loc1, Loc2, T)).
...

% Initial Situation.
robotLocation(cm, s0).
start(s0, 0).
wantsCoffee(sue, 140, 160).
wantsCoffee(bill, 100, 110).
wantsCoffee(joe, 90, 100).
wantsCoffee(mary, 130, 170).
...

```

---

Das in [Fer02] vorgestellte Verfahren kombiniert diese Technik mit einem Verfahren zur Bestimmung von Teilplänen, sodass die Planung nur für einen Ausschnitt aus dem Gesamtplan durchgeführt werden muss. Der nicht mehr gültige Teil des Missionsplanes wird vom Planungsalgorithmus durch einen neuen Teilplan ersetzt, mit dem die Mission fortgeführt werden kann. Das Verfahren kann auch zur Verbesserung eines gültigen Missionsplanes eingesetzt werden, indem ungünstige Sequenzen von Handlungen durch verbesserte Versionen ersetzt werden. Auch für diesen Ansatz gilt, dass sowohl das Missionsmanagement selbst als auch der Missionsplan in der gewählten Programmiersprache des Planungsalgorithmus zu hinterlegen sind.

### 2.1.3 Bewertung

Aus der Vielzahl der verfügbaren Ansätze für ein intelligentes Missionsmanagement wurden exemplarisch einige Verfahren ausgewählt und vorgestellt. Gerade durch die unüberschaubare Menge an autonomen Systemen, deren unterschiedlichste Aufgaben und die meist optimal

daran angepassten Strukturen und Methoden der Fahrzeugführung ist ein objektiver Vergleich untereinander kaum möglich.

Ein Anpassung des Missionsplanes an die aktuelle Situation ist in den wenigsten Systemen zu finden. Üblicherweise wird ein Plan vor dem Start des Systems erzeugt und im Anschluss daran realisiert. Treten Probleme auf, erfolgt ein Missionsabbruch. Weitergehende, vielversprechende Ansätze, wie zum Beispiel die Planungsmethoden aus Abschnitt 2.1.2.4, erfordern stets die Anpassung der Softwarearchitektur an die Spezifik des Planers, sodass eine einfache Erweiterung bestehender mobiler Systeme um ein intelligentes Missionsmanagement nahezu unmöglich ist.

Darüber hinaus benötigen die vorgestellten Verfahren häufig gut ausgebildetes Fachpersonal zur Bedienung, das mit der Programmierung des jeweiligen Systems vertraut sein muss. Gerade im Hinblick auf kommerziell nutzbare autonome mobile Systeme sind derartige Anforderungen jedoch nicht immer erfüllbar. Das in der vorliegenden Arbeit zu entwerfende Konzept eines intelligenten Missionsmanagements versucht, all diese Unzulänglichkeiten auszugleichen und sich dabei nahtlos in eine bestehende Softwarearchitektur zu integrieren.

## 2.2 Softwarearchitektur

Ein in der mobilen Robotik häufig verwendetes Architekturmodell der Software besteht in einer Mehrschichten-Struktur, bei der jede Schicht ein konkret definiertes Aufgabenspektrum zu lösen hat. Diese Einteilung beim Problemlösen kann auf einen grundlegenden Ansatz zurückgeführt werden, der für die Analyse von Entscheidungen menschlicher Operatoren in Fehlersituationen von Rasmussen entworfen wurde [Ras83]. Demzufolge lässt sich die gesamte Bandbreite an Fehlleistungen durch den Menschen in mehrere Ebenen unterteilen [Rea94]. Diese Ebenen sollen nachfolgend vorgestellt werden.

### 2.2.1 Das menschliche Handlungsmodell nach Rasmussen

Das Handlungsmodell nach Rasmussen entwickelte sich in den letzten Jahren zu einem Standard bei der Untersuchung der menschlichen Zuverlässigkeit und spielt eine große Rolle für den Entwurf von Benutzeroberflächen. Dabei besteht das Ziel darin, potentielle Fehlerquellen bei der Bedienung eines Systems aufzudecken und zu beseitigen [SAFG97]. Für den Vergleich dieses Modells mit existierenden Softwarearchitekturen autonomer mobiler Systeme ist eine Betrachtung der auftretenden Fehler und ihrer Ursachen, die einen wesentlichen Aspekt der Arbeit von Rasmussen repräsentieren, nicht notwendig. Deshalb sollen lediglich der Informationsfluss und die kognitiven Aktivitäten des Menschen dargestellt werden.

Die Untersuchungen von Rasmussen anhand verbaler Protokolle von Technikern ergaben, dass die kognitiven Kontrollmechanismen des Menschen in den folgenden drei Ebenen ablaufen (Abbildung 2.5). Auf der *fähigkeitsbasierten Ebene* erfolgen die Handlungen automatisiert und routiniert, sie gehen nicht direkt in das Bewusstsein des Menschen ein. Mit Hilfe einer Zuweisung von Aktionen zu eingehenden Signalen werden Prozesse auf unterstem Niveau gesteuert. Lediglich die bei der Generierung von abstrakten Merkmalen aus den eingehenden Sensorinformationen erzeugten Zeichen spielen eine Rolle für die Auswahl des eingesetzten Verhaltensmusters.

Die nächsthöhere, *regelbasierte Ebene* bestimmt aus den Zeichen eine Folge von abzuarbei-

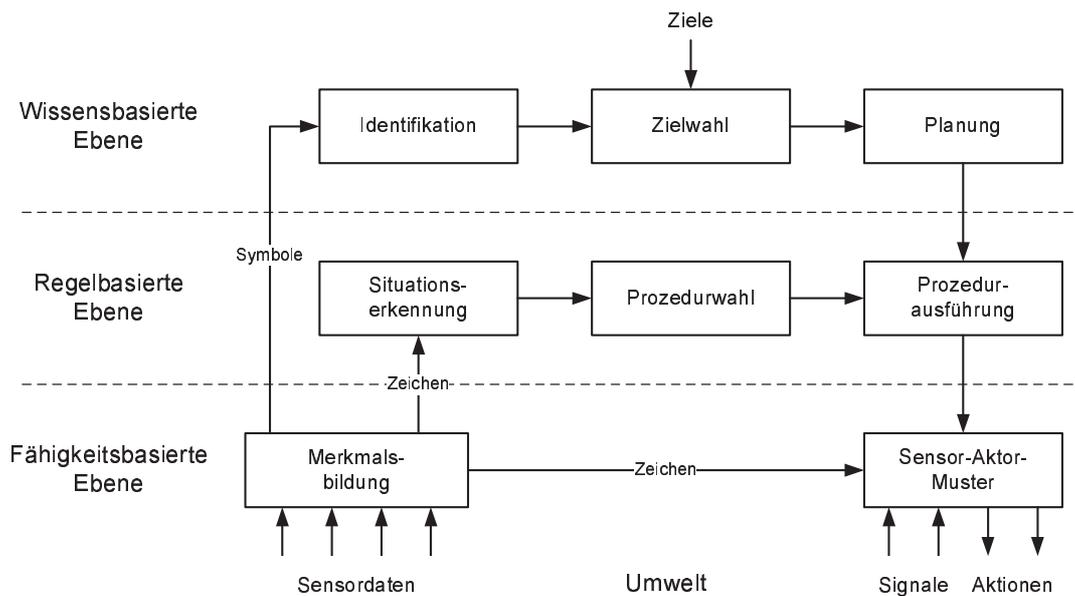


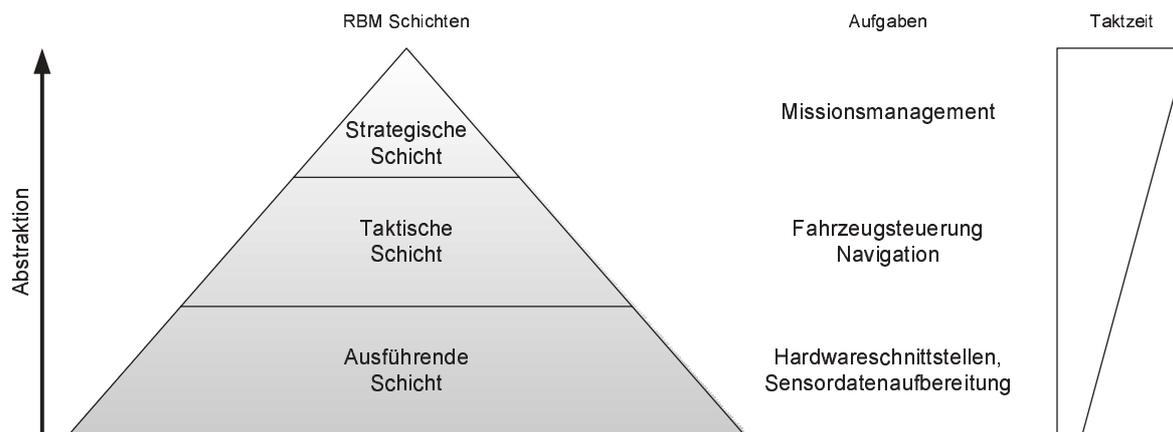
Abb. 2.5: Menschliches Handlungsmodell nach Rasmussen

tenden Regeln, die auch als Prozedur bezeichnet werden kann. Diese Prozedur wird dann auf die aktuelle Situation angewendet. Bekannte Probleme, für die Lösungen in Form von Regeln bereits vorliegen, können auf diese Weise bearbeitet werden.

Liegt eine gänzlich unbekannte, neuartige Situation vor, betrachtet der Mensch das Problem in seiner Gesamtheit und versucht, mit Hilfe diagnostischer und analytischer Methoden ein Schema oder Plan für die Beherrschung der Aufgabe zu generieren. Übergeordnete Zielstellungen spielen dabei eine große Rolle bei der Entwicklung des Planes. Die Prozesse laufen auf der *wissensbasierten Ebene* ab und verwenden symbolische Beschreibungen für die Identifikation der aufgetretenen Situation. Der erzeugte Plan setzt sich aus den auf der regelbasierten Ebene vorhandenen Prozeduren zusammen und wird anschließend als Handlungsschema eingesetzt.

Die auf jeder Ebene verwendeten Informationen stellen jeweils die bestmögliche Beschreibung für die auszuführenden Aktivitäten dar. Die einlaufenden *Signale* als eine kontinuierliche, quantitative Repräsentation der Umweltinformationen dienen zur direkten Steuerung der physikalischen Aktionen des Menschen. Aus ihnen lassen sich *Zeichen* ableiten, die zur Aktivierung oder Modifikation der vordefinierten Prozeduren eingesetzt werden. Sie beziehen sich auf Zustände oder geeignete Verhaltensweisen. Für höhere Denkprozesse sind Zeichen dagegen weniger geeignet. Hier werden *Symbole* als Darstellungsform verwendet. Mit Hilfe einer symbolischen Beschreibung sind kausale Zusammenhänge erkennbar, die bei der Problemlösung eine wesentliche Rolle spielen.

Dieses Modell eignet sich sehr gut zur Abbildung des komplexen menschlichen Verhaltens bei der Problemlösung. Insbesondere in der Auswertung von Unfällen und Katastrophen wird häufig auf derartige Modelle zurückgegriffen, um die verschiedenen Fehler des Menschen, die zu derartigen Situationen führen oder ihre rasche Bewältigung verhindern, aufzudecken und Maßnahmen zu ihrer Vermeidung zu ergreifen [Kle01].



**Abb. 2.6:** Rational Behavior Model

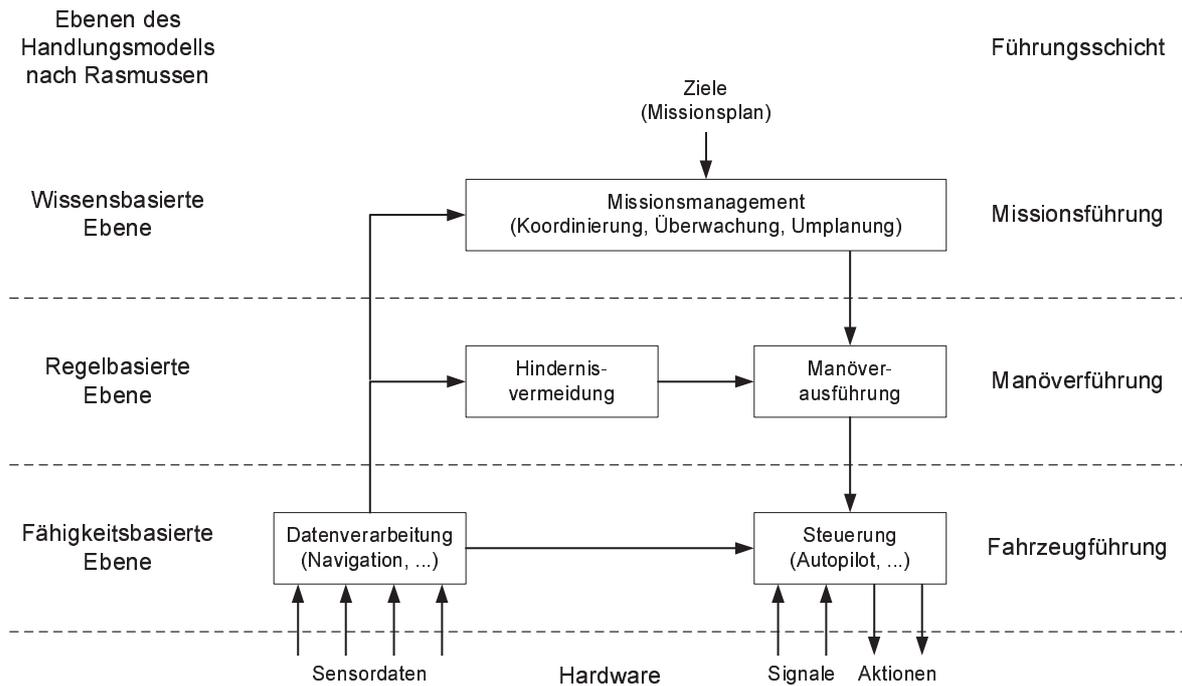
### 2.2.2 Softwarearchitekturen autonomer mobiler Systeme

Softwarearchitekturen für autonome mobile Systeme lassen sich in ein weites Spektrum unterteilen. Dies reicht von hierarchischen bis zu reaktiven Strukturen [Bru94]. Hierarchien sind dabei mit symbolischer Beschreibung, Zielorientierung und geplanten Handlungen verbunden. Nachteilig wirken sich dabei das starre Konzept, ausbleibende Entscheidungen in unerwarteten Situationen sowie die Rechenzeitintensität aus. Reaktive Architekturen arbeiten demgegenüber datengetrieben, häufig unter Einsatz verhaltensbasierter Methoden. Ein Ziel bei der Entwicklung ist auch die Umgehung der Lösung meist komplexer Planungsaufgaben. Damit geht einher, dass die Handlungsweise reaktiver Systeme oft zufallsbestimmt wirkt. Eine Erweiterung im Hinblick auf die Erfüllung komplexer Missionen ist für diese Strukturen mit einem großen Aufwand verbunden.

Durch die Kombination beider Extreme entstehen hybride Architekturen, die die Vorteile beider Strukturen vereinen, ohne auch deren Nachteile zu übernehmen. Ein typischer Vertreter einer derartigen hybriden Softwarearchitektur ist das so genannte Rational Behavior Model [BHMM97]. Es definiert eine Drei-Schichten Softwarearchitektur zur Steuerung von autonomen Systemen (Abbildung 2.6), die sich an den militärischen Führungsebenen in der Marine orientiert [Hol95]. Jede Schicht besitzt hier ein konkret definiertes Aufgabenfeld. So wird ausschließlich in der strategischen Schicht mit dem globalen Ziel der Mission des autonomen mobilen Systems gearbeitet. Aus der strategischen Schicht werden unterlagerte, lokale Aufgaben an die taktische Schicht übergeben, die wiederum konkrete Steuerkommandos für die Aktoren an die ausführende Schicht weiterreicht. Umgekehrt werden Sensordaten bis in die oberste Schicht übergeben, wobei von Schicht zu Schicht zusätzliche Informationen - zum Beispiel durch Sensordatenfusion oder Zustandsschätzung - hinzukommen können.

Charakteristisch für ein solches Strukturmodell ist, dass die Taktzeiten der einzelnen Module mit zunehmender Abstraktion immer größer werden. Damit wird den unterschiedlichen Zeitfenstern zur Entscheidungsfindung Rechnung getragen: Ein Ausweichverfahren zur Verhinderung eines Zusammenstoßes mit einem Hindernis muss beispielsweise in kürzeren Zeiten eine Lösung für das Problem finden als ein Modul zur Anpassung der globalen Missionsziele.

Die drei Schichten des Rational Behavior Model lassen sich den Ebenen des Handlungsmodells



**Abb. 2.7:** *Verwendete Softwarearchitektur der Testplattformen (Auszug)*

dells von Rasmussen direkt zuordnen. So werden die strategischen Aufgaben vom Menschen auf der wissensbasierten Ebene und die taktischen auf der regelbasierten Ebene gelöst. Die ausführende Schicht entspricht schließlich der fähigkeitsbasierten Ebene. In analoger Weise lassen sich auch andere Architekturen auf das Handlungsmodell zurückführen [Gul95], [RYSB00], [XZF04].

Das Modell von Rasmussen ist damit als Grundlage für den Entwurf einer Softwarearchitektur für einen autonomen Roboter direkt geeignet. Auch die für die Anwendung des intelligenten Missionsmanagements ausgewählten Testplattformen verwenden eine dementsprechende Architektur, die im nachfolgenden Abschnitt vorgestellt wird.

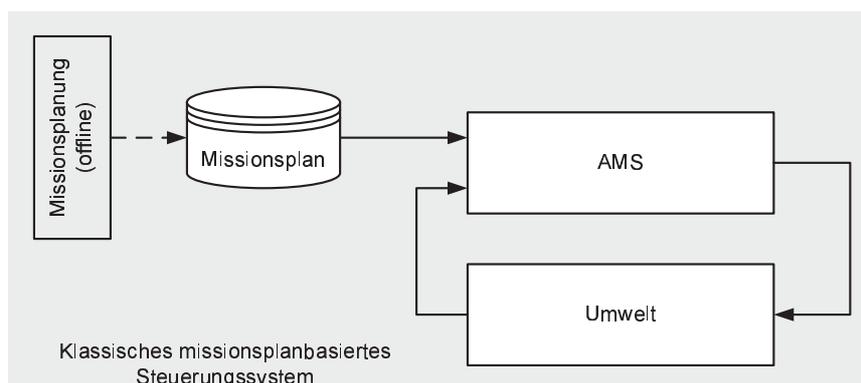
## 2.3 Rahmenbedingungen und Ziele der Arbeit

### 2.3.1 Rahmenbedingungen

Im Rahmen der vorliegenden Arbeit soll eine Erweiterung der vorhandenen Softwarestrukturen autonomer mobiler Systeme um ein intelligentes Missionsmanagement vorgenommen werden. Basis ist das im Abschnitt 2.2.1 vorgestellte Hierarchiemodell nach Rasmussen, nach dem auch die drei Schichten der Softwarearchitektur benannt sind (Abbildung 2.7).

Das autonome mobile System erhält die zu erfüllenden Aufgaben in Form eines Missionsplanes, der aus einzelnen Planelementen (Manövern) besteht. Die verschiedenen Typen von Manövern sind in einem Manöverkatalog definiert und können sich je nach Art des Systems unterscheiden.

Dieser Missionsplan wird mit Hilfe der Manöverausführung umgesetzt, indem die Manö-



**Abb. 2.8:** *Klassisches missionsplanbasiertes Steuerungssystem*

ver sequentiell realisiert werden. Die Manöverausführung generiert aus den Parametern der einzelnen Planelemente Steuerkommandos für die Fahrzeugführung in Form von Sollwerten oder Solltrajektorien. In Sondersituationen, wie zum Beispiel beim Auftauchen von Hindernissen auf der zu fahrenden Route, erfolgt ein Eingriff in die Steuerung, in diesem Fall durch die Hindernisvermeidung. Die geplante Mission wird solange unterbrochen, bis ein gefahrloses Fortsetzen der ursprünglich vorgegebenen Fahrstrecke möglich ist.

In der untersten Schicht werden die Sollvorgaben mit Hilfe von Regelungen umgesetzt. Gleichzeitig erfolgt die Aufbereitung der verfügbaren Sensordaten, die Informationen in unterschiedlicher Ausprägung (als Signale, Zeichen und Symbole) den Module aller drei Schichten bereitstellt.

Der umzusetzende Missionsplan wird vor dem Start der Mission durch das Bedienpersonal erstellt und über eine geeignete Schnittstelle übergeben (Abbildung 2.8). Die automatische Funktion des autonomen mobilen Systems wird durch Anwahl des ersten Manövers in diesem Plan aktiviert.

Damit ergeben sich prinzipiell auch die für die vorliegende Arbeit geltenden Rahmenbedingungen und Anforderungen:

- Die grundlegenden Aufgaben des autonomen mobilen Systems liegen als Handlungsfolge vor, konkret wird ein Missionsplan mit einer Liste von komplexen Manövern übergeben. Dieser Missionsplan ist auch die Arbeitsgrundlage des intelligenten Missionsmanagements.
- Alle Verfahren müssen onlinefähig und einsatzbar auf den Recheneinheiten des mobilen Systems sein, um eine sichere Arbeitsweise auch bei fehlender Kommunikation zu einem menschlichen Operator zu gewährleisten.
- Bedingt durch die Struktur des Missionsplanes müssen Modifikationen derart durchgeführt werden, dass die Kontinuität<sup>1</sup> des Planes aufrecht erhalten wird. Dies erleichtert die Aufbereitung der Fahrkurven des mobilen Systems, die sich aus der Manöverabfolge ergeben.

<sup>1</sup>Siehe Kontinuitätsbedingung, Gleichung 6.2

- Die Architektur des Gesamtsystems sowie Aufbau und Aufgaben der weiteren Softwaremodule stehen entsprechend Abbildung 2.7 fest. Notwendige Anpassungen an bereits bestehenden Strukturen sind auf ein Minimum zu reduzieren.

### 2.3.2 Zielsetzung und Beitrag der Arbeit

Ziel der vorliegenden Arbeit ist der Entwurf und die Realisierung von Modulen für das intelligente Missionsmanagement eines autonomen mobilen Systems. Dabei soll großer Wert auf die einfache Nutzung ohne großen Einarbeitungsaufwand durch nicht speziell geschultes Personal und den universellen Charakter (im Rahmen der Einsatzbedingungen des autonomen mobilen Systems) gelegt werden. Im Gegensatz zu bekannten und veröffentlichten Ansätzen, die teilweise einen sehr großen Planungsaufwand für die Erstellung einer Mission sowie fachspezifische Kenntnisse des Systembedieners erfordern, soll der Einsatz weitestgehend transparent für den Operator möglich sein.

Des Weiteren müssen sich die neu zu erstellenden Module in bestehende Softwarearchitekturen entsprechend Abschnitt 2.2.2 integrieren lassen, ohne weitreichende Anpassungen der Schnittstellen zu erfordern.

Daraus ergeben sich die folgenden zu lösenden Teilprobleme:

- Erweiterung der Planungsparameter des Missionsplanes für das intelligente Missionsmanagement, ohne einen zusätzlichen Einarbeitungsaufwand für das Bedienpersonal zu erfordern,
- Überwachung der Realisierung des Missionsplanes durch das autonome mobile System,
- Erkennung von Situationen, die eine Anpassung des Missionsplanes erfordern,
- Entwurf eines Konzeptes zur Modifikation des Missionsplanes, das sich an unterschiedliche Plattformen anpassen lässt.

Die gefundene Lösung soll auf den Prototypen eines mobilen Kleinroboters (*MauSI*) sowie eines autonomen Unterwasserfahrzeugs (*DeepC*) realisiert und getestet werden. Damit ergeben sich weitere Rahmenbedingungen, die während des Systementwurfs beachtet werden müssen.

Aus den Ausführungen dieses Abschnitts ergeben sich die folgenden neuartigen Beiträge dieser Arbeit:

- Erweiterung der Software-Architektur eines autonomen mobilen Systems um ein intelligentes Missionsmanagement,
- Entwurf eines Verfahrens zur onlinefähigen Optimierung von Missionsplänen bei beschränkten Ressourcen,
- Entwicklung eines Verfahrens zur Planadaption infolge einer Verletzung von geografischen Beschränkungen durch ein digitales globales Umweltmodell,
- Realisierung des Konzepts auf einer kommerziell nutzbaren Plattform, dem autonomen Unterwasserfahrzeug *DeepC* sowie dem mobilen Kleinroboter *MauSI*.

---

Insbesondere der Aspekt des Einsatzes auf einem Unterwasserfahrzeug, das ohne Kommunikationsmöglichkeit seine Mission erfüllen muss, sowie die kommerzielle Nutzbarkeit haben großen Einfluss auf die Lösung der Problemstellung. So wurden alle Verfahren unter dem Blickpunkt der Echtzeitfähigkeit auf den konkreten Recheneinheiten des Systems entworfen. Auf sich dadurch ergebende Einschränkungen wird in den entsprechenden Abschnitten näher eingegangen.

## 2.4 Zusammenfassung

In diesem Kapitel wurden unterschiedliche Ansätze für ein intelligentes Missionsmanagement autonomer mobiler Systeme vorgestellt. Jedes Konzept stützt sich dabei auf eine eigens entwickelte Softwarearchitektur. Eine Anpassung an eine bereits vorhandene, von der favorisierten abweichende Architektur, ist immer mit einem enormen Aufwand verbunden oder gar nicht realisierbar.

Zudem erfordern alle Konzepte eine entsprechende Struktur des Missionsplanes oder aber eine Vorgabe von globalen Missionszielen. Einerseits sind Anpassungen dieser Art für die zur Verfügung stehenden Prototypen durch die Rahmenbedingungen nicht vorgesehen, andererseits ist ein zusätzlicher Einarbeitungsaufwand beziehungsweise ein nicht ganz einfacher Umdenkprozess bei der Erstellung der Missionsbeschreibung durch das Bedienpersonal unerlässlich.

Aus diesem Grund wurde ein neuartiger wissensbasierter Ansatz entwickelt, der nur geringfügige Änderungen an der bestehenden Softwarearchitektur sowie eine nur minimale Erweiterung der vorhandenen Missionsbeschreibung erfordert. Dieser Ansatz wird in den nachfolgenden Abschnitten der Arbeit im Detail vorgestellt.

Besonders hervorzuheben ist die Rückführung bekannter Softwarearchitekturen für autonome mobile Systeme auf das Modell des menschlichen Handelns nach Rasmussen. Durch die vorgefundenen Analogien besitzen diese Architekturen damit eine solide Basis, die eine Abbildung der - unbestritten sehr guten - autonomen Arbeitsweise des Menschen ermöglicht.



## 3 Gesamtkonzeption des Missionsmanagements

Das Kapitel 3 stellt die Gesamtkonzeption des intelligenten Missionsmanagements sowie die Integration in eine vorhandene Softwarearchitektur vor. Im Mittelpunkt stehen dabei die einzusetzenden Verfahren, die sich ergebenden Module sowie die notwendigen Schnittstellen. Abschließend wird auf softwaretechnische Aspekte hingewiesen, die bei der Umsetzung der einzelnen Verfahren zu beachten sind.

### 3.1 Aufbau des Missionsmanagements

Ausgangspunkt für den Entwurf des Missionsmanagementsystems sind die zu lösenden Aufgaben, die sich aus den im Abschnitt 2.3.2 vorgestellten Teilproblemen ergeben:

- Planung der Mission mit Hilfe einer Planungsoberfläche,
- Überwachung der Missionsdurchführung und Erkennung von Situationen, die eine Umplanung erfordern,
- Ermittlung der notwendigen Handlungen für die Anpassung des Missionsplanes,
- Durchführung der Modifikation des Missionsplanes.

Zunächst soll eine Analyse der oben genannten Aufgaben durchgeführt werden. Aus der Analyse ergibt sich schließlich die Struktur des Systems entsprechend Abbildung 3.1.

Jede Mission startet mit einer Planung des Einsatzes des mobilen Systems. Dabei müssen je nach Art der Mission und entsprechend der Struktur des verwendeten Planes unterschiedliche Aufgaben gelöst werden. Die Planung entscheidet wesentlich über Erfolg oder Misserfolg der Mission, dementsprechend ist ihr auch die notwendige Aufmerksamkeit zu widmen. Im Fall einer manuellen Planung durch einen Operator ist eine Simulation des späteren Einsatzes sinnvoll, um die Realisierbarkeit der Mission zu überprüfen.

Die Überwachung der Missionsdurchführung und vor allem die Erkennung von Umplanungssituationen erfordern intelligente Verfahren, um mit Hilfe allgemeinen Wissens derartige Situationen zu detektieren oder auch aus der Ähnlichkeit zu bekannten Situationen auf die Notwendigkeit zur Umplanung zu schließen. Auch die Feststellung, welche Veränderungen am Missionsplan durchzuführen sind, ist abhängig von der aktuellen Situation und benötigt entsprechendes Wissen. Deshalb werden diese beiden Aufgaben gemeinsam von einem Modul - der *Missionsüberwachung* - ausgeführt. Sie stellt damit die eigentliche intelligente Komponente des Systems dar. Ergebnis der Überwachung sind ein oder mehrere Umplanungsbefehle. Diese dienen dem Modul *Missionsumplanung* als Eingangsgrößen. Das Modul führt die Befehle nacheinander aus und gibt den modifizierten Plan zurück.

Die verfügbaren Umplanungsbefehle bestimmen in großem Maße den Charakter des Systems. So vereinfacht eine Festlegung nach den konkreten Erfordernissen einer Roboterplattform insbesondere den Aufbau der Missionsumplanung, da die speziellen Bedürfnisse der

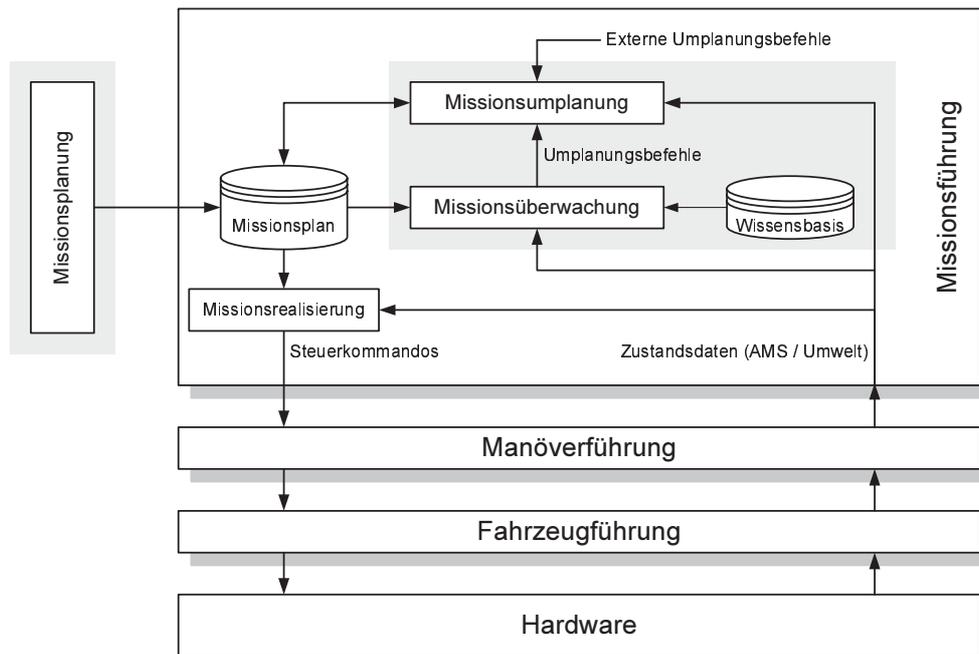
**Tabelle 3.1:** *Verfügbare allgemeingültige Umplanungsbefehle*

Befehl	Beschreibung
<i>Atomare Befehle</i>	
Einfügen	Fügt ein Element in den Missionsplan ein. Elemente können sowohl zwischen andere Elemente als auch in ein Element eingefügt werden, falls der Aufbau des Missionsplanes dies unterstützt.
Löschen	Löscht ein Element aus dem Missionsplan.
Modifizieren	Modifiziert Parameter eines Elementes des Missionsplanes.
<i>Komplexe Befehle</i>	
Abbruch	Bricht die Mission ab und fügt die eventuell dafür notwendigen Anweisungen in den Missionsplan ein.
Optimierung	Optimiert einen Missionsplan bezüglich der verfügbaren Ressourcen des autonomen mobilen Systems.

Plattform direkt berücksichtigt werden können. Nachteilig wirkt sich dabei aus, dass das intelligente Missionsmanagement beim Einsatz auf einem anderen mobilen System möglicherweise neu aufgebaut werden muss. Werden die Befehle hingegen weitestgehend universell definiert, ist unter Umständen ein höherer Aufwand bei der Anpassung des Planes erforderlich. Die Ursache dafür liegt in der Notwendigkeit zur Übersetzung der auszuführenden Planmodifikation in eine Folge allgemeingültiger Befehle. Andererseits ergibt sich durch diese Festlegung als großer Vorteil, dass die Menge an Befehlen für die unterschiedlichen Typen mobiler Systeme und darüber hinaus auch für alle Arten sequentieller Pläne geeignet ist. Deshalb wurde dieser Ansatz in der vorliegenden Arbeit verwendet (Tabelle 3.1).

Insbesondere mit den atomaren Befehlen Einfügen, Löschen und Modifizieren sind beliebige Modifikationen an Missionsplänen möglich. Sie stellen die kleinsten, unteilbaren Einheiten für die Veränderung eines Planes dar. Die komplexen Befehle Abbruch und Optimierung setzen sich im Grunde genommen aus den atomaren zusammen. Sie dienen einerseits der Verlagerung umfangreicher Berechnungen in die Missionsumplanung. Dies ist zum Beispiel bei der Optimierung der Fall. Andererseits wird die Übergabe von Umplanungsbefehlen durch andere Komponenten der Systemsoftware vereinfacht. Diese können über die gleiche Schnittstelle wie die Missionsüberwachung Befehle erteilen, um eine umgehende Reaktion auf bestimmte Ereignisse auszulösen. Denkbar ist zum Beispiel das sofortige Auslösen eines Missionsabbruchs infolge der Lagebeurteilung durch die Fahrzeugeigendiagnose, ohne extra eine Missionsüberwachung durchzuführen.

Das intelligente Missionsmanagement ist entsprechend der in Abschnitt 2.2.2 vorgestellten und in den zur Verfügung stehenden Robotern verwendeten Softwarearchitektur in der Schicht der Missionsführung anzuordnen. Die hier angesiedelten Module besitzen als einzige Zugriff auf den Missionsplan des Fahrzeuges und arbeiten mit den niedrigsten Taktraten im gesamten System. Neben den neu zu schaffenden Modulen Missionsüberwachung und Missionsumplanung ist die Missionsrealisierung ein Kernstück der Software der Missionsführung. Sie zerlegt den Missionsplan in seine Elemente und generiert daraus Vorgaben für die Manöverführung (Abbildung 3.1).



**Abb. 3.1:** Aufbau und Integration des intelligenten Missionsmanagements

Das Missionsmanagement besitzt Zugriff auf alle Zustandsdaten des autonomen mobilen Systems und der Umwelt, die für die Aufgabenerfüllung notwendig sind. Dazu gehören neben den Navigations- und Sensordaten auch diejenigen abgeleiteten Informationen, die von anderen Systemkomponenten erzeugt werden. Mit Hilfe all dieser Daten kann eine Bewertung der aktuellen Situation und gegebenenfalls eine Modifikation des Missionsplanes realisiert werden. Beispielsweise können Diagnosemeldungen wie ein geringer Ladezustand von Batterien zu Veränderungen am Missionsplan, in diesem Fall durch eine Planoptimierung, führen.

Ein weiterer interessanter Aspekt der Missionsumplanung ergibt sich, wenn es in das klassische missionsplanbasierte Steuerungssystem eines autonomen mobilen Systems aus Abbildung 2.8 integriert wird. Durch die Möglichkeit, Korrekturen am Missionsplan durchzuführen, entsteht eine - allen anderen übergeordnete - äußere Rückführung, die auch als Regelkreis aufgefasst werden kann (Abbildung 3.2). Dieser zusätzliche Stelleingriff erhöht die Anpassungsfähigkeit des Roboters sowohl an sich verändernde Umgebungsbedingungen als auch an fahrzeuginterne Probleme (z.B. Hardwareausfälle, unerwarteter höherer Verbrauch verfügbarer Ressourcen).

### 3.1.1 Missionsplanung

Die Missionsplanung ist der Ausgangspunkt für den Einsatz eines autonomen mobilen Systems. Sie liefert den vom Roboter auszuführenden Missionsplan, der je nach Konzeption des Planungsmoduls manuell durch einen Operator oder auch automatisch generiert werden kann. Die Planung stellt damit das erste Element des Missionsmanagements dar und soll aus diesem Grund in einem eigenen Abschnitt erläutert werden. Bei den folgenden Ausführungen wird davon ausgegangen, dass der Missionsplan entsprechend der Definition 1.2



Der Vorteil von Systemen auf der Basis fallbasierten Schließens liegt darin, dass kein allgemeingültiges Wissen bei der Inbetriebnahme zur Verfügung stehen muss. Sie benötigen eine so genannte Falldatensammlung, die verschiedene Fälle unterschiedlichster Situationen beinhalten sollte. Aus dieser Fallbasis wird ein zur vorliegenden Situation ähnlicher Fall ausgewählt, an die aktuellen Gegebenheiten angepasst und als Lösung verwendet. Zunächst muss jedoch eine initiale Datenbasis aufgebaut werden, die sich beispielsweise durch Befragung von Experten oder auch aus vorhandenen Daten ergibt. Im Laufe der Zeit werden immer wieder neue Situationen durchlebt, die die Fallbasis mit weiteren Beispielen auffüllen und damit das *Erfahrungswissen* stetig erweitern.

Regelbasierte Systeme nutzen stattdessen *generisches Wissen* zum Schlussfolgern. Stammt dieses Wissen von einem Experten, kann das System als Expertensystem bezeichnet werden. Der Prozess der Wissensermittlung vom Experten ist komplizierter als beim fallbasierten Schließen, da aus dem Erfahrungswissen allgemeingültige Regeln abzuleiten sind. Im Gegensatz zu fallbasiertem Schließen liefern Expertensysteme zu einer Schlussfolgerung auch gleich eine Argumentationskette, die aus den zur Schlussfolgerung führenden Regeln besteht. Diese Eigenschaft regelbasierter Systeme wird als Erklärungsfähigkeit bezeichnet. Sie ist ein wichtiges Hilfsmittel beim Aufbau der Wissensbasis und ihrer späteren Wartung.

Für die konkreten Anwendungsfälle aus Kapitel 7 stand Wissen in allgemeingültiger Form zur Verfügung. Aus diesem Grund wurde die Missionsüberwachung als regelbasiertes System implementiert. Eine spätere Erweiterung um Methoden des fallbasierten Schließens wurde dabei berücksichtigt. Liegen umfassende Erfahrungen sowie entsprechendes Datenmaterial von praktischen Einsätzen vor, könnten eine (Initial-) Fallbasis erstellt sowie die erforderlichen Verfahren eingebunden werden. Die notwendigen Grundlagen sowie die Beschreibung des Expertensystems sind im Kapitel 5 zu finden.

### 3.1.3 Missionsumplanung

Die Missionsumplanung besitzt durch das vorgeschlagene Konzept einen weitestgehend prozeduralen Charakter. Die übergebenen Umplanungsbefehle werden dabei sequentiell realisiert. Die Methoden zur Umsetzung der atomaren Befehle *Einfügen*, *Löschen*, *Modifizieren* stellen die Basis dar, auf die die komplexen Befehle *Abbruch* und *Optimierung* aufbauen. Durch eine allgemeingültige, unabhängig von den verwendeten Plattformen definierte Beschreibung des Missionsplanes entsprechend Definition 1.2 können die Verfahren zur Planmodifikation derart entworfen werden, dass sie sich in vielfältiger Weise für die Planung von Handlungsabläufen einsetzen lassen.

Nach der Anpassung des Missionsplanes kann mit Hilfe einer Umgebungsdatenbank seine Realisierbarkeit durch die geografische Planprüfung kontrolliert werden. Ergibt sich dabei, dass die in den Plan eingefügten Elemente nicht ausführbar sind, muss eine Anpassung mittels der geografischen Planmodifikation erfolgen. Diese geografischen Verfahren sind gegenüber allen anderen sehr rechenzeitintensiv. So kann es in Abhängigkeit von dem aktuellen Zustand durchaus notwendig sein, nur bestimmte Umplanungen zu überprüfen. Beispiele für derartige Situationen werden im nachfolgenden Abschnitt 3.2 aufgeführt.

Als Ergebnis der Missionsumplanung steht der an die konkrete Situation angepasste Missionsplan zur Verfügung. Er wird an die Plandatenbank zur weiteren Verarbeitung übergeben. Die hier aufgeführten Methoden werden detailliert im Kapitel 6 beschrieben.

### 3.2 Softwaretechnische Aspekte

Wie bereits erwähnt, ist das intelligente Missionsmanagement Bestandteil der Schicht der Missionführung innerhalb der Softwarehierarchie des Fahrzeuges und unterliegt damit generell keinen harten Echtzeitanforderungen. Für die hier ablaufenden Prozesse steht in der Regel genügend Rechenzeit zur Verfügung, sodass auch anspruchsvollere Verfahren eingesetzt werden können.

Im Zuge der praktischen Realisierung ergaben sich trotzdem unterschiedliche Anwendungsfälle, die eine rasche Umplanung erfordern. Da diese Situationen nicht auf die zur Verfügung stehenden Plattformen beschränkt sind, sollen sie hier kurz erläutert werden.

- Beim Einfügen eines Elements in das gegenwärtig realisierte Element des Missionsplanes wird die zu Beginn der Umplanung vorliegende Fahrzeugposition als Startposition verwendet. Durch eine langwierige Planmodifikation würde sich der Roboter bei weiterer Bewegung bereits an einer anderen Position befinden. Wird das eingefügte Planelement nach Abschluss der Umplanung aktiviert, muss erst eine Rückkehr an die geplante Startposition ausgeführt werden.
- Je nach Art des mobilen Systems kann die Anforderung bestehen, dass durch eine erst kurz vor einem Wechsel des aktuellen Planelements detektierbare Situation eine Plananpassung ausgelöst wird. Auch hier ist eine rasche Umplanung notwendig, um vor Ende des aktiven Elements den neuen Plan im System verwenden zu können.
