



Fraunhofer Institut Intelligente Analyse- und Informationssysteme

Wintersemester 2007/2008

DIPLOMARBEIT

SICHERE NAVIGATION EINER AUTONOMEN ROBOTERPLATTFORM

Diplomand: Felix Müller Matr.Nr.: 15146081 Studiengang: Mechatronik

Erstbetreuer: Prof. Dr.-Ing. Heinrich Uhe Zweitbetreuer: Dipl.-Inf. Ulrike Petersen

Danksagung

Die vorliegende Diplomarbeit wurde unter Anleitung von Prof. Dr.-Ing. Heinrich Uhe im Fachbereich Maschinentechnik und Mechatronik der Fachhochschule Lippe und Höxter am Fraunhofer Institut für Intelligente Analyse- und Informationssysteme durchgeführt.

Mein Dank gilt Herrn Prof. Dr.-Ing. Heinrich Uhe für die Betreuung der Arbeit und seine Anregungen sowie für sein stetiges Interesse an der Arbeit.

Ein besonderer Dank gilt Frau Ulrike Petersen vom *ProfiBot*-Team für die Betreuung und Förderung der Arbeit von Seiten des Fraunhofer-Instituts und die hilfreichen Tips.

Weiterhin möchte ich mich bei allen Mitarbeitern des *ProfiBot*-Projekts - Josef, Ulrich, Björn, Thorsten, Robert, Jochen, Christoph B., Christoph P. und Florian - sowie bei Herrn Dr. Hartmut Surmann für die Unterstützung bedanken.

Mein größter Dank gilt meiner Familie und meiner Freundin Jutta.

Kurzfassung

Das vom Fraunhofer Institut für Intelligente Analyse- und Informationssysteme (IAIS) initiierte und vom Bundesministerium für Bildung und Forschung (BMBF) geförderte Projekt *ProfiBot* umfasst ein modulares Roboterbaukastensystem, das in der beruflichen Mechatronikausbildung eingesetzt wird. Das Roboterbaukastensystem wurde als Grundlage für die Entwicklungen in dieser Arbeit verwendet.

In dieser Arbeit wird das Konzept eines mechatronischen Systems beschrieben, mit dem unter Berücksichtiung von Sicherheitsrichtlinien ein autonomer Betrieb der *ProfiBot*-Roboter möglich ist. Das System besteht aus unterschiedlichen Sensoren, aus Modifikationen und Erweiterungen der Roboterhardware und aus einem Steuerungsprogramm.

Diese Arbeit behandelt zunächst eine Analyse und Auswahl vorhandener Navigationsverfahren und Steuerarchitekturen für die Entwicklung eines, für *ProfiBot*-Roboter geeigneten, Navigationssystems. Im Anschluss wird die Auswahl geeigneter Sensoren für eine Objekt- und Kollisionserkennung beschrieben. Zu diesem Zweck wurden verschiedene Sensortypen für eine sinvolle Sensorfusion untersucht. Die Sensordaten werden ausgewertet, um eine redundante Messwertaufnahme zu ermöglichen. Die Navigation besteht aus Verhaltensweisen, die in eine Subsumptionsarchitektur integriert sind und arbeitet mit einer Gewichtung von Navigationsalgorithmen. Das Navigationssystem nutzt Distanzwerte als Echtzeitdaten. Es basiert ausschließlich auf lokaler Navigation, ist einfach strukturiert und benötigt keine aufwändige Kartographie. Das System ist im Steuerungsprogramm integriert und in Fahrversuchen sukzessiv optimiert worden.

Das Steuerprogramm verarbeitet die Sensorsignale und berechnet die Steuerparameter für die Aktoren, die sich aus den Navigationsalgorithmen ergeben. Es ist modular aufgebaut, damit weitere Verhaltensweisen auf einfache Weise integriert werden können. Das Navigationssystem wurde auf einem der *ProfiBot*-Roboter erfolgreich gestestet. Ein *Profi-Bot*-Roboter mit diesem System kann in ebenen, überdachten Räumen unter verschiedenen Licht- und Temperaturverhältnissen eingesetzt werden. Die Materialkosten der Modifikationen sind mit ca. 3000 Euro im Vergleich zu ähnlichen Konzepten preiswert.

Bei Fahrversuchen zeigte sich, dass das System in wenigen, sicherheitsunkritischen Fahrsituationen noch nicht optimal funktioniert. Im Rahmen der Arbeit werden Lösungsvorschläge zur Optimierung des Fahrverhaltens in diesen Situationen erarbeitet.

Es ist geplant den *ProfiBot*-Roboter, der im Rahmen dieser Arbeit modifiziert wurde, als Demonstrator im Foyer von IAIS einzusetzen.

Eidesstattliche Erklärung

- Mir ist bekannt, dass die Diplomarbeit als Pr
 üfungsleistung in das Eigentum Nordrhein-Westfalens
 übergeht. Hiermit erkl
 äre ich mein Einverst
 ändnis, dass Studenten der Fachhochschule Lippe und H
 öxter in diese Pr
 üfungsleistung einsehen d
 ürfen, und dass die Fachhochschule Lippe und H
 öxter die Abschlussarbeit unter Nennung meines Namens als Urheber ver
 öffentlichen darf.
- 2. Ich erkläre hiermit, dass ich diese Diplomarbeit selbständig verfasst, noch nicht anderweitig für andere Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen und Hilfsmittel benützt sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

Lemgo, den 18. Dezember 2007

Aufgabenstellung

Ziel der Diplomarbeit ist es, auf Basis eines ProfiBot-Roboters [57] eine autonome Roboterplattform zu entwickeln, die in der Lage ist, in ebenen Räumen Hindernissen, wie z.B. Personen, Stühlen und Tischen auszuweichen. Grundsätzlich soll die Roboterplattform einen Raum in zufälliger Richtung abfahren und dabei ihre Richtung ändern, sobald ein Hindernis erkennbar ist. Die Hindernisse können entweder statische Objekte sein oder sich selbst im Raum mit einer Geschwindigkeit von bis zu 7 km/h bewegen. Des Weiteren soll gewährleistet werden, dass bei einer unvermeidlichen Kollision die Roboterplattform sofort stoppt. Unvermeidliche Kollisionen können geschehen auf Grund von Ausfällen der Sensorik oder wenn sich ein Objekt schneller als die o.g. Maximalgeschwindigkeit in Richtung der Roboterplattform bewegt. Wesentliche Punkte der Arbeit sind:

- die Auswahl und Einbindung geeigneter Sensorik
- die Entwicklung der zugehörigen Software
- die Konstruktion der erforderlichen Hardware

Inhaltsverzeichnis

\mathbf{D}_{i}	anksa	agung		Ι
K	urzfa	ssung		II
Ei	desst	attlich	ne Erklärung	III
\mathbf{A}	ufgat	oenstel	lung	IV
1	Ein	führun	g	1
	1.1	Gliede	rung	. 1
	1.2	Erläut	erung der Aufgabenstellung	. 2
		1.2.1	Problemstellung	. 2
		1.2.2	Zielsetzung	. 3
		1.2.3	Definition der Umgebung	. 3
2	Gru	ndlage	en	5
	2.1	Begriff	fsdefinitionen	. 5
	2.2	Koord	inatentransformationen	. 6
	2.3	Dead-l	Reckoning Verfahren	. 10
	2.4	Braite	nberg-Vehikel	. 10
	2.5	Sicher	heitsaspekte	. 12
	2.6	State of	of the Art	. 14
		2.6.1	Scitos G5	. 15
		2.6.2	Komm-rein, Mach-was, Also-gut	. 16
		2.6.3	KURT3D	. 17
3	Ana	lyse d	es <i>ProfiBot</i> -Roboters	19
	3.1	Konze	pt	. 20
		3.1.1	ProfiBot in der Mechatronikausbildung	. 20
		3.1.2	Das Baukastenprinzip	. 21
	3.2	Hardw	/are	. 22
		3.2.1	Chassis	. 22
		3.2.2	Energieversorgung	. 23
		3.2.3	Motorcontroller	. 24

		3.2.4	Aktorik	25
		3.2.5	Sensorik	26
	3.3	Softwa	are	27
		3.3.1	Programmierung mit <i>IConnect</i>	28
		3.3.2	Firmware und Regelalgorithmus	32
4	Sicł	nere N	avigation	34
	4.1	Begrif	fseinordnung	34
	4.2	Robot	ernavigation	37
		4.2.1	Kartographie	38
		4.2.2	Lokalisationsfähigkeit	40
		4.2.3	Routen- und Bahnplanung	41
		4.2.4	Karteninterpretation	43
	4.3	Robot	erverhaltensweisen	43
		4.3.1	Subsumptionsarchitektur	45
		4.3.2	Potentialfeldverfahren	46
		4.3.3	Zielloses Umherfahren	48
		4.3.4	Hindernisvermeidung	48
		4.3.5	Notstopp	50
	4.4	Entsch	neidungsfindung	54
5	A 115	swahl o	ler Sensorik	56
0	5.1	Anfor	derungen an die Sensorik	56
	5.2	Redur	idanz durch Sensorfusion	59
	5.3	Sensor	ren für die Hinderniserkennung	59
	0.0	5.3.1	Ultraschallsensoren	59
		5.3.2	Triangulationssensoren	62
		5.3.3	Kapazitive und induktive Wegaufnehmer	64
		5.3.4	Laserscanner	64
		5.3.5	3D - Kameras	67
	5.4	Sensor	cen für Kollisionserkennung	69
		5.4.1	Schaltleisten	69
		5.4.2	Kontaktpuffer	70
		5.4.3	Auswertung des Motorlaststroms	70
	5.5	Entsch	neidungsfindung	72
6	Tec	hnisch	e Realisierung	78
U	61	Physil	kalische Ebene	78
	0.1	6.1.1	Hardwareintegration der Sensorik	78
		6.1.2	Softwareintegration der Sensorik	83
		6.1.3	Modifikationen am Chassis	87
		6.1.4	Mülltonnenaufsatz	91