- Ein Missionsabbruch aufgrund ausgefallener Sensorik oder Aktorik sollte ebenfalls in kürzester Zeit geplant sein, um das autonome mobile System nicht durch längerfristige Planungsprozesse unnötig Gefahrensituationen auszusetzen.

Diese Fälle stellen nur einen Auszug aus einer Menge denkbarer Situationen dar, für die zugunsten einer sicheren oder kontinuierlichen Arbeitsweise des Gesamtsystems auf verschiedene Prüfroutinen wie zum Beispiel die geografische Planprüfung im Anschluss an eine Planmodifikation verzichtet werden kann. Konkret hängt die Entscheidung darüber jedoch unter anderem von der verfügbaren Rechenleistung sowie der Dynamik des mobilen Systems ab. Auch die Art der durchgeführten Planmodifikation beeinflusst die auszuführenden Handlungen. So kann bei dem Befehl zum Modifizieren der Parameter eines Planelementes die geografische Planprüfung unterbleiben, wenn nicht gerade die Start- oder Endposition des Manövers angepasst werden sollen.

### 3.3 Zusammenfassung

In diesem Abschnitt wurde die grundlegende Konzeption des intelligenten Missionsmanagements vorgestellt. Es besteht aus den drei Elementen Missionsplanung, Missionsüberwachung und Missionsumplanung. Das Kernstück ist dabei die Missionsüberwachung, die durch den Einsatz wissensbasierter Systeme die für eine Planmodifikationen notwendige Intelligenz des Systems auf sich vereinigt. Die Missionsplanung definiert den vom Roboter zu realisierenden Plan. Mit ihr wird deshalb der Grundstein für einen erfolgreichen Einsatz des autonomen mobilen Systems gelegt.

---

Die Definition allgemeingültiger Umplanungsbefehle erlaubt die Durchführung beliebiger Planadaptionen. Diese Befehle werden durch die dritte Komponente, die Missionsumplanung, ausgeführt. Geografische Planprüfung und -modifikation können als Abschluss einer Umplanung durchgeführt werden. Damit soll sichergestellt werden, dass der neue Missionsplan realisierbar ist.



## 4 Missionsplanung

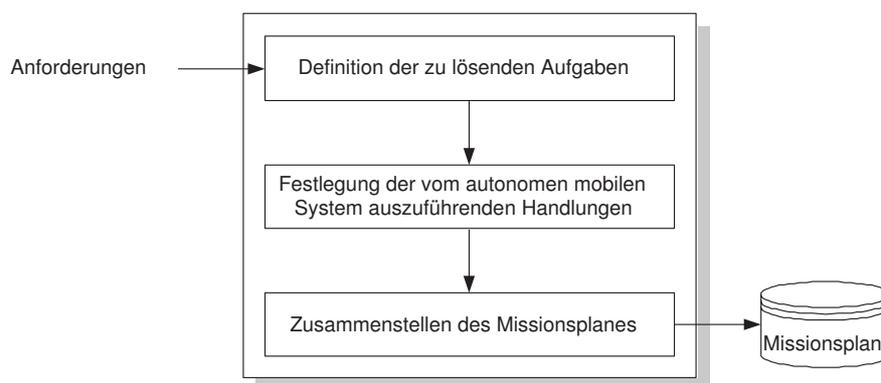
Die Missionsplanung ist der Ausgangspunkt für den Einsatz eines autonomen mobilen Systems. Neben der Prüfung und Vervollständigung der technischen Ausrüstung des Roboters (z.B. Energievorräte oder Sensorkalibrierung) hat sie einen wesentlichen Einfluss auf den Erfolg einer Mission.

Im Rahmen der Missionsplanung sind unterschiedliche Aktionen auszuführen, um zu der gewünschten Handlungsfolge des mobilen Roboters zu gelangen. Sie bilden mit dem Abschnitt 4.1 den Einstieg in dieses Kapitel. Für die Akzeptanz des Gesamtsystems beim späteren Nutzer ist die Gestaltung der Bedienoberfläche von entscheidender Bedeutung. Deshalb werden in Abschnitt 4.2 allgemeine Grundregeln sowie Vorschläge für den ergonomischen Entwurf der Schnittstelle zwischen Mensch und Maschine sowie unterschiedliche Ansätze für die Missionsplanung vorgestellt.

### 4.1 Vorgehensweise bei der Missionsplanung

Im Abschnitt 3.1.1 wurden die zu lösenden Aufgaben bei der Erstellung eines Missionsplanes bereits dargelegt. Ausgangspunkt der Planung ist eine Menge von Anforderungen, die durch den Roboter erfüllt werden sollen. Sie können sowohl einen allgemeinen, verbalen als auch einen bereits die Tätigkeiten spezifizierenden Charakter tragen.

Aus diesen Vorgaben sind im ersten Schritt der Missionsplanung die durch das mobile System zu lösenden Aufgaben zu bestimmen (Abbildung 4.1). Je nach Art der Anforderungen sind dazu unterschiedliche Arbeitsschritte notwendig. So muss zum Beispiel bei einer verbalen Beschreibung der auszuführenden Tätigkeiten deren Inhalt analysiert und in eine weiterverwendbare Form überführt werden.



**Abb. 4.1:** Ablauf der Missionsplanung

Oftmals ist die entstehende Menge an Aufgaben bereits auf die Art und das Leistungsvermögen des mobilen Roboters abgestimmt. Das erleichtert die Festlegung der konkreten

Handlungen des Systems, die die Basis für den Missionsplan darstellen, und reduziert den durchzuführenden Transformationsprozess (Aufgaben  $\rightarrow$  Handlungen) auf ein Minimum.

Der letzte Schritt der Planung ist die Erstellung des zu realisierenden Missionsplanes. Dieser Vorgang besitzt einen direkten Bezug zum mobilen System, da Struktur und Inhalt des Planes von der Implementierung auf dem Roboter abhängen.

Es existieren unterschiedliche Ansätze, den gesamten Planungsprozess durch den autonomen Roboter selbst ausführen zu lassen [BP98]. Ausgangspunkt ist eine Formulierung der Anforderungen in verbaler Form, die automatisch zu dem Missionsplan führt. Für einen eng umrissenen Aufgabenbereich mit fester Syntax der Anforderungen mag diese Vorgehensweise gerechtfertigt sein, so unter anderem für die automatische Wegplanung [TSW01]. Kann diese Bedingung jedoch nicht eingehalten werden, erhöht sich der Aufwand für ein derartiges System beträchtlich.

Die Tätigkeit der Missionsplanung ist für den Menschen nach einer Einarbeitungsphase dagegen ohne größere Schwierigkeiten beherrschbar. Lediglich die Eingabe des Planes mit Hilfe der Bedienoberfläche wird durch deren Gestaltung entscheidend beeinflusst. Die aktive Einbeziehung des Menschen in den Planungsvorgang hat daneben noch weitere entscheidende Vorteile:

- Das benötigte domänenspezifische Wissen für die Erstellung der Mission unter Berücksichtigung der durch die Umwelt vorgegebenen Rahmenbedingungen verbleibt bei dem Operator und muss nicht in ein automatisches Planungssystem integriert werden.
- Eine Anpassung an unterschiedliche Arten der verfügbaren Manöver mobiler Systeme ist für den Menschen ohne größeren Aufwand möglich, während die automatische Plangenerierung an das jeweilige Fahrzeug adaptiert werden muss.
- Die Akzeptanz gegenüber dem Einsatz autonomer mobiler Systeme ist untrennbar mit dem Gefühl des „Beherrschens“ der Technik durch den Menschen verbunden. Eine nicht völlig transparente Zusammenstellung der Handlungen des Systems wird außerhalb des Entwicklerteams häufig nicht zugelassen, sodass weitere Anstrengungen für eine erklärende und begründende Darstellung des Planungsprozesses unternommen werden müssen.
- Durch die manuelle Planung ist der Operator jederzeit über die aktuell ausgeführten Handlungen des mobilen Systems im Bilde und kann, falls erforderlich und möglich, in die Arbeitsweise eingreifen. Das ist besonders für Roboter wichtig, die nicht ständig in Kontakt zu einer Basisstation stehen (z.B. autonome Unterwasserfahrzeuge).

Aus diesen Gründen wird in der vorliegenden Arbeit der Ansatz zur Missionsplanung durch den Menschen mit Hilfe einer an das mobile System angepassten Bedienoberfläche präferiert. Der nachfolgende Abschnitt beinhaltet die für den Entwurf einer derartigen Oberfläche notwendigen Informationen.

## 4.2 Gestaltung der Mensch-Maschine-Schnittstelle

Die bereits angesprochene Akzeptanz eines Robotersystems durch den Menschen wird in entscheidendem Maße durch die Gestaltung der Bedienschnittstelle bestimmt. Häufig ist eine

direkte Anpassung an das konkrete Problemumfeld notwendig. Dazu gehören neben spezifischen grafischen Darstellungen (z.B. topografische oder Straßenkarten bei erdgebundenen und Seekarten bei maritimen Fahrzeugen) auch die Einbeziehung des jeweiligen Sprachumfeldes. Trotzdem existieren allgemeingültige Vorschläge und Richtlinien für den Oberflächenentwurf, die im Folgenden präsentiert werden sollen.

#### 4.2.1 Aspekte zum ergonomischen Entwurf von Bedienoberflächen

Die Erstellung und Bewertung von Benutzeroberflächen ist untrennbar mit dem Begriff der Software-Ergonomie verbunden. Die Ergonomie beschäftigt sich im Allgemeinen mit den Leistungsmöglichkeiten und -grenzen des arbeitenden Menschen. Eine besondere Rolle spielen dabei das Erreichen der bestmöglichen wechselseitigen Anpassung zwischen dem Menschen und seinen Arbeitsbedingungen. Bezogen auf die Gestaltung von Bedienoberflächen steht die benutzerfreundliche und menschengerechte Gestaltung der interaktiven Programme im Mittelpunkt. Mittlerweile existiert eine Reihe von Normen und Richtlinien über den Entwurf von Software, die durch den Entwickler beachtet werden sollten (u.a. DIN EN ISO 9241, DIN 66234).

Die Beurteilung der Qualität einer Bedienoberfläche erfolgt in der Regel auf der Basis unterschiedlicher Kriterien [Boy91]. Zu den Wesentlichsten gehören:

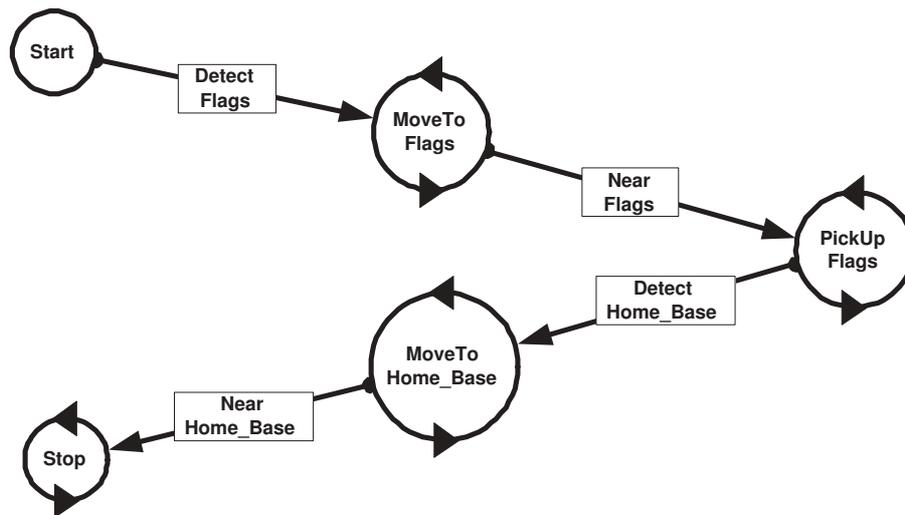
- die Einarbeitungszeit,
- die Dauer der Testphase,
- die entstehenden Kosten für Bedienfehler,
- die subjektive Zufriedenheit des Nutzers sowie
- die Qualität der Arbeit mit dem System bei andauernder Nutzung.

Diese Faktoren beeinflussen die Akzeptanz der Software in großem Maße und entscheiden mit über die Anzahl auftretender Bedienfehler sowie die erzielte Produktivität [Pre94].

Die Analyse menschlicher Fehler und deren Verhinderung durch entsprechend gestaltete Programmoberflächen ist eng verknüpft mit dem menschlichen Handlungsmodell nach Rasmussen, das in Abschnitt 2.2.1 präsentiert wurde. So sind bei den kognitiven Prozessen je nach aktiver Ebene entsprechend des Handlungsmodells unterschiedliche Informationen notwendig (Signale, Zeichen, Symbole). Werden dem Menschen in einer konkreten Situation nicht die entsprechenden Informationsträger präsentiert, ist ein zusätzlicher Arbeitsaufwand bei der Bestimmung der auszuführenden Handlungen notwendig. Das erhöht in beträchtlichem Maße das Risiko, Fehlentscheidungen zu treffen.

#### 4.2.2 Gestaltung der Planungsoberfläche

Grundsätzlich lassen sich bei der Missionsplanung durch den Menschen zwei Arten von Darstellungstechniken unterscheiden, die in verschiedenen Situationen zum Einsatz kommen können. Bei der *strukturbasierten Planung* steht der Aufbau der Missionsplanes im Zentrum der Bedienoberfläche. Der Operator gibt listen- oder graphenorientiert die einzelnen



**Abb. 4.2:** Graphenorientierte Missionsplanung mit MissionLab [ACE99]

Planelemente ein, die schließlich den Gesamtplan formen (Abbildung 4.2). Es entsteht eine übersichtliche Darstellung der Handlungssequenz des Roboters.

Die *geometriebasierte Planung* verwendet hingegen eine auf das Einsatzfeld des Systems bezogene Karte und gestattet eine schnelle grafische Zusammenstellung räumlich ausgedehnter Manöver (Abbildung 4.3 (a)). Sollen mehrere Planelemente in dem gleichen Gebiet realisiert werden, ist eine übersichtliche Darstellung der gesamten Mission nicht mehr möglich. Als Ausweg bietet sich an, nur eine Auswahl an Manövern für die Abbildung zu verwenden. Durch den Einsatz einer Kartenansicht als Hintergrund wird aus der geometriebasierten eine *geografiebasierte Planung*, die sich als besonders vorteilhaftes ergonomisches Hilfsmittel für die Planung empfiehlt (Abbildung 4.3 (b)) [RK00].

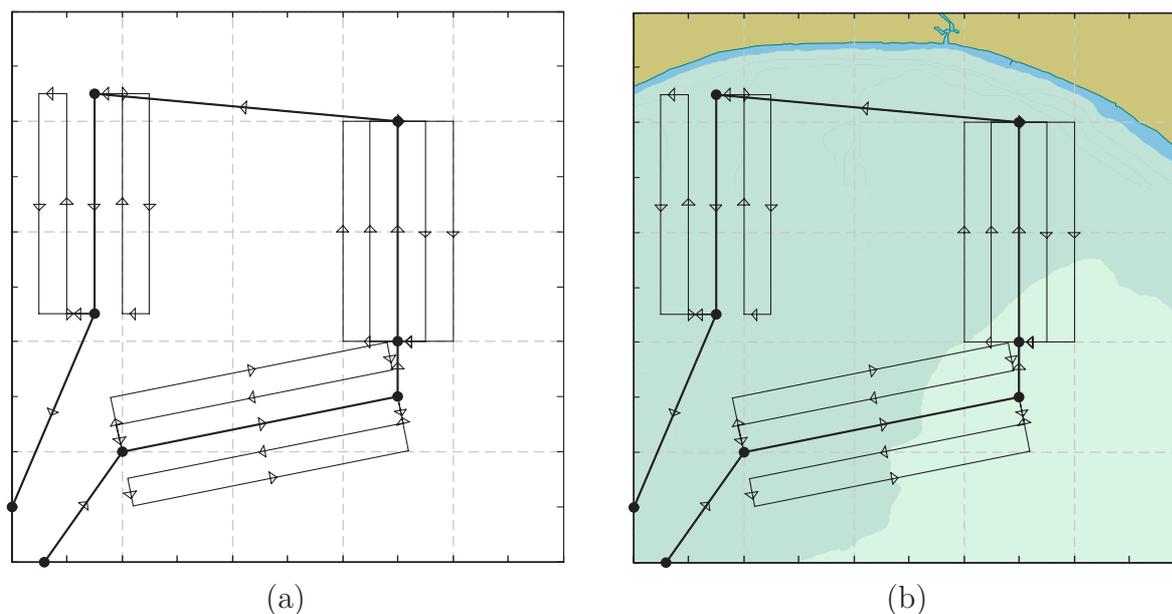
Erst durch die Kombination von struktur- und geometriebasierter Planung steht dem Menschen je nach gegebener Situation und eigener Neigung eine zweckmäßige Ansicht zur Verfügung. Die Planungsoberfläche kann darüber hinaus weitere Elemente enthalten, die die Arbeit des Operators vereinfachen und unterstützen. Beispielsweise ist der Einsatz von erfolgreich absolvierten Missionen als Schablonen für die aktuelle Situation möglich [EMA04].

Als Entscheidungskriterien für die Oberflächenprogrammierung dienen neben den allgemeinen Aspekten aus Abschnitt 4.2.1 unter anderem auch die folgenden wichtigen Faktoren:

- *Kenntnisse des Operators:* Ein kommerziell nutzbares System muss zum Beispiel von einer Vielzahl unterschiedlicher Nutzer bedient werden können. Insbesondere die Verhinderung von Fehlbedienungen, Plausibilitätsprüfungen sowie eine verständliches Hilfesystem sind in diesem Fall wesentliche Bestandteile der Planungssoftware.

Wird die Planung jedoch von einem begrenzten Nutzerkreis mit Einblick in die Arbeitsweise des Roboters durchgeführt, kann neben den allgemein bekannten Informationen noch zusätzliches Wissen über Art und Weise der Missionsplanung vorausgesetzt werden.

- *Einsatzfeld des mobilen Systems:* Häufig ist eine Anpassung an domänenspezifische



**Abb. 4.3:** Geometriebasierte (a) und geografiebasierte Missionsplanung (b)

Darstellungen, Maßeinheiten und Begriffe unerlässlich, um den Einarbeitungsprozess der Bediener effizient zu gestalten. Soll hingegen eine domänenübergreifende Planungsoberfläche erstellt werden, ist eine vereinheitlichende Sprach- und Abbildungsweise zu finden und zu dokumentieren.

Darüber hinaus sind noch die Besonderheiten zu beachten, die durch den unterschiedlichen Aufbau der Steuerungssysteme von Robotern auftreten. Sie fließen in die Gestaltung der Bedienoberfläche und in die zugrunde liegende Programmlogik ein.

### 4.3 Zusammenfassung

Die Missionsplanung ist die direkte Schnittstelle des Menschen zum autonomen mobilen System. Deshalb muss vor allem der Gestaltung der zugehörigen Bedienoberfläche des Entwurfssystems große Aufmerksamkeit gewidmet werden. Neben allgemeinen Gesichtspunkten der Ergonomie spielt auch die Analyse des konkreten Einsatzgebietes des Systems eine große Rolle, um eine bestmögliche Anpassung an die Arbeitsweise des Operators zu erreichen.

Hinsichtlich der Planerstellung ist eine Unterteilung in struktur- und geometriebasierte Planung vorgenommen worden. Durch die Kombination beider Techniken und den Einsatz weiterer unterstützender Methoden kann die Missionsplanung durch einen Operator sehr effizient gestaltet und realisiert werden.



## 5 Wissensbasierte Überwachung der Mission

Die wissensbasierte Überwachung ist das zentrale Element des Missionsmanagements, da sie im Wesentlichen die Intelligenz des Systems beinhaltet. Sie wird entsprechend der Gesamtkonzeption mit Hilfe eines regelbasierten Systems durchgeführt und ermittelt die in einer bestimmten Situation auszuführenden Veränderungen am Missionsplan.

Im Folgenden sollen die wichtigsten Grundlagen dargelegt werden, die für die wissensbasierte Überwachung der Mission notwendig sind. Nach einem Überblick zu dem allgemeinen Aufbau und der Arbeitsweise regelbasierter Systeme im ersten Teil dieses Kapitels wird das neue Konzept der Missionsüberwachung und seine Integration in das intelligente Missionsmanagement vorgestellt.

### 5.1 Aufbau und Arbeitsweise regelbasierter Systeme

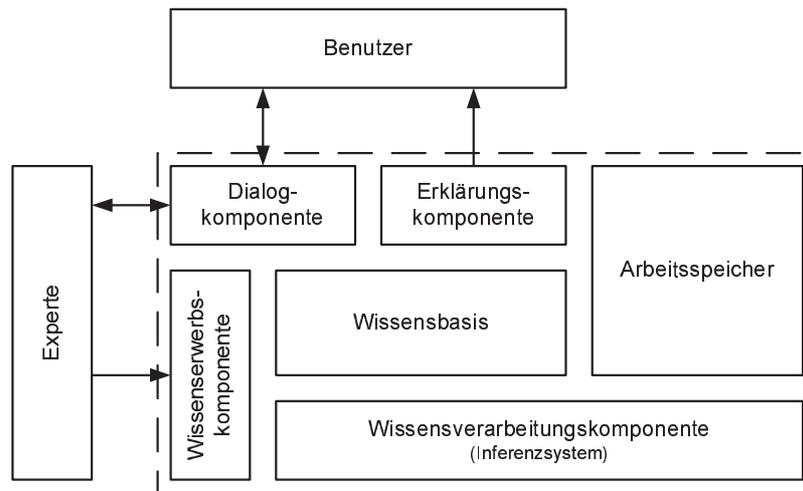
Regelbasierte Systeme und besonders regelbasierte Expertensysteme nutzen spezifisches Wissen, um Probleme analog zu einem menschlichen Experten zu lösen. Ein Experte zeichnet sich durch Sachverstand in einem bestimmten Gebiet aus, der für die Allgemeinheit der Menschen nicht verfügbar ist. Dadurch ist er in der Lage, Probleme einfacher, schneller oder auch genauer zu lösen als jeder andere. Das Wissen eines Experten ist dabei in der Regel auf einen bestimmten Bereich begrenzt (domänenspezifisches Wissen).

Schon frühzeitig wurde erkannt, dass dieses Wissen verloren gehen kann. Der „Verlust“ des Experten durch Abwanderung oder Ruhestand ist zum Beispiel bis in die heutige Zeit ein großes Problem für die Industrie. Auch die Verfügbarkeit der Erfahrung des Experten an verschiedenen Orten gleichzeitig ist zwar durch moderne Kommunikationsmittel erleichtert worden, ein Ausgleich für die Tätigkeit vor Ort ist dies jedoch nicht. Deshalb besteht seit den 70er Jahren der Wunsch, das Wissen und den Sachverstand von Experten in dauerhafter und universell nutzbarer Form zu hinterlegen. Daraus entwickelten sich die ersten Expertensysteme wie DENDRAL oder MYCIN [GR98]. Heutzutage sind regelbasierte Expertensysteme - bedingt durch ihre Vorteile wie Modularität, gute Lesbarkeit sowie einfache Wartbarkeit - die am weitesten verbreitete Form wissensbasierter Systeme [Moh00].

#### 5.1.1 Aufbau

Prinzipiell bestehen regelbasierte Expertensysteme aus mehreren Komponenten (Abbildung 5.1). Die für die eigentliche Aufgabenerfüllung erforderlichen Kernelemente sind [Moh00]:

- *Wissensbasis*: Sie enthält die Regeln und ist im Allgemeinen statisch. Es findet also keine Veränderung der Regeln statt.
- *Arbeitsspeicher*: Er enthält veränderliches Wissen über einen Prozess oder ein System. Häufig wird dieses Wissen fallspezifisch in Form von Fakten hinterlegt.



**Abb. 5.1:** Aufbau eines regelbasierten Expertensystems

- *Wissensverarbeitungskomponente:* Sie enthält den Algorithmus zur Ausführung der Regeln. Der dabei ablaufende Prozess wird auch als Inferenz, die Komponente selbst als Inferenzsystem bezeichnet.

Darüber hinaus besitzen Expertensysteme für die Interaktion mit dem Menschen folgende Schnittstellen:

- *Wissenserwerbskomponente:* Sie dient zur Wissensermittlung vom Experten und zur Veränderung des bereits gespeicherten Wissens.
- *Erklärungskomponente:* Sie beschreibt den Weg in Form der aktivierten Regeln, der beim Schließen aufgrund der vorhandenen Fakten besritten wurde.
- *Dialogkomponente:* Sie dient der Interaktion des Systems mit den verschiedenen Personen (Entwickler, Experte, Nutzer). Darüber wird zum Beispiel der Arbeitsspeicher mit Fakten gefüllt, Regeln eingegeben oder die Steuerung des Systems vorgenommen.

Diese Schnittstellen sind für den Fall, dass ein regelbasiertes System in einem autonomen System eingesetzt werden soll, nur für den Entwurf, die Wissensakquisition sowie zu Test- und Wartungszwecken notwendig. Lediglich Wissensbasis, Arbeitsspeicher und Inferenzsystem werden dann im produktiven Einsatz benötigt. Deshalb sollen die für das Verständnis dieser Komponenten notwendigen Informationen in den nachfolgenden Abschnitten vermittelt werden.

### 5.1.2 Wissensrepräsentation

Die Wissensrepräsentation beschäftigt sich mit der Darstellung, Beschreibung und Präsentation von Wissen [Hau00]. Ziel ist es, die Informationen in maschinenverarbeitbarer Form zu speichern. Gleichzeitig soll es möglich sein, bei Bedarf aus der gewählten Beschreibung

das Wissen wieder zu reproduzieren. Vorhandenes Wissen kann auf verschiedene Weise in einem wissensbasierten System hinterlegt sein. Prinzipiell werden deklarative und prozedurale Darstellungsarten des Wissens unterschieden.

Die *prozedurale Wissensrepräsentation* beinhaltet Verfahren für die Lösung eines Problems in Form von definierten Abläufen. Das Wissen liegt dabei als Gleichungen, Algorithmen oder Programme vor [BW87].

Die *deklarative Wissensrepräsentation* hingegen stellt die vorhandenen Informationen lediglich als Sachverhalte zur Verfügung, es werden keine Aussagen über die Art der Anwendung des Wissens gemacht. Deklaratives Wissen kann sehr gut in Wissenseinheiten strukturiert werden, die die Basis für einen modularen Aufbau von Expertensystemen bilden. Gebräuchliche Methoden für die Wissensdarstellung sind relationale Datenmodelle und objektorientierte Darstellungen wie semantische Netze, Objekt-Attribut-Wert-Tripel oder Frames [Nik97].

Regelbasierte Systeme und darauf basierende Expertensysteme nutzen Regeln für die Speicherung des deklarativen Wissens. Regeln sind nach [BKI00] formalisierte Konditionalsätze der Form

Wenn  $A$  Dann  $B$ ,

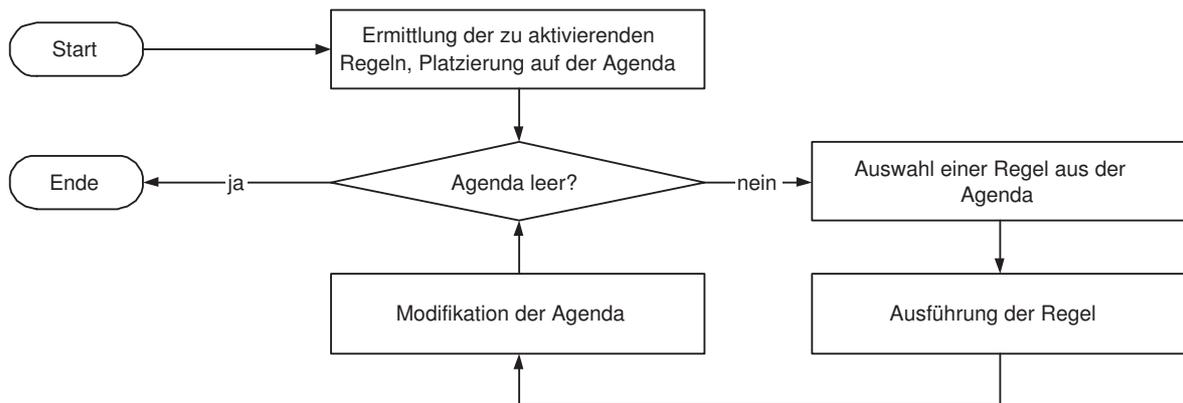
wobei  $A$  als Prämisse (oder Antezedenz) und  $B$  als Konklusion (oder Konsequenz) bezeichnet werden. Da der Konklusionsteil oftmals auszuführende Aktionen beinhaltet, werden derartige Regeln auch als Produktionsregeln und die regelbasierten Systeme als Produktionssysteme bezeichnet. Häufig lassen sich deklaratives und prozedurales Wissen in Produktionssystemen in Kombination anwenden. Dabei können Prozeduren sowohl in der Prämisse als auch in der Konklusion aufgerufen werden. Das erlaubt eine flexible Anpassung an die verschiedenen Formen des vorhandenen Wissens.

### 5.1.3 Wissensverarbeitung

Aufgabe der Wissensverarbeitung ist die Anwendung des vorhandenen Wissens, um aus den enthaltenen Informationen Schlussfolgerungen zu ziehen. Dieser Prozess wird auch als Inferenz bezeichnet, die zugehörige Komponente des wissensbasierten Systems als Inferenzsystem. Für das Schließen existieren unterschiedliche Verfahren, um auch mit großen Wissensbasen effektiv arbeiten zu können.

Die Arbeitsweise der verschiedenen Inferenzstrategien lässt sich prinzipiell folgendermaßen darstellen (Abbildung 5.2). Bei Vorliegen neuer Daten muss zunächst diejenige Menge an Regeln (Konfliktmenge) ermittelt werden, die aufgrund dieser Informationen abzuarbeiten ist. Die so genannte Agenda dient als Speicher für diese Menge an aktivierten Regeln. Anschließend erfolgt die sequentielle Ausführung der Regeln der Agenda, die aufgrund ihres Konklusionsteils zu abgeleitetem Wissen und damit zur Aktivierung weiterer Regeln führen kann. Mit ihrer erfolgreichen Realisierung werden aktivierte Regeln aus der Agenda entfernt. Erst wenn die Agenda komplett geleert ist, wird der Inferenzalgorithmus beendet.

Mit steigender Regelzahl oder auch wachsendem Faktenwissen über die aktuelle Situation wird die Gesamtlaufzeit der Inferenz im Wesentlichen durch den Prozess der Regelauswahl (Platzierung auf der Agenda) sowie die Modifikation der Agenda beeinflusst. Durch eine Vorverarbeitung des Regelwerkes kann dieser Vorgang jedoch erheblich beschleunigt werden.



**Abb. 5.2:** Ablauf der Inferenz

So generiert der Rete-Algorithmus aus den Prämissen des Regelwerks eine Baumstruktur [For82]. Die einzelnen Bestandteile der Prämissen aller Regeln der Wissensbasis sind darin einmalig hinterlegt und werden nach dem Start des Inferenzverfahrens auf ihre Erfüllung hin untersucht. Das Ergebnis wird in den Knoten des Baumes als Zustände gespeichert. Ändern sich nun Fakten, so werden lediglich diejenigen Knoten neu berechnet, die diese Fakten beinhalten. Bedingt durch die Zustandsspeicherung zwischen den Arbeitszyklen ist der Rete-Algorithmus in Situationen, in denen sich gleichzeitig eine große Zahl von Fakten ändert, nicht effizient. Weiterentwicklungen wie Rete 2 oder Rete/UL bieten sich in derartigen Fällen als praktikable Methoden an [For94], [Doo95].

Vor allem die in den letzten Jahren intensiv untersuchten Echtzeit-Expertensysteme benötigen jedoch eine Inferenzstrategie, die innerhalb einer festgelegten Zeit eine Lösung findet. Die vorgestellten Methoden sind dafür weniger geeignet, da für echtzeitfähige Systeme unumgängliche Eigenschaften wie die Interruptfähigkeit und die Vorhersagbarkeit der Laufzeit fehlen. Deshalb wurden spezielle echtzeitfähige Inferenzverfahren entwickelt. Im Mittelpunkt der Methoden steht wiederum die zeitintensive Ermittlung der Konfliktmenge, die zum Beispiel durch Generierung eines optimalen Entscheidungsbaumes (serielle Fakttermittlung) oder durch eine parallele Fakttermittlung mit Hilfe von effizienten Regelkodierungen erfolgen kann [Dun95].

#### 5.1.4 Wissensermittlung

Bevor ein wissensbasiertes System implementiert werden kann, muss zunächst die Erfassung des zu hinterlegenden Wissens erfolgen. Aus dem Blickwinkel des wissensbasierten Systems heißt dieser Prozess *Wissensermittlung* oder *Wissensakquisition*. Wird hingegen primär die Arbeit der wissenserfassenden Person, des Wissensingenieurs, betrachtet, ist der Begriff *Knowledge Engineering* zutreffender. Dieser umfasst neben den reinen Entwicklungsaufgaben auch die insbesondere im produktiven Einsatz wichtigen Wartungstätigkeiten [Hau00].

Schon frühzeitig wurde erkannt, dass die Wissensermittlung den größten Engpass bei der Realisierung eines Expertensystems darstellt [HRWL83]. Neben ihrer hohen Zeitintensität hat sie auch großen Einfluss auf die Qualität des entstehenden Systems. Fehlt zum Beispiel Wissen in der Wissensbasis, weil entweder der Experte oder der Entwicklungsingenieur diese

Informationen als allgemein bekannt oder unwichtig ansehen, kann das Expertensystem unter Umständen nicht in der gewünschten Weise arbeiten.

Zur Überwindung dieses fehlerträchtigen Prozesses werden vermehrt Methoden des maschinellen Lernens eingesetzt, die als Ziel einen vollautomatischen Wissenserwerb verfolgen. Sie benötigen jedoch umfangreiches Datenmaterial zur Wissensextraktion und Verifikation des Gelernten, das nicht für alle möglichen Einsatzgebiete von Expertensystemen zur Verfügung steht. Beispielsweise sind maschinelle Lernverfahren zur Erstellung eines Systems für das Katastrophenmanagement denkbar ungeeignet.

Bei der Wissensermittlung sind verschiedene Schwierigkeiten zu überwinden, die teilweise große Hindernisse innerhalb dieses Prozesses darstellen [Moh00]:

- *Verständigungsprobleme:* Gehört das zu erfassende Wissen nicht zum allgemeinen Lebensbereich der Menschen, sind Probleme aufgrund der unterschiedlichen Sprach- und Erfahrungswelt von Experte und Wissensingenieur an der Tagesordnung. Letzterer hat in einer Einarbeitungsphase grundlegende Informationen über das spezielle Erfahrungsumfeld des Experten zu sammeln, um die Ergebnisse des Wissenserwerbs sinnvoll einsetzen zu können.
- *Verständnisprobleme:* Die wenigsten abzubildenden Prozesse sind derart trivial, dass sie ohne tieferen Einblick in die naturwissenschaftlichen Hintergründe verstanden werden können. Bei besonders komplizierten Vorgängen sind erhebliche Schwierigkeiten bei der Erstellung der Wissensbasis unvermeidbar.
- *Fehlbarkeit:* Auch ein Experte ist nicht unfehlbar, sodass falsche oder unvollständige Informationen in die Wissensbasis gelangen können. Zusätzlich können nicht begründbare oder institutionalisierte Handlungen und ungenaue Angaben den Prozess der Wissensermittlung verkomplizieren.
- *Widerstand:* Nicht immer sind die verfügbaren Experten bereit, ihr Wissen einem Dritten preiszugeben. Die Angst vor dem Verlust des Arbeitsplatzes oder die Überzeugung, dass die in jahrelanger Arbeit entwickelte Expertise durch ein automatisches System nie erreicht werden kann, spielen dabei als Ursachen eine große Rolle. Auch durch den Experten als unqualifiziert abgestempelte Fragen des Wissensingenieurs tragen zur Blockade des Befragten bei.
- *Interpretation:* Die Auswertung des erfassten Wissens ist eng verbunden mit einer Auslegung durch den Wissensingenieur. Unvollständige oder fehlerhafte Interpretation führt ebenfalls zu falschem Wissen im Expertensystem.

Durch eine gut geplante Vorgehensweise bei der Wissensermittlung können die meisten Schwierigkeiten umgangen werden. Ausgangspunkt sollte stets das Studium einschlägiger Literatur zum Einsatzgebiet des Expertensystems sein. Dies schafft ein grundlegendes Verständnis für die Problemdomäne. Für die anschließende Wissenserhebung haben sich verschiedene Verfahren herausgebildet, die im Wesentlichen in Beobachtungs-, Interview- und indirekte Techniken unterschieden werden [Hau00].

*Interviews* erlauben den direkten Wissenstransfer vom Experten zum Wissensingenieur. Zu Beginn der Wissensakquisition werden dabei häufig unstrukturierte Interviews eingesetzt,

um Konzepte und besondere Sprachweise des Problemumfeldes zu erfassen. Daraus ergeben sich unterschiedliche Themen, die in strukturierten Interviews näher zu beleuchten sind. Strukturiert bedeutet dabei, dass Ablauf, Inhalt und Ziel der Befragung im Vorfeld festgelegt werden. Innerhalb der Gespräche können auch Fallstudien durchgeführt werden, um den Lösungsprozess anhand konkreter Probleme zu untersuchen.

Mit Hilfe von *Beobachtungstechniken* wie der Protokollanalyse oder beobachteten Fallstudien wird Wissen durch Erfassen und Auswerten der Vorgehensweise des Experten beim Problemlösen gewonnen. Sie ergänzen die Interviewtechniken, indem verbal durch den Experten schlecht beschreibbare Vorgänge mit ihrer Hilfe darstellbar werden. Hier sind beobachtete Fallstudien ein geeignetes Mittel, um die Vorgehensweise des Experten zu untersuchen. Dabei wird der Experte beim Problemlösen observiert und durch gezielte Fragen zur einer Beschreibung seiner Tätigkeiten animiert.

Unter *indirekten Techniken* sind diejenigen Strategien zusammengefasst, die auf das nicht explizit ermittelbare Wissen abzielen. Solches Wissen wird meist unbewusst eingesetzt und ist damit auch in Befragungen nicht erfassbar. Hier helfen durch den Experten auszuführende Verfahren wie die Ähnlichkeitsanalyse von Konzepten oder die Strukturierung von Fachwissen bei der Wissensermittlung.

Der Prozess der Wissensermittlung wird in unterschiedliche Phasen unterteilt, die die Vorgehensweise des Wissensingenieurs näher beleuchten (Abbildung 5.3) [HRWL83]. Identifikation und der Konzeptionalisierung dienen dem Einstieg in das Problemumfeld. Wesentliche Aufgaben sind hier die Problemdefinition, die Auswahl des einzusetzenden Expertenwissens sowie die Ausarbeitung der grundlegenden Konzepte und Zusammenhänge zur Problembeschreibung. Mit den Phasen Formalisierung, Implementation und Test sind die Bestimmung der Art der Wissensrepräsentation, die Hinterlegung des Wissens in Regeln und die Evaluation des Gesamtsystems verknüpft. Während des Tests ist zu entscheiden, ob die Funktion des Expertensystems der Aufgabenstellung entspricht. Ist das nicht der Fall, ist über Korrekturen des Regelwerkes (*Verfeinerung*), eine Neustrukturierung des erworbenen Wissens (*Redesign*) oder eine *Neuformulierung* des Gesamtproblems eine Anpassung des Expertensystem vorzunehmen. Diese Tätigkeit hat in enger Zusammenarbeit mit dem Experten zu erfolgen.

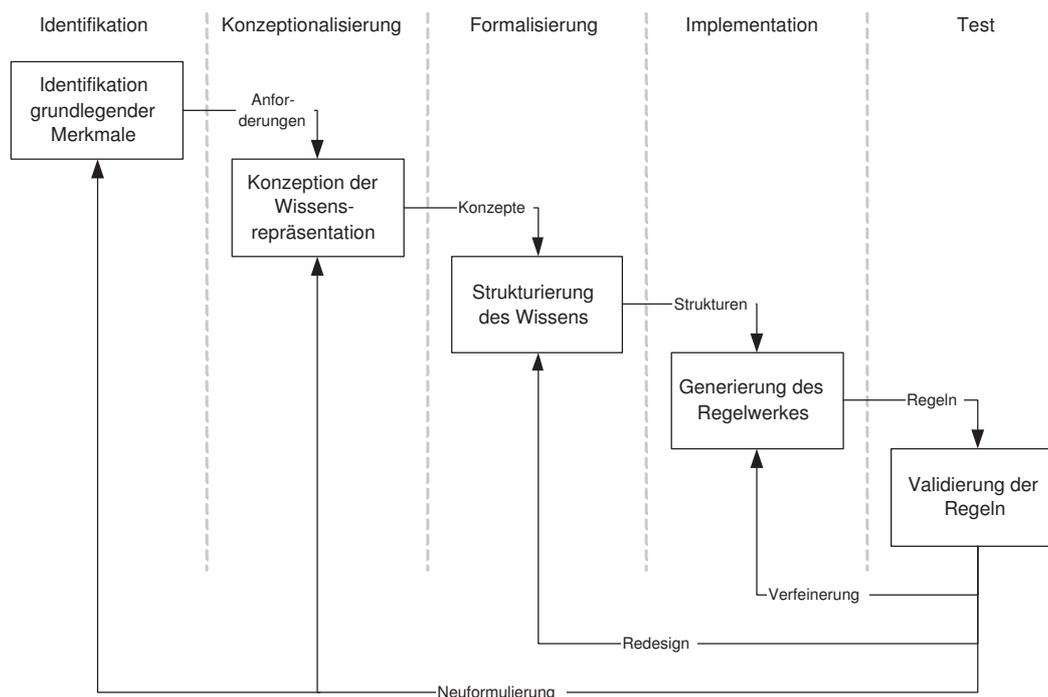
## 5.2 Konzeption des Missionsüberwachungssystems

Nach dem allgemeinen Überblick zu Aufbau und Entwurf von regelbasierten Expertensystemen soll in diesem Abschnitt das Konzept der wissensbasierten Überwachung der Mission vorgestellt werden.

### 5.2.1 Aufgaben der Missionsüberwachung

Grundlegende Aufgabe der wissensbasierten Missionsüberwachung ist die Erfassung von Abweichungen zwischen dem gewünschten und dem realen Zustand des autonomen mobilen Systems [BPRW97]. Real bedeutet hier, dass die Zustandsbeschreibung durch einen externen Beobachter (objektive Realität, Wissen) oder mit Hilfe der systemeigenen Sensorik und geeigneten Auswertungsverfahren (subjektive Realität, Wahrnehmung) erfolgt.

Der gewünschte Systemzustand ergibt sich anhand unterschiedlicher, für die erfolgreiche



**Abb. 5.3:** Phasen der Wissensermittlung nach [HRWL83]

Planerfüllung wesentlicher Kriterien. Diese Merkmale werden aus den einzelnen Manövern des Missionsplanes und den zu erreichenden übergeordneten Vorgaben der Mission abgeleitet.

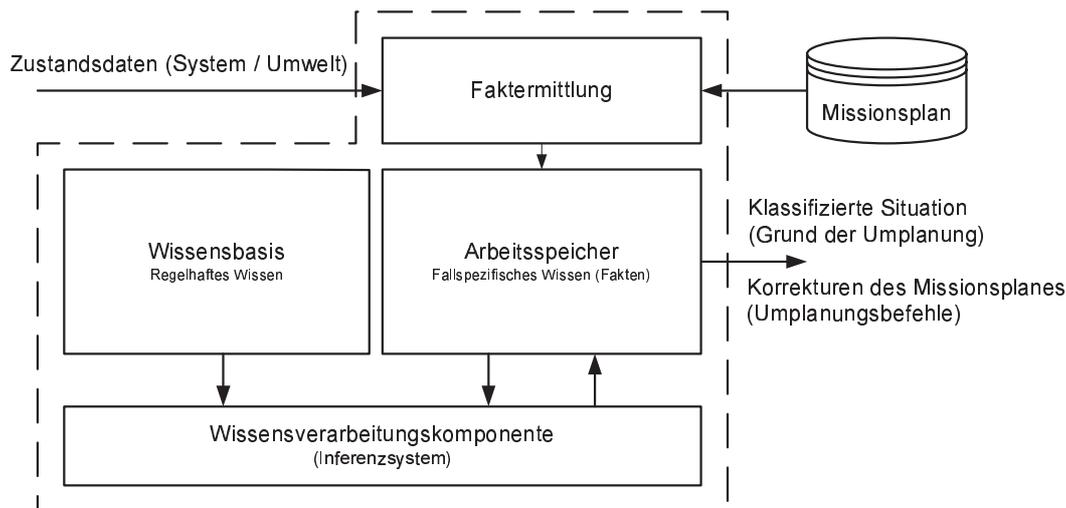
Die Überwachung beschränkt sich auf die Beobachtung und Kontrolle der durch den Missionsplan vorgegebenen Zielstellungen. Die Diagnose von Defekten des mobilen Systems gehört nicht zu ihrem Aufgabenspektrum, erst die sich daraus ergebenden Auswirkungen auf die Durchführung der Mission werden berücksichtigt. Häufig ist ein speziell auf die verwendete Plattform abgestimmtes Diagnosemodul für die Detektion und Beschreibung von hardwarebedingtem Fehlverhalten verantwortlich. Dessen Ergebnisse fließen dann in die Entscheidungsfindung der Missionsüberwachung mit ein.

Ergebnis der Überwachung soll entsprechend der Gesamtkonzeption neben der Detektion eines Fehlverhaltens des mobilen Systems auch ein Vorschlag zur Überwindung dieser Situation sein. Dieser Vorschlag beinhaltet konkrete Angaben über die notwendigen Anpassungen des Missionsplanes. Die zu lösende Aufgabe der wissensbasierten Überwachung der Mission kann somit zusammenfassend folgendermaßen formuliert werden:

*Überwache die Zustandsgrößen des Fahrzeuges sowie die Informationen über die Umgebung und ermittle, ob eine Anpassung des (globalen) Missionsplanes des autonomen mobilen Systems notwendig ist. Bestimme die erforderlichen Korrekturen am Missionsplan.*

### 5.2.2 Aufbau der Missionsüberwachung

Im Abschnitt 5.1.1 wurde bereits erwähnt, dass für den Einsatz eines Expertensystems in einem autonomen System lediglich die Kernkomponenten Wissensbasis, Arbeitsspeicher und



**Abb. 5.4:** Aufbau der Missionsüberwachung

Inferenzsystem benötigt werden. Dementsprechend ergibt sich der prinzipielle Aufbau der Missionsüberwachung aus Abbildung 5.4. Die verfügbaren Informationen über das mobile System, seine Umwelt sowie die Aufgaben der Mission werden durch die Faktermittlung in den Arbeitsspeicher des Produktionssystems übertragen. Dieser Vorgang muss regelmäßig bei Vorliegen neuer Erkenntnisse wiederholt werden.

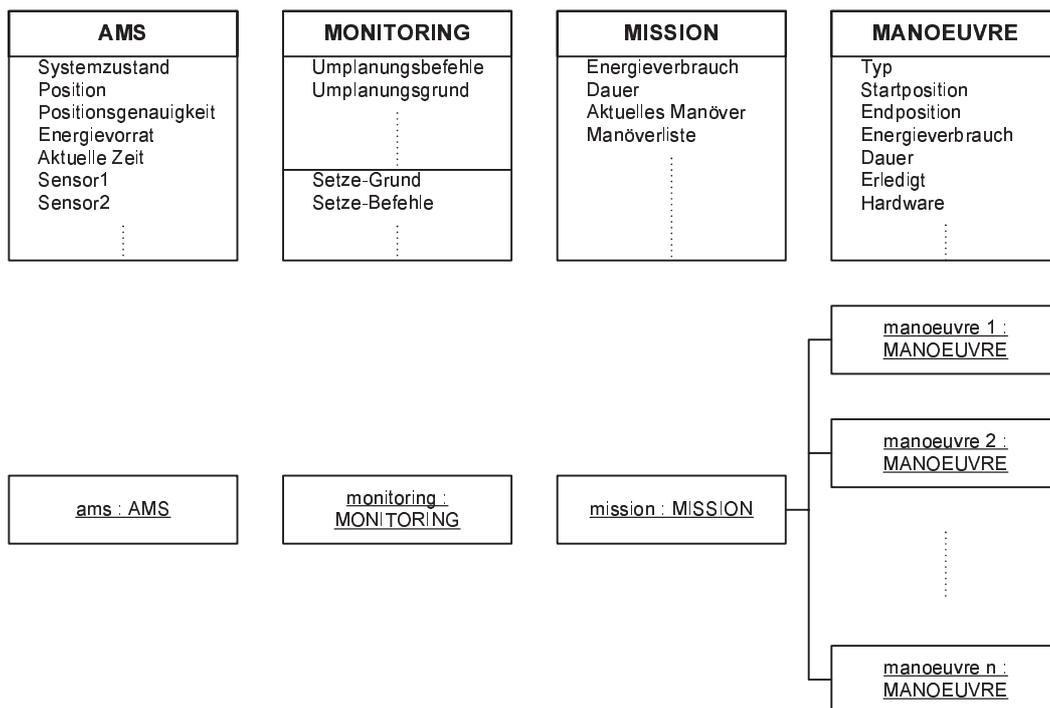
Die Arbeit des Inferenzsystems kann zyklisch oder ereignisgesteuert bei Vorliegen von wichtigen Informationen, zum Beispiel beim Ausfall der Sensorik zur Hindernisvermeidung, gestartet werden. Die Produktionsregeln sind derart aufgebaut, dass nach Ausführung des Konklusionsteils Informationen zu den Gründen einer Planmodifikation sowie den durchzuführenden Korrekturen am Plan vorliegen.

#### 5.2.2.1 Faktermittlung und Arbeitsspeicher

Die Faktermittlung generiert aus den zur Verfügung stehenden Informationen Fakten, die im Arbeitsspeicher abgelegt werden. Zur Wissensrepräsentation eignen sich besonders objektorientierte Techniken, da sie das Wissen sowie die zur Anwendung notwendigen Verarbeitungsroutinen kapseln.

Die für die Missionsüberwachung wesentlichen Elemente des mobilen Systems wurden zu Objekten zusammengefasst und als Klassen modelliert (Abbildung 5.5). Die Attribute der einzelnen Klassen sind dabei von der jeweiligen Plattform des mobilen Systems sowie der Struktur des Missionsplanes abhängig. Bei der Faktermittlung werden nun Instanzen dieser Klassen erzeugt und mit den verfügbaren Informationen gefüllt. Für einen effektiven Mustervergleich, der die Ermittlung der aktivierten Regeln innerhalb der Inferenzstrategie vornimmt, werden nur diejenigen Informationen in den Arbeitsspeicher eingetragen, die sich gegenüber dem Vorzustand geändert haben. Das reduziert den Rechenzeitaufwand, der entsprechend den Aussagen in Abschnitt 5.1.3 wesentlich von der Anzahl der geänderten Fakten innerhalb eines Arbeitszyklus des Inferenzsystems abhängig ist.

Die Klasse MONITORING nimmt innerhalb dieser Struktur eine besondere Stellung ein, da sie die Ergebnisse der Missionsüberwachung beinhaltet. Nach Abschluss eines Zyklus



**Abb. 5.5:** Klassenstruktur und Objektinstanzen des Arbeitsspeichers

der Missionsüberwachung kann über eine Auswertung der Attribute dieser Klasse erkannt werden, ob und welche Modifikationen am Missionsplan durchzuführen sind.

#### 5.2.2.2 Wissensbasis

Die Wissensbasis nimmt die während der Wissensermittlung gewonnenen Informationen in Form von Produktionsregeln auf. Sie ist in einzelne Module aufgeteilt. Die Modularität dient der Abgrenzung unterschiedlicher Wissensseinheiten, damit spezielles Wissen für einen bestimmten Typ von mobilen Systemen von dem allgemeingültigen Wissen unterschieden werden kann. Dadurch ist eine einfache Anpassung der Wissensbasis sowohl an ein anderes mobiles System als auch an weitere Missionsprofile für ein vorhandenes System möglich.

Die Produktionsregeln nutzen die während der Faktermittlung generierten Objektinstanzen, um die Notwendigkeit zur Umplanung zu bestimmen. In den Prämissen der einzelnen Regeln sind dazu die Bedingungen zu formulieren, die zur Aktivierung der jeweiligen Regeln benötigt werden. Der Programmauszug 5.1 zeigt beispielsweise eine Regel zum Entfernen aller Manöver aus dem Missionsplan, die einen bestimmten Sensor (*Sensor 1*) benötigen. Diese Regel verdeutlicht einen wesentlichen Vorteil der regelbasierten Wissensdarstellung gegenüber der prozeduralen Repräsentation. Sind im Missionsplan mehrere Manöver vorhanden, die die Bedingungen der Prämisse erfüllen, wird die Regel für jedes dieser Manöver aktiviert und realisiert. Schleifenläufe wie in prozeduralen Programmiersprachen sind dazu nicht notwendig.

Die Regeln innerhalb der Wissensbasis können mit Hilfe von Prioritäten anhand ihrer Wertigkeit für die Planerfüllung abgestuft werden. Diese Prioritäten finden innerhalb des Inferenzverfahrens zur Einordnung in die Liste der aktivierten Regeln Verwendung. Damit lässt

**Programmauszug 5.1: Regel zum Entfernen von Manövern bei Sensordefekten**


---

```

(defrule HARDWARE:Sensor1-DEFEKT
  ; Priorität der Regel festlegen
  (declare (salience 80))
  ;
  ;
  ; Prämisse
  ;
  ;
  ; Prüfung, ob in der Instanz [ams] Sensor1 als defekt gesetzt ist
  (object (name [ams]) (Sensor1 defekt))
  ; Suchen von Manövern, die die Sensorik Sensor1 verwenden und noch nicht erledigt
  ; wurden. In die Variable ?man-name wird der Name des Manövers eingetragen.
  (object (is-a MANOEUVRE) (Erledigt nein) (Hardware $? "Sensor1" $?)
    (name ?man-name))
=>
  ;
  ; Konklusion
  ;
  ;
  ; Bildschirmausgabe
  (format t "Lösche Manöver %s wegen Defekt von Sensor1.%n" ?man-name)
  ; Eintragen des Umplanungsgrundes in die Instanz [monitoring] der Klasse MONITORING
  (send [monitoring] Setze-Grund "Sensor1-Defekt")
  ; Umplanungsbefehl: Löschen des Manövers mit dem Namen ?man-name
  (send [monitoring] Setze-Befehle (str-cat "DEL " ?man-name))
)

```

---

sich die Reihenfolge der Regelbearbeitung bewusst steuern.

Im Konklusionsteil der Regeln können neue Fakten generiert werden, die der Aktivierung weiterer Regeln dienen. Damit sind auch komplexere Zusammenhänge mit einer größeren Anzahl einfacher Regeln darstellbar, sodass die Wissensbasis zwar umfangreicher, aber die einzelne Regel verständlicher wird.

Des Weiteren sind derartige Fakten auch als Status- oder Zustandsinformationen für die Regeln anwendbar. Das ist unter anderem hilfreich, wenn die Zahl der Regelaktivierungen begrenzt werden soll. Eine Beschränkung ist dabei zeitlich (Anzahl der Aktivierungen pro Zeiteinheit), räumlich (Anzahl der Aktivierungen pro Wegstrecke) oder bezogen auf einzelne Manöver (Anzahl der Aktivierungen pro Manöver) möglich. Beispielsweise kann bei einem Unterwasserfahrzeug die Anzahl der Auftauchmanöver zur Durchführung eines Positionsabgleichs mittels GPS begrenzt werden, um den bei großen Tauchtiefen sehr hohen Zeit- und Energiebedarf für diesen Abgleich zu vermeiden. Außerdem nimmt die Positionsgenauigkeit durch das Abtauchen auf die Zieltiefe wieder ab, so dass eine mehrfache Anforderung eines derartigen Manövers innerhalb kurzer Zeit erfolgen kann. Unter diesen Umständen mag eine Fortsetzung der Mission, auch mit einem größeren Positionsfehler, sinnvoller sein als das häufige Durchführen eines Positionsabgleichs.

Die Auswertung derartiger Fakten in der Regelprämisse kann zur Abweisung einer Umplanungsanforderung führen. Die Menge aller Fakten, die als Statusspeicher dienen, repräsentiert dann den internen Zustand des Expertensystems, der während der Missionsdurchführung regelmäßig gesichert werden muss. Damit ist gewährleistet, dass bei einem Neustart des Missionsmanagements infolge eines Rechnerausfalls der vollständige Status des Expertensystems wieder zur Verfügung steht und eine uneingeschränkte Fortsetzung der Arbeit erfolgen kann.

### 5.3 Zusammenfassung

In diesem Kapitel wurden die Grundlagen und die Arbeitsweise der intelligenten Missionsüberwachung vorgestellt. Kernstück ist ein regelbasiertes Expertensystem, das zur Erkennung von Umplanungssituationen und der zugehörigen Umplanungsbefehle genutzt wird. Die Regeln der Wissensbasis und die Klassenstruktur des Arbeitsspeichers sind auf die Erfordernisse der konkreten mobilen Plattform angepasst, basieren aber auf einer universell verwendbaren Grundstruktur.

Die hinterlegten Regeln sind sowohl für die Erkennung von Umplanungssituationen als auch für die Ermittlung der erforderlichen Planmodifikationen verantwortlich. Damit wird eine Konzentration der notwendigen Anpassungen für den Fall erreicht, dass weitere Zustände erkannt werden, die eine Umplanung erfordern. Allein durch die Aufstellung und den Test der entsprechenden Regeln ist dann eine funktionale Erweiterung des intelligenten Missionsmanagements erreichbar.

Das vorgestellte Grundgerüst für die wissensbasierte Missionsüberwachung wird für die beiden Testplattformen verwendet, die in Kapitel 7 vorgestellt werden.



## 6 Strategien zur Missionsumplanung

In diesem Kapitel erfolgt die Beschreibung der für die Missionsumplanung benötigten Methoden. Nach einer Einführung in die prinzipielle Arbeitsweise der Umplanung werden die atomaren und komplexen Umplanungsbefehle, ihre Auswirkungen auf den Missionsplan sowie die verwendeten Algorithmen vorgestellt. Die geografischen Verfahren zur Planprüfung und Planmodifikation beschließen die Darlegungen.

Soweit möglich wird bei den Ausführungen von den konkreten Gegebenheiten eines autonomen mobilen Systems abstrahiert. Ein Großteil der Strategien ist nicht auf die Klasse autonomer Systeme beschränkt und besitzen ein breiteres Einsatzgebiet. So wird im Folgenden beispielsweise ein allgemeiner Zustandsvektor für die Beschreibung des Systemzustandes verwendet, der bei einem mobilen System unter anderem aus der Positionsangabe gegenüber einem Referenzsystem besteht.

### 6.1 Ablauf der Missionsumplanung

Die Missionsumplanung beginnt mit der Übergabe der zu realisierenden Umplanungsbefehle (Abbildung 6.1). Die Befehle werden zunächst sequentiell ausgeführt. Je nach Art der Planmodifikationen ist es notwendig, im Anschluss den Missionsplan mit Hilfe einer digitalen Geländemodells zu prüfen. So muss zum Beispiel nach dem Einfügen eines neuen Manövers getestet werden, ob das mobile System dieses Planelement befahren kann, ohne die Beschränkungen durch das Gelände zu verletzen.

Die Planprüfung nutzt Routinen zur Schnittberechnung zwischen den einzelnen Planelementen und der geografischen Datenbasis. Dazu wird der Missionsplan in Geradenstücke zerlegt und jede dieser Strecken auf Schnittpunkte mit dem Geländemodell untersucht.

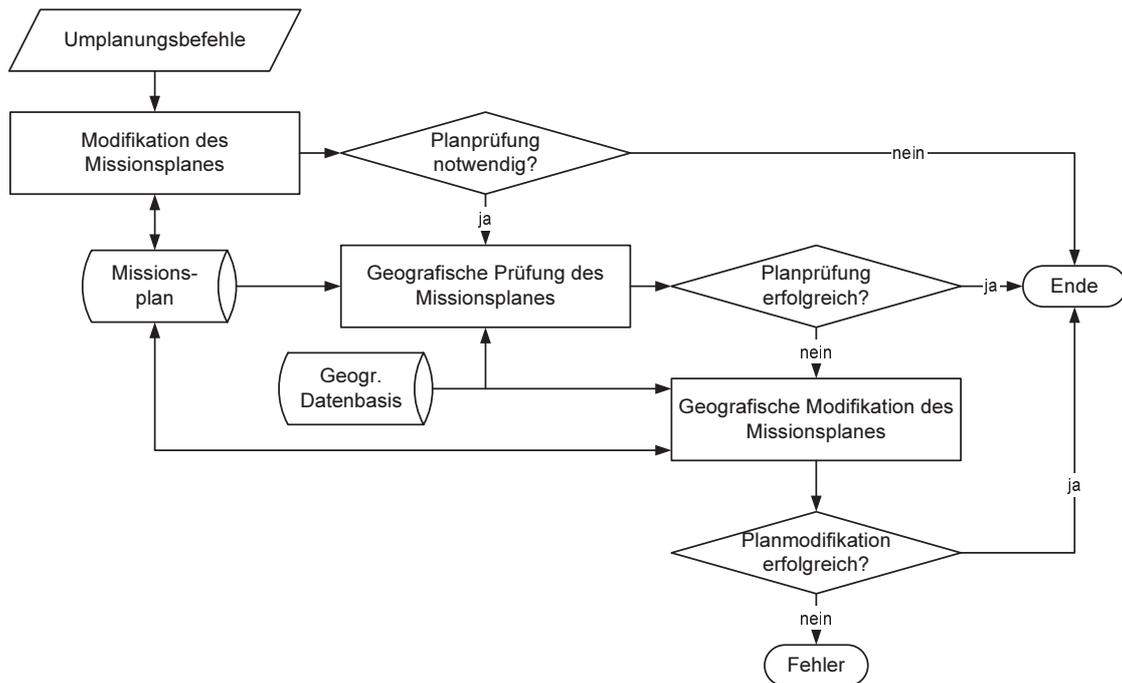
Treten Schnittpunkte auf, soll mit Hilfe der geografischen Planmodifikation für die betroffenen Manöver jeweils eine gültige Route gefunden werden. Dafür ist die geografische Planmodifikation zuständig. Sie nutzt graphenbasierte Suchverfahren, um mögliche fahrbare Strecken zu ermitteln.

### 6.2 Allgemeine Methoden zur Planmodifikation

#### 6.2.1 Voraussetzungen

*Aufbau des Missionsplanes.* Vor der Diskussion der eigentlichen Modifikationsverfahren sollen einige, für das Verständnis wesentliche Grundlagen vermittelt werden. Ausgangspunkt der folgenden Betrachtungen ist die Definition 1.2 des Begriffes *Missionsplan* als eine Folge von Manövern.

Ein Missionsplan ist somit entsprechend Abbildung 6.2 als lineare Sequenz von Planelementen darstellbar. Mit  $n$  wird dabei die Anzahl der in einem Plan vorhandenen Elemente



**Abb. 6.1:** Grundlegender Ablauf der Missionsumplanung

bezeichnet. Die Festlegung, dass die zu betrachtenden Pläne eine sequentielle Anordnung der Manöver besitzen müssen, schränkt in keiner Weise die Möglichkeiten zur Planmodifikation ein, da mit Hilfe der im Kapitel 3 definierten Umlanungsbehele beliebige Veränderungen an einem solchen Plan möglich sind. Ein wesentlicher Vorteil des sequentiellen Planes im Gegensatz zu nebenläufigen oder parallelen Plänen liegt in der Vereinfachung der Planerstellung durch einen menschlichen Operator. Sie weicht zum Beispiel nicht von der Planung des Kapitäns eines Flugzeuges oder Schiffes ab. Die durchzuführenden Aktionen (oder abzufahrenden geografischen Positionen) werden nacheinander geplant und realisiert. Wahrscheinlich gerade deshalb sind sequentielle Missionspläne eine weit verbreitete Planungssprache für autonome Systeme, besonders im Bereich von autonomen Luftfahrzeugen (UAVs, Unmanned Aerial Vehicles) oder autonomen Tauchrobotern (AUVs) [Pri00], [Leo96], [BL99].



**Abb. 6.2:** Aufbau eines sequentiellen Missionsplanes

*Eigenschaften von Planelementen.* Die einzelnen Planelemente des Missionsplanes werden, wie bereits in der Definition 1.2 erwähnt, oft auch als Manöver bezeichnet. Deshalb wird dieser Begriff im Folgenden auch als Synonym verwendet. Jedes Planelement besitzt grundlegende Eigenschaften, die in Analogie zu den Operatoren bei den Planerstellungsmethoden der künstlichen Intelligenz benannt werden sollen (Tabelle 6.1) [Her89].

Bevor ein Planelement durch das System aktiviert werden kann, muss der aktuelle System-

zustand dem *Anfangszustand* des Planelementes entsprechen. Je nach Art des autonomen mobilen Systems können unterschiedliche Variablen des Gesamtzustandsraumes für den Zustandsvektor  $\mathbf{x}$  benutzt werden. So wird ein fliegendes oder tauchendes System zumindest die dreidimensionale geografische Position als Beschreibung des Zustandes verwenden. Mit Aktivierung des Planelementes werden dann die damit verbundenen *Aktionen* ausgeführt. Beeinflusst wird deren Arbeitsweise durch den *Parametervektor*  $\mathbf{p}$ . Dieser beinhaltet charakteristische Größen für die unterschiedlichen Manövertypen (zum Beispiel Geschwindigkeiten, Sicherheitsabstände usw.). Eine besondere Rolle innerhalb der Parameter spielen die so genannten Erfüllungsgrade  $e$ . Davon können bis zu drei implementiert sein:

- Anfangserfüllung  $e_A$  - Wert, bei dem die Ausführung des Planelementes beginnt,
- Enderfüllung  $e_E$  - Wert, bei dem die Realisierung des Planelementes endet,
- aktuelle Erfüllung  $e_{akt}$  - bereits vollendeter Anteil des Manövers, Wertebereich  $e_A \leq e_{akt} \leq e_E$

Die Erfüllungsgrade sind auf die realisierte Zustandsänderung  $\Delta\mathbf{x}$  bezogen. Mit Hilfe von Start- und Enderfüllung ist somit eine Korrektur von Anfangs- und Endzustand des Planelementes möglich, die eine große Rolle sowohl bei dem Einfügen-Befehl als auch bei der Optimierung eines Missionsplanes spielt:

$$\begin{aligned}\Delta\mathbf{x} &= \mathbf{x}_E - \mathbf{x}_A \\ \mathbf{x}_A^* &= f(\mathbf{x}_A, \Delta\mathbf{x}, e_A) \\ \mathbf{x}_E^* &= f(\mathbf{x}_E, \Delta\mathbf{x}, e_E).\end{aligned}\tag{6.1}$$

Über diese Beziehungen kann erreicht werden, dass Planelemente bei einer vordefinierten Anfangserfüllung starten bzw. bei einer bestimmten Enderfüllung enden, ohne dass die geplanten Start- und Endzustände modifiziert werden müssen.

Unter dem Begriff *Ressourcen* werden die für die Ausführung eines Manövers benötigten Betriebsmittel zusammengefasst. Dazu zählen bei einem mobilen System unter anderem der Energie- und Zeitbedarf. Der *Wert* eines Planelementes spiegelt den Nutzen wider, den der Systembetreiber bei erfolgreichem Absolvieren des Manövers erhält. Neben den Ressourcen spielt der Wert eine große Rolle bei der Optimierung eines Missionsplanes.

Der *Endzustand* charakterisiert den Zustandsvektor bei Beendigung des Planelementes. Er dient als Umschaltkriterium zum nachfolgenden Element im Plan, wobei die Enderfüllung berücksichtigt werden muss. Nicht unerwähnt soll hier bleiben, dass besonderes Augenmerk bei der Realisierung eines solchen Systems auf die ordnungsgemäße Beendigung der gesamten Mission zu legen ist. Durch Einnahme eines sicheren Systemzustandes oder Definition eines Abschlussmanövers kann verhindert werden, dass eine Gefährdung für oder durch das mobile System auftritt. Dies ist sowohl bei der Planung der Mission durch den Operator als auch bei der Umplanung durch das intelligente Missionsmanagement zu berücksichtigen und spielt bei einem etwaigen Missionsabbruch eine große Rolle.

**Tabelle 6.1:** *Eigenschaften von Planelementen*

Eigenschaft	Zeichen	Beschreibung
Anfangszustand	$\mathbf{x}_A$	Zustandsvektor bei Aktivierung des Planelements, auch Vorbedingung
Endzustand	$\mathbf{x}_E$	Zustandsvektor bei Beendigung des Planelements, auch Nachbedingung
Aktion(en)	$\mathbf{a}$	auszuführende Handlung(en) im Planelement
Parameter	$\mathbf{p}$	Parametervektor (z.B. Geschwindigkeiten)
Ressourcen	$\mathbf{c}$	benötigte Ressourcen für die Realisierung (z.B. Zeit, Energie), auch als Kosten bezeichnet
Wert	$w$	Erfolgsbewertung des Planelementes, auch Profit genannt

*Kontinuitätsbedingung.* Mit den genannten Eigenschaften lässt sich ein sequentieller Plan folgendermaßen definieren:

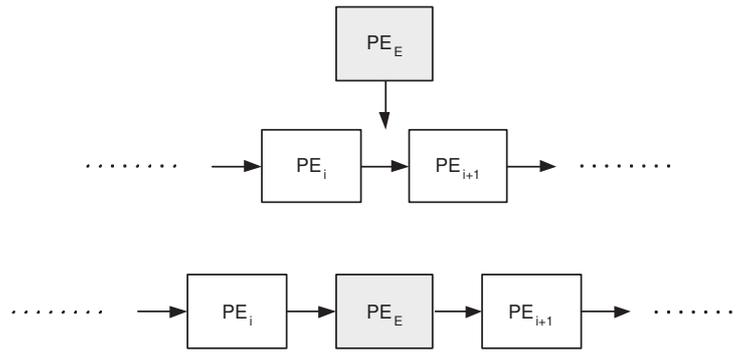
$$\mathbf{x}_A(i+1) = \mathbf{x}_E(i) \quad \forall i = 1, 2, \dots, n-1. \quad (6.2)$$

Dieser auch als Kontinuitätsbedingung bezeichnete Zusammenhang ist die Basis der im Folgenden vorzustellenden Verfahren. Er besagt, dass stets der Endzustand eines Planelementes dem Anfangszustand des nächsten Planelementes entsprechen muss. Diese Festlegung vereinfacht den Umplanungsprozess erheblich, da die fahrdynamischen Gegebenheiten der Roboterplattform nicht berücksichtigt werden müssen. Die Anpassung an das konkrete mobile System ist bei Bedarf in den Schichten der Manöver- und Fahrzeugführung zu realisieren. Für den mobilen Kleinroboter sind die notwendigen Verarbeitungsschritte im Abschnitt 7.1.2.3 dargestellt.

*Füllplanelemente.* Zum besseren Verständnis der nachfolgenden Aussagen soll ein weiterer neuer Begriff, das Füllplanelement, eingeführt werden. Als *Füllplanelemente* (oder auch Füllmanöver) sollen diejenigen Planelemente bezeichnet werden, die lediglich zum Anpassen von Zustandsunterschieden aufeinander folgender Manöver im Plan vorhanden sind. Sie werden zum Beispiel beim Löschen von Planelementen eingesetzt. Nach Durchführung aller Modifikationen an einem Missionsplan werden aufeinander folgende Füllmanöver in der so genannten Planbereinigung automatisch zusammengefasst. Der Typ und damit die Aktionen eines Füllmanövers hängt von der Wahl des Zustandsvektors für die Beschreibung der Planelemente und damit auch von den konkreten Gegebenheiten der Roboterplattform ab. Für ein mobiles System mit der Fahrzeugsposition als Zustandsvektor besteht zum Beispiel ein Füllplanelement in einem Transit von einer Start- zu einer Endposition, für ein AUV stellen darüber hinaus auch Auf- und Abtauchmanöver Füllplanelemente dar.

### 6.2.2 Einfügen von Planelementen

Der atomare Umplanungsbefehl *Einfügen* erlaubt das Hinzufügen von Planelementen zu einem sequentiellen Missionsplan. Er besitzt mehrere zwingend erforderliche Parameter, die den Typ des einzufügenden Planelementes, seine Position im Missionsplan sowie die Art des



**Abb. 6.3:** Einfügen des Planelementes E nach dem Element i

Einfügens definieren. Dabei wird zwischen dem *Einfügen nach* einem Planelement und dem *Einfügen in* ein Planelement unterschieden.

Das Einfügen nach einem Planelement trennt die Verbindung zwischen den betroffenen Elementen und fügt dazwischen das neue Element ein (Abbildung 6.3). Entsprechend der Kontinuitätsbedingung ergibt sich zunächst, dass Anfangs- und Endzustand des neuen Manövers gleich sind:

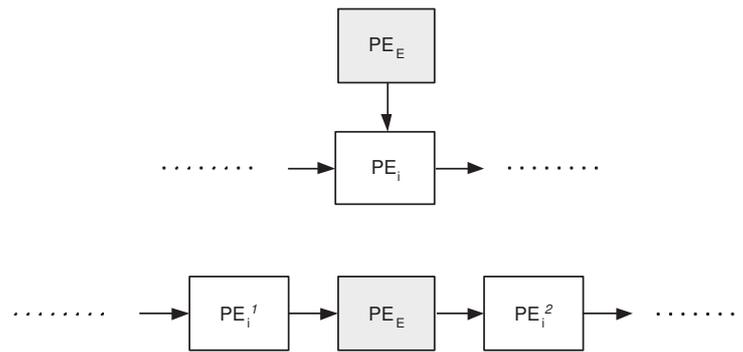
$$\begin{aligned} \text{vor dem Einfügen: } \quad & \mathbf{x}_A(i+1) = \mathbf{x}_E(i) \\ \text{nach dem Einfügen: } \quad & \mathbf{x}_A(E) = \mathbf{x}_E(i) \\ & \mathbf{x}_E(E) = \mathbf{x}_A(i+1). \end{aligned}$$

Je nach Typ des eingefügten Planelementen kann die Gleichheit beider Zustände zulässig sein, ansonsten muss mit Hilfe von Füllmanövern eine Anpassung vorgenommen werden. Dabei kann sowohl ein Füllmanöver vor dem einzufügenden Planelement als auch danach notwendig sein. Das hängt jedoch von der konkreten Implementierung auf dem jeweiligen autonomen mobilen System ab.

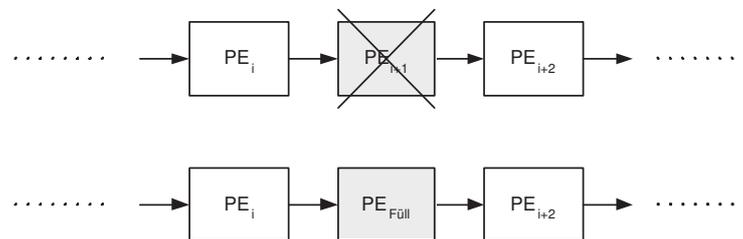
Beim Einfügen in ein Element wird die Arbeit an einem Manöver unterbrochen, das eingefügte Element realisiert und anschließend das unterbrochene an der gleichen Stelle fortgesetzt, an der es beendet wurde. Um dies in einem sequentiellen Plan darstellen zu können, muss eine zweite Instanz des Planelementes  $i$  nach dem Element  $E$  eingefügt werden (Abbildung 6.4). Der im Moment des Abbruchs vorhandene aktuelle Erfüllungsgrad  $e_{akt}$  von  $PE_i$ <sup>1</sup> wird als Starterfüllung  $e_A$  für die Kopie  $PE_i$ <sup>2</sup> eingesetzt, sodass dessen Startposition mit der Unterbrechungsposition der ersten Instanz des Manövers identisch ist. Auch hier kann es notwendig sein, dass Füllmanöver vor und nach dem eingefügten Planelement in den Missionsplan aufgenommen werden müssen.

### 6.2.3 Löschen von Planelementen

Beim Löschen wird das betroffene Manöver aus dem Missionsplan entfernt. Häufig ist daraufhin die Kontinuitätsbedingung verletzt, da für die meisten Planelemente  $\mathbf{x}_A(i+2) \neq \mathbf{x}_E(i)$  gilt. Um die Kontinuität wiederherzustellen, wird an die Position des gelöschten Manövers ein Füllplanelement eingefügt (Abbildung 6.5). Bei einem mobilen System muss beispielsweise nach dem Löschen eines Planelementes ein Transit von dessen Anfangs- zur Endposition



**Abb. 6.4:** Einfügen des Planelementes  $E$  in das Element  $i$

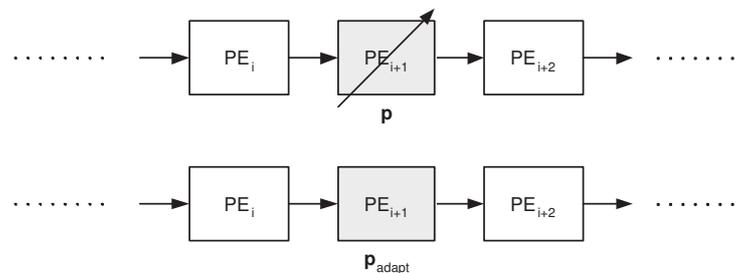


**Abb. 6.5:** Löschen des Planelementes  $i$

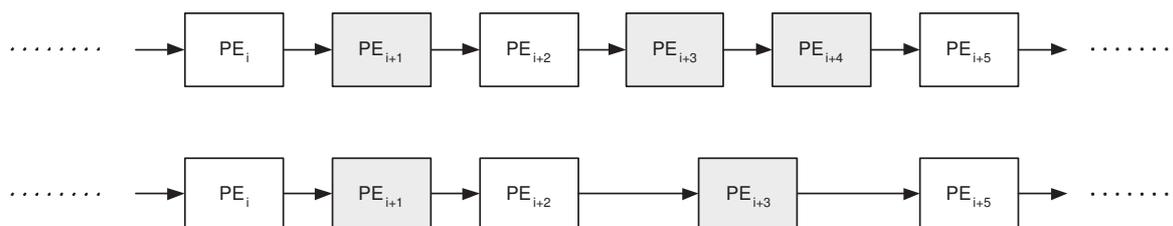
eingefügt werden. Über den Typ des Transitmanövers entscheidet die zu überbrückende Differenz zwischen den beiden Positionen.

#### 6.2.4 Modifizieren von Planelementen

Zahlreiche Situationen erfordern eine Anpassung von Parametern eines Planelementes. Dies geschieht mit dem atomaren Befehl *Modifizieren* (Abbildung 6.6). Dabei können die Einflussmöglichkeiten auf den Parametervektor je nach Implementierung des Missionsplanes unterschiedlich sein. So ist es zum Beispiel häufig sinnvoll, die Fahrgeschwindigkeit einzelner Planelemente auf die vorhandenen Ressourcen abzustimmen, während Anpassungen von einem eventuell in den Manöverparametern vorhandenen Prioritätswert meistens nicht notwendig sind.



**Abb. 6.6:** Modifizieren des Planelementes  $i+1$



**Abb. 6.7:** Bereinigung eines Missionsplanes, Füllplanelemente sind grau hinterlegt

Die Modifikation von Start- oder Endzustand einzelner Planelemente kann mit Hilfe dieses Befehls ebenfalls vorgenommen werden. Dabei ist zu berücksichtigen, dass zur Einhaltung der Kontinuitätsbedingung auch eine Anpassung des Vorgängers oder Nachfolgers im Missionsplan notwendig ist.

### 6.2.5 Planbereinigung

Ziel der Planbereinigung ist es, aufeinander folgende Füllplanelemente aus dem Plan zu entfernen. Das ist möglich, da diese jeweils nur Zustandsübergänge realisieren, die auch durch ein kombiniertes Manöver der gleichen Art erreichbar sind. Abschnitt 7.2.2.3 geht auf Besonderheiten der Planbereinigung ein, die sich darüber hinaus für ein im dreidimensionalen Raum bewegendes mobiles System ergeben. Zur Bereinigung des Planes sind mehrere Arbeitsschritte notwendig:

- Bestimmung der Positionen aller Füllmanöver im Missionsplan,
- Ermittlung aufeinander folgender Füllplanelemente (Startindex und Anzahl der Elemente jeder Folge),
- Löschen der auf den jeweiligen Startindex folgenden Füllplanelemente.

Als Ergebnis entsteht ein Missionsplan, der keine aufeinander folgenden Füllmanöver mehr enthält (Abbildung 6.7). Die Planbereinigung wird stets nach dem Löschen oder Einfügen eines Planelementes und während der Optimierung angewendet, da während dieser Umplanungsmaßnahmen Folgen von Füllplanelementen auftreten können.

Am konkreten Beispiel eines mobilen Systems wird der Nutzen dieses Verfahrens am einfachsten deutlich. Die in der Abbildung 6.7 grau hinterlegten Füllplanelemente stellen auf einer derartigen Plattform jeweils ein Transitmanöver dar, das Anfangs- und Endzustand miteinander verbindet. Zwei aufeinander folgende Elemente wie  $PE_{i+3}$  und  $PE_{i+4}$  stellen dabei nicht zwangsläufig den kürzesten Weg dar, um zum Startzustand von  $PE_{i+5}$  zu gelangen. Durch die Zusammenfassung beider Füllplanelemente wird eine Wegverkürzung erreicht, die besonders bei der Optimierung eines Missionsplanes einen wesentlichen Beitrag zur Verringerung der benötigten Ressourcen des Planes leistet.

## 6.3 Spezielle Verfahren zur Planmodifikation

Die in diesem Abschnitt vorzustellenden Methoden nutzen die allgemeingültigen Verfahren aus Abschnitt 6.2. Sowohl die dort aufgeführten Aussagen als auch die getroffenen Ein-

schränkungen gelten damit ebenfalls für den Missionsabbruch und die Optimierung von Missionsplänen.

### 6.3.1 Missionsabbruch

Der Missionsabbruch stellt einen wichtigen, wenn nicht gar den wichtigsten Umplanungs-befehl dar. Damit soll ein sicherer Abschluss der Mission gewährleistet werden, wenn eine Fortsetzung nicht mehr möglich ist.

Die Ursachen für einen Missionsabbruch können vielfältig sein. Zu nennen sind unter anderem der Ausfall von Sensorik zur Hinderniserkennung, der Ausfall von Aktorik sowie ein extrem schneller Energieabfall. In solchen Situationen ist es erforderlich, die reguläre Planrealisierung zu beenden und stattdessen eine vordefinierte Folge von Planelementen abzuarbeiten, um die Sicherheit des Systems zu gewährleisten.

Voraussetzung für die erfolgreiche Durchführung des Abbruchs einer Mission ist die Verfügbarkeit der dafür notwendigen Hard- und Software des autonomen mobilen Systems. Sind die notwendigen Module nicht funktionsfähig, sollte auf eine, unabhängig von der Steuerung des Systems funktionsfähige Sicherheitseinrichtung zurückgegriffen werden können. So ist zum Beispiel für ein autonomes Luftfahrzeug eine Landung ohne verfügbare Höhenmessung oder Abstandsmessung zum Boden nicht möglich. Die Sicherheitseinrichtung könnte für einen derartigen Roboter aus einem Fallschirm bestehen. Ein AUV benötigt hingegen ein Notaufauchsystem, um genügend Auftrieb für das Erreichen der Wasseroberfläche zu erzeugen. Üblich ist die Mitnahme von Ballast, der in Notsituationen abgeworfen werden kann.

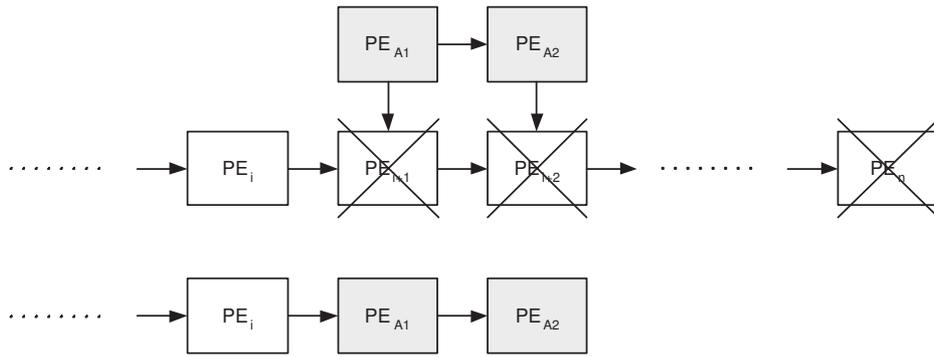
Der geplante Missionsabbruch führt demgegenüber bei Funktionsfähigkeit der benötigten Module folgende Schritte aus, die einen sicheren Systemzustand garantieren (Abbildung 6.8):

- Entfernen aller auf das aktuell ausgeführte Planelement folgenden Elemente,
- Einfügen der für einen Missionsabbruch definierten Manöver  $PE_{Ai}$ , eventuell Parametrieren dieser Manöver.