6.2 Logische Ebene) 4
		6.2.1 Eingesetzte Verhaltensweisen und Drehzahlberechnung 9) 4
		6.2.2 <i>IConnect</i> Signalgraphen)0
7	Mes	sungen und Ergebnisse 10)5
	7.1	Funktionsprüfung der Konturenmessung)5
	7.2	Lenkwinkelgenerierung	10
	7.3	Überprüfung der Robustheit	1
	7.4	Fahrweise	13
		7.4.1 Korrekturen	15
8	\mathbf{Disl}	ussion und Ausblicke 11	.6
	8.1	Diskussion	16
	8.2	Ausblicke (Perspektiven)	17
		8.2.1 Änderung des Regelalgorithmus	17
		8.2.2 Senkung des Rucks	18
		8.2.3 Dockingstation	19
Aı	nhan	g 12	20
In	halt	der CD 12	23
Fo	\mathbf{rmel}	zeichenverzeichnis 12	24
Al	Abkürzungsverzeichnis 1		
$\mathbf{Li}^{\mathbf{r}}$	terat	urverzeichnis 12	29

Kapitel 1

Einführung

1.1 Gliederung

Die Arbeit ist gegliedert in acht Kapitel. Nach einer Einführung und der Aufgabenstellung werden zunächst allgemeine Grundlagen aufgeführt, die zum Verständnis dieser Arbeit beitragen und eine Einführung in die Themenbereiche dieser Arbeit geben (Kap. 2). Weiterhin werden die Sicherheitsrichtlinien und bereits realisierte Konzepte angesprochen und untersucht, inwiefern sie im Rahmen dieser Arbeit anwendbar sind. Der Roboter Profi-Bot, der im Rahmen dieser Arbeit als Basisplattform für die Entwicklungen dient wird vorgestellt (Kap. 3). Es werden die bereits vorhandenen Komponenten beschrieben und die Komponenten ausgewählt, die für diese Arbeit eine Rolle spielen. Des Weiteren sind Lösungsvorschläge zur Modifikation der übrigen Komponenten aufgeführt. Die Einordnung des Begriffs "Navigation" und vorhandene Navigationsverfahren werden erörtert und für die Steuerungsarchitektur ausgewählt (Kap. 4). Anschliessend wird untersucht, welche physikalischen Größen durch Sensoren aufgenommen werden müssen, um eine sichere Navigation zu ermöglichen (Kap. 5). Im Anschluss werden die ensprechenden Sensortypen untersucht und eine passende Auswahl getroffen. Die durchgeführten Modifikationen am ProfiBot-Roboter sowie der Aufbau und die Funktionsweise der erstellten Steuerungssoftware werden dargestellt (Kap. 6). Um das gewählte Konzept auf seine Funktionweise zu untersuchen, wurden Versuche durchgeführt und die Ergebnisse bewertet (Kap. 7). Abschliessend wird eine Gesamtbeurteilung vorgenommen, aus denen Vorschläge für Erweiterungen und Optimierungen abgeleitet werden, die Perspektiven für weitere Untersuchungen aufzeigen (Kap. 8).

1.2 Erläuterung der Aufgabenstellung

ProfiBot ist ein Roboterbaukastensystem, das in der Ausbildung zum Mechatroniker und zur Mechatronikerin als begleitendes Lehrmaterial dient. Im Rahmen dieser Arbeit soll ein Sicherheitskonzepts entworfen und ein *ProfiBot*-Roboter so modifiziert werden, dass eine sichere, autonome Navigation in kritischen Fahrsituationen möglich ist. Kritische Fahrsituationen sind beispielsweise unmittelbar bevorstehende Kollisionen. Das Roboterbaukastensystem liegt in der Basisversion vor. Es besitzt in dieser Version noch keinerlei Systeme zur autonomen und sicheren Navigation. Die Basisversion gilt es derart zu erweitern, dass beim Fahrbetrieb keine Objekte, Personen oder die Roboterplattform selbst beschädigt bzw. verletzt werden können.

Es gibt bereits eine Vielzahl von Verfahren zur sicheren Navigation. Die Verfahren werden ebenfalls im Rahmen dieser Arbeit analysiert. Die Plattform soll für einen Demonstrator (s. Abb. 1.1) genutzt werden. Hierfür soll ein Müllbehälter auf die Plattform montiert werden. Der Roboter soll z.B. in einem Museum oder einem Foyer eingesetzt weden, damit Besucher ihren Abfall einwerfen können.



Abb. 1.1: Der im Rahmen dieser Arbeit entwickelte Demonstrator auf Basis eines ProfiBot-Roboters

1.2.1 Problemstellung

Autonome, mobile Roboter müssen sicher sein, besonders wenn sie mit Menschen in Kontakt kommen. Weil der modifizierte ProfiBot-Roboter relativ große Abmessungen (920mm x 620mm x 560mm) hat, muss die in das Navigationsystem einzubindende Sensorik den *ProfiBot*-Roboter in seiner vollen Größe erfassen.

1.2.2 Zielsetzung

Die wichtigsten Ziele und Anforderungen, die im Rahmen dieser Arbeit an das Sicherheitskonzept gestellt wurden, sind nachfolgend aufgelistet.

- Sichere Fahrweise in möglichst vielen, kritischen Fahrsituationen bezogen auf die Einsatzumgebung
- Hohe Ausfallsicherheit des Systems
- Übertragbarkeit des Sicherheitskonzepts auf andere Robotertypen
- Hohe Bedienfreundlichkeit der Programme und Programmoberflächen
- Übersichtliche, modularisierte Programme um nachträgliche Änderungen zu vereinfachen
- Geringer Energieverbrauch der für das Sicherheitskonzept verwendeten Hardware

1.2.3 Definition der Umgebung

In diesem Abschnitt werden die Definitionen über die Räume und Hindernisse, in denen sich der modifizierte *ProfiBot* bewegen soll, unter Berücksichtigung der Vorgaben aus der Aufgabenstellung erarbeitet. Weil ein autonomer, mobiler Roboter (AMR) nach heutigem Stand der Technik noch immer speziell auf die Umgebung, in der er eingesetzt wird, abgestimmt werden muss, ist eine genaue Definition dieser Umgebung nötig. Nach der Umgebung richtet sich die Auslegung der Sensorik und die Auslegung der mechanischen Konstruktion.

Die Räume, in denen der modifizierte *ProfiBot-Roboter* betrieben werden soll, müssen eben und geschlossen sein. Der *ProfiBot*-Roboter kann nicht im Outdoorbereich eingesetzt werden. Außerdem ist auszuschließen, dass in den Einsatzräumen Rampen oder Treppen sind. Die Einsatzorte können in diesem Zusammenhang z.B.

- Museen
- Empfangshallen
- Fabrikhallen
- Korridore

sein. Des Weiteren muss gewährleistet sein, dass diese Räume durch Wände begrenzt sind. Als Alternative sind z.B. Kennzeichnungen auf dem Boden anzubringen, die mit Reflexionslichtschranken erkannt und als Begrenzung des Raums interpretiert werden können. Innerhalb dieser Räume klassifiziert der modifizierte *ProfiBot*-Roboter alle erkannten Objekte im Rahmen der Sicherheit als Hindernisse, wenn er ihnen ausweichen soll. Unter Berücksichtigung dieser Kriterien werden einige Beispiele von möglichen Hindernissen aufgelistet:

- Inventar (Tische, Stühle, etc.)
- Wände
- Personen
- Fahrzeuge

In den Räumen, in denen der modifizierte ProfiBot-Roboter eingesetzt wird, dürfen des Weiteren folgende Objekte nicht vorhanden sein, sofern der ProfiBot-Roboter mit ihnen kollidieren könnte:

- Gegenstände aus spiegelnden oder transparenten Materialien
- Personen, die nicht dazu in der Lage sind, den Notaus-Schalter zu betätigen

Kapitel 2

Grundlagen

Im folgenden Abschnitt werden die wichtigsten mathematischen und physikalischen Grundlagen behandelt, auf denen die Entwicklungen in dieser Arbeit aufbauen.

2.1 Begriffsdefinitionen

Kinematik:

"Die Untersuchung und Beschreibung von Bewegungen ohne Berücksichtigung der sie verursachenden Kräfte (im Gegensatz zur Dynamik und Kinetik). Die Bewegung eines Massenpunktes wird durch dessen Zeitabhängigen Ortsvektor r, seine Geschwindigkeit $v = \frac{dr}{dt}$ und seine Beschleunigung $a = \frac{dv}{dt}$ beschrieben" (Definition nach Brockhaus [7]).