Je nach Startzustand des ersten Planelementes eines Missionsabbruchs  $PE_{A1}$  kann es erforderlich sein, ein Füllplanelement davor einzufügen, um den Systemzustand zum Abbruchzeitpunkt daran anzupassen. Zum Abschluss dieser Umplanung wird das erste neu eingefügte Manöver aktiviert und vom Führungssystem des autonomen mobilen Systems anschließend realisiert.

### 6.3.2 Optimierung eines Planes

Eine Optimierung des Missionsplanes wird durch die Missionsüberwachung angefordert, wenn die vorhandenen Ressourcen des mobilen Systems nicht für die Erfüllung der verbleibenden Planelemente ausreichen. Das Ziel der Optimierung ist, den Plan derart zu modifizieren, dass er sicher verwirklicht werden kann und dabei den größten realisierbaren Nutzen (oder Wert)  $W$  der Mission repräsentiert. Diese Art der Optimierung versucht demnach, einen bestehenden Plan an die neue Situation anzupassen. Damit soll sichergestellt



**Abb. 6.8:** Abbruch einer Mission bei der Ausführung des Planelementes  $i$

werden, dass die im Vorfeld bei der Planung durch den Menschen vorgegebene Zielstellung der Mission weitestgehend erhalten bleibt.

### 6.3.2.1 Optimierungskriterien und -variablen

Ausgangspunkt für die Lösung einer Optimierungsaufgabe ist stets die Definition des Problems. Wie bereits erwähnt, ist das Ziel dieser Plananpassung, den größtmöglichen Nutzen aus der Mission zu ziehen. Der Wert  $W$  kann mit Hilfe der einzelnen Planelemente folgendermaßen dargestellt werden:

$$f(\mathbf{x}) = W(\mathbf{x}) = \sum_{k=i}^{n(\mathbf{x})} w_k \cdot \mathbf{x}_k, \quad (6.3)$$

wobei  $\mathbf{x}$  der Vektor der Optimierungsvariablen und  $i$  der Index des aktuellen Manövers sind. Es ist zu beachten, dass die Anzahl der Planelemente eines Missionsplanes von den Optimierungsvariablen abhängen kann, was durch  $n(\mathbf{x})$  repräsentiert wird. Das zunächst unbeschränkte Optimierungsproblem ergibt sich damit zu:

$$\max \{f(\mathbf{x})\}. \quad (6.4)$$

Als Beschränkungen für diese Aufgabe treten die verfügbaren Ressourcen des mobilen Systems  $\mathbf{c}_{max}$  auf. Die benötigten Vorräte an Betriebsmitteln sind für jedes Planelement im Vektor  $\mathbf{c}$  hinterlegt, für den Gesamtaufwand  $\mathbf{C}(\mathbf{x})$  der verbleibenden Planelemente der Mission gilt dann:

$$\mathbf{C}(\mathbf{x}) = \sum_{k=i}^{n(\mathbf{x})} \mathbf{c}_k(\mathbf{x}_k). \quad (6.5)$$

Das Funktional der Beschränkungen kann nun folgendermaßen dargestellt werden:

$$\mathbf{g}(\mathbf{x}) = \mathbf{C}(\mathbf{x}) - \mathbf{c}_{max} = \sum_{k=i}^{n(\mathbf{x})} \mathbf{c}_k(\mathbf{x}_k) - \mathbf{c}_{max}. \quad (6.6)$$

Aus dieser Beziehung wird ersichtlich, dass zum Erfüllen der Beschränkungen  $\mathbf{g}(\mathbf{x}) \leq 0$  gelten muss. Das vollständige, beschränkte Optimierungsproblem ergibt sich damit zu:

$$\max \{f(\mathbf{x}) : \mathbf{g}(\mathbf{x}) \leq 0\}. \quad (6.7)$$

Die Wahl der Optimierungsvariablen beeinflusst wesentlich die Art und den Ablauf der Optimierung. Deshalb müssen vor der Auswahl des zu nutzenden Verfahrens diejenigen Größen festgelegt werden, über die ein Missionsplan an die konkrete Situation angepasst werden kann. Dafür bieten sich verschiedene Parameter an, die direkten Einfluss auf die benötigten Ressourcen des Missionsplanes eines autonomen mobilen Systems haben:

- Über eine *Auswahl der zu realisierenden Planelemente* kann der Betriebsmittelverbrauch mit den verfügbaren Vorräten abgeglichen werden. Auswahl, oder besser gesagt *Aktivierung*, bedeutet hier, dass für jedes verbleibende Planelement entschieden wird, ob es verwirklicht wird. Diese Entscheidung wird mit Hilfe einer booleschen Variable mit dem Wertebereich  $\{0, 1\}$  repräsentiert.
- Über die *Parameter der einzelnen Planelemente* kann unter Umständen ebenfalls der Ressourcenbedarf von Manövern reduziert werden. Für ein mobiles System würde sich dafür beispielsweise die Fahrgeschwindigkeit anbieten. Sie beeinflusst über die Leistungs-Geschwindigkeits-Kennlinie den Energieverbrauch sowie über die direkte Verknüpfung mit der Fahrzeit den Zeitaufwand der Gesamtmission. Parameter dieser Art besitzen meist einen kontinuierlichen Wertebereich  $p_{min} \leq p \leq p_{max}$ .

Bei alleiniger Betrachtung der *Aktivierung der Planelemente* ergibt sich ein diskretes Optimierungsproblem. Diese sind dadurch gekennzeichnet, dass der zulässige Bereich aus einer endlichen oder abzählbar unendlichen Menge  $M$  besteht [BG93]. Für den hier dargestellten Fall ist  $M$  als Menge von binären Vektoren definiert:  $M \subseteq \{0, 1\}^n$ . Da sich die Anzahl der Elemente von  $M$  durch Kombinatorik ermitteln lässt, wird häufig auch von kombinatorischer Optimierung gesprochen. Für einen Satz von  $n$  Aktivierungswerten ergibt sich  $2^n$  als Mächtigkeit von  $M$ .

Werden die *Parameter der einzelnen Planelemente* als Variable für die Optimierung näher untersucht, so treten verschiedene Probleme auf. Zunächst ist die Möglichkeit der Einflussnahme auf den Missionsplan durch Kenngrößen einzelner Manöver abhängig von dem konkreten Aufbau und den Aufgaben des mobilen Systems. Es kann also nicht generell davon ausgegangen werden, dass solche Parameter existieren. Andererseits erhöht der meist kontinuierliche Wertebereich solcher Kenngrößen den Aufwand zur Lösung des Problems, sodass durch Nutzung von Manöverparametern als Optimierungsvariablen zusätzlich zur Aktivierung der Planelemente aus einem diskreten ein gemischt ganzzahliges Optimierungsproblem wird. Zugunsten der Allgemeingültigkeit des zu entwerfenden Verfahrens soll aus den genannten Gründen nur die Aktivierung der einzelnen Planelemente als Variable der Optimierung herangezogen werden.

### 6.3.2.2 Analogiebetrachtungen und Lösungsansätze

Probleme der im letzten Abschnitt geschilderten Art treten in der Realität häufig auf und stellen deshalb einen wesentlichen Forschungsschwerpunkt der Mathematik und Theoretischen Informatik dar. Ein Standardproblem, an dem oft unterschiedliche Algorithmen der diskreten Optimierung getestet werden, ist das so genannte Rucksack-Problem. Es weist eine große Ähnlichkeit mit der zu lösenden Optimierungsaufgabe auf und soll deshalb im Folgenden näher betrachtet werden.

Speziell das eindimensionale lineare Rucksack-Problem ist zum Vergleich sehr gut geeignet und geht von folgender Situation aus. Es sind  $n$  verschiedene Gegenstände verfügbar, wobei jeder einen Wert  $w_k$  und ein Gewicht (gleich Kosten)  $c_k$  besitzt. Ziel ist, diejenigen Artikel auszuwählen, die in einen Rucksack mit dem zulässigen Gesamtgewicht  $c_{max}$  passen und dabei den maximalen Gesamtwert  $W_{max}$  erzielen. Steht nun jeder Gegenstand nur ein Mal zur Verfügung, so ergibt sich das so genannte 0-1 Rucksack-Problem in der folgenden Weise:

$$\begin{aligned} \text{Maximiere} \quad & W(\mathbf{x}) = \sum_{k=1}^n w_k \cdot \mathbf{x}_k \\ \text{unter den Nebenbedingungen:} \quad & \sum_{k=1}^n c_k \cdot \mathbf{x}_k \leq c_{max} \\ & \mathbf{x}_k \in \{0, 1\} \quad k = 1, \dots, n \end{aligned} \quad (6.8)$$

In der Praxis treten derartige Aufgaben häufig auf, so zum Beispiel bei der Finanzplanung<sup>1</sup> oder Projektauswahl<sup>2</sup> [Dom95]. Obwohl dies das einfachste (beschränkte) diskrete Optimierungsproblem darstellt, gehört es zur Komplexitätsklasse NP [GJ79]. Im Gegensatz zur Klasse P der effizient lösbaren Probleme, für die Algorithmen in einer durch ein beliebiges Polynom begrenzten Rechenzeit ein Ergebnis liefern, sind NP-Probleme lediglich effizient überprüfbar. Das Rucksackproblem ist darüber hinaus ein Vertreter der NP-vollständigen Probleme [Lag96]. Derartige Aufgaben gehören zu den schwierigsten der Problemklasse NP und benötigen exponentielle Zeit zur Lösungsfindung. Durch Anwendung von Heuristiken und Approximationsverfahren mit annähernd optimalen Ergebnissen kann jedoch die Suche nach eine Lösung für viele dieser Probleme stark vereinfacht werden [ACG<sup>+</sup>99].

Gerade für das Rucksack-Problem existieren effiziente Methoden, die eine Lösung für eine überschaubare Menge von Variablen (oder Gegenständen) mit vertretbarem Zeitaufwand liefern [MT90], [Mur95]. Dazu gehören unter anderem Verfahren der Dynamischen Programmierung, Branch-and-Bound, Tabu Search oder auch Suchverfahren wie Genetische Algorithmen.

Das Optimierungsproblem aus Gleichung 6.7 besitzt das gleiche Gütefunktional wie das Rucksack-Problem. Die Nebenbedingungen unterscheiden sich jedoch grundsätzlich. Während das Rucksack-Problem lineare eindimensionale Bedingungen besitzt, stellt die in dieser Arbeit zu lösende Aufgabe ein nichtlineares mehrdimensionales Optimierungsproblem dar. Das ergibt sich aus der Ermittlung des Kostenvektors eines Planelementes, da die Nichtrealisierung ( $x_k = 0$ ) in Analogie zum Löschen eines Planelementes meist ein Füllplanelement erfordert:

$$\mathbf{c}_k(\mathbf{x}_k) = \begin{cases} \mathbf{c}_k & (\mathbf{x}_k = 1) \quad \text{Kosten des Planelementes} \\ \mathbf{c}_{k \text{ FPE}} & (\mathbf{x}_k = 0) \quad \text{Kosten des Füllplanelementes} \end{cases} \quad (6.9)$$

Wird weiterhin berücksichtigt, dass eine Planbereinigung entsprechend Abschnitt 6.2.5 im Anschluss an die Zusammenstellung des sich durch den Optimierungsvektor  $\mathbf{x}$  ergebenden Missionsplanes durchgeführt wird, so können die Kosten für die Nichtrealisierung eines Manövers ohne Beachtung der vorhergehenden und nachfolgenden Planelemente nicht bestimmt

<sup>1</sup> Aufteilen einer Geldmenge auf unterschiedliche Anlagen

<sup>2</sup> Entscheidung für die Realisierung bestimmter Projekte aus einer verfügbaren Menge

werden. Das liegt daran, dass durch ein mögliches Zusammenfassen mehrerer Füllplanelemente zu einem kombinierten Element die Kosten nicht mehr nur vom betrachteten Manöver abhängen.

Aus diesem Grund sind die für eine Lösung des linearen Rucksack-Problems verfügbaren Verfahren nicht auf diese Problemklasse übertragbar. Es existieren zwar auch Ansätze für nichtlineare Rucksack-Probleme wie zum Beispiel in [HJM93], doch sind diese wiederum auf spezielle Arten der Nichtlinearität zugeschnitten. Alle anderen Fälle werden häufig mit Hilfe von Suchverfahren oder angepassten Heuristiken gelöst.

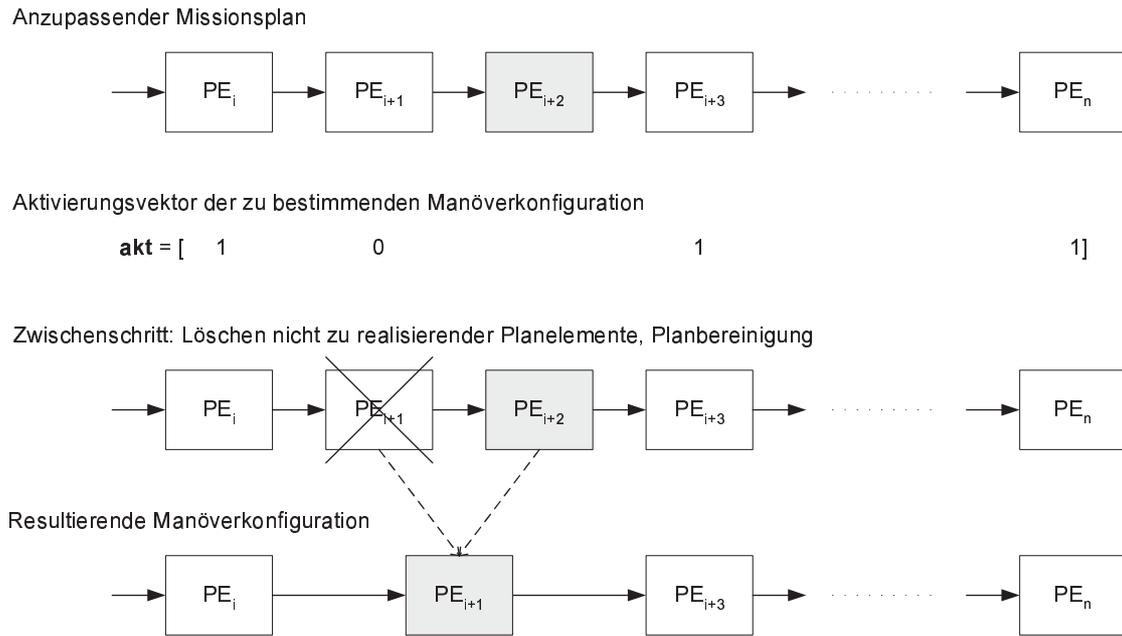
### 6.3.2.3 Lösung des Optimierungsproblems

Ausgangspunkt der folgenden Betrachtungen sind die Einsatzbedingungen des zu implementierenden Verfahrens. Es soll auf einem autonomen mobilen System eingesetzt werden und dabei in akzeptabler Zeit eine Lösung präsentieren. Die Anzahl der zu berücksichtigenden Planelemente an sich ist aufgrund unterschiedlicher Überlegungen bereits qualitativ beschränkt:

- Die Anzahl der Planelemente ist durch die Vorgehensweise bei der Missionsplanung begrenzt. Dies liegt an der manuellen Definition der Mission durch den Menschen, der bestrebt ist, die Menge der sequentiell zu planenden Handlungen überschaubar zu halten.
- Eine Optimierung des Missionsplanes wird normalerweise nicht gleich zu Beginn der Mission durchgeführt. Erst nach einiger Zeit mit einem gegenüber der Planung ungleich höheren Ressourcenverbrauch wird eine Anpassung des Planes erforderlich. Zusätzlich werden gewöhnlich bereits bei der Missionsplanung gewisse Reserven an Betriebsmitteln vorgesehen, sodass die Anzahl der Manöver bei Start einer Optimierung geringer als die ursprüngliche Gesamtzahl im Missionsplan sein wird.
- Planelemente, die nur eine Zustandsänderung  $\mathbf{x}_A \rightarrow \mathbf{x}_E$  bewirken, stellen laut Definition Füllplanelemente dar. Es kann davon ausgegangen werden, dass derartige Manöver in jedem Missionsplan aufgefunden werden. Diese Füllmanöver können bei der Optimierung unberücksichtigt bleiben, da ihre Nichtrealisierung zum Ersetzen durch ein (gleichartiges) Füllplanelement führt. Auf das Ergebnis der Optimierung hat die Beachtung dieser Manöver keinen Einfluss, sie erhöht lediglich die Komplexität des Gesamtproblems.

Daraus lässt sich jedoch keine quantitative Obergrenze für die Größe des Optimierungsvektors ableiten. Aus diesem Grund wurden zwei Strategien zur Problemlösung entworfen. Während die erste Verfahrensweise bei einer geringen Anzahl von Planelementen zur Optimallösung führt, gewährleistet die für höhere Manöverzahlen entworfene einfache Heuristik eine schnelle Ermittlung eines Ergebnisses. Sie garantiert zwar nicht das mögliche Optimum, liefert aber unter den gegebenen Bedingungen akzeptable Resultate. Ein Vergleich beider Verfahren wird für das AUV *DeepC* im Abschnitt 7.2.2.4 durchgeführt.

*Auffinden der Optimallösung.* Zunächst muss der Vektor der Optimierungsvariablen  $\mathbf{x}$  ermittelt werden. Er setzt sich aus der Aktivierung  $akt_k$  derjenigen Planelemente zusammen,



**Abb. 6.9:** Erzeugen einer Manöverkonfiguration, die grau hinterlegten Planelemente stellen Füllplanelemente dar

die keine Füllmanöver darstellen:

$$\mathbf{x} = \mathbf{akt} = [akt_i, akt_{i+1}, \dots, akt_m] \quad (6.10)$$

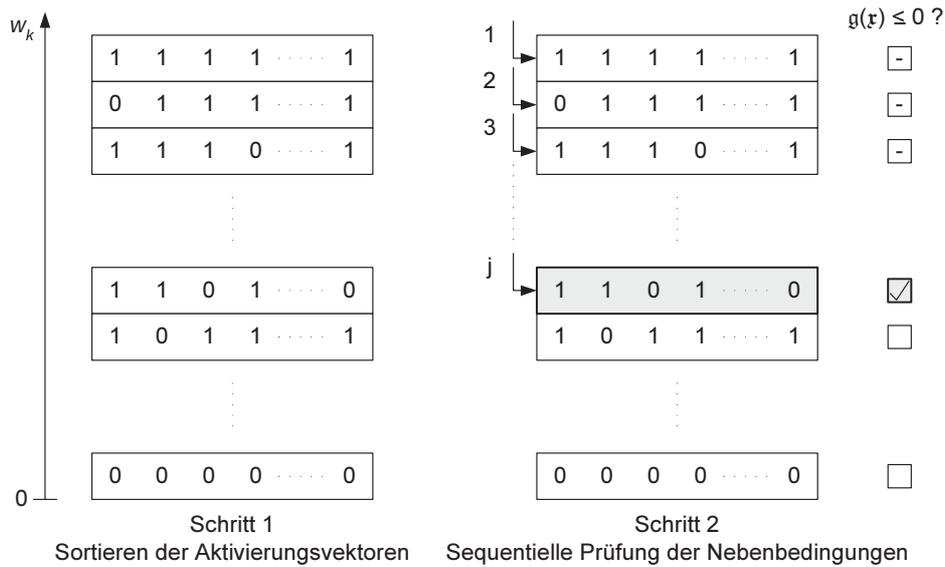
mit  $akt_k \in \{0, 1\}$   $k = [i, \dots, m]$

wobei  $i$  das aktuelle Planelement und  $m$  der Index des letzten zu berücksichtigenden Manövers ist.

Zur Optimallösung führt ein Enumerationsverfahren, das eine vollständige Aufzählung aller möglichen Kombinationen der Einzelwerte von  $\mathbf{akt}$  generiert. Jeder Aktivierungsvektor repräsentiert dabei eine mögliche Manöverkonfiguration, die auf ihren Wert und ihre Kosten untersucht werden muss. Für eine effiziente Arbeitsweise erfolgt dies in einem zweistufigen Prozess. Während die Ermittlung des Wertes einer Konfiguration einfach durch Bildung des Skalarproduktes  $W = \mathbf{w} \cdot \mathbf{akt}$  geschieht, muss für die Kostenberechnung die Manöverkonfiguration durch Löschen der nicht zu realisierenden Planelemente und anschließende Planbereinigung erzeugt werden (Abbildung 6.9).

Im ersten Schritt der Optimierung werden die Aktivierungsvektoren entsprechend dem zugehörigen Wert der Manöverkonfiguration absteigend sortiert (Abbildung 6.10). Beginnend mit der höchstwertigsten Konfiguration werden nun im zweiten Schritt der Reihe nach die Nebenbedingungen berechnet und geprüft. Die erste Konfiguration, die alle Bedingungen erfüllt, ist die gesuchte optimale Lösung. Sie besitzt den höchsten Wert unter den vorhandenen Beschränkungen.

*Heuristik zur schnellen Problemlösung.* [Ree95] definiert eine Heuristik als Technik, die mit vertretbarem Rechenaufwand gute approximative Lösungen sucht. Sie ist jedoch nicht imstande, die Optimalität und Zulässigkeit des gefundenen Ergebnisses zu garantieren. Für das



**Abb. 6.10:** Zweistufiges Konzept der Optimierung

Rucksack-Problem existiert eine Vielzahl von Heuristiken, die bekannteste und einfachste ist der so genannte Greedy-Algorithmus<sup>3</sup>. Er besitzt einen geringen, linear mit der Problemgröße anwachsenden Zeitbedarf.

Dieses Verfahren wird auch für die Planoptimierung im Fall einer großen Anzahl von Plan-elementen verwendet. Ausgangspunkt ist analog zum Rucksackproblem eine absteigend sortierte Liste relativer Wertigkeiten  $w_{rel}$ , die auf die Kosten bezogen werden:

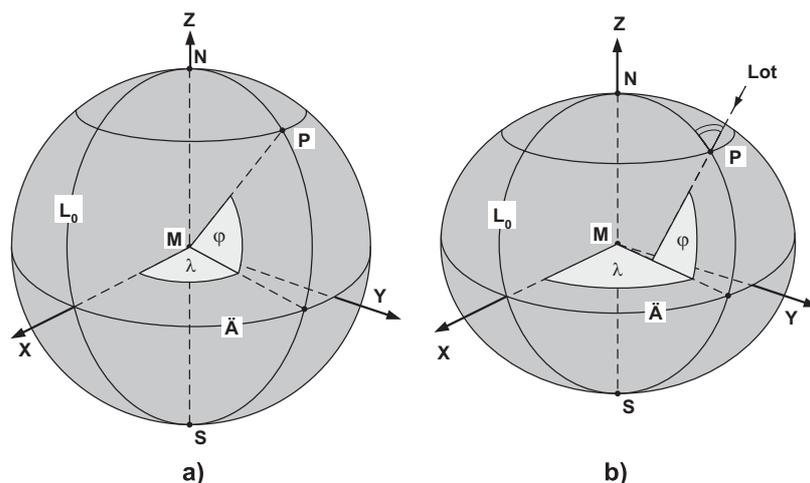
$$w_{rel}(k) = \frac{w_k}{\|\mathbf{c}_k\|} \quad k = [i, \dots, m]. \quad (6.11)$$

In jedem Schritt des Greedy-Algorithmus wird nun aus der Liste der verfügbaren Elemente das mit dem höchsten Wert ausgewählt, die Erfüllung der Nebenbedingungen der Optimierung laut Gleichung 6.6 geprüft und daraufhin entschieden, ob dieses Element in der Auswahlliste verbleibt oder nicht benutzt wird. Sobald kein weiteres Element mehr vorhanden ist, terminiert das Verfahren und die Lösung ist gefunden. Hier zeigt sich auch der Charakter dieser Heuristik, der im Namen zum Ausdruck kommt: In jedem Schritt wird das wertvollste verfügbare Element genommen, ohne die Gesamtsituation zu berücksichtigen. Der große Vorteil dieses Algorithmus liegt darin, *eine* Lösung in verhältnismäßig kurzer Zeit zu finden. Die Qualität des Ergebnisses und der Zeitbedarf des Verfahrens zeigen sich eindrucksvoll anhand der durchgeführten Untersuchungen für das AUV *DeepC*.

## 6.4 Geografische Planprüfung und -modifikation

Um eine sichere Navigation eines sich in seiner Umwelt bewegenden Roboters zu gewährleisten, sollte der Missionsplan vor dem Start der Mission und nach einer Umplanung auf Einhaltung der Beschränkungen durch seine Umgebung überprüft werden. Die zu berücksichtigenden Restriktionen können vielfältig sein. So muss ein in Gebäuden operierendes

<sup>3</sup>greedy = gierig, gefräßig



**Abb. 6.11:** Geografisches Koordinatensystem (a) auf der Kugel und (b) auf dem Rotationsellipsoid (nach [HGM02])

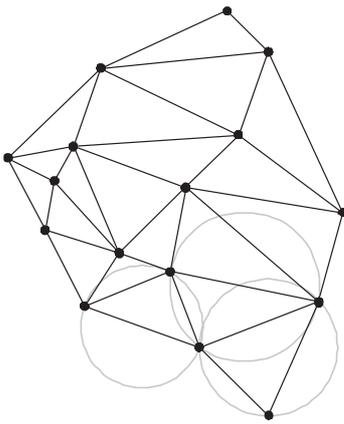
System eine Karte der Flure, Räume und Hindernisse besitzen, um sinnvolle Bewegungen durchführen zu können [Kni91]. Ein im Freien arbeitender Roboter benötigt hingegen ein Geländemodell und - falls er auf befestigte Wege angewiesen ist - die Informationen über verfügbare Fahrstrecken.

Die hier vorzustellenden Verfahren zur geografischen Planprüfung und -modifikation sind auf Systeme ausgerichtet, die sich in der natürlichen Umwelt „fliegend“ bewegen. Dabei kann es sich sowohl um Flug- als auch Tauchroboter handeln. Sie sind dadurch gekennzeichnet, dass auf Verletzung der Umgebungsbeschränkungen mit dreidimensionalen Manövern reagiert werden kann. Dies spielt insbesondere bei der Planmodifikation eine große Rolle. Voraussetzung für die Anwendung der folgenden Algorithmen ist das Vorhandensein der geografischen Position des autonomen mobilen Systems im Zustandsvektor, was im Allgemeinen durch den Begriff *mobil* impliziert wird.

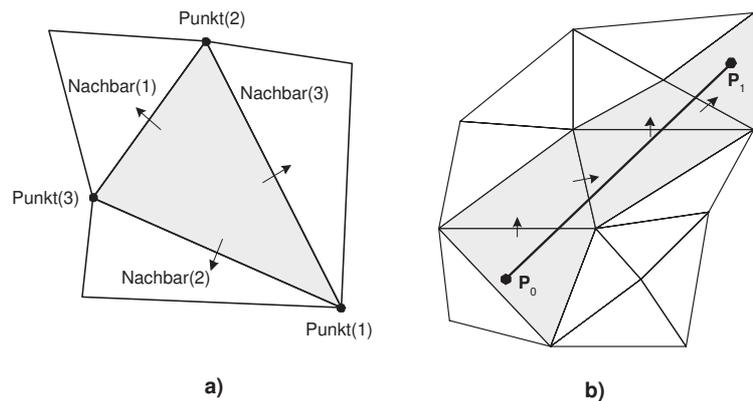
Damit die Position eines sich im Freien bewegenden Roboters festgelegt ist, muss ein eindeutiges Koordinatensystem verwendet werden. Für die Lagebeschreibung auf der Erdoberfläche bieten sich das geografische und das geozentrische Koordinatensystem an [HGM02]. Letzteres ist ein erdfestes dreidimensionales rechtwinkliges System  $(X, Y, Z)$ , das seinen Ursprung im Erdmittelpunkt hat (Abbildung 6.11 (a)). Gebräuchlicher sind jedoch die geografischen Flächenkoordinaten geografische Breite  $\lambda$  und geografische Länge  $\varphi$ , die auch in dieser Arbeit benutzt werden. Die Flächenkoordinaten sind immer bezogen auf eine bestimmte Beschreibung der Erdfigur, meist wird der Geoid<sup>4</sup> durch eine Kugel oder einen Rotationsellipsoid angenähert (Abbildung 6.11 (b)). Aufgrund der weiten Verbreitung des Global Positioning System (GPS) ist mittlerweile der als WGS84 (World Geodetic System) bezeichnete Ellipsoid der am häufigsten verwendete.

Die Beschreibung mittels Flächenkoordinaten gestattet lediglich die Festlegung einer Position auf dem damit assoziierten Ellipsoid, erst die Hinzunahme einer Höhe über (bzw. Tiefe

<sup>4</sup>Niveaulfläche (Äquipotentialfläche) des Erdschwerefeldes, die in mittlerer Höhe des Meeresspiegels verläuft



**Abb. 6.12:** Delaunay Triangulation mit Umkreisen



**Abb. 6.13:** Datenstruktur eines Dreiecks (a) und Navigation innerhalb des dreiecksvermaschten Netzes (b)

unter) diesem Referenzobjekt erlaubt eine eindeutige Bestimmung der Lage eines Roboters. Der verwendete Positionsvektor ergibt sich damit zu  $(\lambda, \varphi, h)$ .

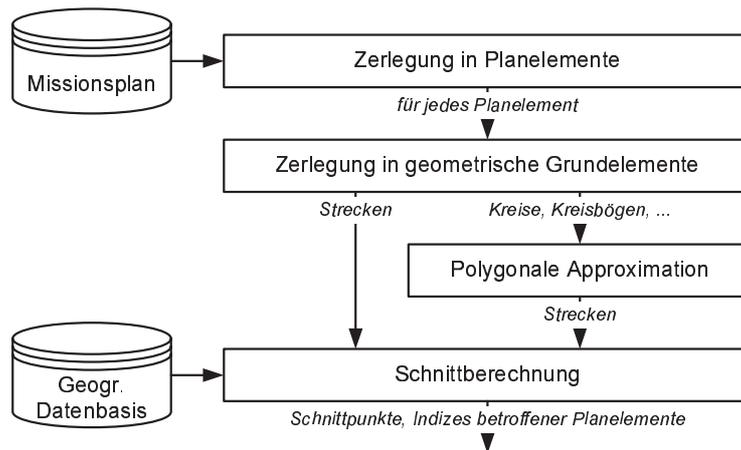
#### 6.4.1 Datenbasis

Grundlage der im Folgenden vorzustellenden Verfahren ist eine auf diesen Koordinaten basierende digitale Karte des Einsatzgebietes des Roboters. In dieser Karte ist die Erdoberfläche als Höhenmodell hinterlegt. Bedingt durch die Ausrichtung auf nicht erdgebundene Fahrzeuge (Flug- und Tauchroboter) sind keine künstlichen Fahrwege in das Geländemodell integriert worden.

Als Datenstruktur für eine effiziente Prüfung des Missionsplanes werden dreiecksvermaschte Netze (engl. TIN = Triangulated Irregular Network) verwendet. Diese bestehen aus einer Menge ungleichmäßig verteilter Punkte, die durch Verbindungen untereinander Dreiecke formen. Der Vorteil eines TIN gegenüber einer regelmäßigen Verteilung der Punktdaten besteht darin, dass Bereiche mit unterschiedlicher Informationsdichte gleichzeitig dargestellt werden können. Die Erdoberfläche ist besonders dadurch gekennzeichnet, dass sich häufig großflächige Ebenen geringer Höhenänderungen mit vielfältig strukturierten Gegenden abwechseln. Dreiecksvermaschte Netze sind geradezu prädestiniert, derartige Formen detailgetreu abzubilden.

Jeder Geländepunkt im TIN ist charakterisiert durch seine geografischen Koordinaten  $(\lambda, \varphi)$  und besitzt die Höhenangabe  $h$  als Attribut. Die Dreiecke werden mit Hilfe der zwei-dimensionalen Delaunay Triangulation über  $\lambda$  und  $\varphi$  erzeugt [PS90]. Bei diesem Verfahren wird das so genannte Umkreiskriterium als Grundlage für die Dreiecksbildung verwendet. Es besagt, dass im Umkreis eines Dreiecks kein weiterer Punkt liegen darf (Abbildung 6.12). Durch diese Bedingung wird stets der kleinste Innenwinkel der Dreiecke maximiert.

Die interne Datenstruktur des dreiecksvermaschten Netzes ist auf eine effiziente Arbeitsweise optimiert [Dev98]. Jedes Dreieck besitzt Verweise auf die drei Eckpunkte sowie die benachbarten Dreiecke (Abbildung 6.13 (a)). Damit ist eine schnelle Navigation innerhalb der Datenbasis möglich, was besonders wichtig für die Prüfung eines Missionsplanes ist. So kann in kürzester Zeit ein den Punkt  $\mathbf{P}_0 = [\lambda, \varphi]$  umschließendes Dreieck gefunden und dar-



**Abb. 6.14:** Ablauf der geografischen Prüfung eines Missionsplanes

aufhin sehr effektiv eine Liste der Dreiecke bestimmt werden, die die Strecke von  $P_0$  nach  $P_1$  berühren (Abbildung 6.13 (b)).

## 6.4.2 Prüfung des Missionsplanes

Ziel der Missionsplanprüfung ist es, eine Aussage über die Fahrbarkeit einer geplanten Mission zu treffen. Dazu ist eine Reihe von Operationen notwendig (Abbildung 6.14). Der Plan wird zunächst in seine Planelemente aufgespalten. Je nach Art eines Elementes können unterschiedliche geometrische Figuren als Beschreibungsform gefunden werden, die die Fahrkurve des mobilen Systems bei der Planrealisierung repräsentieren. Um die Prüfung der Fahrkurven gegen die Datenbasis effizient zu gestalten, werden nur Geraden (genauer gesagt Strecken) zur Schnittberechnung verwendet. Dafür existiert eine Vielzahl leistungsfähiger Verfahren. Alle weiteren geometrischen Objekte werden - zum Beispiel über polygonale Approximation - durch Geradenstücke angenähert. Der nachfolgende Abschnitt stellt eine Möglichkeit zur Approximation von Kreisen oder Kreisbögen vor, die nach Vorgabe eines maximalen Fehlers einen Polygonzug erstellt. Für eine effiziente Schnittberechnung zwischen Strecken und Dreiecken bietet sich die Transformation der Dreiecke in baryzentrische Koordinaten an, die in Abschnitt 6.4.2.2 näher erläutert wird.

### 6.4.2.1 Polygonale Approximation von Kreisen und Kreisbögen

Kreise und Kreisbögen stellen ein gängiges Beschreibungsmittel für Fahrkurven mobiler Roboter dar. Meist werden sie als Übergangselemente zwischen geradlinigen Abschnitten verwendet. Es existieren aber auch Systeme, die nur mit Kreisabschnitten arbeiten [Edl97]. Strecken werden dabei als Kreisbögen mit dem Radius  $r = \infty$  behandelt. Aufgrund der weiten Verbreitung dieser geometrischen Grundelemente soll hier ein Ansatz für ihre Annäherung unter Einsatz von Geradenstücken entwickelt werden.

Ausgangspunkt ist die Festlegung eines zulässigen maximalen Fehlers  $f_{max}$ . Er ist ein anschauliches Maß für die Genauigkeit der Approximation, da damit die maximale lotrechte Abweichung des Näherungspolygons vom Kreis oder Kreisabschnitt beschrieben wird (Abbildung 6.15).

Ein zweckmäßiges Verfahren stellt die Annäherung der Kreisform durch Vielecke dar. Bei Vorgabe von  $f_{max}$  und dem Kreisradius  $r$  lässt sich auf einfachem Wege die Zahl der Eckpunkte  $n$  des Vielecks sowie der Öffnungswinkel  $\alpha_n$  zwischen zwei benachbarten Stützstellen des Polygons berechnen:

$$\begin{aligned} r &= f_{max} + a = f_{max} + r \cdot \cos(\alpha_n/2) & (6.12) \\ f_{max} &= r(1 - \cos(\alpha_n/2)) \\ \alpha_n &= 2 \cdot \arccos\left(1 - \frac{f_{max}}{r}\right) \\ n &\geq \frac{2\pi}{\alpha_n} \quad n \text{ ganzzahlig.} & (6.13) \end{aligned}$$

Soll ein vollständiger Kreis angenähert werden, ist nur noch die Ermittlung der Stützstellen des Polygons  $\mathbf{P}_i$  ausgehend von einem Startpunkt  $\mathbf{P}_s$  unter Berücksichtigung von  $\alpha_n$  mit Hilfe trigonometrischer Beziehungen durchzuführen.

Für einen Kreisbogen des Öffnungswinkels  $\gamma$  entsprechend Abbildung 6.16 sind auf dieser Grundlage zwei Ansätze zur Approximation verwendbar:

- Benutzung des für einen Kreis ermittelten Winkels  $\alpha_n$ , dabei entsteht bei einem nicht ganzzahligen Verhältnis von  $\gamma/\alpha_n$  ein gegenüber den anderen Streckenabschnitten des Polygons kürzeres Teilstück.
- Durch Berechnung von  $\alpha_{\hat{n}}$  als ganzzahliger Teiler von  $\gamma$  ergibt sich eine äquidistante Aufteilung des Kreisbogens. Dazu ist die Teilungszahl  $\hat{n}$  folgendermaßen zu berechnen:

$$\hat{n} \geq \frac{\gamma}{\alpha_n} = \frac{\gamma}{2 \cdot \arccos\left(1 - \frac{f_{max}}{r}\right)} \quad \hat{n} \text{ ganzzahlig,} \quad (6.14)$$

$\alpha_{\hat{n}}$  kann aus dem Verhältnis  $\gamma/\hat{n}$  bestimmt werden.

Über trigonometrische Beziehungen ergeben sich nun in Analogie zum Kreis die Stützstellen des Polygonzuges.

#### 6.4.2.2 Schnittberechnungen

Die Kollisionsprüfung des Missionsplanes kann mit Hilfe der vorgestellten Verfahren zur Planzerlegung auf Schnittberechnungen zwischen Strecken und Dreiecken zurückgeführt werden. Da sich Dreiecke mit üblichen x-y-Koordinaten schwer beschreiben lassen, werden so genannte baryzentrische Koordinaten  $(\mathbf{u}, \mathbf{v}, \mathbf{w})$  verwendet. Mit ihrer Hilfe kann ein beliebiger Punkt in der Dreiecksebene als Linearkombination der Eckpunkte  $\mathbf{P}_0$ ,  $\mathbf{P}_1$  und  $\mathbf{P}_2$  dargestellt werden:

$$\mathbf{P}(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \mathbf{w}\mathbf{P}_0 + \mathbf{u}\mathbf{P}_1 + \mathbf{v}\mathbf{P}_2. \quad (6.15)$$

Liegt  $\mathbf{P}$  innerhalb des Dreiecks, erhalten  $(\mathbf{u}, \mathbf{v}, \mathbf{w})$  eine physikalische Bedeutung. Sie entsprechen dem Flächeninhalt der Dreiecke  $\mathbf{P}\mathbf{P}_1\mathbf{P}_2$ ,  $\mathbf{P}_0\mathbf{P}\mathbf{P}_2$  und  $\mathbf{P}_0\mathbf{P}_1\mathbf{P}$ . Bei einer Zuweisung  $\mathbf{u} = \mathbf{v} = \mathbf{w} = 1/3$  befindet sich aufgrund der Flächengleichheit der Punkt  $\mathbf{P}$  im Schwerpunkt des Dreiecks. Aus diesem Grund werden baryzentrische Koordinaten auch als

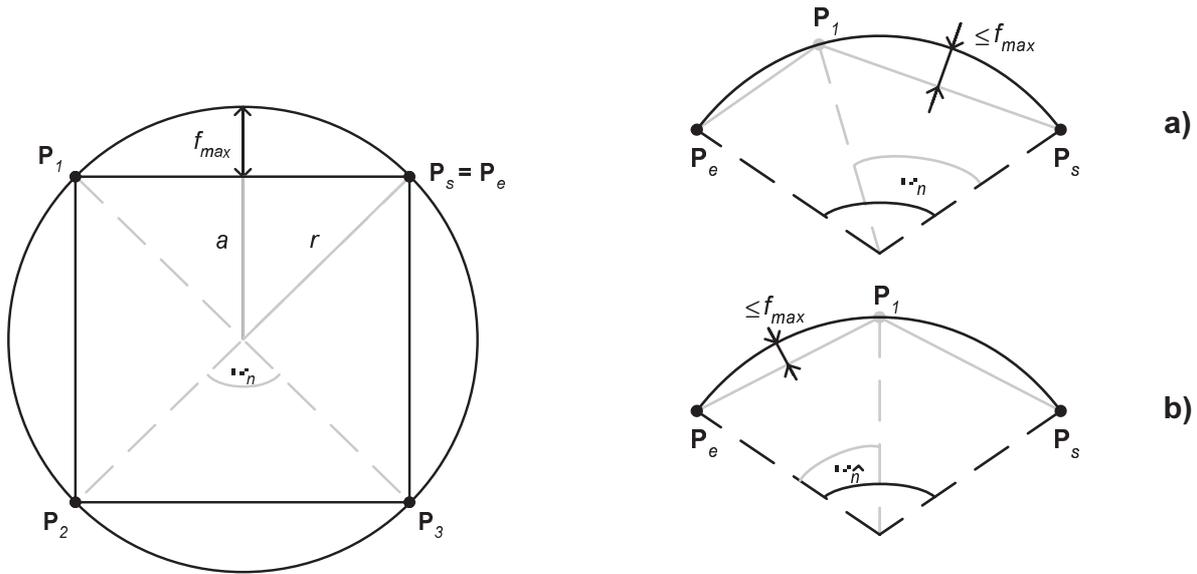


Abb. 6.15: Kreisapproximation mit Vieleck

Abb. 6.16: Approximation eines Kreisbogens; (a) mit  $\alpha_n$ , (b) mit  $\alpha_{\hat{n}}$

Schwerpunktkoordinaten bezeichnet. Sie sind ein weit verbreitetes Beschreibungsmittel in der Finite-Element-Methode [GRT93] und transformieren jedes Dreieck in ein Referenzdreieck, wenn  $\mathbf{w} = 1 - \mathbf{u} - \mathbf{v}$  gesetzt wird (Abbildung 6.17). Diese Festlegung wird bei den Schnittberechnungen ebenfalls genutzt.

Baryzentrische Koordinaten sind von dem den Punkten  $\mathbf{P}_i$  zugrunde liegenden Koordinatensystem unabhängig und besitzen deshalb günstige Eigenschaften für die Schnittberechnung. So kann anhand der Werte für  $(u, v)$  sofort entschieden werden, ob der Punkt  $\mathbf{P}$  zur Dreiecksfläche  $\mathbf{P}_0\mathbf{P}_1\mathbf{P}_2$  gehört oder nicht:

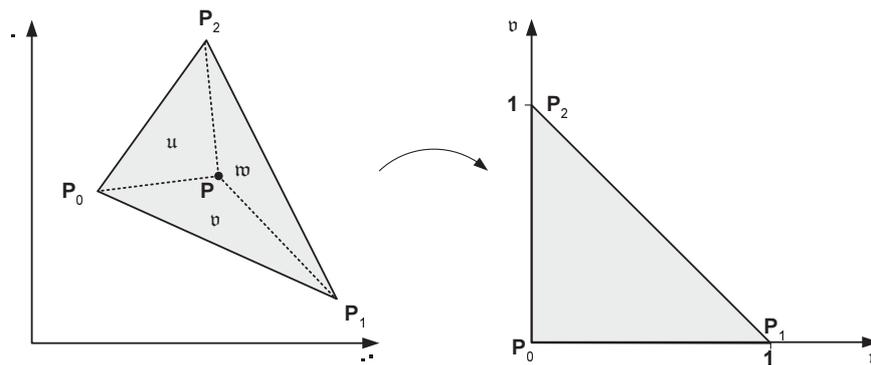
$$\mathbf{P}(u, v) \text{ gehört zum Dreieck} \leftrightarrow \begin{cases} u \geq 0 \\ v \geq 0 \\ u + v \leq 1. \end{cases} \quad (6.16)$$

In [MT97] wird ein Verfahren zur Schnittpunktberechnung zwischen Dreiecken und Geraden im dreidimensionalen Raum vorgestellt, dass diese Bedingungen verwendet. Darin wird eine Gerade durch die Punkte  $\mathbf{G}_0$  und  $\mathbf{G}_1$  mit Hilfe eines Parameters  $t$  folgendermaßen dargestellt:

$$\mathbf{G}(t) = \mathbf{G}_0 + t(\mathbf{G}_1 - \mathbf{G}_0) \quad t \in \mathfrak{R}. \quad (6.17)$$

Schnittpunkte existieren genau dann, wenn die Lösung  $(u', v', t')$  des Gleichungssystems  $\mathbf{P}(u, v, 1 - u - v) = \mathbf{G}(t)$  den Zusammenhang (6.16) erfüllt. Da für die Planprüfung keine Geraden, sondern Strecken mit Start- und Endpunkt verwendet werden, muss dieser Algorithmus um die Bestimmung der Zugehörigkeit zu der Strecke  $\overline{\mathbf{G}_0\mathbf{G}_1}$  ergänzt werden. Mit Hilfe des Parameters  $t$  ist die Entscheidung über die Zugehörigkeit zur Strecke einfach zu treffen:

$$\mathbf{G}(t) \text{ gehört zur Strecke} \leftrightarrow 0 \leq t \leq 1. \quad (6.18)$$



**Abb. 6.17:** Baryzentrische Koordinaten im Dreieck

Mit diesem Ansatz steht eine sehr effiziente Methode zur Verfügung, um eine Planprüfung in kürzester Zeit durchführen zu können. Dabei wird für jede Strecke, die aus den Manövern des Missionsplanes gebildet werden kann, eine Liste der überdeckten Dreiecke entsprechend Abbildung 6.13 ermittelt. Schnittberechnungen zwischen den betroffenen Dreiecken und dem zugehörigen Streckenabschnitt liefern dann die Aussage, ob eine Verletzung der Beschränkungen durch das digitale Höhenmodell vorliegt oder nicht. Dieses Ergebnis wird anschließend in der geografischen Planmodifikation für die Auswahl der neu zu berechnenden Fahrstrecken genutzt.

### 6.4.3 Modifikation des Missionsplanes

Wird bei der Prüfung des Missionsplanes festgestellt, dass Verletzungen der Beschränkungen durch das Terrain auftreten, generiert die geografische Planmodifikation einen Vorschlag für die Anpassung des oder der betroffenen Planelemente. Im Ergebnis entsteht dabei eine neue Route zwischen geografischer Start- und Endposition dieser Manöver.

Die im Abschnitt 6.4.1 präsentierte Struktur zur Hinterlegung eines digitalen Höhenmodells ist für die Berechnung eines neuen Weges nicht gut geeignet. Eine wesentliche Ursache dafür liegt in der fehlenden Information über die Fahrbarkeit der Strecken zwischen einzelnen Knotenpunkten des Netzes. Außer für Wege zwischen den drei Eckpunkten eines Dreiecks, die sich immer als direkte Verbindung ergeben, kann keine Route ohne größeren Rechenaufwand bestimmt werden. Darüber hinaus steigt der Aufwand mit zunehmender Zahl der Knoten des Netzes. Aus diesem Grund ist eine Vorverarbeitung der Daten und damit eine Erweiterung der Datenbasis unumgänglich.

Als geeignete Verfahren für die Suche des nach Möglichkeit minimalen Weges zwischen zwei Punkten im Gelände bieten sich graphenbasierte Methoden an, die im schlechtesten Fall polynomial von der Größe des Graphen abhängen [BG93]. Sie garantieren eine effiziente Modifikation des Missionsplanes und werden im Abschnitt 6.4.3.2 vorgestellt.

#### 6.4.3.1 Erweiterung der Datenbasis

Aufgabe der Vorverarbeitung ist die Generierung der für die dreidimensionale Wegplanung benötigten Informationen und deren Integration in die Datenbasis. Für graphenbasierte Verfahren wird ein Graph  $G = (V, A)$  benötigt, der als Knoten  $V$  die Stützstellen des TIN und

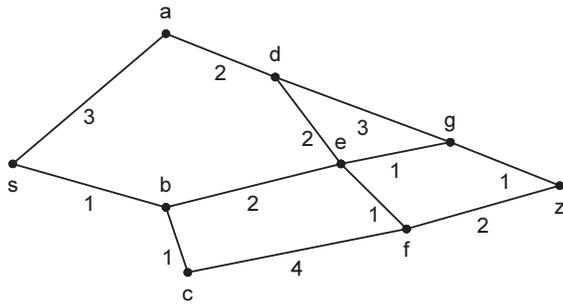
als Kanten  $A$  die Kosten für den Weg zwischen den Knoten beinhaltet. Der Graph kann dabei sowohl gerichtet als auch ungerichtet sein. Ein gerichteter Graph verknüpft die Kosten für den zurückzulegenden Wegabschnitt mit der Bewegungsrichtung, sodass der Aufwand für die Fahrt von Knoten  $a$  nach  $b$  anders als der von  $b$  nach  $a$  sein kann. Ein ungerichteter Graph besitzt demgegenüber Kantenbewertungen, die unabhängig von der Richtung der Bewegung sind. Im Hinblick auf den Einsatz der Verfahren zur geografischen Planmodifikation für das autonome Unterwasserfahrzeug *DeepC* wird hier ein ungerichteter Graph verwendet (Abbildung 6.18). Die Ursache dafür liegt in dem ähnlichen Aufwand, der für Tiefenänderungen in positiver und negativer Richtung notwendig ist. Durch das Vorhandensein eines statisch wirkenden Trimmsystems ist im Gegensatz zu fliegenden Systemen für beide Fälle ein höherer Energiebedarf als für die Bewegung in der Horizontalen vorhanden.

Die Kosten  $c_{i,j}$  für die Fahrstrecke vom Knoten  $i$  zum Knoten  $j$  ergeben sich über eine Bewertungsfunktion, die im Fall eines mobilen Roboters immer positive Werte liefert (entsprechend dem Ressourcenverbrauch beim Zurücklegen der Strecke). Nicht fahrbare Strecken besitzen ungültige Kostenangaben (z.B.  $\infty$ ). Dieser Graph wird in Form einer Distanzmatrix in die Datenbasis aufgenommen (Abbildung 6.19). Die Kosten unterhalb der Diagonalen entsprechen bei einem ungerichteten Graphen wie dem hier dargestellten denen im oberen Dreieck des Distanzmatrix:  $c_{i,j} = c_{j,i}$ .

Für nicht erdgebundene Fahrzeuge, die sich frei im dreidimensionalen Raum bewegen können, ist eine einmalige, vollständige Bestimmung des Graphen nicht möglich. Die Ursache dafür liegt in dem Sachverhalt, dass die Fahrbarkeit zwischen zwei dreidimensionalen Punkten im Gelände von deren Abstand zum Boden abhängig ist. Ein in 10 Kilometer Höhe fliegender Roboter benötigt zum Beispiel außer bei Start und Landung keine Informationen über die Struktur des Terrains, da bekanntlich diese Höhe an keinem Ort der Erdoberfläche erreicht werden kann. Soll er jedoch eine Strecke fliegen, deren Start- und Endposition im Abstand von 250 Meter zum Boden liegen, sind vielfältige Beschränkungen innerhalb der verfügbaren Routen (und damit auch im Graphen) zu erwarten. Daraus ergibt sich prinzipiell die Notwendigkeit, den Graphen anhand der Höhenangaben im Missionsplan immer wieder neu zu erzeugen.

Die Ermittlung des Graphen erfordert jedoch durch Berechnung der Fahrbarkeit (oder auch Sichtbarkeit) zwischen den einzelnen Knoten des TIN einen großen Rechenaufwand. Deshalb kann häufig nicht während einer Mission eine komplette Neugenerierung des Graphen durchgeführt werden. Als Ausweg wird eine Strategie vorgeschlagen, die den Gesamtprozess in eine Offline- und eine Online-Phase aufteilt:

- *Offline-Phase:* Es erfolgt die Fahrbarkeitsprüfung zwischen allen Paaren von Knoten des Graphen mit minimal zulässigem Abstand zum Gelände. Dies ergibt eine geringe Anzahl von fahrbaren Strecken, die aber in jeder beliebigen Höhe genutzt werden dürfen. Die Berechnungen sind sehr zeitaufwendig, da für jedes Knotenpaar Schnittberechnungen mit dem digitalen Höhenmodell notwendig sind. Die Kosten werden über den zurückzulegenden dreidimensionalen Weg ermittelt.
- *Online-Phase:* Die Höheninformationen von Anfangs- und Endpunkt einer zu suchenden Route werden in den Graphen integriert. Dabei werden die in der offline Phase gefundenen Routen daraufhin überprüft, ob der Roboter bei ihrer Benutzung Höhenänderungen durchzuführen hat. Auftretende Änderungen werden mit zusätzlichen



**Abb. 6.18:** Graph mit positiven Kantenbewertungen

	s	a	b	c	d	e	f	g	z
s	0	3	1	.	.	.	.	.	.
a		0	.	.	2	.	.	.	.
b			0	1	.	2	.	.	.
c				0	.	.	4	.	.
d					0	2	.	3	.
e						0	1	1	.
f							0	.	2
g								0	1
z									0

**Abb. 6.19:** Distanzmatrix zum Graph aus Abbildung 6.18

Kosten belegt. Der Rechenaufwand für diese Schritte ist wesentlich geringer als der der offline Phase, da nur eine geringe Anzahl von Knotenpaaren untersucht werden muss. Bei den in Experimenten untersuchten Daten ergab sich, dass ca. 10 bis 20 % der Gesamtzahl der Knotenpaare die offline durchgeführte Fahrbarkeitsprüfung bestanden haben und damit in den Graphen aufgenommen wurden.

Zusätzlich werden Start- und Endpunkt der Route in den Graphen integriert, wenn sie nicht bereits als Knoten vorhanden oder in geringem, parametrierbaren Abstand zu einem vorhandenen Knoten gelegen sind.

Durch die offline Vorbereitung des Graphen ist die zur Verfügung stehende Routenzahl gegenüber der theoretisch möglichen eingeschränkt, da nur die in allen Höhen befahrbaren Verbindungen zwischen den Knoten ermittelt werden. So wird nicht unbedingt der optimale Weg durch das Gelände gefunden, sondern die mit der vorhandenen (vorverarbeiteten) Datenbasis kürzeste Route. Demgegenüber steht als großer Vorteil, dass diese Verfahrensweise auf den oft begrenzten Rechenkapazitäten eines mobilen Roboters eingesetzt werden kann.

#### 6.4.3.2 Graphenbasierte Routensuche

Mit Hilfe von Suchmethoden kann die kürzeste Route in einem Graphen  $G$  sehr effektiv ermittelt werden. Als Standardverfahren für ein derartiges Problem hat sich im Fall eines vollständig bekannten Graphen der nach seinem Erfinder E. W. Dijkstra benannte Dijkstra-Algorithmus etabliert [Dij59]. Er findet in polynomialer Zeit iterativ eine Route, falls diese in dem Graphen existiert.

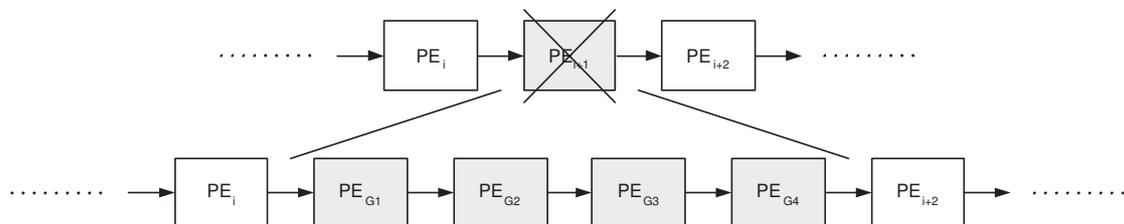
Ausgehend von einem Startknoten  $s$  werden beim Dijkstra-Algorithmus alle benachbarten Knoten (Knoten, die von  $s$  aus erreichbar sind, also eine Kantenbewertung ungleich  $\infty$  besitzen) besucht und die jeweiligen Kosten bestimmt (Abbildung 6.20). Referenzknoten  $k_{ref}$  für die zweite Iteration wird immer derjenige, der die niedrigsten Kosten zum Startknoten besitzt (Knoten  $b$ ). Es erfolgt wiederum eine Kostenermittlung zu den Nachbarknoten von  $k_{ref}$ , die zu den bereits aufgelaufenen, auf  $s$  bezogenen Kosten hinzugefügt werden. Iterativ werden nun nacheinander die Knoten mit den jeweils niedrigsten Gesamtkosten als Referenzknoten festgelegt, wobei bereits besuchte als erledigt markiert und nicht nochmals ausgewählt werden.

Im Ergebnis entsteht ein Kostenvektor, der den Aufwand für alle Strecken ausgehend von  $s$  repräsentiert. Gleichzeitig ergibt sich eine Liste derjenigen Knoten, die den kürzesten

	Knoten									erledigt
	s	a	b	c	d	e	f	g	z	
Start	0	.	.	.	.	.	.	.	.	-
Iteration 1	0	3	1	.	.	.	.	.	.	s
Iteration 2	0	3	1	2	.	3	.	.	.	s,b
Iteration 3	0	3	1	2	.	3	6	.	.	s,b,c
Iteration 4	0	3	1	2	5	3	6	.	.	s,b,c,a
Iteration 5	0	3	1	2	5	3	4	4	.	s,b,c,a,e
Iteration 6	0	3	1	2	5	3	4	4	6	s,b,c,a,e,f
Iteration 7	0	3	1	2	5	3	4	4	5	s,b,c,a,e,f,g
Iteration 8	0	3	1	2	5	3	4	4	5	s,b,c,a,e,f,g,d

**Abb. 6.20:** Ablauf des Dijkstra-Algorithmus für den Graph aus Abbildung 6.18

Weg vom Start zum Ziel bilden. Mit Hilfe ihrer Positionsangaben wird beispielsweise das in Abbildung 6.21 dargestellte, nicht fahrbare Manöver  $PE_{i+1}$  des Missionsplanes durch eine Sequenz von Planelementen ( $PE_{G1}, \dots, PE_{Gn}$ ) ersetzt.



**Abb. 6.21:** Geografische Planmodifikation

## 6.5 Zusammenfassung

Unter den Strategien zur Missionsumplanung sind alle Verfahren zur Modifikation des Missionsplanes eines autonomen mobilen Systems zusammengefasst. Die Methoden sind in allgemeine und spezielle Verfahren unterteilt. Mit Hilfe der allgemeingültigen atomaren Umplanungsbefehle kann nahezu jede beliebige Veränderung des Planes durchgeführt werden. Sie bilden damit die Basis für die speziellen Verfahren.

Die Definition komplexer Befehle vereinfacht die Schnittstelle von der Missionsüberwachung zur Missionsumplanung erheblich. Die damit verbundenen speziellen Verfahren der Planmodifikation erlauben umfangreiche Korrekturen des Missionsplanes. So kann mit Hilfe des Missionsabbruchs eine Folge von für den sicheren Abschluss des Einsatzes notwendigen Planelementen erzeugt werden. Die Optimierung des Missionsplanes ermittelt die im Fall eines Ressourcenmangels zu realisierenden Manöver. Aufgrund der Komplexität des Problems wurden zwei unterschiedliche Methoden entwickelt. Das Optimierungsverfahren findet die globale Lösung in einem zweistufigen Prozess, der mit zunehmender Manöveranzahl zu einem für die Online-Anwendung unverträglich hohen Rechenaufwand führt. Die entworfene Heuristik ist demgegenüber für große Manöverzahlen sehr gut geeignet, da bei ihr der zu durchsuchende Lösungsraum linear von der Anzahl der Optimierungsvariablen abhängt.

Mit Hilfe von Verfahren zur geografischen Planprüfung und -modifikation kann der veränderte Missionsplan auf die Einhaltung von Beschränkungen durch das Einsatzgebiet des Roboters überprüft und gegebenenfalls daran angepasst werden. Neu entworfene, auch für große Einsatzgebiete effiziente Methoden gestatten eine Online-Anwendung auf autonomen mobilen Systemen.

## 7 Anwendung und Ergebnisse

In diesem Kapitel wird die Implementierung der neuen Strategien auf zwei unterschiedlichen Plattformen vorgestellt. Für jedes dieser mobilen Systeme soll eine Beschreibung der hard- und softwaretechnischen Rahmenbedingungen, der Einbindung des intelligenten Missionsmanagements sowie der Ergebnisse aus den Tests vorgenommen werden.

Die vorgestellten intelligenten Strategien wurden vorrangig für das autonome Unterwasserfahrzeug *DeepC* entwickelt und in Simulationen erprobt. Der Hauptteil dieses Kapitels beschäftigt sich deshalb mit diesem mobilen System. Zur Demonstration des universellen Charakters erfolgte darüber hinaus der Aufbau eines komplexen Missionsführungssystems für den mobilen Kleinroboter *MauSI*. Nachfolgend sollen für beide Roboter die konkreten Rahmenbedingungen, die Struktur der Missionspläne sowie die Architektur der Software vorgestellt werden. Verschiedene Umplanungsszenarien beleuchten den Einsatz des Missionsumplanungssystems unter unterschiedlichen Aspekten.

### 7.1 Der mobile Kleinroboter MauSI

*MauSI* ist die Abkürzung für „**M**odell eines **a**utonomen **S**ystems mit variabler **I**ntelligenz“ und bezeichnet eine Klasse von mobilen Kleinrobotern, die entweder eine übergebene Bewegungsaufgabe selbständig realisieren oder per Funkverbindung von einer Leitstelle aus vollständig ferngesteuert werden [AA01].

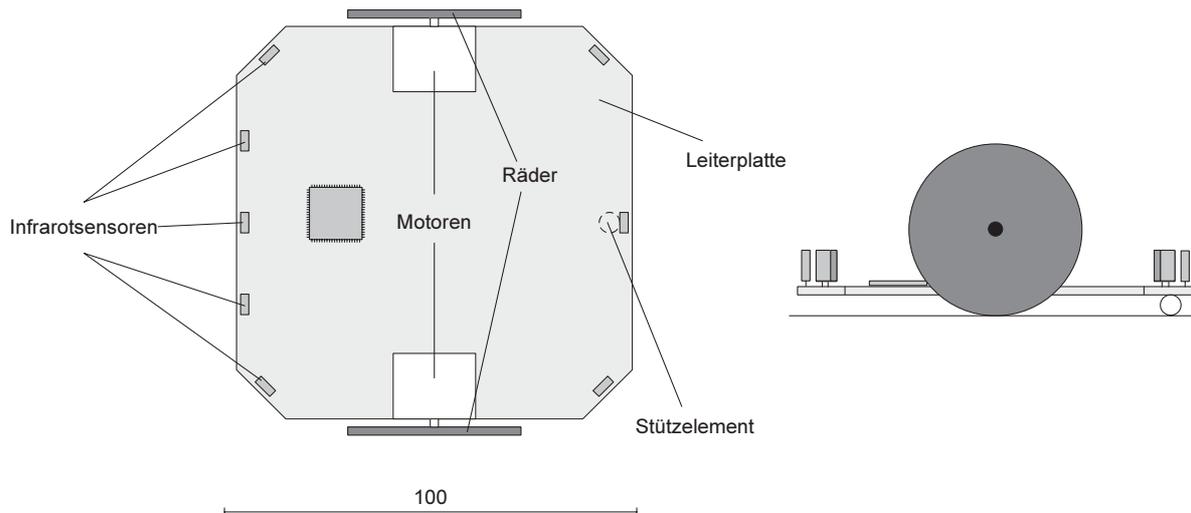
Aufgrund der beschränkten Rechenkapazitäten der Roboter wird das intelligente Missionsmanagement gemeinsam mit einem Großteil der Softwarearchitektur zur Steuerung des Systems auf der zentralen Leitstelle verwirklicht. Dadurch besteht die Möglichkeit, auch anspruchsvolle Verfahren zur Fahrzeugführung testen zu können. Die Autonomie des Systems ist damit jedoch nur in Kombination mit dem Leitrechner gegeben.

#### 7.1.1 Aufbau und Steuerung

Die Roboter besitzen einen konzentrischen Aufbau (Abbildung 7.1) und werden durch zwei getrennt steuerbare Antriebe bewegt. Ein sich im hinteren Bereich befindendes Stützelement stabilisiert die Lage der Roboter. Mit Hilfe von acht Infrarotsensoren können Informationen über die Umwelt aufgenommen werden. Bedingt durch die Art der Sensorik sind die erfassbaren Umweltdaten auf Hindernisse im Nahbereich begrenzt.

Roboter des Typs *MauSI* gehören aufgrund ihres Aufbaus zu den nicht-holonomen Fahrzeugen, da ihre Bewegungsfreiheit zumindest in Achsrichtung der angetriebenen Räder eingeschränkt ist. Durch die zentrale Anordnung der Antriebseinheiten spielen diese Beschränkungen jedoch kaum eine Rolle. Die Art der Konstruktion erlaubt eine Drehung auf der Stelle, was Manöver auf engstem Raum ermöglicht.

Das integrierte Funkmodem der Roboter gestattet die Verbindungsaufnahme zur Sollwertübermittlung und Zustandsabfrage von einer Leitstation aus. Innerhalb des Funkprotokolls



**Abb. 7.1:** Schematischer Aufbau des Roboters MauSI

wird der zu adressierende Roboter durch eine Identifikationsnummer ausgewählt. Dadurch können mehrere derartiger Systeme gleichzeitig betrieben werden, was eine wesentliche Grundlage für spätere Experimente mit Schwärmen mobiler Systeme bildet.

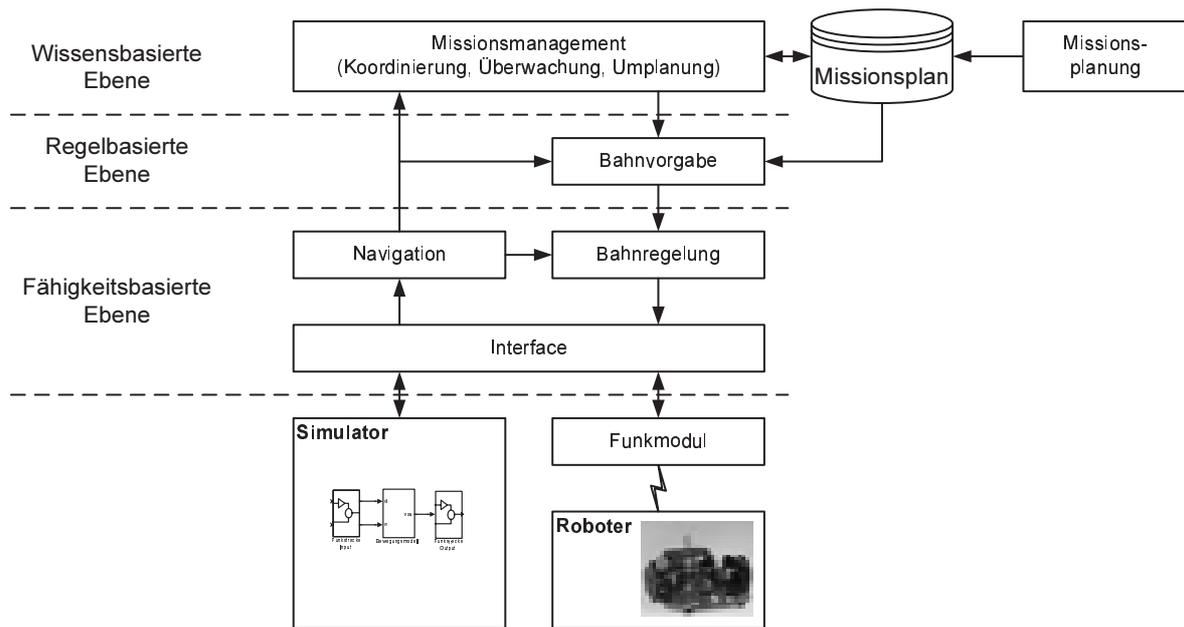
Über die Funkstrecke kann im Fernsteuermodus die Drehzahl beider Räder getrennt gesteuert werden. Als Rückmeldung wird die aktuelle Ist-Drehzahl der Räder gesendet, sodass mit Hilfe eines Beobachters auf dem angeschlossenen Rechner eine Positionsschätzung möglich ist. Für einen späteren Test von Hindernisvermeidungsroutinen werden auch die Messwerte der Infrarotsensoren übertragen.

### 7.1.2 Softwarearchitektur

Voraussetzung für eine reibungslose Einbindung des intelligenten Missionsmanagements ist der Aufbau einer Softwarearchitektur entsprechend Abschnitt 2.3.1. Die Architektur ist dabei in die drei bekannten Ebenen aufgeteilt (Abbildung 7.2) [Pfü04]. Die gesamte Struktur wird unter der blockorientierten Simulationsumgebung SIMULINK des Programmpaketes MATLAB entworfen und getestet. Dieser Ansatz erlaubt eine schnelle Entwicklung des Prototyps und einen einfachen und zweckmäßigen Test seiner Funktionalität.

Die fähigkeitsbasierte Ebene beinhaltet die Schnittstelle zu dem Funkmodul, um Kontakt mit einem Roboter aufnehmen zu können. Weiterhin ist eine Verbindung zu einem Simulator vorhanden, damit Tests auch ohne reale Roboter möglich sind. Für die Fahrzeugsteuerung sind auf dieser Ebene die Navigation und die Bahnregelung wesentliche Module. Die Navigation ermittelt aus den über die Funkstrecke übertragenen oder vom Simulator generierten Drehzahlen der Räder eine Positions- und Orientierungsschätzung. Diese Koordinaten werden von der Bahnregelung verwendet, um die Abweichung von der gewünschten Sollbahn und daraus die neuen Soll-Drehzahlen der Räder zu berechnen.

Auf der regelbasierten Ebene wird durch das Modul Bahnvorgabe die Solltrajektorie für die Bahnregelung aus dem aktuell zu realisierenden Manöver des Missionsplanes errechnet und an die Regelung übergeben. Dieses Modul erkennt auch das Erreichen des Endpunktes einer Bahn und schaltet dann auf das nachfolgende Teilstück um. Die Bahnvorgabe verwendet



**Abb. 7.2:** Softwarearchitektur für die Steuerung von MauSI

dazu eine angepasste Darstellung des Missionsplanes, die im Abschnitt 7.1.2.4 vorgestellt wird.

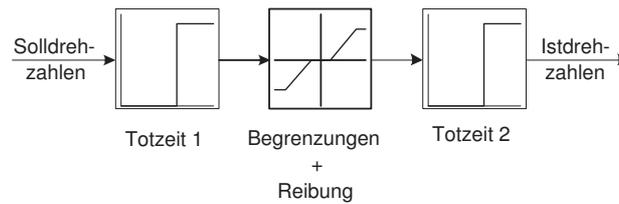
Das intelligente Missionsmanagement befindet sich auf der wissensbasierten Ebene und überwacht die Missionsausführung. Erkannte Abweichungen vom Missionsplan führen zur Anpassung des Planes. Der modifizierte Missionsplan wird anschließend von der Bahnvorgabe in Form von Sollbahnen aufbereitet und mit Hilfe der Bahnregelung umgesetzt.

Die nachfolgenden Abschnitte beschreiben den Aufbau und die Schnittstellen der Kernelemente der Softwarearchitektur, die als Basis für das intelligente Missionsmanagement neu geschaffen wurde. Diese Informationen sind für das Verständnis des Missionsmanagements von grundlegender Bedeutung. Weitergehende Untersuchungen zu diesen neu entworfenen Konzepten und Strategien wurden im Rahmen von Diplom- und Studienarbeiten durchgeführt [Koc03a], [Brü02], [Koc03b].

#### 7.1.2.1 Simulator

Bedingt durch den einfachen Aufbau der Roboter ist der Aufwand zur Modellierung des Fahrzeuges relativ gering. Das Simulationsmodell muss lediglich die per Funk übergebenen Drehzallsollwerte in die zurückzumeldenden Istwerte überführen. In der ersten Ausbaustufe des Modells ist die Integration einer Strategie zur Hindernisvermeidung nicht geplant, sodass auf die Generierung künstlicher Sensordaten für die im realen Fahrzeug vorhandenen Infrarotsensoren verzichtet wurde.

Untersuchungen in [Koc03a] zeigten, dass das dynamische Verhalten des Systems im Wesentlichen durch die Totzeiten der Funkübertragung bestimmt wird. Eine Berücksichtigung der Übertragungskennlinien der Antriebseinheiten wurde aus diesem Grund nicht vorgenommen. Bedingt durch die geringen Beschleunigungs- und Verzögerungswerte, die mittels der vorhandenen Motoren erreicht werden, kann auch kein Schlupf auftreten. Somit ist auch



**Abb. 7.3:** Vereinfachtes Simulationsmodell des Kleinroboters MauSI

dieser Einfluss vernachlässigbar. Das derart vereinfachte Simulationsmodell ergibt sich damit zu der in Abbildung 7.3 gezeigten Struktur. Neben den Totzeiten der Funkübertragung spielen nur die Begrenzungen der Motordrehzahlen sowie Reibungsverluste eine Rolle.

### 7.1.2.2 Navigation

Mit den vom Roboter zurückgelieferten Istdrehzahlen wird eine Schätzung der Position des Systems durch die Navigation durchgeführt. Der Aufbau als „differential drive“ Fahrzeug erlaubt eine einfache Berechnung des zurückgelegten Weges und der Änderung der Fahrzeugorientierung aus den Geschwindigkeiten des linken Rades  $v_l$  sowie des rechten Rades  $v_r$  (Abbildung 7.4). Aus diesen beiden Werten werden zunächst die Lineargeschwindigkeit  $v$  und die Winkelgeschwindigkeit  $\omega$  ermittelt:

$$\begin{aligned} v(t) &= \frac{v_r + v_l}{2} \\ \omega(t) &= \frac{v_r - v_l}{b}, \end{aligned} \quad (7.1)$$

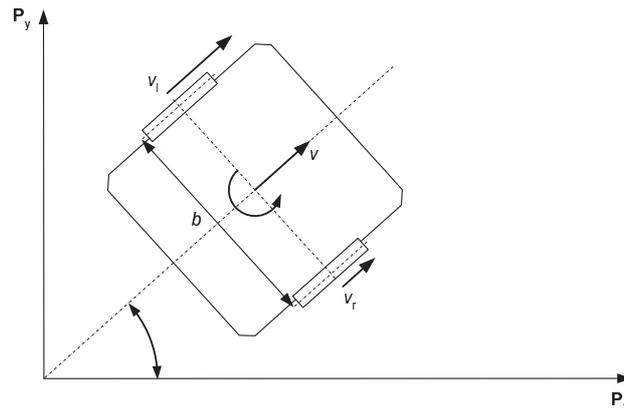
wobei  $b$  die Spurbreite des Fahrzeuges ist. Aus der Winkelgeschwindigkeit ergibt sich über Integration der Lagewinkel  $\alpha$ :

$$\alpha(t) = \int_0^t \omega(\tau) d\tau + \alpha(0). \quad (7.2)$$

Dabei stellt  $\alpha(0)$  den Anfangswert des Lagewinkels zum Zeitpunkt  $t = 0$  dar. Die Koordinaten der aktuellen Fahrzeugposition  $\mathbf{P}(t)$  können damit folgendermaßen berechnet werden:

$$\begin{aligned} \mathbf{P}_x(t) &= \int_0^t \cos[\alpha(\tau)] v(\tau) d\tau + \mathbf{P}_x(0) \\ \mathbf{P}_y(t) &= \int_0^t \sin[\alpha(\tau)] v(\tau) d\tau + \mathbf{P}_y(0), \end{aligned} \quad (7.3)$$

der Vektor  $\mathbf{P}(0) = [\mathbf{P}_x(0), \mathbf{P}_y(0)]$  repräsentiert hier die Position des Roboters zu Beginn der Mission. Dieser einfache Ansatz liefert bei fehlerfreier Messung der Drehzahlen, der exakten Ausrichtung und Positionierung des Roboters vor dem Beginn und einer schlupf- und kollisionsfreien Arbeit des Systems gute Ergebnisse.



**Abb. 7.4:** Modellierung des Bewegungsverhaltens des Kleinroboters MauSI

Für einen realen Einsatz des mobilen Roboters ist diese Betrachtungsweise jedoch zu einfach. Aufgrund der großen Totzeiten der Funkstrecke ist die Positionsschätzung nur mit Hilfe der Drehzahlen mit einem sehr großen Fehler verbunden. Ein möglicher Ausweg ist die Berechnung von Position und Orientierung direkt im Roboter, da damit die Verzögerungszeiten minimiert werden. Zusätzlich bietet es sich an, einen Abgleich der berechneten Position anhand unterschiedlicher, vom Roboter zu detektierender Merkmale der Umwelt oder durch ein externes, den Roboter beobachtendes System durchzuführen [EPSW01]. Diese auch als Positionsstützung bezeichnete Fusion von Sensordaten unterschiedlicher Herkunft ist ein wesentlicher Schwerpunkt weiterer Arbeiten an dem Robotersystem.

#### 7.1.2.3 Bahnregelung

Die Bahnregelung erhält einen Bahnabschnitt von der Bahnvorgabe, auf dem sich der Roboter bewegen soll. Als Bahnabschnitte werden dabei kontinuierlich fahrbare Teilstrecken übergeben, da damit die durch den nicht-holonomen Aufbau von MauSI bedingten, unnötigen Stellgliedbelastungen der Antriebe vermieden werden können.

Jeder Bahnabschnitt ist gekennzeichnet durch zwei so genannte Vektorpunkte, die Position, Ausrichtung und Geschwindigkeit des Roboters an Start- und Endpunkt des Teilstücks repräsentieren (Abbildung 7.5). Die Vektorpunkte werden auf den geradlinigen Teilstrecken zwischen zwei Bahnpunkten derartig fixiert, dass sie sich in einem abhängig vom Wenderradius des Roboters festzulegenden Abstand von den einzelnen Bahnpunkten befinden. Die durch den Bahnregler einzuhaltende Sollbahn ergibt sich als ein Polynom vierten Grades und verläuft durch jeweils zwei Vektorpunkte [Edl97]. Der Bahnregler bestimmt die Abweichung der aktuellen Roboterposition von einer Referenzposition auf der Sollbahn und daraus die notwendigen Solldrehzahlen der angetriebenen Räder (Abbildung 7.6).

#### 7.1.2.4 Bahnvorgabe

Die Bahnvorgabe übergibt das abzufahrende Bahnstück in Form zweier Vektorpunkte an den Bahnregler. Intern führt sie einen Zeiger auf das aktuelle Paar von Vektorpunkten, dessen Weiterschaltung bei Erreichen der jeweiligen Endposition ausgeführt wird.

Wird der Missionsplan durch das Missionsmanagement verändert und betrifft diese Änderung das gerade ausgeführte Manöver, so muss an die Bahnvorgabe ein neuer Zeiger für das

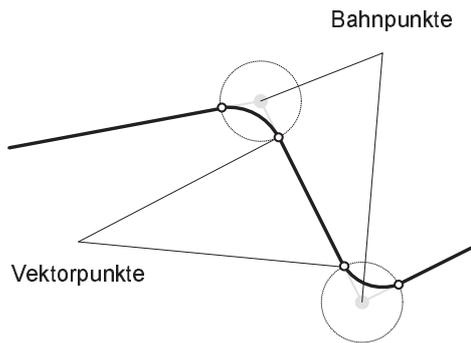


Abb. 7.5: Bahnpunkte und Vektorpunkte

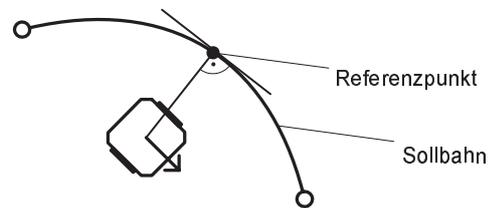


Abb. 7.6: Bahnregelung mit Vektorpunkten

zu nutzende Paar von Vektorpunkten übergeben werden. Das erfolgt über die Schnittstelle vom Missionsmanagement an die Bahnvorgabe.

#### 7.1.2.5 Manöverkatalog

Nachfolgend sollen die für den Kleinroboter zur Verfügung stehenden Manöver vorgestellt werden. Die drei implementierten Typen von Planelementen entsprechend Tabelle 7.1 ergeben sich aus den wesentlichen Aufgaben eines sich im zweidimensionalen Raum bewegenden mobilen Systems. Diese umfassen neben reinen Transportaufträgen auch Überwachungs- und Suchtätigkeiten. Deshalb wurden neben dem einfachen, geradlinigen Planelement *Gerade* zwei unterschiedliche Suchfunktionen erstellt. Mit einem *Mäander* wird ein Suchgebiet ausgehend von einer Längskante mit parallelen Geraden erfasst, während sich eine *Q-Route* spiralförmig beginnend an der Mittellinie des Suchbereichs nach außen vorarbeitet.

Die beiden Suchmanöver werden von der Bahnvorgabe in einzelne Teilelemente zerlegt, die dann sequentiell mit Hilfe des Bahnreglers abgefahren werden. Das aktive Manöver gilt als beendet, wenn das letzte Teilstück realisiert ist. Die Detektion des Endes eines Suchmanövers muss also im Gegensatz zu einem Manöver des Typs *Gerade* nicht mit dem geplanten Endpunkt des Planelementes, sondern mit der aus den Kenngrößen berechneten Endposition der letzten Teilstrecke erfolgen.

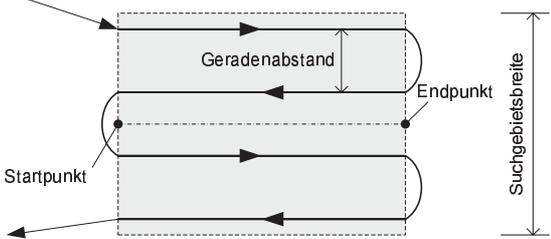
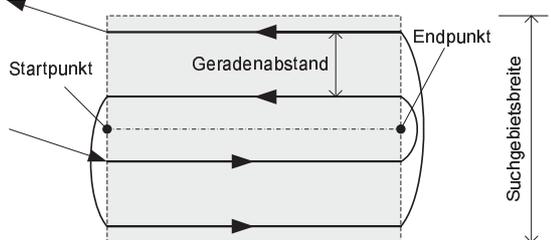
### 7.1.3 Intelligentes Missionsmanagement

In diesem Abschnitt werden die einzelnen Bestandteile des intelligenten Missionsmanagements für die mobilen Kleinroboter vorgestellt. Eine große Rolle spielen dabei die notwendigen Anpassungen an die verwendete Plattform sowie die erzielten Ergebnisse.

#### 7.1.3.1 Missionsplanung

Die Missionsplanung ist der Ausgangspunkt eines jeden Einsatzes autonomer mobiler Systeme. Für die Roboter des Typs *MauSI* müssen die Manöver des im vorangegangenen Abschnitt vorgestellten Manöverkatalogs in der Planungsoberfläche verfügbar sein. Die Abbildung 7.7 zeigt den Missionsplaneditor, der wie auch die gesamte Softwarearchitektur unter MATLAB erstellt wurde. Entsprechend der Einteilung aus Kapitel 4 erfolgt die Planung geometriebasiert.