Pose:

"In der Pose sind die Position, Rotation und Geschwindigkeit des Objekts zusammengefasst. Existiert zu dem Objekt eine Wahrscheinlichkeitsverteilung, so kann diese zusätzlich angegeben werden" (Definition nach Laue [24]).

Robotik:

"Wissenschaft von der Entwicklung und dem Betreiben automatischer Maschinen (Roboter) zur Handhabung verschiedenartiger Güter oder zur Realisierung unterschiedlicher, auf den Menschen bezogener Funktionen, die vor allem in den Wirschaftsbereichen (Produktion, Dienstleistung), speziell in der Industrie (Industrieroboter) eingesetzt werden können" (Definition nach Brockhaus [7]).

2.2 Koordinatentransformationen

Das in dieser Arbeit verwendete Koordinatensystem für die Messwertberechnungen und Darstellungen ist ein roboterzentrisches, kathesisches Koordinatensystem, bei dem die Ordinate die Distanz in x-Richtung darstellt und die Abszisse die Distanz in y-Richtung, wie in Abbildung 2.1 dargestellt. Das gewählte Koordinatensystem wird sehr oft in der mobilen Robotik angewendet, weil es anschaulich ist. Die Ordinate ist gleichzeitig kongruent mit der Symmetrieachse des *ProfiBot*-Roboters und ebenso mit dem Orientierungswinkel 0°. Dadurch werden Berechnungen vereinfacht, die eine Orientierungsänderung des *Profi-Bot*-Roboters bewirken. Im Gegensatz zu dem gewählten Koordinatensystem liegt in der Mathematik häufig die x-Richtung auf der Abszisse und die y-Richtung auf der Ordinate.



Abb. 2.1: roboterzentrisches, kartesisches Koordinatensystem

Distanzwerte von Sensoren liegen immer als Polarkoordinaten vor, bei denen die Distanz zum gemessenen Objekt der Betrag ist und der Polarwinkel der Winkel zwischen der Sensorachse und der Orientierung eines AMR. Die Distanzmessung, die für die Hinderniserkennung notwendig ist, bezieht sich auf ein Sensorzentrum S, das der Ursprung eines dazugehörigen Bezugssystems ist. Die Distanzwerte werden durch einen Sensor erfasst, der im Sensorzentrum positioniert ist. In Abbildung 2.2 ist ein Sensor dargestellt, der Hindernisse auf einer zum Untergrund parallelen, euklidischen Ebene erfassen kann. Im Rahmen dieser Arbeit wurde ein entsprechender Sensor benutzt. Der Sensor sollte hinsichtlich der sicheren Navigation einen möglichst weiten Bereich der Umgebung erfassen, damit viele Hindernisse erkannt werden. Dieser Bereich wird einerseits durch die Reichweite r_{Smax} der Sensoren begrenzt, andererseits durch den Erfassungswinkel γ (s. Abb. 2.3). In Abbildung 2.3, in der vereinfacht angenommen wird, dass der AMR eine lineare Bahn abfährt, wird dies verdeutlicht. Mit dem Erfassungswinkel γ' ist der Erfassungskegel zu klein um das Hindernis zu erkennen. Hindernisse können nur innerhalb der Spur mit der Breite b' erkannt werden. Der AMR würde mit dem Hindernis in der Abbildung 2.3 kollidieren. Mit dem Erfassungswinkel γ und der resultierenden Spur der Breite b kann das Hindernis registriert werden und der Roboter ausweichen. b muss wesentlich breiter als der Roboter selbst sein.



Abb. 2.2: Sensorzentrum S nach [39]

Abb. 2.3: Erfassungswinkel

Sind zusätzlich ein weiterer Sensor mit dem Zentrum S' an einer anderen Position befestigt, wird eine Transformation durchgeführt (s. Abb. 2.4). In der nachfolgenden Beschreibung wird davon ausgegangen, dass dem AMR ein roboterzentrisches, kartesisches Bezugssystem zugrunde liegt. Für die Transformation muss die Position der Sensoren relativ zum Sensorzentrum bekannt sein.

Die Transformation lässt sich bezüglich Abb. 2.4 mit folgender Vektorgleichung ausdrücken:

$$\vec{d_S} = \vec{r_S} - \vec{r_S}', \ \vec{d_S} = \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$
 (2.1)

 $\vec{r_S}'$ ist die Distanz von Sensor zum erfassten Punkt, $\vec{r_S}$ die Distanz vom Sensorzentrum zum Punkt und $\vec{d_S}$ die Distanz vom Sensorzentrum zum Sensor.



Abb. 2.4: Transformation in das Sensorzentrum

Abb. 2.5: Transformation in das kinematische Zentrum in Anlehnung an [39]

Die Transformation bezieht sich auf Sensoren, die eindimensional messen, weil solche im Rahmen dieser Arbeit eingesetzt wurden. Es werden die Distanzwerte mit nachfolgenden Gleichungen in ein Koordinatensystem transformiert, das den jeweiligen Sensor als Ursprung hat.

$$\Delta x_S = r_S \cdot \cos\varphi_s$$

$$\Delta y_S = r_S \cdot \sin\varphi_s$$
(2.2)

Durch Addition von $\vec{d_S}$ wird der erfasste Punkt schließlich im Bezug zum Sensorzentrum dargestellt.

$$x_S = \Delta x'_S + \Delta x$$

$$y_S = \Delta y'_S + \Delta y$$
(2.3)

Die Transformation ist dementsprechend eine Addition der Entfernung des Sensors zu den Distanzwerten. In Polarform ist $r_S = \sqrt{x_S + y_S}$ mit dem Winkel $\varphi_S = \arctan(\frac{y_S}{x_S})$.

Wenn $\vec{r_S}'$ bekannt ist, erfolgt eine Transformation auf das kinematische Zentrum K des AMR. Das kinematische Zentrum liegt beim *ProfiBot*-Roboter auf der Drehachse. Ohne diese Transformation wäre keine genaue Steuerung möglich. An Abbildung 2.5 und den nachfolgenden Gleichungen wird die Transformation verdeutlicht. φ_S und r_S werden als bekannt vorausgesetzt, φ_K und r_K sind die im kinematischen Zentrum benötigten Größen in polarer Form nach der Transformation.

$$\Delta x_S = \sin\varphi_S \cdot r_S \tag{2.4}$$

$$\Delta x_K = \sin\varphi_K \cdot r_K \tag{2.5}$$

Aus 2.4 und 2.5 folgt:

$$r_K = \frac{\sin\varphi_S \cdot r_S}{\sin\varphi_K} \tag{2.6}$$

$$y_S = \cos\varphi_S \cdot r_S \tag{2.7}$$

$$\Delta y_K = y_S + \Delta y = y_S + \overline{KS} \tag{2.8}$$

Aus 2.4, 2.7 und 2.8 folgt:

$$\varphi_K = \arctan\left(\frac{\sin\varphi_S \cdot r_S}{\cos\varphi_S \cdot r_S + \overline{KS}}\right) \tag{2.9}$$

Wird eine Transformation von Sensorzentrum in das kinematische Zentrum in einem roboterzentrischen, kartesischen Koordinatensystem durchgeführt, ist es notwendig das Sensorzentrum um die Strecke $-\overline{KS}$ auf der Ordinate zu verschieben.

Distanzwerte können direkt in das kinematische Zentrum transformiert werden. Eine Lösung ist Δx und Δy auf die Distanzwerte zu addieren, wie in Abbildung 2.6 zu sehen ist.



Abb. 2.6: Direkte Transformation in das kinematische Zentrum

2.3 Dead-Reckoning Verfahren

Das Dead-Reckoning Verfahren wird oft in der mobilen Robotik unterstützend zur Navigation angewendet und wird deswegen im nachfolgenden Abschnitt beschrieben. Das Dead-Reckoning ist ein Verfahren, das kleine Positionsveränderungen summiert und anhand dieser Änderungen die Folgeposition eines AMR ermitteln kann, z.B. über die Auswertung der Odometrie. Wenn die Pose $P_{k-1}(x_{k-1}, y_{k-1}, \theta_{y,k-1})$ eines AMRs zum Zeitpunkt k-1bekannt ist, kann die aktuelle Pose durch die Posenänderung mit Hilfe der Gleichungen 2.10 bis 2.12 bestimmt werden.

$$\theta y_k = \theta_{y,k-1} + \delta \theta_y \tag{2.10}$$

$$x_k = x_{k-1} + \delta x \cdot \cos(\theta_{y,k-1}) + \delta_y \cdot \sin(\theta_{y,k-1})$$
(2.11)

$$y_k = y_{k-1} + \delta x \cdot \sin(\theta_{y,k-1}) + \delta_y \cdot \cos(\theta_{y,k-1})$$
(2.12)

2.4 Braitenberg-Vehikel

Um ein Einblick in Steuerungarchitekturen und Verhaltensweisen für AMR zu gewährleisten, wird im Folgenden ein einfaches Beispiel vorgestellt. Braitenberg-Vehikel sind Fahrzeuge, die komplizierte Verhalten darlegen, obwohl sie einfache Steuerungsstrukturen haben. Valentino Braitenberg stellt diese Vehikel in seinem Buch Experimente mit künstlichen Wesen [5] vor. Durch eindimensional messende Sensoren, die sich an der rechten und linken Frontseite eines Braitenberg-Vehikels befinden, werden Umgebungseinflüsse direkt als Steuerparameter zu einem Differentialantrieb weitergeleitet.