**Tabelle 7.1:** Manöverkatalog für MauSI

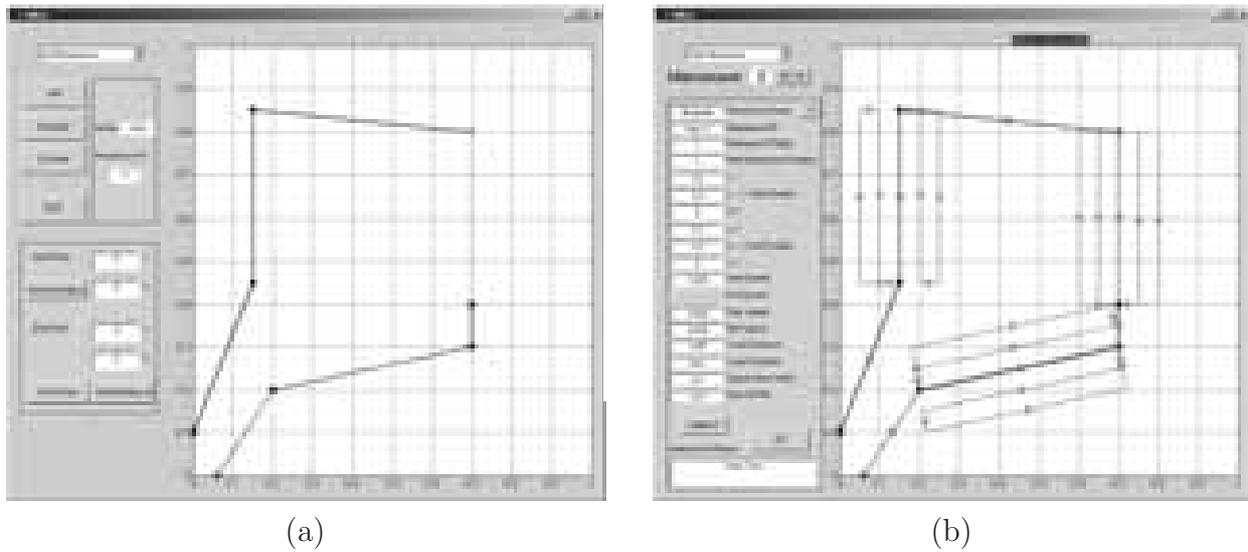
Manövertyp	Kenngroßen	Darstellung
Gerade	Startpunkt, Endpunkt	
Mäander	Startpunkt, Endpunkt, Suchgebietsbreite, Geradenabstand	
Q-Route	Startpunkt, Endpunkt, Suchgebietsbreite, Geradenabstand	

Das Koordinatensystem zeigt das Einsatzgebiet des Roboters. Hier können ebenfalls im voraus bekannte Hindernisse dargestellt werden. Die Planung der Mission erfolgt graphisch, wobei alle möglichen Parameter der einzelnen Manöver auch über Eingabefelder gesetzt werden können. Der in der Abbildung 7.7 (a) dargestellte Missionsplan beinhaltet nur die Start- und Endpunkte der Manöver. Für eine zweckmäßige Beurteilung des gesamten Planes und insbesondere der Suchmanöver fehlt die Ansicht der einzelnen Teilstrecken der Planelemente. Abbildung 7.7 (b) zeigt den gleichen Missionsplan unter Einbeziehung aller vom Roboter abzufahrenden Wegstrecken. Hier können dann Sicherheitsabstände zu eventuellen Hindernissen oder Überlappungen verschiedener Suchgebiete beurteilt werden. Einen abschließenden Überblick über die benötigten Ressourcen der Mission erhält der Planer durch die integrierte Kalkulation des zurückzulegenden Weges, der Dauer der Mission sowie des geschätzten Energieverbrauchs.

Einmal erstellte Missionspläne können gespeichert, geladen und bearbeitet werden. Das ermöglicht die Reduktion des erforderlichen Planungsaufwandes, in dem unterschiedliche Typen von Missionen in Form von Schablonen vorbereitet und in der entsprechenden Situation parametrisiert und umgesetzt werden.

### 7.1.3.2 Missionsüberwachung

Aufgrund des einfachen Aufbaus des Roboters und der nur beschränkt zur Verfügung stehenden Informationen über den Zustand von Sensorik und Aktorik wurde die Missionsüberwa-



**Abb. 7.7:** Grafischer Missionsplaneditor für MauSI-Roboter, mit (a) vereinfachter und (b) komplexer Darstellung

**Tabelle 7.2:** Getestete Umplanungssituationen für MauSI

Situation	Umplanungsaufgaben
Ausfall der Nutzlast	Entfernen der Manöver aus dem Missionsplan, die die Nutzlast verwenden
Ausfall wesentlicher Module	Abbruch der Mission an der aktuellen Position (Löschen aller verbleibenden Manöver)
Ausfall unkritischer Module	Sofortige Rückkehr zum geplanten Endpunkt der Mission
Mangel an Ressourcen	Optimierung des Missionsplanes bei Unterschreiten der benötigten Ressourcen

chung mit Hilfe künstlich generierter Zustandsmeldungen als Eingangsgrößen durchgeführt. Dabei wird der Begriff „Nutzlast“ für eine zusätzliche Sensorik oder Aktorik eingesetzt, die nicht zwingend für den sicheren Betrieb des Roboters benötigt wird. Aufgabe der Nutzlast kann die Erfassung von Daten zur späteren Auswertung während der Ausführung einzelner Manöver oder auch die Aufnahme und der Transport von gefundenen Objekten sein. Der Ausfall dieser Nutzlast führt dann zum Entfernen derjenigen Manöver, die die Nutzlast benötigen.

Für den Test des Missionsmanagements erfolgte eine Auswahl von Situationen, die unterschiedliche Methoden zur Missionsumplanung nutzen (Tabelle 7.2). Für jede dieser Situationen wurden die notwendigen Regeln für das Expertensystem zur Missionsüberwachung analog dem Beispiel aus Programmauszug 5.1 aufgestellt. Während der Missionsrealisierung wird die Überwachung in regelmäßigen Zeitabständen ausgeführt. Ergibt sich aus den Regeln, dass der Missionsplan verändert werden muss, erfolgt die Missionsumplanung mit den von der Überwachung generierten Umplanungsbefehlen.

### 7.1.3.3 Missionsumplanung

Die Missionsumplanung startet sofort nach dem Erkennen einer Umplanungssituation. Die von der Missionsüberwachung generierten Umplanungsbefehle werden zunächst auf Beeinflussungen untereinander überprüft. So kann zum Beispiel die Anforderung zur Optimierung des Missionsplanes beim gleichzeitigen Auftrag zum Entfernen eines Manövers aus dem Plan ignoriert werden, da eventuell durch das Löschen genügend Ressourcen für den restlichen Missionsplan vorhanden sind. Ist das nicht der Fall, erfolgt eine erneute Anforderung zur Planoptimierung im nächsten Überwachungszyklus.

Die verbliebenen Umplanungsbefehle werden dann sequentiell realisiert, der resultierende Missionsplan mit Hilfe der Planbereinigung (Abschnitt 6.2.5) von überflüssigen Füllplanelementen befreit und abschließend in die für die Bahnvorgabe notwendige Folge von Vektorpunkten überführt. Die Bahnvorgabe erhält darüber hinaus einen aktualisierten Zeiger auf den zu befahrenden Sollbahnabschnitt.

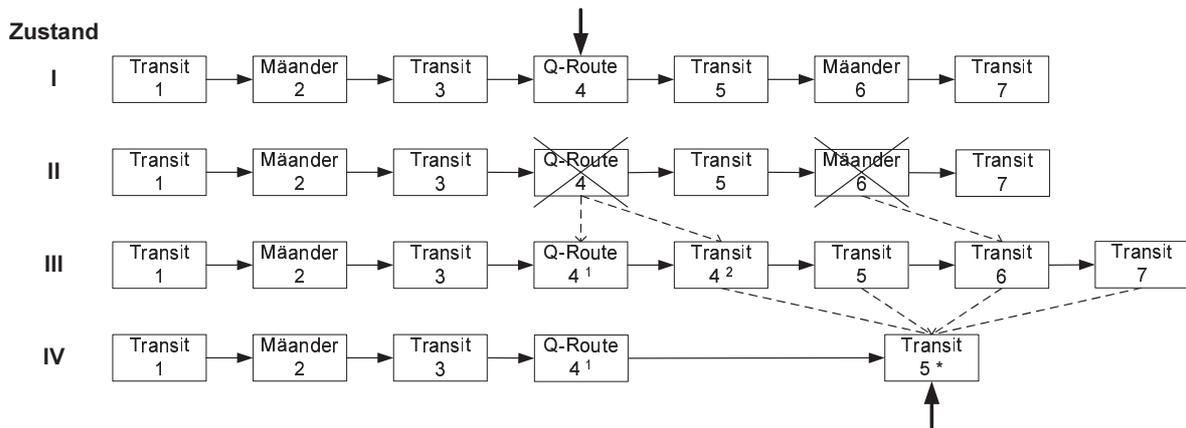
Die Verfahren der geografischen Planprüfung und -modifikation sind für die Kleinroboter nicht genutzt worden. Aussagen zu ihrer Anwendung sind dem Abschnitt zum autonomen Unterwasserfahrzeug *DeepC* zu entnehmen.

### 7.1.3.4 Beispiele und Ergebnisse

Die unterschiedlichen Umplanungssituationen wurden mit dem in Abbildung 7.7 dargestellten Missionsplan getestet. Der Plan besteht aus drei Suchmanövern mit dazwischen liegenden Transitelementen. Die Transite sind dabei als Füllplanelemente gedacht, um auch die Planbereinigung in den Tests einsetzen zu können. Zu jeder Umplanungssituation werden im Folgenden jeweils der Originalplan und der modifizierte Missionsplan blockorientiert sowie die Fahrkurve des Roboters aus der Simulation dargestellt.

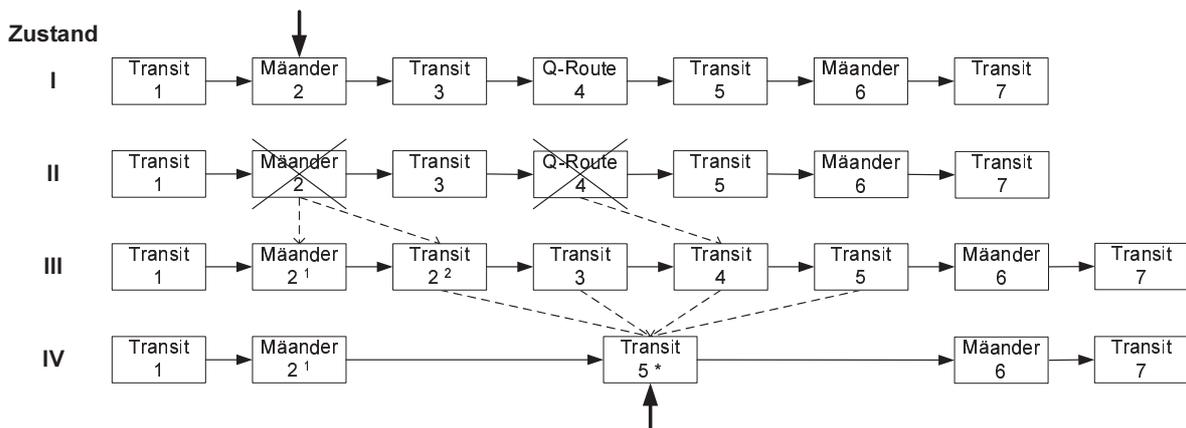
*Ausfall der Nutzlast.* Diese Situation wird anhand zweier Szenarien vorgestellt. Im Szenario I ist die Nutzlast während der Ausführung aller drei Suchmanöver aktiviert und der Ausfall wird während der Realisierung des zweiten Suchmanövers festgestellt. In Abbildung 7.8 ist der gesamte Umplanungsvorgang dargestellt. Der Zustand I kennzeichnet die Situation vor der Detektion des Fehlers, das Manöver *Q-Route 4* ist aktiviert. Die Regeln des Überwachungssystems generieren bei Ausfall der Nutzlast die Befehle *Löschen 'Mäander 6'* und *Löschen 'Q-Route 4'* (Zustand II). Zustand III zeigt die Situation nach dem Löschen der Planelemente. Beim Entfernen des aktuellen Manövers verbleibt der bereits realisierte Anteil im Missionsplan (*Q-Route 4*<sup>1</sup>), während für das restliche Manöver ein Füllplanelement (*Transit 4*<sup>2</sup>) eingefügt wird. Das zweite zu löschende Manöver wird lediglich durch einen Transit ersetzt (*Transit 6*). Zum Abschluss der Umplanung wird die Planbereinigung durchgeführt (Zustand IV), wobei alle aufeinander folgenden Transit-Manöver (*Transit 4*<sup>2</sup> bis *Transit 7*) zu einem einzelnen Füllplanelement zusammengesetzt werden. Dieses kombinierte Manöver stellt daraufhin das aktuelle Manöver für die Bahnvorgabe dar. In Abbildung 7.10 (a) ist die resultierende Fahrkurve des Roboters während dieser Mission dargestellt.

Im Szenario II ist die Nutzlast für das Manöver *Mäander 6* nicht erforderlich, sodass dieses Planelement nicht entfernt wird (Abbildung 7.9). Der Ausfall wird hier bereits während der Realisierung von *Mäander 2* detektiert. Dieses Manöver wird abgebrochen und der noch zu bearbeitende Anteil wiederum durch einen Transit ersetzt, während *Q-Route 4* vollständig entfernt wird. Die Planbereinigung fasst auch hier die überflüssigen Füllplanelemente



**Abb. 7.8:** Umplanung bei Ausfall der Nutzlast, Szenario I

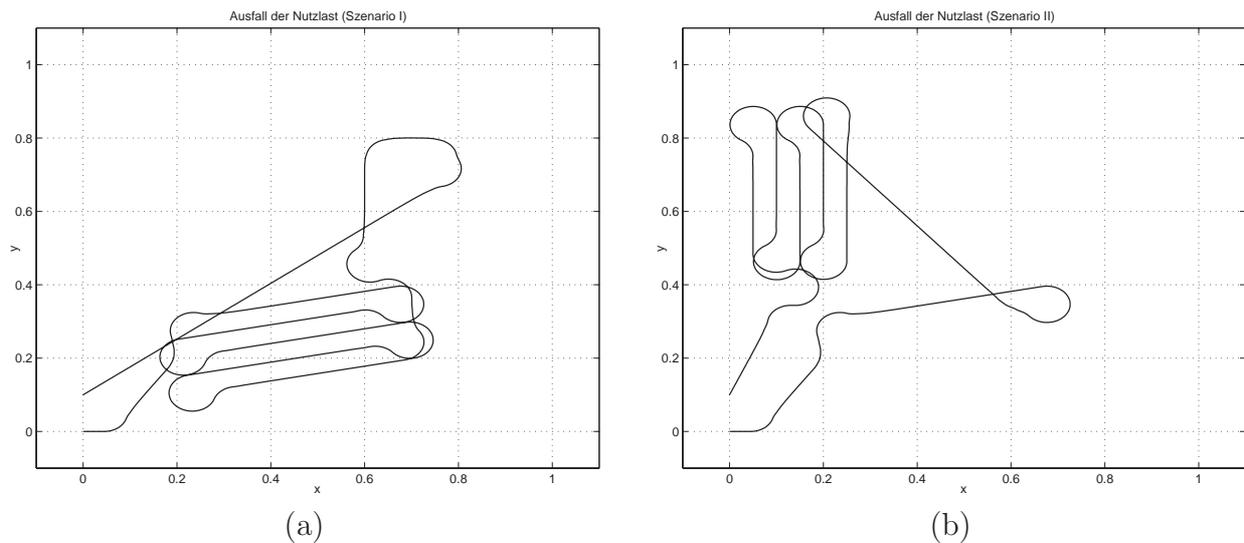
zusammen. Der Roboter legt in diesem Szenario die Fahrstrecke aus Abbildung 7.10 (b) zurück.



**Abb. 7.9:** Umplanung bei Ausfall der Nutzlast, Szenario II

*Ausfall wesentlicher Module.* Bei einem Ausfall von wesentlichen Komponenten des Roboters wie zum Beispiel der Hindernisvermeidesensorik soll die Mission umgehend beendet werden. Für den vorliegenden Kleinroboter kann dies einfach durch Entfernen aller noch zu realisierenden Manöver geschehen. An das aktuell angearbeitete Planelement wird ein kurzes Transitstück *Transit Abbr.* angefügt, um an dessen Ende einen regulären Missionsabschluss zu erzielen (Abbildung 7.11).

*Ausfall unkritischer Module.* Im Gegensatz zum Ausfall wesentlicher Komponenten kann der Roboter in dieser Situation die Fahrt fortsetzen, sollte aber aus Sicherheitsgründen umgehend die geplante Abschlussposition der Mission anfahren. Analog zu den Handlungen des zuletzt genannten Szenarios werden hier ebenso alle verbleibenden Manöver aus dem Missionsplan entfernt. Als Abschluss wird ein Transit zum Endpunkt der Mission in den Plan aufgenommen. Die Arbeitsweise während der Umplanung entspricht dem in Abbildung 7.11 dargestellten Ablauf. Dabei ergibt sich die Fahrtrajektorie in Abbildung 7.13.



**Abb. 7.10:** Fahrtrajektorie bei Ausfall der Nutzlast, (a) Szenario I und (b) Szenario II

*Mangel an Ressourcen.* Die wichtigste Ressource des Roboters ist die verfügbare elektrische Energie, die während der gesamten Mission überwacht wird. Der aktuelle Energieverbrauch wird während der Mission aus den Motordrehzahlen abgeleitet und daraus können wiederum die verfügbaren Reserven berechnet werden. Unterschreiten die Reserven den Bedarf der Mission, wird eine Optimierung durchgeführt. Entsprechend der im Abschnitt 6.3.2 vorgestellten Verfahrensweise müssen zunächst diejenigen Manöver ermittelt werden, die keine Transite darstellen. Im vorliegenden Beispiel sind das die Manöver *Mäander 2*, *Q-Route 4* und *Mäander 6*. Anhand des Wertes dieser Manöver können dann die möglichen Konfigurationen für die Optimierung aufgestellt und bezüglich ihres Gesamtwertes entsprechend Abbildung 7.12 absteigend sortiert werden. Die Überprüfung der Nebenbedingungen ergibt schließlich die Optimallösung, bei der das Manöver *Mäander 6* entfernt werden muss. Der derart modifizierte Missionsplan wird vom Fahrzeug realisiert, ohne das weitere Ressourcenprobleme auftreten (Abbildung 7.14).

Für die mobilen Kleinroboter wird nur das Optimierungsverfahren angewendet, das stets zum globalen Optimum führt. Ein Vergleich mit der einfacheren Greedy-Heuristik zur schnellen Problemlösung ist im Abschnitt 7.2.2.4 für das autonome Unterwasserfahrzeug *DeepC* vorgenommen worden.

*Bewertung.* Die dargestellten Umplanungssituationen zeigen beispielhaft unterschiedliche Aufgaben des intelligenten Missionsmanagements. Darüber hinaus sind noch vielfältige weitere Szenarien denkbar, in denen durch das Missionsmanagement eine sinnvolle Modifikation des Missionsplanes vorgenommen werden kann. Durch den Einsatz eines regelbasierten Expertensystems zur Überwachung und Generierung von Umplanungsbefehlen ist eine einfache und flexible Erweiterung des Systems möglich.

Der Einsatz des intelligenten Missionsmanagements vereinfacht darüber hinaus die gesamte Softwarearchitektur der Kleinroboter, da unterschiedliche Reaktionen auf Ausfälle von Sensorik und Aktorik über eine einheitliche Schnittstelle ausgelöst werden können. Ohne diese Erweiterung wäre eine Anpassung unterschiedlicher Module notwendig, um zumindest einen Teil der Aufgaben übernehmen zu können. Beispielsweise könnte ein unmittelbarer Missions-

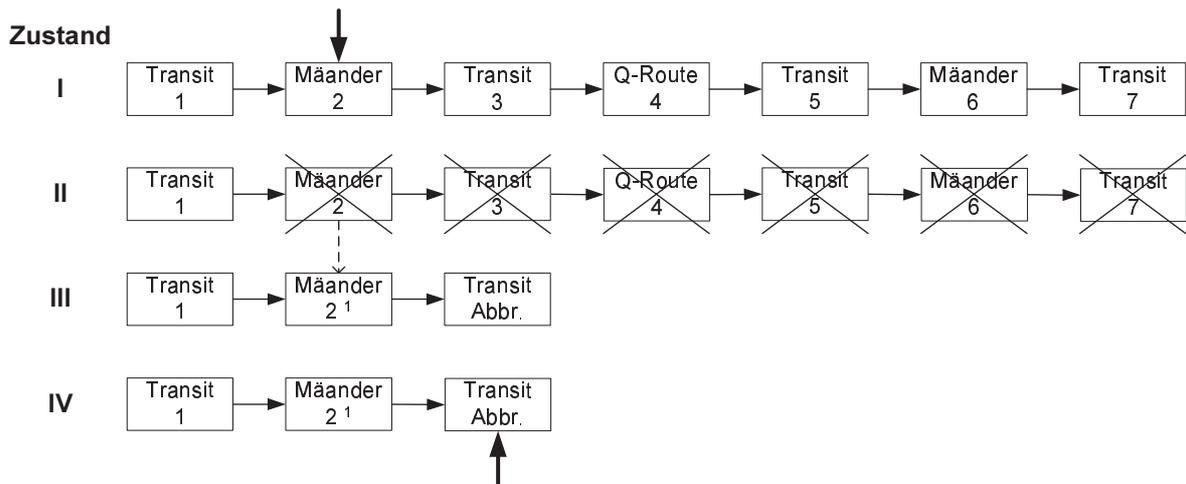


Abb. 7.11: Umplanung bei Ausfall wesentlicher Fahrzeugmodule

	Mäander 2	Q-Route 4	Mäander 6	Wert der Konfiguration	Nebenbedingungen erfüllt?
Wert	1.827123	1.846154	1.833333		
1	1	1	1	5.506610	—
0	1	1	1	3.679487	—
1	1	0	0	3.673277	✓
1	0	1	1	3.660456	
0	1	0	0	1.846154	
0	0	0	1	1.833333	
1	0	0	0	1.827123	
0	0	0	0	0.000000	

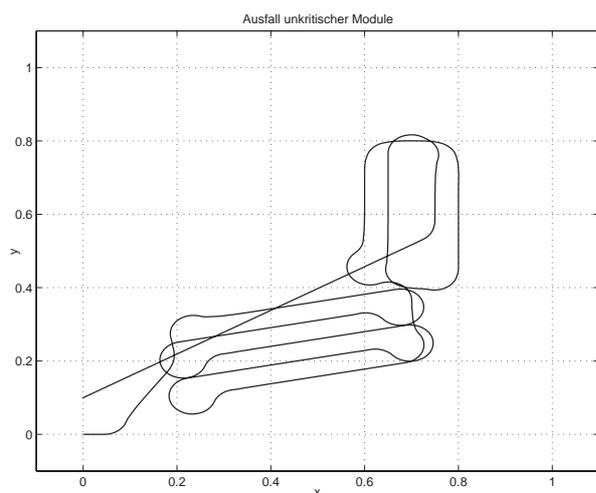
Abb. 7.12: Optimierung des Missionsplanes bei Ressourcenmangel

abbruch bei Ausfall wesentlicher Module auch durch die Bahnregelung ausgeführt werden. Dazu muss diesem Softwaremodul über eine zusätzliche Schnittstelle die entsprechende Diagnosemeldung übergeben werden, um im Anschluss daran das Fahrzeug anzuhalten.

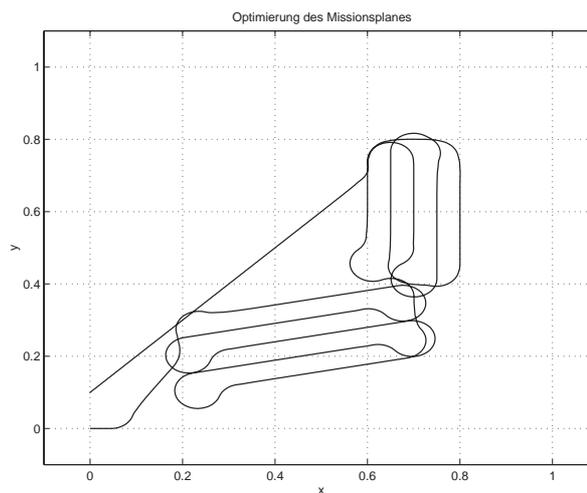
## 7.2 Das autonome Unterwasserfahrzeug DeepC

Das Unterwasserfahrzeug *DeepC* entsteht im Rahmen eines BmBF-Verbundprojektes zwischen verschiedenen Partnern aus Forschung und Industrie [Pfü03b]. Ziel des Projektes ist es, ein modernes, modular aufgebautes AUV zu entwickeln und zu erproben, dass mit herausragenden technischen Daten einen Innovationsschub für die beteiligten Unternehmen erzeugt. Die mit Hilfe des Prototypen zu sammelnden Erfahrungen sollen in einem kommerziell zu vermarktenden System münden.

Als primäre Einsatzszenarien für das AUV wurden Vermessungs-, Inspektions- und Suchaufgaben festgelegt. Dies ergab sich aus einer Analyse der möglichen Einsparpotentiale in



**Abb. 7.13:** Fahrkurve bei Ausfall unkritischer Module



**Abb. 7.14:** Fahrtrajektorie unter Einsatz der Optimierung

typischen Anwendungsfällen autonomer Unterwassersysteme. Ein System, bestehend aus einem ferngesteuerten Fahrzeug oder einem Schleppschlitten in Verbindung mit einem Oberflächenschiff, verursacht hohe Kosten über die gesamte Missionsdauer. Durch den Einsatz eines AUVs können enorme Kostenvorteile erzielt werden, da insbesondere bei langfristigen Einsätzen ein zusätzlicher Nutzen durch die freiwerdenden Kapazitäten des Schiffes entsteht [Pfü03a].

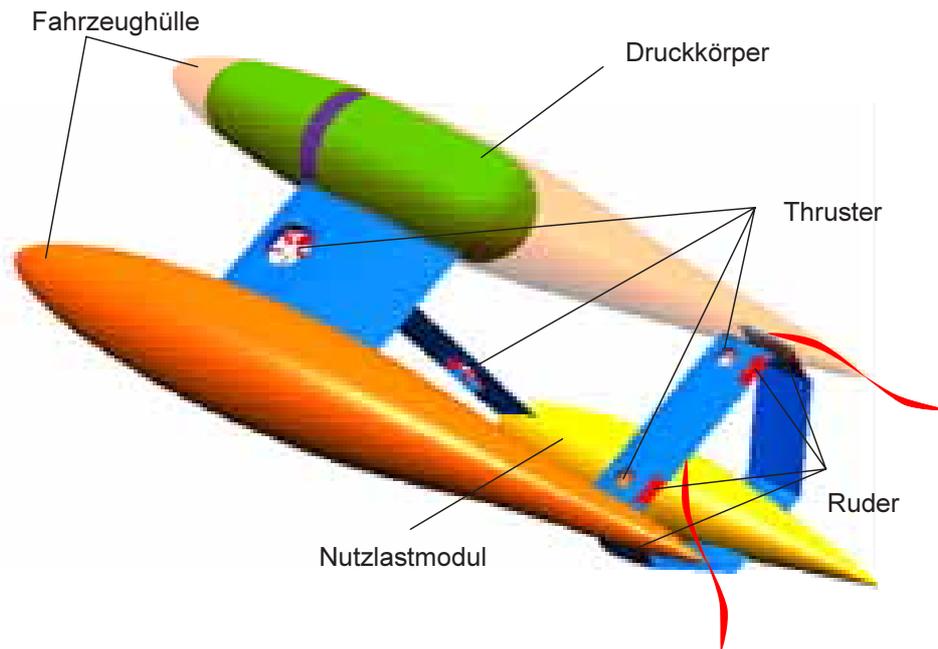
Aus einer Analyse von Anforderungen seitens potentieller Nutzer des Systems und von Erfahrungen mit existierenden autonomen Unterwasserfahrzeugen ergaben sich sowohl die technischen Daten als auch der Aufbau von Hard- und Software (Tabelle 7.3). Die geplante Tauchtiefe und -dauer sind für den Einsatz zur Vermessung von Kabeltrassen und die Inspektion vorhandener Seekabel erforderlich, da ein stetig wachsender Bedarf an schnellen Datenverbindungen insbesondere zwischen den Kontinenten zu verzeichnen ist. Typischerweise liegen die Seekabel in den Ozeanen in Tiefen größer 2000 Meter, sodass ein dafür geeignetes Tauchsystem in diesem Tiefen operieren können muss. Die hohe Einsatzdauer gewährleistet darüber hinaus eine effiziente Arbeitsweise, da großflächige Missionen realisiert und dadurch die Anzahl von Einsätzen, zum Beispiel beim Vermessen eines Meeresbodenabschnitts, reduziert werden können.

Diese Anforderungen haben einen großen Einfluss auf die Auslegung aller Fahrzeugsysteme. Für die Software besteht das primäre Ziel in einer zuverlässigen und sicherheitsorientierten Arbeitsweise des AUV. Trotzdem soll mit Hilfe moderner Verfahren ein intelligentes Verhalten des Systems ermöglicht werden. Das intelligente Missionsmanagement versucht, diese Probleme auf der höchsten Ebene innerhalb der Softwarearchitektur zu lösen.

## 7.2.1 Aufbau und Steuerung

### 7.2.1.1 Aufbau

*DeepC* besteht im Wesentlichen aus zwei identischen Fahrzeugeinheiten und einem Nutzlastmodul (Abbildung 7.15). In den beiden Einheiten sind die redundant aufgebauten Druckkör-



**Abb. 7.15:** Aufbau des autonomen Unterwasserfahrzeuges DeepC

per untergebracht, die unter anderem jeweils einen Energiespeicher, ein Modul zur Energieerzeugung sowie die Rechnermodule beinhalten. Als Energiequelle kommen Brennstoffzellen zum Einsatz, die mit Wasserstoff und Sauerstoff aus Flüssiggaspeichern versorgt werden. Durch das Redundanzkonzept ist gewährleistet, dass eine sichere Rückkehr zur Wasseroberfläche zum Beispiel bei Ausfall eines Energieerzeugungssystems oder eines Rechnermoduls erfolgen kann.

*Aktorik.* Die Hauptantriebe befinden sich am Ende der zwei Basiseinheiten im gefluteten Bereich. Sie sind druckneutral aufgebaut, um nur geringfügige Einbußen beim Wirkungsgrad gegenüber einem Einsatz in Luft hinnehmen zu müssen. Als weitere Stelleinrichtungen sind Thruster zur Aufbringung von Schubkräften senkrecht zur Hauptantriebsrichtung sowie Ruder im hinteren Teil des Fahrzeuges vorhanden. Während die Ruder der dynamischen Beeinflussung des Fahrzeugverhaltens dienen, sind die Thruster vorrangig für Positionierungsaufgaben und langsame Bewegungsabläufe verantwortlich.

*Sensorik.* Die Positionsbestimmung eines Unterwasserfahrzeuges stellt auch mit den verfügbaren modernen Sensoren eine große Herausforderung dar [Pfü00]. Die Ursachen dafür liegen unter anderem in den kaum abschätzbaren äußeren Einflüssen auf das Fahrzeug (z.B. Strömung) sowie in der schwierigen bis unmöglichen absoluten Positionsbestimmung in der Tiefsee. Kernstück der Navigationssensorik ist das inertielle Navigationssystem (INS), das mit Hilfe von optischen Kreisel und Beschleunigungsmessern eine Abbildung der Fahrzeugbewegung durchführt. Nachteilig wirkt sich hier die Drift aus, die bei langfristigen Positionsschätzungen zu erheblichen Abweichungen der vermuteten Fahrzeugposition von der tatsächlichen führt. So erzielen in U-Booten eingesetzte, hochgenaue inertielle Navigationssysteme einen mittleren Fehler von 926 Meter in 24 Stunden [Hin02]. Derartige Größenordnungen sind für die beabsichtigten Einsatzfelder von DeepC wie die exakte Vermessung

**Tabelle 7.3:** *DeepC: Wesentliche technische Daten*

Kenngröße	Wert
Maße (LxBxH)	5,75 m x 2,30 m x 1,70 m
Gewicht	2400 kg
Maximale Tauchtiefe	4000 m
Maximale Reichweite	400 km
Einsatzdauer mit Marschgeschwindigkeit	45 h
Maximalgeschwindigkeit	6 kn (3,1 m/s)
Marschgeschwindigkeit	4 kn (2,1 m/s)

des Meeresbodens nicht zulässig. Deshalb wird das INS durch ein Doppler-Log (DOLOG) unterstützt, das bei entsprechendem Abstand zum Meeresboden eine zuverlässige Geschwindigkeitsschätzung der so genannten „Fahrt über Grund“ ermöglicht. Der Navigationsfehler kann damit auf wenige Meter pro Stunde Fahrt reduziert werden.

Mit einem GPS-Empfänger ist schließlich ein absoluter Positionsabgleich des inertialen Navigationssystems an der Wasseroberfläche möglich. Dafür ist im Manöverkatalog ein komplexes Planelement *GPS-Update* vorgesehen. Ausgelöst wird das GPS-Update, wenn der vom Navigationsmodul geschätzte Positionsfehler einen zulässigen Wert überschreitet. Der Grenzwert kann dabei für jedes komplexe Manöver bei der Missionsplanung einzeln festgelegt werden.

Zur Vermeidung von Kollisionen besitzt *DeepC* ein Sonar kürzerer Reichweite, dessen Scanrichtung und Öffnungswinkel dynamisch an Geschwindigkeit und Richtung der Fahrzeugbewegung angepasst werden können. Drei Kameramodule mit zugehöriger Beleuchtung komplettieren die Standardsensorik des Unterwasserfahrzeugs.

Der Prototyp besitzt darüber hinaus als Nutzlastmodul ein Fächerlot, das eine hochauflösende Vermessung des Meeresbodens durch eine fächerförmige Abstrahlung von Schallwellen quer zur Fahrtrichtung erlaubt. Für jede der ausgesendeten Schallwellen wird die Laufzeit des Signals gemessen und daraus der Abstand zum AUV berechnet. Aus diesen Informationen sowie den zugehörigen Navigationsdaten des Fahrzeuges können Konturkarten oder 3D-Bilder des Meeresbodens erstellt werden.

*Rechner.* In jedem der Druckkörper ist ein Rechnermodul untergebracht. Bedingt durch das Redundanzkonzept führen beide Module die gleiche Software aus. Jedes Modul besteht aus mehreren vernetzten Rechnern, auf die die für den Betrieb des Fahrzeuges notwendigen Softwareeinheiten verteilt werden. Die Kommunikation zwischen den verschiedenen Programmen erfolgt dabei entsprechend der CORBA-Spezifikation (**C**ommon **O**bject **R**equest **B**roker **A**rchitecture), die einen standardisierten, objektorientierten Ansatz für das Schreiben verteilter Anwendungen darstellt [Obj02].

#### 7.2.1.2 Softwarearchitektur

*DeepC* besitzt ebenso wie der Kleinroboter *MauSI* eine Softwarearchitektur entsprechend Abschnitt 2.3.1. In Abbildung 7.16 sind die für das intelligente Missionsmanagement wesentlichen Module und ihre Verknüpfungen dargestellt.

Die fähigkeitsbasierte Ebene beinhaltet die grundlegenden Module für die Fahrzeugsteuerung. Dazu zählen neben dem Autopiloten auch die Module zur Aufbereitung und Fusion der von den verschiedenen Sensoren gelieferten Informationen. Die Schnittstellen zur Hardware des Fahrzeuges werden über Interface-Module realisiert. Anstelle der Hardware kann auch eine Simulationsumgebung integriert werden, die im Vorfeld der praktischen Erprobung eine weitreichende Untersuchung der erstellten Softwaremodule erlaubt. Darüber hinaus sind damit manuelle Fahrten in einer virtuellen Welt möglich, die für die Ableitung von Handlungsfolgen aus den aufgezeichneten Fahrkurven des Operators, beispielsweise zur Bewältigung von Hindernissituationen genutzt werden können [EPWZ00]. Die Navigation verknüpft die Informationen der unterschiedlichen Sensoren und generiert daraus den Navigationsdatensatz. Dieser Datensatz ist eine wesentliche Grundlage für die Arbeit der anderen Softwareeinheiten. Die vom Sonar gelieferten Messwerte werden durch die Hinderniserkennung ausgewertet, um Position, Größe und Geschwindigkeit von potentiellen Hindernissen zu ermitteln. Dafür sind entsprechende Bildverarbeitungsmethoden entwickelt worden.

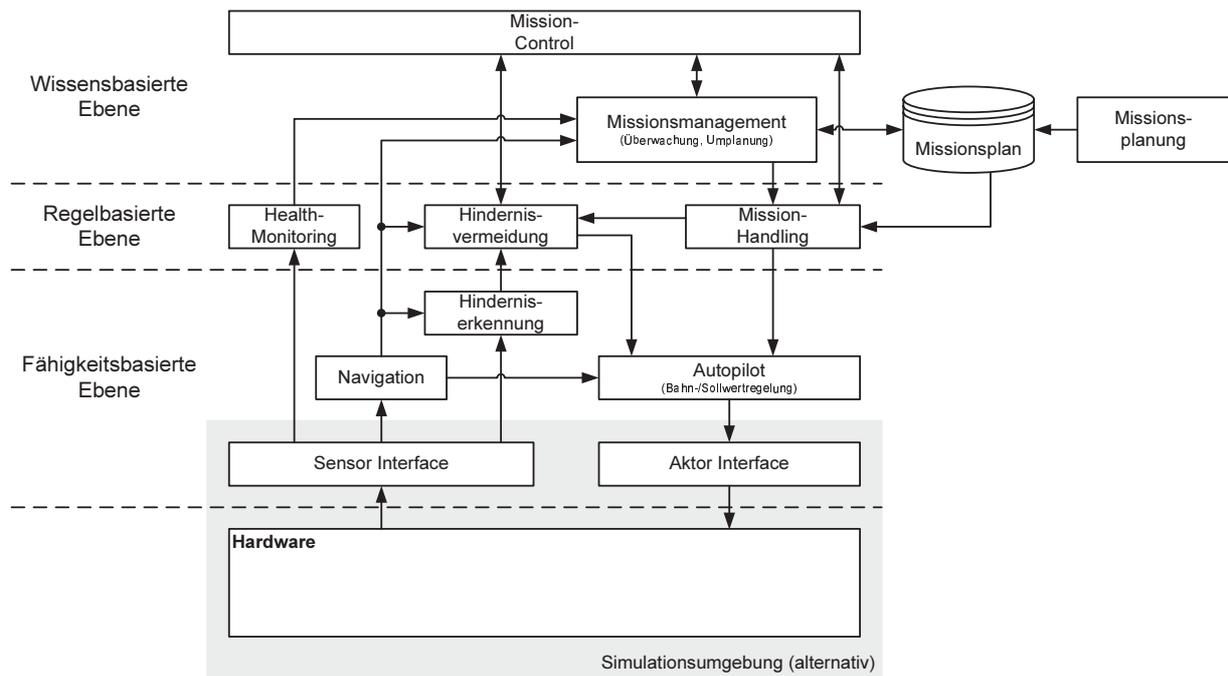
Das Mission-Handling teilt auf der regelbasierten Ebene die komplexen Manöver in zugehörige Basismanöver auf, die an den Autopiloten übergeben werden. Als Basismanöver werden die für die komplexen Planelemente notwendigen geometrischen Grundkonstrukte (z.B. Gerade, Kreis, Kreisbogen) bezeichnet. Darüber hinaus beinhaltet das Mission-Handling wichtige Verwaltungsfunktionen für den Missionsplan. Die Hindernisvermeidung bewertet die von der Hinderniserkennung erfassten Objekte im Umfeld des Fahrzeuges hinsichtlich möglicher Kollisionen [Eic04]. Dabei werden auch die nächsten zu realisierenden Basismanöver einbezogen. Das Health-Monitoring führt eine Diagnose der Hardware des Fahrzeuges durch und erzeugt in verschiedenen Situationen Vorschläge für die durchzuführenden Handlungen bei Auftreten defekter Subsysteme im AUV. Diese Vorschläge und die Ergebnisse der Diagnose fließen in das intelligente Missionsmanagement ein.

Die wissensbasierte Ebene beinhaltet analog zum Kleinroboter *MauSI* das intelligente Missionsmanagement, das in Abschnitt 7.2.2 näher vorgestellt wird. Als zentrale Entscheidungsinstanz des gesamten Systems ist das Modul Mission-Control für die Koordinierung der Arbeit der verschiedenen Programme verantwortlich. So muss zum Beispiel vor der Durchführung von Modifikationen des Missionsplanes eine entsprechende Genehmigung von Mission-Control vorliegen. Darüber hinaus überwacht dieses Modul alle ablaufenden Prozesse, um frühzeitig ein Fehlverhalten einzelner Einheiten des AUV zu erkennen.

### 7.2.1.3 Manöverkatalog

Analog zu den im Abschnitt 7.1.2.5 vorgestellten Manövern für den Kleinroboter *MauSI* sollen hier die für *DeepC* verfügbaren Manöver präsentiert werden. Alle Planelemente besitzen dabei dreidimensionale Positionsangaben. Die zusätzlich vorhandene Tiefe gestattet sowohl eine absolute (Tiefe unter der Wasseroberfläche) als auch eine relative Ausrichtung der Manöver (Höhe über dem Meeresgrund). Die relative Höhenangabe spielt besonders für Vermessungsaufgaben eine große Rolle, da ein sich ständig ändernder Abstand zum zu vermessenden Objekt aufwendige Skalierungsrechnungen während der Datenaufbereitung erfordert.

Neben den bereits für *MauSI* implementierten drei Manövertypen *Gerade*, *Mäander* und *Q-Route* (Tabelle 7.1)) sind bedingt durch den dreidimensionalen Raum und die vorhandene Sensorik weitere Manöverarten erforderlich (Tabelle 7.4). Speziell für das Fächerlot wurde



**Abb. 7.16:** Softwarearchitektur des AUV DeepC (Auszug)

ein Manöver zur Vermessung des Profils der Wasserschallgeschwindigkeit in Abhängigkeit von der Tiefe (*Sound Velocity Profile, SVP*) eingeführt, damit dieser Sensor kalibriert werden kann. Das Manöver wird vor dem Einsatz des Sensors auf Anforderung in den Plan aufgenommen und umgehend realisiert.

Zum Anfahren unterschiedlicher Tiefen durch das AUV sind sowohl ein *Auftauch-* als auch ein *Abtauchmanöver* im Katalog verfügbar. Zur Verringerung des Navigationsfehlers während der Tiefenänderung werden diese Manöver in Form einer liegenden Acht ausgeführt. Sie dienen lediglich einer vertikalen Positionsanpassung und gehören damit nach der im Abschnitt 6.2.1 zu findenden Festlegung zu den Füllplanelementen. Als Konsequenz sind auch Ab- und Auftauchmanöver von einer Planbereinigung betroffen. Die dabei zu beachtenden Besonderheiten sind im Abschnitt 7.2.2.3 dargestellt. Bei allen nachfolgenden Abbildungen werden der Einfachheit halber die horizontalen Transitmanöver des Typs *Gerade* als *Transite* und die vertikalen mit ihrem zugehörigen Manövertyp (*Auftauchen, Abtauchen*) bezeichnet.