Beim Einsatz von lichtempfindlichen Sensoren, wie bespielsweise Photodioden, kann ein Braitenberg-Vehikel entweder zu der Lichtquelle hinfahren, oder von ihr weg. Das entsprechende Verhalten wird durch eine Bewertung der Steuerparameter mit entweder einer hemmenden (drehzahlsenkenden) Wirkung, oder einer stimulieren (drehzahlerhöhenden) Wirkung realisiert¹. Die Verhalten werden des Weiteren durch eine unterschiedliche Verschaltung beeinflusst. Als Beispiel sind zwei Vehikel mit den von *Braitenberg* gegebenen Bezeichnungen 2a und 2b in Abbildung 2.7 dargestellt.

Das Vehikel 2a in Abbildung 2.7 entfernt sich von der Lichtquelle, die als Kreis dargestellt ist, das Vehikel 2b fährt zu ihr hin. Die Berechnung der Motordrehzahlen n_l und n_r in Abhängigkeit von den Sensorwerten s_l und s_r Vehikel kann durch die Gleichungen 2.13 (Vehikel 2a) und 2.14 (Vehikel 2b) erfolgen. k_n ist ein Verstärkungsfaktor.

$$n_l = s_l \cdot k_n$$

$$n_r = s_r \cdot k_n$$
(2.13)

¹eine hemmende Wirkung wird in [5] mit einem -, eine stimulierende Wirkung mit einem + bezeichnet



Abb. 2.7: Braitenberg-Vehikel 2a (links) und 2b (rechts) (Quelle: [40])

$$n_r = s_l \cdot k_n \tag{2.14}$$
$$n_l = s_r \cdot k_n$$

Das Vehikel 3c, das in Abbildung 2.8 dargestellt ist, hat zwei Sensorpaare, s_1 (rot) und s_2 (grün), die durch Symbole dargestellt sind und auf die enstprechenden Umwelteinflüsse reagieren. Die Drehzahlen für die Motoren können durch die Gleichungen 2.15 berechnet werden. Das rote und grüne Symbol wirkt jeweils anziehend auf das Vehikel 3c.



Abb. 2.8: Braitenberg-Vehikel 3c (Quelle: [40])

$$n_r = s_{l,1} \cdot k_{n,1} - s_{l,2} \cdot k_{n,2}$$

$$n_l = s_{r,1} \cdot k_{n,1} - s_{l,2} \cdot k_{n,2}$$
(2.15)

Mit den Braitenberg-Vehikeln wurden bereits frühzeitig autonome, mobile Roboter realisiert, die Hindernissen ausweichen konnten. Allerdings waren die Umgebungsbedingungen, in denen die Vehikel untersucht wurden, sehr einfach. In der heutigen Zeit werden Braitenberg-Vehikel u.a. für Lehrzwecke eingesetzt. Mit dem Roboter *Boe-Bot* vom Hersteller *Parallax* kann das Verhalten eines Braitenberg-Vehikels simuliert werden. Bei AMR mit geringen Abmessungen ist eine Hindernisvermeidung nach dem vorgestellten Verfahren ausreichend. AMR mit großen Abmessungen, wie beispielsweise der *ProfiBot*-Roboter, benötigen mehr als zwei eindimensional messende Sensoren und eine damit verbundene, komplexere Kontrollarchitektur, um die geometrische Auflösung der Umgebung zu erhöhen.

2.5 Sicherheitsaspekte

In diesem Abschnitt werden Sicherheitkriterien für einen mobilen Roboter erläutert. Außerdem wird geprüft, in welchem Maß die Sicherheitkriterien auf das Sicherheitskonzept des modifizierten *ProfiBot*-Roboters angewendet werden können.

Der Grad an Sicherheit eines Roboters hängt davon ab, in welchem Maß die Vermeidung von Schäden an Personen und Gegenständen sowie am Roboter selbst in allen Situationen gewährleistet ist. Weiterhin sind folgende Begriffe aufgeführt, die auch bei Adamy und Bechtel [1] zu finden sind:

1. Sicherheit:

Freiheit von unvertretbaren Risiken.

2. Verlässlichkeit:

Eigenschaft eines Systems, eine vertrauenswürdige Systemleistung zu erbringen.

3. ZUVERLÄSSIGKEIT:

Beschaffenheit einer Einheit bezüglich ihrer Eignung, während oder nach vorgegebenen Zeitspannen bei vorgegebenen Anwendungsbedingungen die Zuverlässigkeitsanforderungen zu erfüllen.

4. STATIONÄRE VERFÜGBARKEIT:

Die Wahrscheinlichkeit, in der die Betriebsbereitschaft eines Systems zu einem beliebigen Zeitpunkt gegeben ist. Für die stationäre Verfügbarkeit gilt $\frac{T_b}{T_b+T_s}$. Dabei ist T_b die mittlere Betriebsdauer zwischen zwei Ausfällen und T_s die mittlere Störungsdauer.

5. INSTANDHALTBARKEIT:

Beschaffenheit einer Einheit bezüglich ihrer Eignung für die Instandhaltung bei festgelegten Mitteln und Verfahren.

Momentan existieren noch keine Maschinenrichtlinien für mobile Roboter, die sich auf die genannten Definitionen zur Sicherheit beziehen. Es gilt grundsätzliche Richtlinien zur Sicherheit (z.B. für Fahrerlose Transportsysteme) auch auf mobile Roboter anzuwenden. Es dürfen z.B. keine Schäden entstehen, wenn der Roboter sich nicht in einem ordnungsgemäßen Zustand befindet oder wenn ein nicht bestimmungsgemäßer Gebrauch des Roboters vorliegt. Weiterhin ist zwischen

- Bauart des Roboters
- Aufgaben des Roboters
- Einsatzgebieten

zu unterscheiden um das Maß der Sicherheitsanforderungen festzulegen [1].

Niedrige Sicherheits- anforderungen		Hohe Sicherheits- anforderungen
Bauart/Typ des Roboters:	 Geringes Gewicht Radgetriebener Roboter 	 Hohes Gewicht zweibeiniger Laufroboter
Aufgaben des Roboters:	 Ungefährliche Aufgaben Einfache Aufgaben Ferngesteuerter Roboter 	 Gefährliche Aufgaben Komplexe Aufgaben Autonomer Roboter
Einsatzgebiet/ Umgebungs- bedingungen:	 Ungefährliche Atmosphäre Ungefährliche Umgebung Innen 	 Gefährliche Atmosphäre Gefährliche Umgebung Außen
	 getrennt von Menschen 	mit Menschen Zusammenarbeit mit Menschen

Abb. 2.9: Sicherheitsanforderungen mobiler Roboter (Quelle: [1])

Bezogen auf Abbildung 2.9, in der die genannten Sicherheitskriterien konkret dargestellt sind, sind für die Roboterplattform Sicherheitsanforderungen in Tabelle 2.1 dargestellt. Alle Kriterien in der Abbildung beziehen sich auf die zu erwartenden Anforderungen der zu entwickelnden Roboterplattform. Die Bewertung der Sicherheitskriterien geschieht auf einer Skala zwischen 0 und 10, wobei 10 für höchste Sicherheitsanfordungen steht. Die Sicherheitskriterien in Tabelle 2.1 wurden nochmals unterteilt, um eine vereinfachte Bewertung zu ermöglichen.

Sicherheitskriterien	Eigenschaften	Bewertung
BAUART	radbetrieben	1
Gewicht	hohes Gewicht	8
Aufgaben	einfache Aufgaben	2
Betrieb	autonom	10
Einsatzgebiet	innen, ebener Raum	2
Umgebungsbedingunen	ungefährlich	3
Einfluss auf Personen	menschlicher Kontakt	6

Tabelle 2.1: Sicherheitsanforderungen an die Roboterplattform

In dieser Arbeit werden vorwiegend diejenigen Sicherheitskriterien behandelt, die sich auf

eine sichere Navigation beziehen. Dazu zählt hauptsächlich das Vermeiden einer Gefährdung durch mechanisch-kinetische Energie, z.B. durch Kollisionen, die Quetschungen und Scherungen zur Folge haben können. Hierzu ist die Sensorik vornehmlich auf die Vermeidung von Kollisionen mit Personen auszurichten, da Personenschutz oberste Priorität hat (s. [50] und [1]). Die Sensorik ist so gewählt, dass sie die Vermeidung von Personenschäden auf ein geringes Restrisiko reduziert. Eine vollständige Risikobeurteilung des entwickelten Sicherheitskonzepts wird im Rahmen dieser Arbeit nicht erbracht, sie kann zweckmäßig nur von einer *Benannten Stelle*² durchgeführt werden. Sensoren können entweder berührungslos oder berührungsbehaftet Kollisionen erkennen [50]. In der Praxis wird die Kombination aus beiden Systemen bevorzugt. Diese Kombination wird auch in dieser Arbeit betrachtet.

Für geringe Folgeschäden einer Kollision sollte weiterhin die mechanische Konstruktion so ausgelegt sein, dass keine gefährlichen Kanten am AMR vorhanden sind. Die Software muss so ausgelegt sein, dass der sichere Betriebszustand zu jeder Zeit überwacht wird.

Weiterhin muss an jeder Art von Maschinen eine Notstopp- bzw. Nothalt-Funktion vorhanden sein. Nach DIN EN 60204-1 Anschnitt 9.2.2 sind Stopp-Funktionen in drei Kategorien aufgeteilt (s. [4]). Nach Kategorie 0 muss eine Maschine durch sofortiges Abschalten der Energie stillgesetzt werden. Nach den Kategorien 1 und 2 erfolgt ein gesteuertes Stillsetzten der Maschinen ohne Energieunterbrechung, wobei in Kategorie 1 die Energiezufuhr nach dem Stillsetzen unterbrochen wird. Das Abschalten des ProfiBot-Roboters nach Kategorie 0 wird im Rahmen dieser Arbeit durch ein Notaus-Schalter realisiert, bei allen anderen Notstopp-Verfahren bleibt die Energiezufuhr erhalten (Kategorie 2). Somit ist eine schnellere Wiederaufnahme des Fahrbetriebs ermöglicht und die stationäre Verfügbarkeit wird dadurch erhöht.

2.6 State of the Art

In dem folgenden Abschnitt werden einige Robotertypen vorgestellt, die im Rahmen dieser Arbeit auf ihr Sicherheitskonzept hin untersucht wurden. Einige Roboter besitzen ausgeprägte Sicherheitsmechanismen, wie z.B. autonome Navigation mit einem Verhalten zur Hindernisvermeidung³. Des Weiteren gibt es umfangreiche Algorithmen zur Navigation autonomer Roboter. Einige dieser Algorithmen sind sehr komplex, so dass schnelle Rechner und Schnittstellen vorhanden sein müssen, um eine ausreichend kurze Reaktionszeit auf Hindernisse gewährleisten zu können. In den folgenden Abschnitten sind einige Beispiele von Robotern augeführt, die verschiedene Methoden der Hindernisvermeidung anwenden, um Hindernissen auszuweichen. Dabei werden die Einsatzorte der Roboter erläutert und der Grund, weshalb die Hindernisvermeidung in diesen Einsatzgebieten wichtig ist. Außer-

 $^{^{2}}Benannte Stellen$ sind neutrale, unabhängige und kompetente Stellen, die die Konformitätsbewertung von Produkten des freien Warenverkehrs (Harmonisierungsrichtlinien nach Art. 95 des EG-Vertrages) durchführen.

³engl. obstacle avoidance

dem wird beschrieben, mit welcher Sensorik die Roboter ausgestattet sind. Die Methoden der Hindernisvermeidung selbst werden in Kapitel 4 beschrieben.

2.6.1 Scitos G5

Scitos G5, ist ein autonomer Roboter, der gemeinsam vom Fachgebiet Neuroinformatik und Kognitive Robotik der *TU Ilmenau* und der Roboter-Firma *MetraLabs* entwickelt wurde (s. Herstellerbeschreibung [56]). Er wird momentan hauptsächlich für die Unterhaltung von Personen verwendet.

Scitos C5, (s. Abb. 2.11), wird als interaktive Kommunikationsplattform in der Baumarkt-Kette Toom unter der Bezeichnung Toomas eingesetzt. Kunden eines Baumarktes können mit Hilfe dieses Roboters zu den gewünschten Baumarktartikeln geleitet werden.





Abb. 2.10: Scitos G5 (Quelle: [56])

Abb. 2.11: Scitos C5 (Quelle: [56])

Beide Roboter erkennen mit 24 Ultraschallsensoren und einem 2D-Laserscanner, die an den Robotern ca. 100mm über Bodenhöhe befestigt sind, die Begrenzungen des Roboterfahrweges durch Hindernisse. Zusätzlich sind unterhalb dieser Sensoren Kontaktpuffer angebracht, die es den Robotern ermöglichen, Kollisionen zu erkennen.

Toomas hat ständigen Personenkontakt, weil Personen nah an ihn herantreten müssen, um die gewünschten Baumarktartikel auf dem Touchscreen des Roboters einzugeben. Der Roboter muss daher mit einem festgelegten Sicherheitsabstand vor den Personen zum Stillstand kommen.

Das Konzept der bodennahen Einbauposition eines 2D-Laserscanners ist für den Pro-fiBot-Roboter brauchbar.

2.6.2 Komm-rein, Mach-was, Also-gut

Die drei Roboter *Komm-rein*, *Mach-was* und *Also-gut*, die in Abb. 2.12 zu sehen sind, wurden im Jahr 2000 im Fraunhofer Institut für Produktionstechnik und Automatisierung (Fraunhofer IPA) entwickelt und sind zur Unterhaltung der Besucher und Besucherinnen im Berliner Museums für Kommunikation bestimmt.



Abb. 2.12: Museumsroboter: Komm-rein, Mach-was und Also-gut (Quelle: [14])

Alle drei Roboter haben unterschiedliche Reaktionsweisen auf Menschen: *Komm-rein* animiert Personen in das Museum einzutreten, *Mach-was* spielt mit einem Ball und *Also-gut* erläutert die Sehenswürdigkeiten des Museums. Weil die Roboter Personenkontakt haben, ist eine Hindernisvermeidung notwendig, damit die Roboter keine Gefahr für die Museumsbesucher und -besucherinnen darstellen.

Die Museumsroboter sind jeweils mit Laserscannern ausgestattet, die 150mm über dem Boden montiert wurden. Ein 2D-Laserscanner hat eine Abtastrate von 30Hz und kann in einem Erfassungswinkel von 180° einen Bogen mit dem Radius von 15m abtasten. Personen werden von dem Begrüßungsroboter *Komm-rein* anhand ihrer Beinpaare erkannt [14]. Durchmesser, Form und Abstand von Objekten werden mit Hilfe einer Fuzzy-Regel ausgewertet, um zu entscheiden, ob es sich um Beinpaare handelt. Die Roboter sind in der Lage sich selbst zu lokalisieren und verfügen über Algorithmen, um einem Pfad zu folgen (s. [13]. Durch den Abgleich der Distanzmessungen mit einer vorgegebenen Karte des Museums errechnet sie sich einen Pfad, der an statischen und dynamischen Hindernissen vorbeiführt und sie innerhalb eines Areals umherfahren lässt, das in den Karten festgelegt wurde. Dabei haben sie eine Maximalgeschwindigkeit von ca. 2km/h, wie Schraft et. al. [42] beschreiben.

Die Konstruktion der Museumsroboter basiert auf einem Roboterassistenten mit der

Modellbezeichnung *Care-O-Bot*, der ebenfalls vom Fraunhofer Institut für Produktionstechnik und Automatisierung entwickelt wurde. Er wurde konzipiert als Roboterassistent für Haushalt und Pflege, der Hilfsbedürftigen einfache Arbeiten im Haushalt abnehmen soll. Das Nachfolgemodell *Care-O-Bot II*, das in Abb. 2.13 zu sehen ist, kann derner als Gehhilfe benutzt werden und verfügt nach einer Beschreibung von Hans [17] zusätzlich über einen 2D-Laserscanner, der in seinem neigbaren Kopf integriert ist. Durch das Neigen wird ein dreidimensionales Entfernungsbild der Umgebung erstellt.



Abb. 2.13: Roboterassistent Care-O-Bot II (Quelle: [45])

Care-O-Bot II besitzt, ebenso wie die Museumsroboter, Kontaktpuffer zur Kollisionserkennung. Kontaktpuffer oder Schaltleisten sollte auch der modifizierte *ProfiBot*-Roboter besitzen.

2.6.3 KURT3D

 $KURT^43D$ (s. Abb. 2.14) ist ein vom Fraunhofer Institut für Intelligente Analyse- und Informationssysteme (Fraunhofer Institut IAIS) entwickelter Erkundungsroboter. Er wurde in seiner ursprünglichen Version für die Inspektion von Abwasserkanälen entwickelt und kann bei der Erkundung von schwer zugänglichem Terrain verwendet werden [52]. Im Wettbewerb "Robo Rescue Real Robot League", bei dem in einem nachgestellten Katastrophenszenario (z.B. Erdbeben) verschüttete Personen gefunden werden müssen, ist das Team um KURT3D 2004 Vizeweltmeister geworden. Sowohl die beiden 90W DC-Motoren, als auch der Motorcontroller von KURT3D, sind identisch im *ProfiBot*-Roboter vorhanden.

 $^{{}^{4}\}mathrm{Kanal}\text{-}\mathrm{Untersuchungs}\text{-}\mathrm{Roboter}\text{-}\mathrm{Testplattform}$

KURT3D kann eine Fahrgeschwindigkeit von bis zu 5,4 $\frac{m}{s}$ erreichen. Für die Navigation verwendet KURT3D einen 2D-Laserscanner der auf einer Nickvorrichtung montiert ist und die Umgebung somit dreidimensional in einem Winkel von 180° in horizontaler Scanrichtung und 120° in vertikaler Scanrichtung erfassen kann. Die Kartenerstellung einer unbekannten Umgebung erfolgt nach SLAM⁵, dem gleichzeitigen Erfassen und Kartieren der Umgebung. Für die Hindernisvermeidung wurde von Lingemann et. al. [25] auf KURT3D ein Navigationsalgorithmus untersucht, der die Messdaten des 3D-Laserscanners nach Fuzzy-Regeln gewichtet und in Kapitel 4.3.4 beschrieben ist. Dieser Algorithmus wurde neben anderen Algorithmen auch im Rahmen dieser Arbeit verwendet.