Das *Circling*-Manöver dient der Ortsstabilisierung des AUV und wird eingesetzt, um vom Oberflächenschiff aus einen Kontakt zum Fahrzeug herzustellen. Manöver dieser Art sind üblicherweise bei der Missionsplanung an unterschiedlichen Stellen im Plan vorhanden, um auch während der Missionsdurchführung Kontakt zum AUV aufnehmen zu können.

Das *GPS-Update* setzt sich aus einem Auftauch-, einem Update- und einem Abtauchmanöver zusammen. Das eigentliche Update-Manöver besteht aus einer kreisförmigen Bewegung analog dem *Circling*-Manöver. Durch die Festlegung der Fahrkurven für das Auf- und Abtauchmanöver startet und endet ein *GPS-Update* stets an der gleichen Position.

Verschiedene Manöver des Manöverkatalogs können als so genannte *Event-Manöver* bei der Missionsplanung in ihrer Geometrie beschrieben und im späteren Einsatz bei dem Missions-

**Tabelle 7.4:** Manöverkatalog für DeepC, Erweiterungen gegenüber MauSI

Manövertyp	wesentliche Kenngrößen	Darstellung <i>Draufsicht</i>	<i>Tiefenprofil</i>
GPS-Update	Startpunkt, Endpunkt, CEP-Sollwert		
Auftauchen	Startpunkt, Endpunkt		
Abtauchen	Startpunkt, Endpunkt		
Sound Velocity Profile	Startpunkt, Endtiefe		
Circling	Startpunkt, Endpunkt, Radius		

management angefordert werden. Damit kann bei Auftreten bestimmter, vor dem Missionsstart bekannter Situationen mit vordefinierten Aktivitäten des AUV reagiert werden. Ein Beispiel dafür ist die Durchführung einer Kabelinspektion, bei der bei einem Verlust des zu inspizierenden Kabels durch die entsprechende Sensorik ein Suchmanöver an der letzten bekannten Position mit gültigem Signal ausgeführt wird. Der zu nutzende Manövertyp und die Ausdehnung des Suchgebietes werden durch den Operator in Form eines Event-Manövers vor dem Missionsstart festgelegt, Positionierung und Ausrichtung erfolgen dann anhand der vorhandenen Navigationsdaten durch das intelligente Missionsmanagement.

Diese komplexen Manöver werden von dem Modul Mission-Handling in eine Folge von Basismanövern zerlegt. Der Autopilot realisiert die einzelnen Basismanöver, überwacht das Erreichen ihres Endpunktes und fordert daraufhin das nachfolgende Element an.

## 7.2.2 Intelligentes Missionsmanagement

Durch den universellen Charakter des in dieser Arbeit vorgestellten Konzeptes zum intelligenten Missionsmanagement für autonome mobile System konnte für das AUV *DeepC* die gleiche Struktur wie für die mobilen Kleinroboter *MauSI* eingesetzt werden. Die wesentlichen Bestandteile sind damit die Missionplanung zum Erstellen des Missionsplanes, die Missionsüberwachung, verantwortlich für die Detektion von Umplanungssituationen sowie die Missionsumplanung, die die Modifikationen am Plan vornimmt.

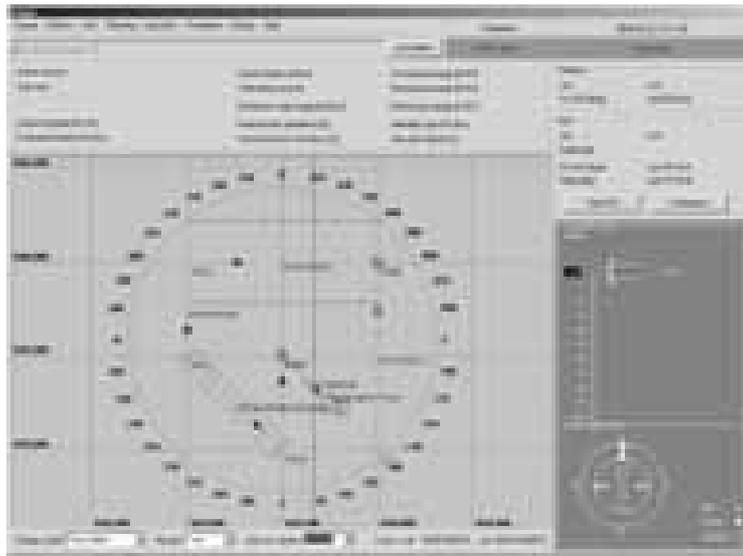
### 7.2.2.1 Missionsplanung

Die Missionsplanung wird innerhalb des Projektes *DeepC* von einem mit der Oberflächen-gestaltung im maritimen Bereich vertrauten Partner realisiert. Diese Aufgabenteilung erfordert die Definition exakter Schnittstellen, um in paralleler Arbeit die einzelnen Elemente des Missionsmanagements erstellen zu können. Als Schnittstelle ergibt sich im vorliegenden Fall der Missionsplan, der um die für das intelligente Missionsmanagement benötigten Parameter ergänzt wurde (z.B. Manöverprioritäten für die Planoptimierung).

Die Erstellung des Missionsplanes erfolgt entweder als geometrie- oder geografieorientierte Planung in Abhängigkeit von der Verfügbarkeit von Seekarten des Einsatzgebietes. Abbildung 7.17 zeigt die Planungsoberfläche für die geometriebasierte Planung mit dem in den nachfolgenden Tests eingesetzten Missionsplan. Für jede Mission können so genannte Recovery-Positionen festgelegt werden. Diese geografischen Koordinaten dienen bei einem unerwarteten Missionsabbruch als bevorzugte Wartepositionen des AUV und werden deshalb in entsprechenden Umplanungssituationen verwendet.

Nach Abschluss der Planung wird der Missionsplan in das Fahrzeug geladen, das ab diesem Zeitpunkt einsatzbereit ist. Der Start der Mission kann unmittelbar nach dem Aussetzen des AUV in das Wasser erfolgen. Mit dem Start wird die Missionsüberwachung aktiviert.

Der für die Tests des intelligenten Missionsmanagement verwendete Missionsplan ist in Abbildung 7.18 schematisch in Draufsicht und Tiefenprofil dargestellt. Er besteht im Wesentlichen aus verbindenden Geraden und zwei Suchmanövern (*Mäander 3*, *Q-Route 5*), die die Nutzlast des AUV benötigen. Die Geraden sind als *Transite* bezeichnet, da sie lediglich Füllplanelemente darstellen. Die notwendige Tiefenanpassung wird über die Manöver *Abtauchen 2* und *Auftauchen 7* durchgeführt. Als Abschluss der Mission ist ein *Circling-Manöver* vorhanden. Das Fahrzeug verbleibt auf dieser Kreisbahn, bis von einem Oberflächenschiff aus



**Abb. 7.17:** Geometriebasierte Planung für DeepC

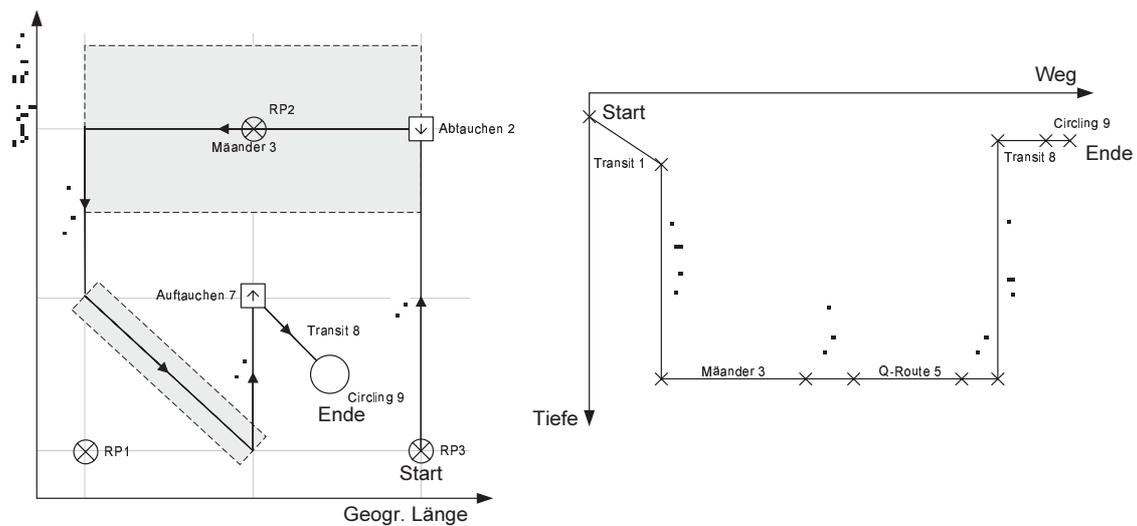
eine Kontaktaufnahme mit dem AUV erfolgt. Die für diesen Plan vorgesehenen Recovery-Positionen sind in der Abbildung mit *RP1* bis *RP3* bezeichnet. Anhand dieses Planes sollen im Abschnitt 7.2.2.4 unterschiedliche Umplanungssituationen und die resultierenden Modifikationen beschrieben werden.

#### 7.2.2.2 Missionsüberwachung

Die Missionsüberwachung wertet Informationen unterschiedlicher Herkunft aus, um eine Entscheidung zum Umplanen der Mission zu treffen. Als wesentlichste Informationsquellen dienen dabei der Navigationsdatensatz und die Diagnoseergebnisse des Health-Monitoring. Diese Daten werden an das integrierte Expertensystem der Missionsüberwachung übergeben und während des nachfolgenden Prozesses der logischen Inferenz zur Aktivierung der betroffenen Regeln genutzt. In der Tabelle 7.5 ist eine Übersicht einiger durch die Missionsüberwachung zu bewältigender Umplanungssituationen dargestellt.

Den Schwerpunkt bilden dabei Reaktionen auf Defekte der Fahrzeugsensorik. Ziel ist es, möglichst wenig Einbußen an den geplanten Missionszielen bei gleichzeitiger minimaler Gefährdung des AUV hinnehmen zu müssen. So wird zum Beispiel bei einem Defekt des inertialen Navigationssystems die Missionsausführung erst nach dem Ansteigen des geschätzten Navigationsfehlers über das definierte zulässige Maß abgebrochen. Die Positionsschätzung erfolgt in einer derartigen Situation mit Hilfe einer einfacheren Sensorik zur Lage- und Beschleunigungsmessung, die eine Weiterfahrt des Fahrzeuges ermöglicht.

Die Überwachung der vorhandenen Ressourcen beschränkt sich für *DeepC* auf die Betrachtung der elektrischen Energie, da die zeitlichen Aspekte einer Mission bereits durch das Modul Mission-Control beobachtet werden. Ein ständig durchgeführter Vergleich der vorhandenen Energiereserven mit den vom Missionsplan benötigten Energiemengen soll frühzeitig Defizite aufdecken, die mittels der Planoptimierung zu beheben sind. Daneben ist auch die aktuelle Leistungsaufnahme des Fahrzeuges zu kontrollieren, da ein langfristiger, zu hoher Strombedarf das Energieerzeugungssystem unzulässig belasten würde. Diese Über-



**Abb. 7.18:** DeepC: Missionsplan für die Tests des intelligenten Missionsmanagements, Draufsicht und Tiefenprofil

wachung erfolgt durch das Health-Monitoring und führt gegebenenfalls zur Anforderung einer Geschwindigkeitsreduktion. Das intelligente Missionsmanagement muss daraufhin die Geschwindigkeiten des aktuellen sowie eventuell der nachfolgenden Manöver anpassen. Vom Health-Monitoring werden dazu Geschwindigkeit und Dauer der Reduktion berechnet und mit der Anforderung übermittelt.

Durch das Health-Monitoring können darüber hinaus auch ein sofortiger Missionsabbruch sowie eine Auftauchempfehlung an das Missionsmanagement übergeben werden. Beide Entscheidungen resultieren aus der Diagnose der Fahrzeugsysteme. Die Auswahl der durchzuführenden Handlung erfolgt durch das Diagnosemodul abhängig von der Schwere der vorliegenden Störung.

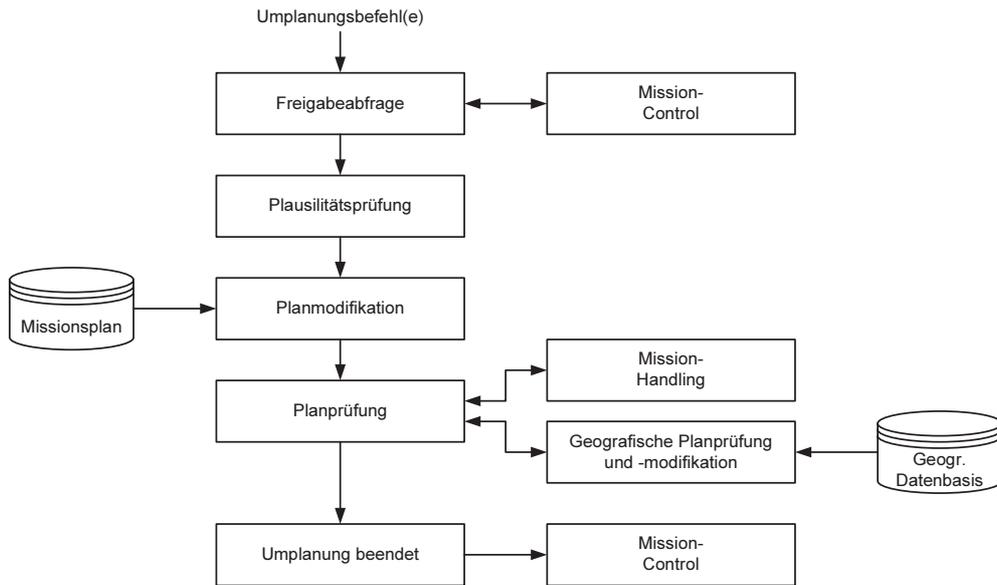
Das Nutzlastmodul ist ebenfalls in der Lage, einen begrenzten Einfluss auf den Missionsplan zu nehmen. Dazu sind während der Missionsplanung Event-Manöver mit ihrer Geometrie zu definieren, die durch das Missionsmanagement bei Bedarf parametrisiert und in den Plan aufgenommen werden. Während der Missionsdurchführung kann die Nutzlast nun in unterschiedlichen Situationen ein Event-Manöver anfordern. Ein Beispiel für ein derartiges Szenario ist die Durchführung einer Meeresbodenvermessung für das Verlegen von Seekabeln. Die dazu notwendige Sensorik erstellt laufend das zu den eingehenden Daten gehörige Profil des Bodens und kann damit auch auf eine Überschreitung definierter Grenzwerte für die Kabelverlegung (z.B. Neigungswinkel des Meeresbodens) reagieren. In einer derartigen Situation fordert die Nutzlast ein lokal begrenztes Manöver an der aktuellen Position an. Aus den zusätzlich gewonnenen Daten kann bei der Planung des Kabelverlaufs eine Umgehung des betroffenen Geländeabschnittes eingearbeitet werden.

Abweichungen vom Missionsplan<sup>1</sup> erfordern unter Umständen ebenfalls Modifikationen des Planes. Exemplarisch sind in Tabelle 7.5 die notwendigen Korrekturen für einen zu großen Navigationsfehler sowie eine vertikale Abweichung von der geplanten Fahrstrecke aufgeführt.

<sup>1</sup>z.B. infolge ungenauer Navigation oder schlechter Regelgüte (bedingt durch Strömung)

**Tabelle 7.5:** Umplanungssituationen für DeepC (Auszug)

Situation	Umplanungsaufgaben
<i>Defekte der Sensorik</i>	
Defekt der Nutzlast	Entfernung aller Manöver aus dem Plan, die die Nutzlast verwenden
Defekt des INS	Abbruch der Mission, wenn der Navigationsfehler zu groß wird (mit Überprüfung der Erreichbarkeit einer Recovery-Position)
Defekt des DOLOG	Sofortiger Abbruch der Mission mit Überprüfung der Erreichbarkeit einer Recovery-Position
Akustik-Modem ist defekt	Verschieben eines den Missionsplan abschließenden Circling-Manövers an die Wasseroberfläche
<i>Ressourcenbezogene Umplanungen</i>	
Mangel an Ressourcen (Energie)	Optimierung des Missionsplanes bei Unterschreiten der benötigten Ressourcen
Zu hohe Leistungsaufnahme	Anpassung der Fahrgeschwindigkeit der Manöver für eine begrenzte Zeit
<i>Anforderungen durch das Health-Monitoring</i>	
Unbedingter Missionsabbruch	Abbruch der Mission an der aktuellen Position, Einplanung der für den Abbruch benötigten Manöver
Auftauchempfehlung	Abbruch der Mission mit Überprüfung der Erreichbarkeit einer Recovery-Position
<i>Anforderungen durch weitere Module</i>	
Anforderung eines Event-Manövers	Einfügen des gewünschten Manövers in das aktuell ausgeführte Manöver
<i>Detektierte Abweichungen vom Missionsplan</i>	
Navigationsfehler zu groß	Einfügen eines GPS-Updates in das aktuelle Manöver
Zu große vertikale Trackablage	Einfügen eines Auftauch- / Abtauchmanövers von der aktuellen Tiefe auf die Endtiefe des aktuellen Manövers



**Abb. 7.19:** DeepC: Ablauf der Missionsumplanung

Einige der in der Tabelle aufgeführten Umplanungsaufgaben werden durch die Missionsüberwachung mit Hilfe der Umplanungsbeefehle bereits vollständig aufbereitet an die Missionsumplanung übergeben. Deren Tätigkeit besteht dann im Ausführen der einzelnen Umplanungsbeefehle und in der Prüfung des entstehenden Missionsplanes. In den anderen Fällen sind durch die Umplanung zunächst die durchzuführenden Modifikationen zu ermitteln, die anschließend realisiert werden.

Einen Teil der zu einer Umplanung führenden Datensätze empfängt das intelligente Missionsmanagement zyklisch. Damit eine bereits in Bearbeitung befindliche Umplanung nicht nochmals ausgelöst wird, muss der aktuelle Zustand der betroffenen Regeln des Expertensystems gespeichert werden. Das erfolgt über den einzelnen Regeln zugeordnete Fakten, die entsprechend Abschnitt 5.2.2.2 als Zustandspeicher verwendet werden.

### 7.2.2.3 Missionsumplanung

Die Missionsumplanung ist analog zu der des Kleinroboters *MauSI* für die Ausführung der notwendigen Veränderungen am Missionsplan und deren Prüfung verantwortlich. Aufgrund des gleichen Umfangs an Umplanungsbeefehlen ist auch der prinzipielle Aufbau weitestgehend identisch. Die interne Arbeitsweise unterscheidet sich im Wesentlichen durch die Einbindung des Missionsmanagements in die Softwarearchitektur des AUV, da alle durchzuführenden Aktivitäten einer Freigabe durch die zentrale Entscheidungsinstanz Mission-Control bedürfen. Zusätzlich ist der modifizierte Plan einer Kontrolle zu unterziehen, die sich aus der geografischen Planprüfung und -modifikation sowie einem Plausibilitäts-Check durch das Mission-Handling zusammensetzt. Erst wenn diese Prüfungen positiv abgeschlossen wurden, kann der neue Plan dem System zur Realisierung übergeben werden (Abbildung 7.19).

Die geografische Planprüfung erfordert entsprechend den Aussagen aus Abschnitt 6.4 eine digitale Karte des Einsatzgebietes des autonomen mobilen Systems. Diese Karte muss aus den verfügbaren Informationen generiert und in der von dem intelligenten Missionsmanage-



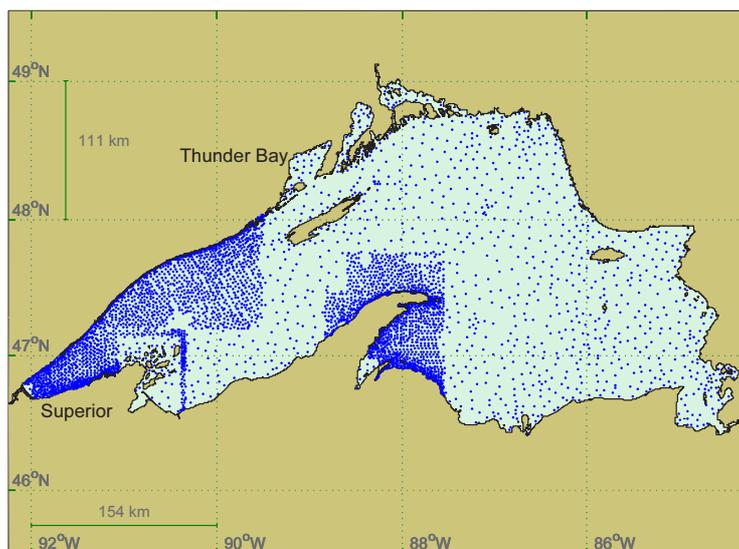
**Abb. 7.20:** Digitale Seekarte der Eckernförder Bucht

ment benötigten Struktur abgelegt werden.

Das im Forschungsprojekt vorgesehene erste Einsatzgebiet befindet sich in der Ostsee. Als Datenquelle für dieses Gebiet stehen digitale Höhenmodelle, häufig mit äquidistant angeordneten Stützstellen und digitale Navigationskarten für die Schifffahrt (Electronic Navigational Chart, ENC) zur Verfügung. Letztere beinhalten genaue Daten für die Navigation von Überwasserschiffen und besitzen ein standardisiertes Austauschformat [Int00]. In das zugrunde liegende objektorientierte Datenmodell sind alle für die Schifffahrt relevanten Informationen integriert. Die Datenbasis ist in Form von Zellen organisiert, die die gesamte Erdoberfläche in unterschiedlichen Genauigkeitsstufen abdecken. Jede Zelle enthält dabei die sie beschreibenden Objekte mit ihrer Geometrie und den zugehörigen Attributen. Abbildung 7.20 zeigt einen Ausschnitt aus einer Seekarte, der aus Gründen der Übersichtlichkeit lediglich die grundlegenden topografischen Elemente beinhaltet.

Digitale Seekarten garantieren eine hohe Genauigkeit der eingebetteten Daten. Sie sind jedoch durch die Ausrichtung auf die konventionelle Schifffahrt nur bedingt für die Unterwassernavigation einsetzbar, da der Großteil der enthaltenen Datensätze das Relief des Meeresbodens im Flachwasserbereich beschreibt. In Gegenden ohne jegliche Gefährdung für Schiffe ist demgegenüber die Informationsdichte verhältnismäßig gering. Abbildung 7.21 zeigt beispielsweise die unterschiedlichen Konzentrationen der Messpunkte für den Superior See (USA / Kanada). In der Nähe der bedeutenden Häfen des Sees steht eine exakte Vermessung des Reliefs des Bodens zur Verfügung, während der Großteil der Fläche des Sees nur mit größeren Abständen zwischen den Echolotungen hinterlegt wurde.

Für das vorgesehene Einsatzgebiet von DeepC sind die Daten für Bereiche geringer Informationsdichte aus einem digitalen Höhenmodell [STK01] entnommen und mit der Seekarte aus Abbildung 7.20 kombiniert worden. Die abschließende Triangulation beinhaltet gleichzeitig eine Datenreduktion, um die Anzahl der äquidistant angeordneten Stützstellen des verwen-



**Abb. 7.21:** Messpunkte der digitalen Navigationskarten des Superior Sees

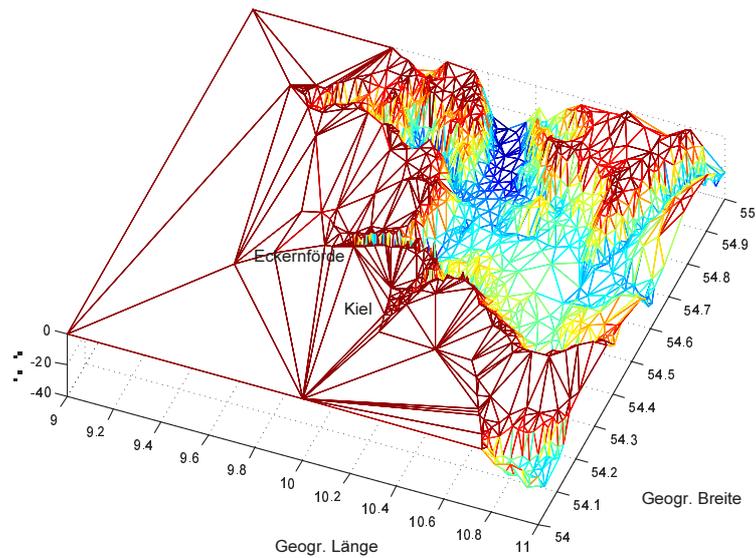
**Tabelle 7.6:** Daten des digitalen Höhenmodells

Kenngröße	Wert
West-Ost-Ausdehung	09°00,00'E bis 11°00,00'E (ca. 130 km)
Süd-Nord-Ausdehung	54°00,00'N bis 55°00,00'N (ca. 110 km)
Wassertiefe	0 - 33 m
Anzahl der Stützstellen	1775 = $n$
Anzahl der fahrbaren Wegstrecken	230793 = 14,6% der maximal möglichen Strecken $1/2 \cdot n \cdot (n - 1) = 1574425$

deten Höhenmodells an die jeweiligen Bodenverhältnisse anzupassen. Dabei werden über den Gradient des Meeresbodenprofils diejenigen Messpunkte entfernt, die keinen maßgeblichen Beitrag zur Abbildung des Geländes leisten [SMK95]. Als Resultat liegt der benötigte Bereich der Ostsee als trianguliertes Höhenmodell vor (Abbildung 7.22).

Die geografische Planmodifikation benötigt zusätzlich die Sichtbarkeits- oder auch Distanzmatrix der Höhenpunkte. Sie enthält Kostenbewertungen für die Strecken zwischen jeweils zwei Höhenpunkten, die bei einer Planprüfung als gültige Wege erkannt werden. Die Generierung des Graphen erfolgt entsprechend den Aussagen aus Abschnitt 6.4.3.1. In Tabelle 7.6 sind die wesentlichen Kenngrößen der resultierenden geografischen Datenbasis des intelligenten Missionsmanagements aufgeführt.

Bei der Vorstellung des Manöverkataloges wurde bereits erwähnt, dass bei der Planbereinigung bedingt durch die freie Bewegung im dreidimensionalen Raum und die Unterscheidung von drei unterschiedlichen Transitmanövern Erweiterungen der in Abschnitt 6.2.5 vorgestellten Vorgehensweise notwendig sind. Im Einzelnen sind folgende Einschränkungen bei der Bereinigung eines Missionsplanes zu berücksichtigen:



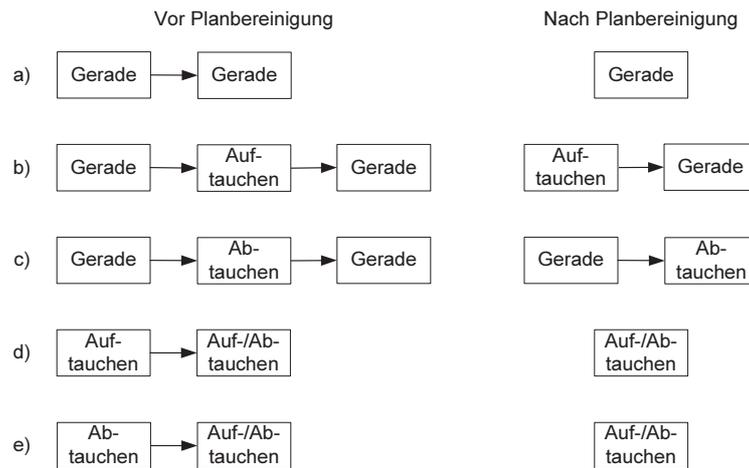
**Abb. 7.22:** Höhenmodell für die geografischen Umplanungsverfahren

- Manöver des Typs Gerade dürfen einen maximalen Neigungswinkel nicht überschreiten. Diese Begrenzung ergibt sich aus dem zulässigen Nickwinkel des AUV.
- Auf- und Abtauchmanöver gleichen nur Höhenunterschiede zwischen Anfangs- und Endposition aus, die Angaben zu Längen- und Breitengraden müssen stets identisch sein.

Die Planbereinigung selbst läuft als iterativer Prozess ab und fasst die folgenden Kombinationen von Transitmanövern zusammen (Abbildung 7.23):

- Zwei oder mehrere Transitmanöver des Typs Gerade werden zu einem Manöver gleichen Typs kombiniert.
- Ein zwischen zwei horizontalen Transitmanövern angeordnetes Auftauchmanöver wird vor dem kombinierten Manöver des Typs Gerade ausgeführt, um das AUV aus Sicherheitsgründen mit größerem Abstand zum Meeresboden fahren zu lassen.
- Analog zur vorangehenden Verfahrensweise erfolgt ein Abtauchen erst nach der Durchführung des horizontalen Transits.
- und e) Folgen vertikaler Transitmanöver werden abhängig von dem zu überwindenden resultierenden Höhenunterschied als einzelnes Abtauch- oder Auftauchmanöver ausgeführt.

Damit ist es möglich, vertikale Transitmanöver vollständig aus dem Plan zu entfernen. So wird bei der Planbereinigung geprüft, ob ein Höhenunterschied mit Hilfe eines bereits im Missionsplan vorhandenen horizontalen Transits unter Beachtung des maximalen Neigungswinkels überwunden werden kann. In diesem Fall wird das vorhandene Auf- oder Abtauchmanöver mit diesem Transit kombiniert, der dann auch den Ausgleich der vertikalen Positionsunterschiede zwischen Start- und Endpunkt vornimmt.



**Abb. 7.23:** Vorgehensweise bei der Planbereinigung

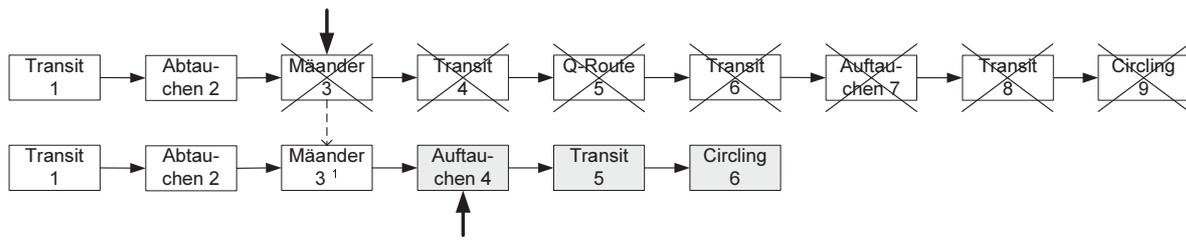
#### 7.2.2.4 Beispiele und Ergebnisse

Die in der Tabelle 7.5 aufgeführten Umplanungssituationen wurden beispielhaft mit dem Missionsplan aus Abbildung 7.18 untersucht. Dieser Plan befindet sich nicht in dem für die geografischen Planungsverfahren vorgesehenen Einsatzgebiet, da durch die in der Ostsee vorherrschende geringe Wassertiefe Einschränkungen bei den unterschiedlichen Umplanungen auftreten würden. Das genutzte Missionsgebiet besitzt demgegenüber eine ausreichende Wassertiefe, um auch mit Auftauch- und Abtauchmanövern operieren zu können. Für die Darstellung der Ergebnisse der geografischen Verfahren ist ein zweiter Missionsplan für das Gebiet der Eckernförder Bucht generiert worden, der am Ende dieses Abschnitts vorgestellt wird.

Die verschiedenen Umplanungssituationen werden hier nicht einzeln dargestellt, sondern entsprechend der Gruppierung aus Tabelle 7.5 zusammengefasst. Dadurch können ähnliche oder gleichartige Planmodifikationen gemeinsam erläutert werden.

*Defekte der Sensorik.* Ein Ausfall der Fahrzeugsensorik stellt unter Umständen einen kritischen Systemzustand dar. Insbesondere bei den primären Navigationssensoren ist deshalb ein Abbruch der aktuell gefahrenen Mission und, wenn der Missionsabbruch in einer vorgegebenen maximalen Entfernung um die geplanten Recovery-Positionen liegt, das Anfahren der nächsten dieser Positionen vorgesehen. Anhängig von der konkreten Situation sind unterschiedliche Manöver in den Missionsplan zu integrieren. In dem Beispiel aus Abbildung 7.24 sind nach dem Auftauchen ein Transit zu einer erreichbaren Recovery-Position und das abschließende Circling-Manöver als durchzuführende Folge von Planelementen erzeugt worden.

Beim Ausfall der Nutzlast sind alle darauf angewiesenen Manöver aus dem Missionsplan zu entfernen. Durch die sich an die eigentliche Umplanung anschließende Planbereinigung werden dann die vorhandenen Transitmanöver zusammengefasst. Sind im überarbeiteten Plan noch zu realisierende Nicht-Transit-Manöver vorhanden, so kann die Mission mit deren Bearbeitung fortgesetzt werden. Beinhaltet der Plan hingegen lediglich Transitmanöver, wird durch das intelligente Missionsmanagement versucht, den geplanten Missionsendpunkt



**Abb. 7.24:** Beispiel für die Umplanung im Fall defekter Sensorik

auf kürzestem Weg zu erreichen. Die Vorgehensweise entspricht damit der für den mobilen Kleinroboter *MauSI*.

*Ressourcenbezogene Umplanungen.* Die Optimierung des Missionsplanes ist die bedeutendste ressourcenbezogene Umplanung. Sie wird gestartet, wenn ein Defizit beim Vergleich der vorhandenen Energiereserven mit den benötigten Ressourcen der noch zu realisierenden Manöver entdeckt wird. Die Missionsüberwachung fordert daraufhin eine Optimierung des Planes an, die von der Missionsumplanung umgehend realisiert wird. Aufgrund der in Abschnitt 6.3.2 dargestellten Komplexität des Optimierungsproblems und dem damit verbundenen hohen Zeitbedarf muss jederzeit ein Abbruch des Algorithmus möglich sein, um auf anderweitige, höherpriorisierte Umplanungsanforderungen reagieren zu können. Die abgebrochene Optimierung wird im Anschluss an die durchgeführte Planmodifikation neu gestartet, wenn das Ressourcendefizit noch vorhanden ist.

Aufgrund der Komplexität des Problems wurde neben dem eigentlichen Optimierungsverfahren eine Heuristik entworfen, die insbesondere bei großen Manöverzahlen eine erhebliche Beschleunigung des Optimierungsprozesses erzielt. Für einen Vergleich beider Methoden sind umfangreichere Pläne durch zwei-, fünf- und zehnfache Aneinanderreihung der Manöver 1 bis 8 des Testplanes aus Abbildung 7.18 erzeugt worden. Alle Pläne werden mit dem abschließenden Kontaktmanöver *Circling 9* beendet. Die Wiederholung der Manöversequenz stellt hohe Anforderungen an die Methoden der Planbereinigung, da das Entfernen von den in größeren Tiefen ausgeführten Manövern *Mäander 3* und *Q-Route 5* auch zur Kombination typischer Sequenzen von Abtauch-, Transit- und Auftauchmanövern zu einem einzelnen horizontalen Transit führen kann.

Den Maßstab für die Bewertung stellen der unter einem bestimmten Energiedefizit erreichte relative Wert der Mission  $W_{rel}(\mathbf{x})$  bezogen auf den Wert des ursprünglichen Planes

$$W_{rel}(\mathbf{x}) = \frac{W(\mathbf{x})}{W(\mathbf{x}^*)} \quad (7.4)$$

$$\mathbf{x}_k^* = 1, \quad k = 1, \dots, n$$

sowie die Laufzeit der Optimierung, gemessen über die Anzahl der Schritte bis zum Auffinden der Lösung, dar. Die Ergebnisse beider Optimierungsmethoden sind in Tabelle 7.7 gegenübergestellt. Die vermerkte Anzahl der Nicht-Transit-Manöver stellt dabei gleichzeitig den Umfang der zu optimierenden Aktivierungswerte und damit die Zahl der Optimierungsvariablen dar.

**Tabelle 7.7:** Ergebnisse und relativer Zeitbedarf der Missionsplanoptimierung

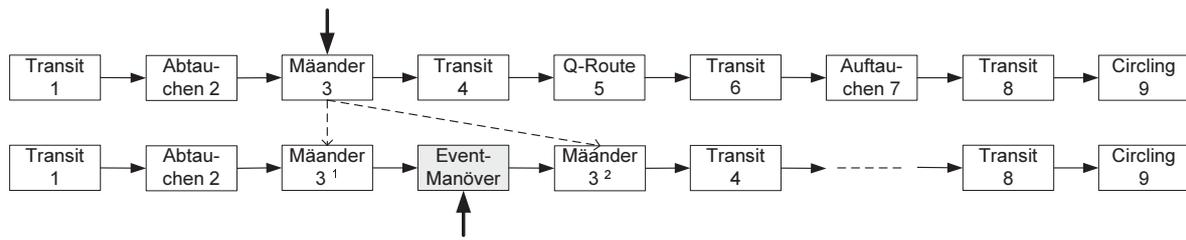
Missionsplan	Energie- defizit in %	Optimierungsverfahren		Heuristik	
		$W_{rel}(\mathbf{r})$ in %	Schritte	$W_{rel}(\mathbf{r})$ in %	Schritte
Einfach (9 Manöver, 2 Nicht-Transit)	10,0	78,5	2	78,5	2
	25,0	78,5	2	78,5	2
	50,0	78,5	2	78,5	2
	75,0	0,0 <sup>1</sup>	2	0,0 <sup>1</sup>	2
Zweifach (17 Manöver, 4 Nicht-Transit)	10,0	89,2	2	89,2	4
	25,0	89,2	2	89,2	4
	50,0	78,5	4	78,5	4
	75,0	39,2	11	39,2	4
Fünffach (41 Manöver, 10 Nicht-Transit)	10,0	95,7	2	95,7	10
	25,0	91,4	7	91,4	10
	50,0	87,1	20	87,1	10
	75,0	78,5	62	78,5	10
Zehnfach (81 Manöver, 20 Nicht-Transit)	10,0	95,7	11	95,7	20
	25,0	91,4	140	91,4	20
	50,0	84,9	1637	84,9	20
	75,0	78,5	12024	78,5	20

Im Vergleich der beiden Methoden wird deutlich, dass Heuristik und Optimierungsverfahren gleichwertige Lösungen liefern. Diese Aussage lässt sich jedoch nicht verallgemeinern, da die Greedy-Heuristik im Gegensatz zu dem Optimierungsverfahren keine optimale Lösung garantiert. Durch die Anwendung der in Abschnitt 6.3.2.3 vorgestellten relativen Wertigkeiten wird jedoch stets ein gutes Ergebnis erreicht.

In den Fällen mit einem geringen Energiedefizit kann durch das Optimierungsverfahren die Lösung mit einer kleineren Anzahl von Schritten gefunden werden, da bedingt aufgrund der Sortierung aller möglichen Manöverkonfigurationen die Lösung des Problems bereits nach wenigen Berechnungen vorliegt. Mit zunehmendem Defizit und damit auch mit zunehmendem Abstand der Lösung von der Startkonfiguration innerhalb der sortierten Liste erhöht sich der Berechnungsaufwand, bei großen Manöverzahlen sogar drastisch. Die Ursache dafür liegt in der bereits angesprochenen Komplexität des Problems.

Die Heuristik arbeitet im Gegensatz dazu mit einer der Anzahl an Optimierungsvariablen entsprechenden Schrittzahl. Bei steigender Manöverzahl oder größer werdendem Energiedefizit liegt die Laufzeit dieser Methode erheblich unter der des Optimierungsverfahrens. Die Qualität der Lösung, also der Wert des entstehenden Missionsplanes, ist dabei jedoch identisch oder geringfügig schlechter. Das Optimierungsverfahren erzielte in den Testbeispielen durch die Auswahl anderer Manöverkonfigurationen maximale Verbesserungen von

<sup>1</sup>Der Missionsplan wird an der aktuellen Position abgebrochen, deshalb besitzt der verbleibende Anteil des Planes den Wert 0,0.



**Abb. 7.25:** Einfügen eines Event-Manövers in den Missionsplan

ca.  $2 \cdot 10^{-5}$  Prozent gegenüber der Heuristik.

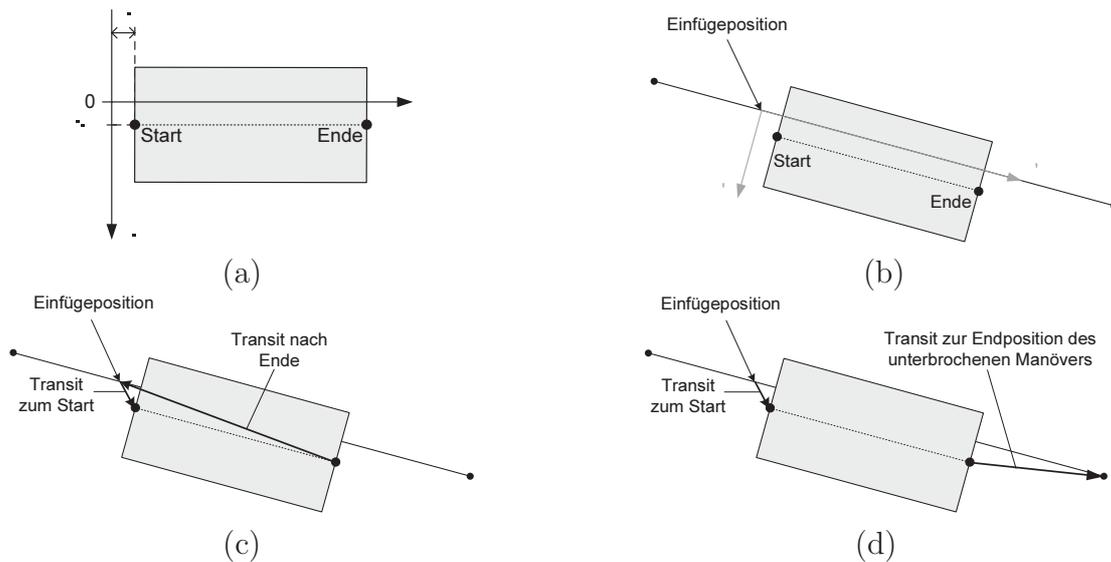
Die Umschaltung zwischen den beiden Methoden erfolgt für das AUV *DeepC* abhängig von der zu optimierenden Manöveranzahl. Damit ist garantiert, dass eine Lösung in jeder Situation in möglichst kurzer Zeit berechnet werden kann. Gleichzeitig erlaubt dieses Umschaltkriterium eine einfache Anpassung an die verfügbare Rechenleistung der in einem autonomen mobilen System eingesetzten Rechentechnik, da sie maßgeblich die Berechnungsdauer beeinflusst.