Abb. 2.14: KURT3D auf Erkundungsfahrt (Quelle: [52])

⁵simultaneous localisation and mapping

Kapitel 3

Analyse des *ProfiBot*-Roboters

Die in dieser Diplomarbeit behandelten Entwicklungen basieren auf dem vom Fraunhofer Institut für Intelligente Analyse- und Informationssysteme entwickelten Roboterbaukastensystem *ProfiBot*. Daher werden in diesem Kapitel Konzept, Aufgaben und Funktionen von *ProfiBot* näher erläutert.

ProfiBot ist ein Roboterbaukastensystem, bestehend aus einem Basissystem, s. Abb. 3.1, und modularen Erweiterungen¹.



Abb. 3.1: ProfiBot Basissystem

Ein großer Vorteil des Baukastensystems ist, dass durch die modulare Struktur von *Pro-fiBot* die Montage und Applikation weiterer Komponenten leicht möglich ist. Dies vereinfacht Personen, die mit *ProfiBot* arbeiten, Ideen und Eigenentwicklungen in der Robotik in das Gesamtsystem von *ProfiBot* zu integrieren und Tests durchzuführen. Die nachfol-

¹Auf die Erweiterungen wird hier nicht näher eingegangen, da sie für die Entwicklungen in dieser Arbeit keine Rolle gespielt haben

genden Beschreibungen von *ProfiBot* beziehen sich auf das Basissystem, das Grundlage der Entwicklungen in dieser Arbeit ist.

3.1 Konzept

Das Roboterbaukastensystem $ProfiBot^2$ wurde konzipiert, um Auszubildenden der Mechatronik die Komponenten und den Aufbau mechatronischer Systeme zu vermitteln. ProfiBotwird u.a. in Berufskollegs eingesetzt, wo Berufsschüler und Berufsschülerinnen durch die Montage und Inbetriebnahme von ProfiBot im Unterricht theoretische Lerninhalte der Mechatronikausbildung in der Praxis vertiefen können. Nach der Montage und der Inbetriebnahme von ProfiBot können die Schüler und Schülerinnen Übungen mit Hilfe von ProfiBot durchführen.





Abb. 3.2: *Roberta*-Roboter

Abb. 3.3: sechsrädriger VolksBot

Ein ähnliches Projekt des Fraunhofer Instituts IAIS ist das Projekt *Roberta*³, das vor allem für Schülerinnen konzipiert wurde, um sie für die Wahl eines technischen Berufs zu begeistern, s. Abb. 3.2. Das *ProfiBot*-System ist eine Weiterentwicklung von *VolksBot*⁴ und beruht auf den gleichen Prinzipien. *VolksBot* ist ein Roboterbaukastensystem, das für Roboter-Prototyping in Forschung und Lehre entwickelt wurde. *VolksBot* kann durch den modularen Aufbau für Indoor- sowie Outdooreinsätze, s. Abb. 3.3, angepasst werden.

3.1.1 ProfiBot in der Mechatronikausbildung

Das *ProfiBot*-System soll möglichst alle Lernfelder, die im Rahmen der Ausbildung zum Mechatroniker und zur Mechatronikerin unterrichtet werden, abdecken. Die Lernfelder sind nachfolgend aufgelistet.

²http://www.profibot.de

³http://www.roberta-home.de

⁴http://www.volksbot.de

- 1. Analysieren von Funktionszusammenhängen in mechatronischen Systemen
- 2. Herstellen mechatronischer Teilsysteme
- 3. Installieren elektrischer Betriebsmittel unter Beachtung sicherheitstechnischer Aspekte
- 4. Untersuchungen der Energie- und Informationsflüsse in elektrischen, pneumatischen und hydraulischen Baugruppen
- 5. Kommunizieren mit Hilfe von Datenverarbeitungssystemen
- 6. Planen und Organisieren von Arbeitsabläufen
- 7. Realisieren mechatronischer Teilsysteme
- 8. Design und Erstellen mechatronischer Systeme
- 9. Untersuchen des Informationsflusses in komlexen mechatronischen Systemen
- 10. Planen der Montage und Demontage
- 11. Inbetriebnahme, Fehlersuche und Instandsetzung
- 12. Vorbeugende Instandhaltung
- 13. Übergabe von mechatronischen Systemen an Kunden

Alle diese Bereiche in ihrer Gesamtheit zu erfassen um daraus ein Lernobjekt zu entwickeln, hat den Vorteil, dass die Themen in der Mechatronikausbildung in diesem Fall fächerübergreifend angewendet werden, so wie sie auch im Berufsleben Anwendung finden.

Gegenwärtig werden von einigen *ProfiBot*-Projektpartnern Praxisversuche durchgeführt, in denen es darum geht, das Konzept von *ProfiBot* zu erproben und Verbesserungen und Weiterentwicklungen vorzuschlagen.

3.1.2 Das Baukastenprinzip

Die *ProfiBot*-Hardware wird als Bausatz ausgeliefert. Aus den Einzelkomponenten wird der Roboter schrittweise aufgebaut.

- Chassis
- Antriebseinheiten
- Schaltleisten und Sensoren
- Elektronische Komponenten

Nach dem Aufbau folgt die Softwareinstallation und der Betriebstest.

Die einzelnen Komponenten von *ProfiBot* werden in den folgenden Abschnitten dieses Kapitels beschrieben. Das Baukastensystem bietet viele Vorteile. So haben Personen, die die Montage von *ProfiBot* durchgeführt haben, eine bessere Kenntnis über den Aufbau und so das nötige Wissen, um Änderungen durchführen zu können. Weil *ProfiBot* vorwiegend in der Ausbildung eingesetzt wird, wurde bei der Entwicklung darauf geachtet, dass durch die anschauliche Bauweise ein leichtes Erkennen der Funktionsweise vieler Komponenten möglich wird.

3.2 Hardware

Die Hardware, bestehend aus Chassis, Motorcontroller, Sensorik, Aktorik und Steuerungskomponenten, wird im folgenden beschrieben. Zum Betrieb von *ProfiBot* ist ein Notebook, das den Systemanforderungen der Programmieroberfläche *IConnect*⁵ genügt, ausreichend.

3.2.1 Chassis

Das Chassis von *ProfiBot* besteht aus verschiedenen Komponenten (s. Abb. 3.4), die in einzelnen Arbeitsschritten montiert werden müssen. Für die Montage ist nur eine geringe Auswahl an Werkzeugen erforderlich, da der Roboterrahmen aus vielen gleichartigen Komponenten besteht. Die gute Übersichtlichkeit und Anschaulichkeit der Funktionalität lässt das Chassis auf den ersten Blick etwas unförmig aussehen.



Abb. 3.4: Chassis (1. Sensoraufbau, 2. Basisrahmen, 3. Bedienfeld, 4. Schublade)

Der Roboterrahmen ist aus Aluminiumprofilstangen aufgebaut (s. Abb. 3.5), die auf die erforderlichen Längen zugeschnitten sind. Die Profilstangen stammen aus einem handelsübli-

⁵http://www.micro-epsilon.de/de-de/Software/ICONNECT/

chen Verbindungssystem der Firma $Item^6$. Es ist auch möglich Profile und Verbinder der Firma $Bosch-Rexroth^7$ zu nutzen. Verbunden werden diese Profile mit Automatikverbindern⁸ (s. Abb. 3.5). Anbauten und Erweiterungen am Chassis des Basissystems, die z.B. auch im Rahmen dieser Diplomarbeit durchgeführt wurden, lassen sich durch die Montage weiterer Profilstangen auf einfache Weise durchführen.



Abb. 3.5: Verbund aus Profilstange (1) mit Automatikverbindern (2)

3.2.2 Energieversorgung

Die Versorgungspannung für fast alle elektronischen Komponenten, die im Lieferumfang des Basissystems enthalten sind, beträgt 24V. Die Spannung wird durch zwei in Reihe geschalteten Blei-Gel-Akkumulatoren erzeugt, die im Chassis eingebaut sind. Viele Sensoren und Aktoren, die im Rahmen von Weiterentwicklungen verwendet werden, benötigen jedoch eine andere Betriebsspannung als 24V, die durch Spannungswandler⁹ bereitgestellt werden. Die Versorgungsspannung der beiden Akkumulatoren beträgt meistens nicht genau 24V, sondern schwankt im Betrieb je nach Ladezustand.



Abb. 3.6: ProfiBot-Bedienfeld

Am Bedienfeld (s. Abb. 3.6) befindet sich der Hauptschalter, über den sich die Betriebs-

⁶http://www.item-international.com

⁷http://www.boschrexroth.com

⁸Automatikverbinder bestehen aus einer Hülse zum Eindrehen in die Nut und einem Nutenstein sowie einer Schraube mit M4-Gewinde.

⁹s. Kapitel 6

zustände Ein, Aus und Charge¹⁰ einstellen lassen. Der Betriebsschalter, ein Taster mit Leuchtsignal, mit dem der Motorcontroller angeschaltet wird, befindet sich ebenfalls dort. Die Spannungsversorgung der Motoren und der Auswertelektronik für die Schaltleisten wird durch den Motorcontroller durchgeschleift. Über eine LED auf dem Bedienfeld kann erkannt werden, ob der Ladezustand der Batterien noch für eine Versorgungsspannung von 24V ausreicht. Sobald die Versorgungsspannung der Akkumulatoren unter 23,7V absinkt, wird der Motorcontroller automatisch ausgeschaltet, jedoch nicht die anderen Verbraucher. Gleichzeitig wird durch Funktionslosigkeit des Hauptschalters gewährleistet, dass der Roboter aus Sicherheitsgründen nicht mehr eingeschaltet werden kann. Andererseits sind die Akkumulatoren in diesem Fall nicht vor Tiefenentladung geschützt.