Liegt eine zu hohe Leistungsaufnahme des Gesamtsystems vor, wird durch das Health-Monitoring eine Reduktion der Fahrgeschwindigkeit angefordert. Das Regelwerk des integrierten Expertensystems der Missionsüberwachung ermittelt aus der gewünschten Dauer der Reduktion die Anzahl der anzupassenden Manöver und erzeugt eine Reihe von atomaren Befehlen zum Modifizieren der Manövergeschwindigkeiten. Die ursprünglich geplanten Geschwindigkeitswerte werden als Fakten im Expertensystem gesichert, um sie nach Ablauf oder Rücknahme der Reduktion wieder anwenden zu können. Laufende Reduktionen können jederzeit hinsichtlich ihrer Dauer und zulässiger Fahrgeschwindigkeit parametrisiert werden. Damit steht ein flexibles Instrument zur Reaktion auf kurzfristige Energieengpässe zur Verfügung.

*Anforderungen durch das Health-Monitoring.* Kritische Situationen, die durch Ausfälle fahrzeugeigener Module oder die vorherrschenden Umgebungsbedingungen auftreten, werden durch das Diagnosesystem des Moduls Health-Monitoring erkannt und bewertet. Da die Sicherheit des Fahrzeuges im Vordergrund aller Aktivitäten des Führungssystems steht, wird in der Regel ein Missionsabbruch kommandiert. Die Unterscheidung in einen unbedingten Abbruch und eine Auftauchempfehlung ermöglicht eine angemessene Reaktion in Abhängigkeit vom Grad der Gefährdung für das Fahrzeug.

*Anforderungen durch weitere Module.* Eine für die erfolgreiche Missionsdurchführung wesentliche Umplanungsaktion besteht in dem Einfügen von im Abschnitt 7.2.1.3 vorgestellten Event-Manövern. Diese Manöver werden bei der Missionsplanung erzeugt und bis auf die geografischen Positionen parametrisiert. Das intelligente Missionsmanagement passt die fehlenden Angaben an die zum Anforderungszeitpunkt existierenden Bedingungen an und fügt das Planelement an der aktuellen Position in den Missionplan ein (Abbildung 7.25).

Start- und Endposition können während der Planung mit relativen Verschiebungsvektoren belegt sein, die eine Verlegung des Event-Manövers bezogen auf die Einfügeposition erlauben (Abbildung 7.26 (a)). Beim Einfügen werden dann die absolute Position sowie die Ausrichtung des Manövers berechnet (Abbildung 7.26 (b)). Ist das Event-Manöver gegenüber der



**Abb. 7.26:** Berechnung der absoluten Koordinaten bei Event-Manövern ((a) und (b)), mögliche Anpassungen des Plans für Event-Manöver ((c) und (d))

eigentlichen Einfügeposition verschoben, so hat zunächst ein Transit an die Startposition zu erfolgen (Abbildung 7.26 (c)). Analog dazu muss nach dem Ende des eingefügten Planelementes ein Transit zurück an die Einfügeposition durchgeführt werden. Ist das unterbrochene Manöver selbst ein Transit, so kann anstelle des Re-Transits zur Einfügeposition sofort die Endposition des unterbrochenen Manövers angefahren werden (Abbildung 7.26 (d)).

*Detektierte Abweichungen vom Missionsplan.* Im Fall eines zu großen Navigationsfehlers wird durch die Missionsüberwachung das Einfügen eines GPS-Updates in den Missionsplan angefordert. Die auszuführenden Schritte bei der Missionsumplanung entsprechen den in der Abbildung 7.26 dargestellten Maßnahmen beim Einplanen eines Event-Manövers. Dabei wird das aktuelle Manöver unterbrochen, das GPS-Update eingefügt und anschließend der verbleibende Teil des noch nicht vollständig realisierten Manövers fortgesetzt.

Wird eine zu große vertikale Entfernung des AUV von der geplanten Fahrstrecke erkannt, muss ein Tiefenausgleich mit Hilfe eines vertikalen Transits durchgeführt werden. Die zugehörige Anforderung wird durch den Autopiloten kurz vor Beendigung eines Manövers erzeugt, wenn die geplante Endtiefe nicht erreicht werden kann. Die Missionsumplanung generiert daraufhin je nach Notwendigkeit ein Auf- oder Abtauchmanöver im Anschluss an das aktuelle Planelement.

*Geografische Planprüfung und -modifikation.* Die geografische Planprüfung wird im Anschluss an jede Umplanung durchgeführt, um eine Gefährdung des AUV durch die Modifikation des Planes auszuschließen. Als Beispiel zur Darstellung der Arbeitsweise der Planprüfung und einer sich im Fall auftretender Kollisionen anschließenden geografischen Planmodifikation wurde ein Missionsplan erstellt, der in der Eckernförder Bucht beginnt und in die offene See führt (Abbildung 7.27). Bei einer angenommenen Entfernung des zweiten Manövers *Q-Route 2* und der sich aus der anschließenden Planbereinigung ergebenden Zusammenfassung mit dem Manöver *Transit 1* entsteht eine nicht fahrbare Route für das



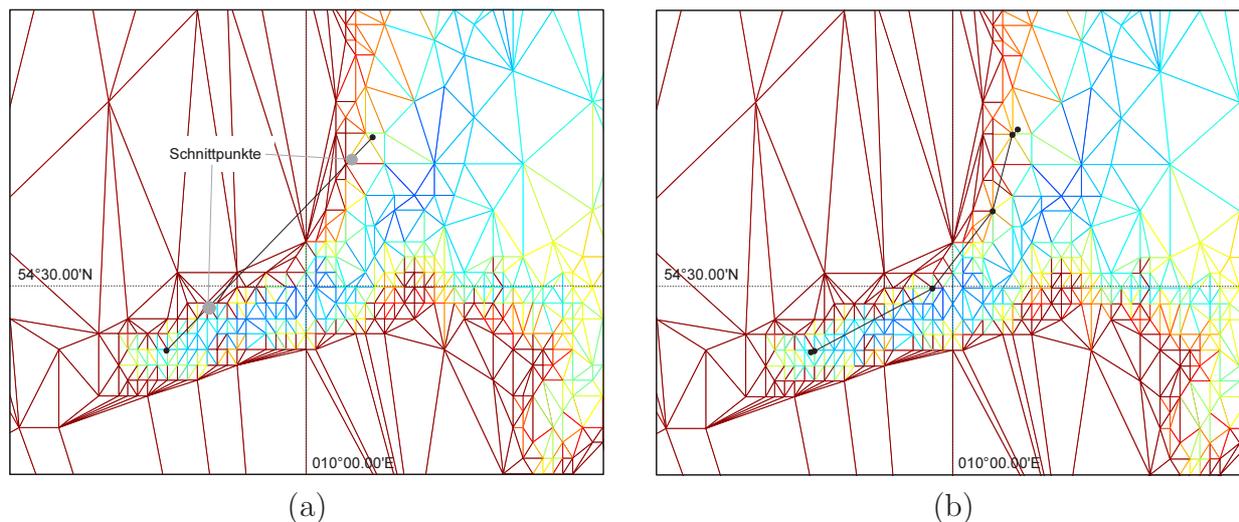
**Abb. 7.27:** Missionsplan für den Test der geografischen Planungsverfahren

AUV. Bei der geografischen Planprüfung werden dann zwei Schnittpunkte mit dem digitalen Höhenmodell festgestellt (Abbildung 7.28 (a)).

In einer derartigen Situation kann mit Hilfe der geografischen Planmodifikation eine fahrbare Route für diese Strecke berechnet werden. Das Ergebnis ist in Abbildung 7.28 (b) dargestellt. Zwischen dem eigentlichen Start- und Endpunkt des nicht fahrbaren Transits wird eine Folge von Stützstellen des digitalen Höhenmodells eingefügt, um eine manövrierbare Strecke zu generieren. Der ursprünglich vorhandene Transit wird dann durch eine Sequenz von Transit-Manövern zwischen den einzelnen Positionen entsprechend Abbildung 6.21 ersetzt und dieser gültige Missionsplan in die Datenbank aus Abbildung 7.16 eingetragen. Alle den Plan nutzenden Module der Fahrzeugsoftware arbeiten ab diesem Zeitpunkt mit dem aktualisierten Missionsplan.

Die Verfahren zur geografischen Planprüfung und -modifikation sind auch während der Missionsplanung einsetzbar, um den Operator bei seiner Tätigkeit zu unterstützen. So kann innerhalb der Planungsphase eines Einsatzes mit Hilfe der Planmodifikation eine automatische Wegplanung durchgeführt werden, um eine sichere Route durch schwieriges Gelände zu generieren. Die Aufgabe des Operators beschränkt sich dabei auf die Vorgabe von Start- und Endposition des Manövers.

*Weitere Umplanungssituationen.* Neben den ausführlich dargestellten Umplanungen sind verschiedene weitere Bedingungen definiert, die eine Modifikation des Missionsplanes erfordern. Teilweise sind diese trivial oder können analog zu einer der geschilderten Vorgehensweisen ausgeführt werden. Deshalb soll auf eine Beschreibung derartiger Umplanungssituationen verzichtet werden.



**Abb. 7.28:** Beispiel für die geografische Planprüfung (a) und -modifikation (b)

### 7.3 Zusammenfassung

In diesem Kapitel wurde die Anwendung des intelligenten Missionsmanagements für zwei unterschiedliche mobile Systeme dargestellt. Beide Systeme verwenden die als Grundlage des Gesamtkonzeptes festgelegte Softwarearchitektur, die eine nahezu transparente Integration des Missionsmanagements in die Menge bereits bestehender Softwaremodule erlaubt.

Diese Architektur wurde für den mobilen Kleinroboter *MauSI* neu aufgebaut und mit einer Basismenge an Umplanungsregeln ausgestattet. Der Roboter ist damit in der Lage, längerfristige Missionen mit komplexem Charakter auszuführen. Die gezeigten Beispiele zur Anpassung des Missionsplanes demonstrieren die Anwendbarkeit auf einem erdgebundenen System.

Der Großteil der Ausführungen bezog sich auf das autonome Unterwasserfahrzeug *DeepC*, das mit der vorgestellten Funktionalität eine neue Klasse von AUVs begründet. Gerade die Adaption von Missionsplänen ist in den derzeit vorhandenen existierenden Unterwasserfahrzeugen noch nicht vorhanden. Aufgrund der Vielfalt möglicher Umplanungsszenarien konnte mit den gezeigten Beispielen nur exemplarisch auf die wesentlichsten Methoden näher eingegangen werden. Die ausgewählten Situationen offenbaren jedoch die enorme Flexibilität des Konzeptes des intelligenten Missionsmanagements.

Die Optimierung des Missionsplanes stellt einen Schwerpunkt der entworfenen Verfahren dar. Durch die Implementierung zweier unterschiedlicher Methoden ist ein Einsatz direkt auf dem AUV ermöglicht worden. Der Vergleich von globaler Optimierung und Greedy-Heuristik zeigt die Leistungsfähigkeit des Näherungsverfahrens, wenn die Rahmenbedingungen sinnvoll gewählt werden.

Die geografischen Planungsverfahren garantieren schließlich die Fahrbarkeit der neu erstellten oder angepassten Manöver. Dazu ist neben der Prüfung eines Missionsplanes auch die Bestimmung eines navigierbaren Weges notwendig. Anhand eines im späteren Einsatzgebiet des AUV *DeepC* gelegenen Missionsplanes konnte die prinzipielle Vorgehensweise der entworfenen Algorithmen und der Aufbau der Datenbasis dargestellt werden.



## 8 Zusammenfassung und Ausblick

### 8.1 Zusammenfassung

Das intelligente Management der globalen Missionsziele eines autonomen mobilen Systems ist ein wichtiger Bestandteil seiner Softwarearchitektur. Während für erdgebundene Roboter vielfältige Ansätze zur Lösung dieser Aufgabe zu finden sind, fehlen insbesondere bei fliegenden und tauchenden mobilen Systemen häufig derartige Verfahren. Als wesentliche Ursachen dürften die fehlenden manuellen Eingriffsmöglichkeiten des Bedienpersonals und das hohe Gefahrenpotential im Fall von Fehlfunktionen gelten.

In der vorliegenden Arbeit wurde, ausgehend von den Anforderungen für ein innovatives autonomes Unterwasserfahrzeug, ein neuartiger Ansatz für das intelligente Missionsmanagement entwickelt, der auch auf andere mobile Systeme übertragbar ist. Ausgangspunkt war dabei die Festlegung eines Missionsplanes als Schnittstelle für das Missionsmanagement, der als lineare Sequenz von komplexen Planelementen aufgebaut ist.

Die Recherche des internationalen Standes zur Anwendung intelligenter Verfahren für das Management der Mission ergab, dass weltweit eine Vielzahl unterschiedlichster Methoden zum Einsatz kommen. In einer Analyse wurden typische Vertreter ausgewählt, systematisiert sowie ihre Arbeitsweise beschrieben und bewertet.

Ein wesentlicher Aspekt für den Entwurf des intelligenten Missionsmanagements ist der Aufbau der Softwarearchitektur des autonomen mobilen Systems. Sie beeinflusst gleichermaßen die Schnittstellen und die Entscheidung für die einzusetzenden Verfahren. Es konnte gezeigt werden, dass sowohl die in der vorliegenden Arbeit verwendeten als auch viele in der Literatur vorgefundenen Architekturen auf ein Handlungsmodell des Menschen nach Rasmussen zurückgeführt werden können. Die Unterteilung der Softwaremodule in drei hierarchisch angeordnete Ebenen, die mit unterschiedlichen Taktzyklen voneinander abgegrenzte Aufgaben zu erfüllen haben, ist damit ein geeigneter Ansatz für autonom agierende, mobile Roboter.

Das intelligente Missionsmanagement wurde als Kombination einer wissensbasierten Missionsüberwachung und eines Moduls zur Missionsumplanung konzipiert. Als Schnittstelle zwischen diesen beiden Elementen sind Umplanungsbefehle definiert, die sich in atomare und komplexe Befehle unterteilen lassen. Durch den universellen Charakter dieser Befehle ist ein Einsatz in unterschiedlichen mobilen Systemen ermöglicht worden.

Ausgangspunkt einer Mission aus dem Blickwinkel des Missionsmanagements ist die Missionsplanung. In der vorliegenden Arbeit wurde die manuelle Planung durch einen Operator gegenüber automatischen Planungsverfahren favorisiert. Neben der Anwendbarkeit der vorgestellten unterschiedlichen Planungskonzepte in verschiedensten Einsatzgebieten der mobilen Robotik ist damit auch die Frage nach der Verantwortung für die Handlungen des autonomen Roboters eng verbunden.

Die wissensbasierte Missionsüberwachung bildet den intelligenten Bestandteil des Missionsmanagements. Sie beinhaltet ein regelbasiertes Expertensystem, das eine Bewertung der

unterschiedlichen Eingangsdaten vornimmt. Dabei fließen sowohl direkt aufgenommene und aufbereitete Sensormesswerte als auch abgeleitete Größen beispielsweise von der Fahrzeugdiagnose ein. Als Ergebnis liefern die Regeln des Expertensystems die Information, ob eine Umplanung notwendig ist, und die dazu erforderlichen Umplanungsbefehle. Dieses Konzept erleichtert zukünftige Erweiterungen des Aufgabenfeldes des Systems, da für die Erkennung und Bewältigung zusätzlicher Umplanungssituationen lediglich die Anpassung der Wissensbasis des Expertensystems notwendig ist.

Die Umplanungsbefehle werden anschließend von der Missionsumplanung realisiert. Dazu wurden Methoden für die atomaren und komplexen Befehle entworfen. Sie gestatten nahezu beliebige Veränderungen an dem sequentiellen Plan. Die Planbereinigung detektiert und repariert Folgen von Füllplanelementen, die durch die Modifikationen des Missionsplanes entstehen. Gerade für mobile Systeme, die sich frei im dreidimensionalen Raum bewegen können, ist das ein bedeutendes Verfahren zur Vermeidung unnötiger Fahrstrecken. Ebenso wichtig ist die rechtzeitige Reaktion auf einen Ressourcenmangel. Die dafür verantwortliche Optimierung des Missionsplanes erfordert aufgrund der Komplexität des Problems die Entwicklung unterschiedlicher Methoden. So wurde ein Verfahren zum Auffinden der globalen Lösung des diskreten nichtlinearen Optimierungsproblems entwickelt, das bei einer geringen Anzahl an Optimierungsvariablen eine hohe Arbeitsgeschwindigkeit aufweist. Mit einer größer werdenden Menge an Variablen ist dieses Verhalten jedoch nicht mehr gewährleistet, sodass dann eine effiziente Greedy-Heuristik angewendet wird. Deren Entwurf basiert auf Analogiebetrachtungen zum so genannten Rucksack-Problem, einer intensiv untersuchten Aufgabenstellung der kombinatorischen Optimierung.

Im Anschluss an eine Modifikation des Missionsplanes wird dessen Fahrbarkeit mit Hilfe der geografischen Planprüfung kontrolliert. Dazu sind eine auf den onlinefähigen Einsatz optimierte Datenbasis des Einsatzgebietes und die darauf arbeitenden Algorithmen erforderlich. Beides wurde unter Nutzung effizienter Methoden neu entwickelt. Ergebnis der Planprüfung ist die Aussage, ob der Missionsplan realisierbar ist. Im Fehlerfall werden auch die betroffenen Planelemente und die Schnittpunkte mit dem Gelände zurückgeliefert. Die geografische Planmodifikation nutzt diese Informationen, um alternative Routen für die nicht fahrbaren Manöver zu generieren. Dafür werden graphenbasierte Verfahren und eine dementsprechende Datenstruktur eingesetzt.

Das intelligente Missionsmanagement wurde auf zwei vollkommen unterschiedlichen mobilen Systemen implementiert. Der mobile Kleinroboter *MauSI* gehört zur Klasse der erdgebundenen Roboter. Hier war es notwendig, eine Softwarearchitektur entsprechend des Handlungsmodells nach Rasmussen auf einem Leitrechner neu zu implementieren. Mit Hilfe einzelner Beispiele konnte die Arbeitsweise des intelligenten Missionsmanagements gezeigt werden.

Für die zweite Testplattform, das autonome Unterwasserfahrzeug *DeepC*, wurde eine umfangreiche Menge an Umplanungssituationen definiert. Hier kommt das volle Spektrum der entwickelten Methoden zum Einsatz. So ist für die geografischen Prüf- und Umplanungsverfahren eine Datenbasis des ersten Erprobungsgebietes des AUV hinterlegt und getestet worden. Anhand einiger ausgewählter Experimente konnte die Anwendung der Umplanungsbefehle und die Auswirkungen der zugehörigen Umplanungsalgorithmen auf den Missionsplan demonstriert werden. Ein durchgeführter Vergleich der entworfenen Verfahren zur Planoptimierung rundet die Beschreibung der Anwendungsbeispiele ab.

Die konzipierte Struktur für das intelligente Missionsmanagement autonomer mobiler Systeme konnte auf den beiden verfügbaren Prototypen erfolgreich umgesetzt werden. Die Anwendung gestaltet sich nahezu transparent für den Operator, der die Planung des Einsatzes vornimmt. Durch die konsequente Ausrichtung aller entwickelten Verfahren auf den Online-Einsatz auf einem Roboter ist eine effiziente Vorgehensweise bei der Umplanung einer Mission gewährleistet.

## 8.2 Ausblick

Die im Rahmen der vorliegenden Arbeit entwickelte Struktur für das intelligente Missionsmanagement ist in vielfältiger Weise erweiterbar. Durch fortlaufende Tests mit den beiden Prototypen ist ein Ausbau der Wissensbasis der Missionsüberwachung zur Erkennung und Behandlung zusätzlicher Umplanungssituationen möglich. Dieser Prozess kann auch während des späteren produktiven Einsatzes des AUV *DeepC* fortgesetzt werden und zu einer Spezialisierung einzelner Fahrzeuge auf ihre typischen Missionen führen.

Darüber hinaus kann das intelligente Missionsmanagement an andere Klassen mobiler Systeme angepasst werden, um weitere Erfahrungen sammeln zu können. Das dabei zu implementierende Wissen ergänzt die für die beiden Prototypen erstellte Wissensbasis um neue Situationen, die mit Hilfe einer Umplanung zu bewältigen sind. Die Erweiterung der Menge an vorhandenen Umplanungsbefehlen um komplexe Anweisungen, die den Notwendigkeiten der jeweiligen Plattform Rechnung tragen, sollte in zukünftigen Projekten ebenfalls untersucht werden. Dabei gilt es zu beachten, dass der universelle Charakter der Befehle erhalten bleibt.

Ein sehr interessanter Aspekt zukünftiger Arbeiten ist in der Untersuchung der kooperativen Handlungsweise mehrerer autonomer Roboter zu sehen. Diese Thematik ist von hoher Aktualität und gehört zu den sehr intensiv untersuchten Gebieten der Robotik. Ideen und Ansätze für ein teamfähiges Arbeiten sind mittlerweile für unterschiedliche Architekturen mobiler Systeme zu finden, zum Beispiel in [SSD<sup>+</sup>02]. Eine Erweiterung des intelligenten Missionsmanagements um ein wissensbasiertes, koordinierendes Modul und die Synchronisation der Missionspläne der verschiedenen mobilen Systeme kann das in dieser Arbeit vorgestellte Konzept auch für den koordinierten Einsatz einer Gruppe von Robotern befähigen und zu einer neuartigen Betrachtungsweise der Problematik führen.



## Literaturverzeichnis

- [AA01] ALTENBURG, J. ; ALTENBURG, U.: Roboter mit Köpfchen - Modell eines autonomen Systems mit variabler Intelligenz (MauSI). In: *Elektronik* (2001), Nr. 22, S. 54–58
- [ACE99] ARKIN, R. C. ; COLLINS, T. R. ; ENDO, Y.: Tactical Mobile Robot Mission Specification and Execution. In: *Proceedings for SPIE: Mobile Robots XIV* 3838 (1999), S. 150–163
- [ACG<sup>+</sup>99] AUSIELLO, G. ; CRESCENZI, P. ; GAMBOSI, G. ; KANN, V. ; MARCHETTI-SPACCAMELA, A. ; PROTASI, M.: *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Springer Verlag, Berlin, Heidelberg, 1999
- [BCF<sup>+</sup>98] BURGARD, W. ; CREMERS, A. B. ; FOX, D. ; HÄHNEL, D. ; LAKEMEYER, G. ; SCHULZ, D. ; STEINER, W. ; THRUN, S.: The Interactive Museum Tour-Guide Robot. In: *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, 1998
- [Bee02] BEETZ, M.: *Plan-Based Control of Robotic Agents*. Springer Verlag, Berlin, Heidelberg, 2002
- [BG93] BOMZE, I. M. ; GROSSMANN, W.: *Optimierung - Theorie und Algorithmen*. BI-Wissenschaftsverlag, Mannheim, Wien, Zürich, 1993
- [BHMM97] BRUTZMAN, D. ; HEALEY, T. ; MARCO, D. ; MCGHEE, B.: The Phoenix Autonomous Underwater Vehicle. In: *AI-Based Mobile Robots*. MIT/AAFI Press, 1997
- [BKI00] BEIERLE, C. ; KERN-ISBERNER, G.: *Methoden wissensbasierter Systeme*. Vieweg Verlagsgesellschaft mbH, Braunschweig, Wiesbaden, 2000
- [BL99] BARROUIL, C. ; LEMAIRE, J.: Advanced Real-Time Mission Management for an AUV. In: *NATO Symposium on Advanced Mission Management and System Integration Technologies for Improved Tactical Operations*. Florence, Italy, sep 1999
- [BNP<sup>+</sup>03] BACKES, P. G. ; NORRIS, J. S. ; POWEEL, M. W. ; VONA, M. A. ; WICK, J. V.: *Maestro User's Guide*. NASA (National Aeronautics and Space Administration), 2003
- [Boy91] BOY, G.: *Intelligent Assistant Systems*. Academic Press Ltd., London, 1991

- [BP94] BLAASVAER, H. ; PIRJANIAN, P.: *AMOR: An Autonomous Mobile Robot Navigation System*, Laboratory of Image Analysis, Aalborg University, Denmark, Diplomarbeit, 1994
- [BP98] BEETZ, M. ; PETERS, H.: Structured Reactive Communication Plans - Integrating Conversational Actions into High-level Robot Control Systems. In: *Proceedings of the 22nd German Conference on Artificial Intelligence (KI 98)*, 1998, S. 177–188
- [BPC94] BLAASVAER, H. ; PIRJANIAN, P. ; CHRISTENSEN, H. I.: AMOR: An Autonomous Mobile Robot Navigation System. In: *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics* Bd. 3, 1994, S. 2266–2271
- [BPRW97] BRETSCHNEIDER, P. ; PFÜTZENREUTER, T. ; RAUSCHENBACH, T. ; WERNSTEDT, J.: DIMOVOR-Expert - ein modulares System zur Lösung von Diagnose-, Monitoring- und Prognoseaufgaben. In: *Tagungsband des 42. Internationalen Wissenschaftlichen Kolloquiums, Band 3, Ilmenau*, 1997
- [Bru94] BRUTZMAN, D. P.: *A Virtual World for an Autonomous Underwater Vehicle*, Naval Postgraduate School, Monterey, USA, Dissertation, 1994
- [Brü02] BRÜCKNER, I.: *Simulation von Bewegung und Steuerung des autonomen mobilen Robotersystems MauSI 2 in virtuellen Räumen*, Technische Universität Ilmenau, Diplomarbeit, 2002
- [BS85] BUCHANAN, B. G. ; SHORTLIFFE, E. H.: *Rule-based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley, 1985
- [BW87] BÖHME, D. ; WERNSTEDT, J.: Entwurfskonzepte für Beratungssysteme zur Lösung kybernetischer Aufgaben. In: *messen - steuern - regeln (msr)* 30 (1987), 12, S. 535–539
- [Dev98] DEVILLERS, O.: Improved Incremental Randomized Delaunay Triangulation. In: *Proceedings of the Fourteenth Annual Symposium on Computational Geometry*, 1998, S. 106–115
- [Dij59] DIJKSTRA, E. W.: A Note on Two Problems in Connexion with Graphs. In: *Numerische Mathematik, Mathematisch Centrum, Amsterdam* 1 (1959), S. 269–271
- [Dom95] DOMSCHKE, W.: *Einführung in Operations Research*. Springer-Verlag Berlin, Heidelberg, New York, 1995
- [Doo95] DOORENBOS, R.: *Production Matching for Large Learning Systems*, Carnegie Mellon University, Pittsburgh, USA, Dissertation, 1995
- [Dun95] DUNG, L. T.: *Entwurf und Realisierung von echtzeitfähigen Inferenztechniken für wissensbasierte Systeme*, Technische Universität Ilmenau, Dissertation, 1995

- [Edl97] EDLINGER, Thomas: *Hierarchische Steuerung für einen mobilen Roboter zur autonomen Erkundung seiner Einsatzumgebung*. VDI Verlag, Düsseldorf, 1997
- [Eic04] EICHHORN, M.: An Obstacle Avoidance System for an Autonomous Underwater Vehicle. In: *Proceedings of the 2004 International Symposium on Underwater Technology, Taipei, Taiwan, 2004*
- [EMA04] ENDO, Y. ; MACKENZIE, D C. ; ARKIN, R. C.: Usability Evaluation of High-Level User Assistance for Robot Mission Specification. In: *Special Issue on Human-Robot Interaction, Transactions on Systems, Man and Cybernetics Part C, Vorabdruck*. Institute of Electrical and Electronics Engineers (IEEE), 2004
- [EPSW01] EICHHORN, M. ; PFÜTZENREUTER, T. ; SCHMIDT, M. ; WALTL, B.: Modellierung, Regelung und Navigation von teilautonomen, ferngesteuerten Unterwasserfahrzeugen. In: *at - Automatisierungstechnik* (2001), 8, S. 373–380
- [EPWZ00] EICHHORN, M. ; PFÜTZENREUTER, T. ; WALTL, B. ; ZERBE, V.: Manuelle Führung von Unterwasserfahrzeugen in virtuellen Welten. In: *Tagungsband des 45. Internationalen Wissenschaftlichen Kolloquiums, Ilmenau, 2000*
- [Fer02] FERRER, G. J.: *Anytime Replanning Using Local Subplan Replacement*, University of Virginia, Dissertation, 2002
- [For82] FORGY, C. L.: Rete: A Fast Algorithm for the Many Pattern / Many Object Pattern Match Problem. In: *Artificial Intelligence* 19 (1982), S. 17–37
- [For94] FORGY, C. L.: RAL/C and RAL/C++: Rule-Based Extensions to C and C++. In: *OOPSLA Workshop*, 1994
- [GJ79] GAREY, M. R. ; JOHNSON, D. S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979
- [Gow97] GOWDY, J.: SAUSAGES: Between Planning and Action. In: HEBERT, M. (Hrsg.) ; THORPE, C. (Hrsg.) ; STENTZ, A. (Hrsg.): *Intelligent Unmanned Ground Vehicles: Autonomous Navigation Research at Carnegie Mellon*. Kluwer Academic Publishers, Norwell, USA, 1997, S. 33–52
- [GR98] GIARRATANO, J. ; RILEY, G.: *Expert Systems - Principles and Programming*. PWS Publishing Company, Boston, USA, 1998
- [Gro02] GROSSKREUTZ, H.: *Towards More Realistic Logic-Based Robot Controllers in the GOLOG Framework*, Rheinisch-Westfälische Technische Hochschule Aachen, Diss., 2002
- [GRT93] GOERING, H. ; ROOS, H.-G. ; TOBISKA, L.: *Finite-Element-Methode*. Akademie Verlag, Berlin, 1993
- [Gul95] GULDNER, J.: *Intelligentes hierarchisches Regelungskonzept für autonome mobile Robotersysteme: globale Planung - lokale Kollisionsvermeidung - robuste Bewegungsregelung*. VDI-Verlag, Düsseldorf, 1995

- [Hau00] HAUN, M.: *Wissensbasierte Systeme*. expert verlag, Renningen, 2000
- [Hel66] HELMER, O.: *50 Jahre Zukunft*. Hamburg, 1966
- [Her89] HERTZBERG, J.: *Planen: Einführung in die Planerstellungsmethoden der künstlichen Intelligenz*. BI-Wissenschaftsverlag, Mannheim, Wien, Zürich, 1989
- [HGM02] HAKE, G. ; GRÜNREICH, D. ; MENG, L.: *Kartographie*. Walter de Gruyter, Berlin, New York, 2002
- [Hin02] v. HINÜBER, E.: New Approaches in High-Performance Navigation Solutions for AUVs and ROVs / iMAR GmbH. 2002. – Übersichtsaufsatz
- [HJM93] HANSEN, P. ; JAUMARD, B. ; MATHON, V.: Constraint Nonlinear 0-1 Programming. In: *Journal on Computing, Operations Research Society of America* 5 (1993), Nr. 2, S. 97–119
- [HMO<sup>+</sup>96] HEALEY, A. J. ; MARCO, D. B. ; OLIVEIRA, P. ; PASCOAL, A. ; SILVA, V. ; SILVESTRE, C.: Strategic Level Mission Control - An Evaluation of CORAL and PROLOG Implementations for Mission Control Specifications. In: *Proceedings of the 1996 IEEE Symposium on Autonomous Underwater Vehicle Technology*, 1996
- [Hol95] HOLDEN, M. J.: *ADA Implementation of Concurrent Execution of Multiple Tasks in the Strategic and Tactical Levels of the Rational Behavior Model for the NPS Phoenix AUV*, Naval Postgraduate School, Monterey, USA, Diplomarbeit, 1995
- [HRWL83] HAYES-ROTH, F. ; WATERMAN, D. A. ; LENAT, D. B.: *Building Expert Systems*. Addison-Wesley Publishing Company, Inc., Reading, USA, 1983
- [Int00] INTERNATIONAL HYDROGRAPHIC ORGANIZATION: *IHO Transfer Standard for Digital Hydrographic Data - Publication S-57*. International Hydrographic Bureau, 2000
- [Kle01] KLETZ, Trevor: *An Engineer's View of Human Error*. Institution of Chemical Engineers, Rugby, UK, 2001
- [Kni91] KNIERIEMEN, T.: *Autonome Mobile Roboter - Sensordateninterpretation und Weltmodellierung zur Navigation in unbekannter Umgebung*. BI-Wissenschaftsverlag, Mannheim, Wien, Zürich, 1991
- [Koc03a] KOCH, D.: *Realisierung einer Mehrschicht-Softwarearchitektur für mobile Kleinroboter*, Technische Universität Ilmenau, Diplomarbeit, 2003
- [Koc03b] KOCH, D.: *Realisierung eines Missionsplan Editors unter MATLAB*, Technische Universität Ilmenau, Studienjahresarbeit, 2003
- [LA03] LENNON, J. ; ATKINS, E.: A Rule-based Strategy for Astronaut Following Operations. In: *Technical Report of AAI Spring Symposium on Human Interaction with Autonomous Systems in Complex Environments*, 2003, S. 123–128

- [Lag96] LAGOUDAKIS, M. G.: The 0-1 Knapsack Problem, An Introductory Survey / Duke University, Department of Computer Science. 1996. – Forschungsbericht
- [Leo96] LEONHARDT, B.: *Mission Planung and Mission Control Software for the Phoenix AUV*, Naval Postgraduate School, Monterey, USA, Diplomarbeit, 1996
- [LRL<sup>+</sup>97] LEVESQUE, H. J. ; REITER, R. ; LESPERANCE, Y. ; LIN, F. ; SCHERL, R. B.: GOLOG: A Logic Programming Language for Dynamic Domains. In: *Journal of Logic Programming* 31 (1997), Nr. 1-3, S. 59–83
- [Moh00] MOHAN, C. K.: *Frontiers of Expert Systems: Reasoning with Limited Knowledge*. Kluwer Academic Publishers, Norwell, USA, 2000
- [MT90] MARTELLO, S. ; TOTH, P.: *Knapsack problems: algorithms and computer implementations*. John Wiley and Sons, New York, Chichester, Brisbane, Toronto, 1990
- [MT97] MÖLLER, Tomas ; TRUMBORE, Ben: Fast, Minimum Storage Ray-Triangle Intersection. In: *Journal of Graphics Tools* 2 (1997), Nr. 1, S. 21–28. – ISSN 1086–7651
- [Mur95] MURTY, K. G.: *Operations research: deterministic optimization models*. Prentice-Hall, Englewood Cliffs, 1995
- [NAS04] NASA (NATIONAL AERONAUTICS AND SPACE ADMINISTRATION). *Mars Exploration Rover Landings, Press Kit*. 2004
- [Nik97] NIKOLOPOULOS, C.: *Expert Systems: Introduction to First and Second Generation and Hybrid Knowledge Based Systems*. Marcel Dekker, Inc., New York, 1997
- [Obj02] Object Management Group, Inc.: *The Common Object Request Broker: Architecture and Specification*. Juli 2002. – Version 3.0
- [OPSS98] OLIVEIRA, P. ; PASCOAL, A. ; SILVA, V. ; SILVESTRE, C.: Mission control of the Marius autonomous underwater vehicle – system design, implementation and sea trials. In: *International Journal of Systems Science* 29 (1998), Nr. 10, S. 1065–1080
- [Orc98] ORCHARD, R. A.: *FuzzyCLIPS version 6.04A user's guide*. Institute for Information Technology, National Research Council Canada, 1998
- [Pfü00] PFÜTZENREUTER, T.: Navigationsverfahren für ferngesteuerte Unterwasserfahrzeuge. In: *Interner Bericht, TU Ilmenau*, 2000
- [Pfü03a] PFÜTZENREUTER, T.: Advanced Mission Management for Long-Range Autonomous Underwater Vehicles. In: *Proceedings of the MTS/IEEE Oceans 2003, San Diego, USA*, 2003, S. 2266–2271

- [Pfü03b] PFÜTZENREUTER, T.: An Onboard Mission Replanning System for Autonomous Underwater Vehicles. In: *Proceedings of the 11th IEEE Mediterranean Conference on Control and Automation, Rodos, Greek*, 2003
- [Pfü04] PFÜTZENREUTER, T.: Intelligent Mission Management für Autonomous Mobile Systems. In: *Tagungsband des 49. Internationalen Wissenschaftlichen Kolloquiums, Ilmenau. In Vorbereitung*, 2004
- [Pre94] PREECE, J.: *Human-Computer Interaction*. Addison-Wesley Publishing Company, 1994
- [Pri00] PRITCHARD, D. E.: *Dynamic Route Replanning and Retasking of Unmanned Aerial Reconnaissance Vehicles*, Air Force Institute of Technology, Wright-Patterson AFB Ohio, USA, Diplomarbeit, 2000
- [PS90] PREPARATA, F. P. ; SHAMOS, M. I.: *Computational geometry: An Introduction*. Springer Verlag, New York, 1990
- [Ras83] RASMUSSEN, J.: Skills, Rules and Knowledge; Signals, Signs and Symbols, and Other Distinctions in Human Performance Models. In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-13 (1983), S. 257–266
- [Rea94] REASON, James: *Menschliches Versagen*. Spektrum Akademischer Verlag, Heidelberg, Berlin, Oxford, 1994
- [Ree95] REEVES, C. R. (Hrsg.): *Modern heuristic techniques for combinatorial problems*. McGraw-Hill, London, 1995
- [RK00] ROBINSON, H. ; KEARY, A.: Remote Control of Unmanned Undersea Vehicles. In: *International Unmanned Undersea Vehicle Symposium*. Newport, USA, 2000
- [RYSB00] RIDAO, P. ; YUH, J. ; SUGIHARA, K. ; BATLLE, J.: On AUV Control Architecture. In: *Proceedings of the International Conference on Robots and Systems*. Takamatsu, Japan, 2000
- [SAFG97] SANDBLAD, B. ; ANDERSSON, A. W. ; FREJ, I. ; GIDEON, A.: The role of Human-Computer Interaction in design of new Train Traffic Control Systems. In: *Proceedings of the World Congress on Railway Research*, 1997, S. 777–783
- [SKLC96] SCHNEIDER, M. ; KANDEL, A. ; LANGHOLZ, G. ; CHEW, G.: *Fuzzy Expert System Tools*. John Wiley and Sons, Chichester, England, 1996
- [SMK95] SILVA, C. T. ; MITCHELL, J. S. B. ; KAUFMAN, A. E.: Automatic Generation of Triangular Irregular Networks using Greedy Cuts. In: *Proceedings of the 6th IEEE Visualization 1995 Conference (VIS '95)*, 1995
- [SSD+02] SIMMONS, R. ; SMITH, T. ; DIAS, M. B. ; GOLDBERG, D. ; HERSHBERGER, D. ; STENTZ, A. ; ZLOT, R. M.: A Layered Architecture for Coordination of Mobile Robots. In: *Multi-Robot Systems: From Swarms to Intelligent Automata, Proceedings from the 2002 NRL Workshop on Multi-Robot Systems*, Kluwer Academic Publishers, 2002

- [STK01] SEIFERT, T. ; TAUBER, F. ; KAYSER, B.: A high resolution spherical grid topography of the Baltic Sea - revised edition. In: *Proceedings of the Baltic Sea Science Congress, 2001*
- [TSW01] TOMPKINS, P. ; STENTZ, T. ; WHITTAKER, W.: Automated Surface Mission Planning Considering Terrain, Shadows, Resources and Time. In: *Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS '01), 2001*
- [Tur95] TURING, A. M.: Computing Machinery and Intelligence. In: FEIGENBAUM, E. A. (Hrsg.) ; FELDMAN, J. (Hrsg.): *Computers and Thought*, AAAI Press / The MIT Press, Menlo Park, 1995, S. 11–35
- [VG94] VASUDEVAN, C. ; GANESAN, K.: Intelligent mission planner for autonomous underwater vehicles. In: BUNTINE, W. (Hrsg.) ; FISHER, D. H. (Hrsg.): *Knowledge-Based Artificial Intelligence Systems in Aerospace and Industry* Bd. 2244, SPIE-The International Society for Optical Engineering, 1994, S. 118–128
- [XZF04] XU, H. ; ZHANG, Y. ; FENG, X.: Discrete Hierarchical Supervisory Control for Autonomous Underwater Vehicles. In: *Proceedings of the 2004 International Symposium on Underwater Technology, Taipei, Taiwan, 2004*

# Index

- Autonomes mobiles System, 3
- DeepC, 2, 75
  - Überblick, 86
  - Aufbau, 87
  - Manöverkatalog, 90
  - Missionsüberwachung, 94
  - Missionsplanung, 93
  - Missionsumplanung, 97
  - Softwarearchitektur, 89
- Delaunay Triangulation, 66
- DENDRAL, 39
- Dijkstra-Algorithmus, 72
- dreiecksvermaschte Netze, 66
- Ergonomie, 35
- Expertensystem, 29, 39
- Füllplanelement, 54
- Faktermittlung, 42
- Fallbasiertes Schließen, 9, 14, 29
- Geografische Planmodifikation, 70, 99
- Geografische Planprüfung, 67, 97
- Geoid, 65
- GOLOG, 15
- Handlungsmodell nach Rasmussen, 17, 35
- Inferenzsystem, 41
- Intelligentes Missionsmanagement, 4
- Kreisapproximation, 67
- Maschinelles Lernen, 9
- MauSI, 2, 75
  - Manöverkatalog, 80
  - Missionsüberwachung, 81
  - Missionsplanung, 80
  - Missionsumplanung, 83
  - Softwarearchitektur, 76
- Missionsüberwachung, 39
  - Aufbau, 45
  - Aufgaben, 44
- Missionsmanagement, 4
  - Aufbau, 25
  - intelligentes, 4
  - Konzeption, 25
  - Rahmenbedingungen, 20
  - Zielsetzung, 22
- Missionsplan, 3
- Missionsplanung, 27, 33
- Missionsumplanung, 51
  - Ablauf, 51
  - Allgemeine Methoden, 51
- MYCIN, 9, 39
- Planbereinigung, 54, 57, 63, 83, 99
- Planen, 9, 15
- Planmodifikation
  - Ablauf, 51
  - Allgemeine Methoden, 51
  - Geografische, 70, 99
  - Spezielle Verfahren, 57
- Planprüfung
  - Geografische, 67, 97
- Polygonale Approximation, 67
- Produktionssystem, 9, 41
- Rahmenbedingungen, 20
- Rasmussen
  - Handlungsmodell, 17, 35
- Rational Behavior Model, 19
- Regelbasiertes System, 9, 29, 39
- SAUSAGES, 11
- Softwarearchitektur, 17
- STRIPS, 15
- Wissen, 4
- Wissensbasierte Systeme, 8
- Wissensbasis, 39
- Wissensrepräsentation, 40
- Wissensverarbeitung, 41
- Zielsetzung, 22