Das Bedienfeld besitzt an der Rückseite verschiedene Buchsen zur Spannungsversorgung von Roboterkomponenten, insbesondere für den Motorcontroller.

3.2.3 Motorcontroller

Der Motorcontroller von *ProfiBot* ist ein eigens für den Einsatz in autonomen Robotern entwickelter Motorcontroller mit der Modelbezeichnung TMC200¹¹. Er wurde vom Fraunhofer Institut IAIS im Rahmen der *VolksBot*-Arbeiten entwickelt. Eine Auswahl der Spezifikationen ist nachfolgend aufgelistet:

Kontinuierlicher Maximalstrom pro Kanal:	8	[A]
PWM-Ausgangsspannung:	24	[V]
PWM-Frequenz:	20	[kHz]
Maximale Motorleistung je Motor:	150	[W]
Schnittstelle:	RS232	
Analoge Ausgänge:	3	
Encodereingänge:	3	

Im *ProfiBot*-Basissystem sind zwei der drei analogen Ausgänge durch die Motoren belegt, der dritte Ausgang kann für andere Zwecke verwendet werden (z.B. Schaltleisten). Weitere Vorteile des TMC200 für die Anwendung in der Robotik sind:

- Interne Drehzahlsynchronisation der angeschlossenen Motoren¹²
- Baugruppe zur Motorstrommessung
- Relativ geringe Kosten, da Eigenproduktion

 $^{^{10}\}mathrm{Im}$ Betriebsmodus **Charge** werden die Akkus durch ein angeschlossendes, externes Ladegerät geladen $^{11}\mathrm{Triple}$ Motor Controller 200

¹²Ohne Drehzahlsynchronisation der Motoren kann der Roboter bei gewähltem Antriebskonzept zur Seite driften

Die Motorspannung lässt sich über eine Pulsweitenmodulation einstellen, die über eine H-Brücke¹³ pro Ausgang moduliert wird. Um eine relativ gut geglättete Ausgangsspannung zu erhalten sind verschiedene Baugruppen wie Glättungskondensatoren und Ausgangsfilter zusätzlich auf der Platine untergebracht. Die Firmware des TMC200 läuft auf einem Mikrocontroller von Infineon¹⁴. Der TMC200 wird über die serielle Schnittstelle¹⁵ mit dem Notebook, auf dem die Steuerprogramme für den Betrieb des Roboters laufen, verbunden. Der TMC200 wird über eine Buchse an der Bedienfeldrückseite mit Spannung versorgt.

3.2.4 Aktorik

ProfiBot hat einen Differentialantrieb, der aus zwei bürstenbehafteten Gleichstrommotoren¹⁶ (DC-Motoren) besteht. Nachfolgend sind die wichtigsten Motordaten aufgelistet.

Nennspannung	30	[V]
Nenndrehzahl	6420	$[\min^{-1}]$
Nennmoment	$92,\!4$	[mNm]
Nennstrom	2,5	[A]

Zum Einsatz der Gleichstrommaschine im Projekt *ProfiBot* ist von großer Bedeutung, dass die Gleichtrommaschine für didaktische Zwecke geeignet ist, weil sie einfach aufgebaut ist. An einer Gleichstrommaschine kann die Funktionsweise einer elektrischen Maschine leicht veranschaulicht werden.

Auf jeder der beiden Motorwellen der Gleichstrommaschinen sitzt ein Planetengetriebe¹⁷, welches wiederum über eine Kupplung mit der Antriebskette verkoppelt ist. Davon angetrieben werden zwei Lufträder (s. Abb.3.7).



Abb. 3.7: Antriebseinheit (1 Luftrad, 2 Kupplung, 3 Getriebe, 4 DC-Motor, 5 Encoder)

¹³4-Quadranten-Steller

¹⁴Der Infineon C167 ist ein 16Bit Controller mit 20MHz Taktrate

¹⁵Momentan ist lediglich eine RS232-Schnittstelle implementiert, eine CAN-Schnittstelle ist vorgesehen
¹⁶Maxon RE 35

¹⁷Standartuntersetzung: 1:74, bei ProfiBot in dieser Arbeit: 1:43

Als Navigationsbasis für den Roboter können Odometrie-Daten verwendet werden. Diese Daten werden von einem Digital-Encoder geliefert, der fest mit den DC-Motoren. Der Encoder hat eine Auflösung von $500 \frac{\text{Ticks}}{\text{Umdrehung}}$. Die gefahrene Strecke wird mit Hilfe der Gleichungen 3.1 bis 3.5 berechnet, die u.a. im Skript zur Vorlesung *Verhaltensbasierte Robotik* von T. Bechtel [3] erläutert werden.

Drehzahl der Getriebewelle:

$$\omega_{mech} = I \cdot \omega_M$$

$$\omega_M = \text{Motordrehzahl in } \frac{\text{rad}}{\text{s}}, I = \text{Untersetzungsverhältnis}$$
(3.1)

Geschwindigkeit eines Luftrades:

۵

$$v_R = r_R \cdot \omega_{mech} \tag{3.2}$$

 $r_R = \text{Radius des Luftrades in m}$

Gefahrene Strecke eines Luftrades:

$$s_R = \int_{t_2}^{t_1} v_R \, dt = v_R \cdot t \tag{3.3}$$

Umrechnung der Encoderticks in die aktuelle Motordrehzahl:

$$\omega_M = \frac{C_e}{N_e \cdot t} \tag{3.4}$$

 $C_e =$ Encoderauflösung in Ticks, $N_e =$ aktuelle Ticks

Aus 3.1 bis 3.4 folgt für die gefahrene Strecke:

$$s_R = \frac{I \cdot C_e \cdot r_R}{N_e} \tag{3.5}$$

3.2.5 Sensorik

Die Sensorik beschränkt sich beim Basismodell des *ProfiBot*-Roboters auf zwei Schaltleisten¹⁸ (s. Abb. 3.8), die auf der Front- und Rückseite vom *ProfiBot*-Roboter angebracht wurden. Diese sind zum sofortigen Stopp bei Kollisionen vorgesehen. Das Funktionsprinzip von Schaltleisten ist in Abschnitt 5.4.1 beschrieben.

Die Schaltleisten werden über eine Auswertelektronik, die am dritten Encodereingang des Motorcontrollers angeschlossen ist, kontrolliert. Ein Kupferband liegt auf einem Potential von 5V, das andere Kupferband ist an GND^{19} angeschlossen. Nach dem Feststellen einer Kollision läuft eine Not-Stopp-Routine ab, welche die Hindernisse oder auch den Roboter vor weiteren Schäden schützt. Damit ein Leitungsbruch detektiert werden kann, fließt bei offenem Schalter ein Strom (s. Abb. 3.9). Dieser Strom lässt sich nach dem Ohmschen Gesetz wie folgt berechnen: $\frac{5V}{1.5k\Omega} = 3.3mA$.

¹⁸Die Gummiprofile und Bandschalter werden von der Firma *Tapeswitch* hergestellt

¹⁹Erdpotential



Abb. 3.8: Schaltleiste (1 Montageschiene, 2 Bandschalter, 3 Schutzleiste)

Abb. 3.9: Bandschalter - Ersatzschaltbild

Die Schaltleisten sind wegen den großen Abmessungen des *ProfiBot*-Roboters unterdimensioniert. Die Aufprallfläche der Schaltleisten ist lediglich 25 mm breit und sie sind in einer Höhe von 170 mm vorne und 105 mm hinten über Grund montiert. Hindernisse können nur erkannt werden, wenn diese so hoch sind wie die Montagehöhe der Schaltleisten. In einer solchen Position können z.B. keine Füße erkannt werden, die der Roboter überfahren würde. Um eine höhere Sicherheit beim Navigieren zu gewährleisten, ist es notwendig, die Position der Schaltleisten zu verändern.

Für den Anschluss weiterer Sensoren ist ein Buskoppler²⁰ vorgesehen. Zur Auswahl sind zwei analoge und acht digitale Eingänge verfügbar. Die Sensordaten werden über ein Ethernet-Protokoll zur Auswertung an das Notebook versendet, die Software zur Auswertung wird in Kapitel 3.3 beschrieben. Alternativ zum Buskoppler kann man auch ein preisgünstigeres USB-Experimentierboard²¹ benutzen, um Messwerte von Sensoren zu erfassen. Hierbei ist eine Einbindung in die Programmierumgebung *IConnect* leider noch nicht vollständig gegeben.

3.3 Software

Die Hardware von *ProfiBot* wird über die geeignete Software bedient. Hierbei muss man unterscheiden, ob *ProfiBot* von Anwendern und Anwenderinnen bedient wird oder von Entwicklern und Entwicklerinnen (s. Tabelle 3.1).

Ein Anwender und eine Anwenderin bedient *ProfiBot* hauptsächlich mit der grafischen Programmierumgebung *IConnect*²². Da man in einer solchen Umgebung schnell komplette Benutzungsoberflächen erstellen kann, benötigt man keine besonderen Programmierkenntnisse. Es können auch Roboterverhaltensweisen programmiert werden, die den Roboter

²⁰Buskoppler FL IL 24 BK-B-PAK von der Firma Phoenix Contact, http://www.phoenix-contact.de

 $^{^{21}}$ Vellemann-Bausatz, http://www.vellemann.be

 $^{^{22}{\}it IConnect}$ wird von der Firma Micro-Epsilon vertrieben.

z.B. auf bestimmte Ereignisse reagieren lassen. Dafür muss die entsprechende Hardware vorhanden und in die Programmieroberfläche eingebunden worden sein.

Die einfache Handhabung ist ein großer Vorteil, da sich so in der Ausbildung Übungsgruppen und Praktika effektiver gestalten lassen. Auf der Anwendungsebene müssen Kenntnisse über das Gesamtsystem vorhanden sein, jedoch können Eigenschaften und Funktionen von *ProfiBot* nur geringfügig beeinflusst werden, was bei der beruflichen Mechatronik-Ausbildung ausreicht.

Wenn die *ProfiBot*-Plattform als Basis für Entwicklungen im Bereich der Robotik eingesetzt und neue Hardware integriert wird, reichen die Kenntnisse auf Anwendungsebene nicht mehr aus. Entwickler und Entwicklerinnen brauchen detaillierte Kenntnisse über den Aufbau und die Funktionsweise der Hardware von *ProfiBot* sowie über die Programmiersprachen C und C++. Weiterhin benötigen sie Kenntnisse über die Vernetzung der vorhandenen Hardware. Sie müssen außerdem wissen, welche Applikationen für die Hardware vorhanden sind, und sie müssen im Stande sein, diese modifizieren zu können.

	Anwender/Anwenderinnen	ENTWICKLER/ENTWICKLERINNEN
EINSATZORTE	Lehrwerkstätten, Berufskollegs	Hochschulen, Institutionen
Einsatz	Schulungszwecke	Roboter-Prototyping
Kenntnisse	Gesamtsystem, IConnect	gute Programmierkenntnisse
		Hardwarekomponenten

Tabelle 3.1: Einsatzmöglichkeiten von ProfiBot

3.3.1 Programmierung mit IConnect

Roboter und Maschinen werden vorwiegend mit anwendungsfreundlichen Programmen bedient und gesteuert. So genannte grafische Programmierumgebungen bieten sich dazu an. Früher wurden Steuerprogramme für Roboter und Maschinen hauptsächlich in textbasierten Programmiersprachen geschrieben. Grafische Programmierumgebungen übernehmen in den meisten Fällen auch die Schnittstelle zwischen der Sensorik und Aktorik von Robotern und Maschinen und dem Computer. Das geschieht mit Hilfe von Steckkarten für den Computer, z.B. A/D-Wandlern, die meistens vom gleichen Hersteller sind wie das Programm selbst. Sind Karten und Programm von unterschiedlichen Herstellern, ist in manchen Fällen ein größerer, aber einmaliger Programmieraufwand nötig.

IConnect ist eine solche grafische Programmieroberfläche, die nach dem Signalflussprinzip arbeitet. Hierbei werden die Programme nicht in Textform, sondern mit Hilfe sogenannter Module verknüpft, die in einem Blockschaltbild bzw. so genannten *Signal*graphen graphisch dargestellt werden. Mehrere Module können zu einem einzigen Modul,
einem sognannten Macro, zusammengefasst werden, um den Signalgraphen übersichtlicher zu machen. In *IConnect* gibt es eine Vielzahl werksseitig programmierter Module für die Signalflusssteuerung, beispielsweise ein PID-Reglermodul, oder Module zum Einlesen von Daten eines CAN²³-USB Interfaces vom Hersteller EMS. Mit dem Modul DisplayManager können Bedien- und Anzeigeoberflächen erstellt werden. Des Weiteren gibt es die Möglichkeit, den Modulen Prioritäten von 1 (höchste Priorität) bis 255 (niedrigste Priorität) zu geben. Eine nähere Beschreibung von IConnect ist im Handbuch [30] zu finden. Für den Entwickler und die Entwicklerin bietet IConnect die Möglichkeit, selbst Module zu entwickeln. Mit dem Modul Interpret können Ein- und Ausgänge selbst definiert werden und die Datenverarbeitung der Signale in einer simplen, textbasierten Programmiersprache realisieren. Fast alle im Rahmen dieser Arbeit entwickelten Module wurden mit dem Modul Interpret erstellt. Außerdem gibt es die Möglichkeit, Module in der Sprache C++ zu programmieren. Hierbei können alle Vorteile objektorientierter Programmierung ausgenutzt werden, beispielsweise das Benutzen von Strukturen und Klassen. Weil IConnect für viele Eigenschaften keine entsprechenden Module bietet, beispielsweise für den Datenempfang von Buskopplern, sind selbstgeschriebene Module eine Möglichkeit, die geeigneten Funktionen zu realisieren. Im Rahmen dieser Arbeit wurden solche Module mit Hilfe eines werksseitigen Beispiel-Moduls erstellt.

Um intelligentes Verhalten in der Robotersteuerung von ProfiBot einzusetzen, sind selbst geschriebene Module notwendig. Intelligente Roboterverhaltensweisen können durch Zustandsautomaten realisiert werden, oder durch eine Subsumptionsarchitektur (s. Kapitel 4.3).

IConnect wird auch als Steuerungssoftware für das *ProfiBot*-Basissystem eingesetzt, hierfür ist es auf dem Notebook von *ProfiBot* installiert. Die Software wird auch in dem modifizierten *ProfiBot*-Roboter weiterhin als Programmierumgebung verwendet. Die nachfolgenden Absätze beschreiben einige für das *ProfiBot*-Basissystem entwickelte *IConnect*-Module. Weiterhin wird untersucht, inwiefern diese Module in dem für die vorliegende Arbeit modifizierten *ProfiBot*-Roboter eingesetzt werden können.

²³controler area network

TMC-Modul



Abb. 3.10: Screenshot IConnect-Modul TMC200

Das Modul für den TMC200 besitzt drei Eingänge für die jeweiligen Soll-Drehzahlen der Motoren. Modulausgangsseitig ist eine Ausgabe verschiedener Zustandsgrößen der Motoren realisiert worden, die wichtigsten Größen für die Realisierung der Modifikationen im Rahmen dieser Arbeit sind die absoluten und relativen Ticks der Encoder und der momentane Motorstrom. Dieses Modul entnimmt die jeweiligen Daten für alle drei Motoren aus dem Datenpuffer des Application Programming Interface (API), der Schnittstelle zu dem TMC200. Hardwareseitig ist eine Aktualisierung jedes einzelnen Datentripels alle 10ms realisiert, d.h. wenn drei Zustandsgrößen abgefragt werden, wird ein Datentripel alle 30ms aktualisiert. Für die Ausgabe der einzelnen Zustandsgrößen sind an dem TMC-Modul Eingänge vorhanden. Mittels binärem Signal an den jeweiligen Eingängen des TMC200 kann die Ausgabe der Zustandsgrößen ein- und ausgeschaltet werden. Das TMC-Modul kann auch für den zu entwickelnden *ProfiBot*-Roboter eingesetzt werden.

Buskoppler-Modul



Abb. 3.11: Screenshot IConnect-Modul ILBETH

Das Modul *ILBETH* unterstützt den Betrieb des Buskopplers *FL IL 24 BK-B-PAK* des Herstellers *Phoenix Contact* (s. Abb. 3.12). Das Modul gibt die Messdaten der an dem Busskoppler angeschlossenen Sensoren aus. Für das Auslesen der Sensor-Ausgangssignale gibt es die Möglichkeit, den Buskoppler mit Inline-Klemmen zu erweitern. Zum *ProfiBot*-Basissystem gehört eine Klemme mit acht digitalen Eingängen, die für den Betrieb von Sensoren mit Schaltausgängen vorgesehen ist. Die digitalen Eingangssignale werden über hochohmige Widerstände eingelesen. Bei einer Messspannung von 15V...24V wird das Eingangssignal als High-Pegel, bei einer Messpannung von 0...5V als Low-Pegel interpretiert. Des Weiteren gibt es die Möglichkeit, Inline-Klemmen mit einem A/D-Wandler an den Buskoppler anzuschliessen und Spannungen im wählbaren Bereich von 0V...10V, $\pm 10V$ oder Ströme im Bereich von 0mA...20mA, 4mA...20mA oder $\pm 20mA$ einzulesen. Die im *ProfiBot*-Roboter benutzte Inline-Klemme²⁴ besitzt zwei analoge Eingänge (s. Abb. 3.13).





Abb. 3.12: Buskoppler (Quelle: [36])

Abb. 3.13: A/D Inline-Klemme (Quelle: [37])

Weitere Beschreibungen, wie die Anschlussbelegungen des Buskopplers und die der Inline-Klemmen, finden sich in den Handbüchern [36] und [37]. Das Modul *ILBETH* empfängt die Daten über eine Ethernet-Verbindung. Analoge Daten werden in *IConnect* als digitale Signale in der Einheit mV bzw. mA ausgegeben, bei digitalen Daten ist 0 ein Low-Pegel und 1 ein High-Pegel. Das Modul ILBETH ist für den Empfang der Sensordaten im *ProfiBot*-Roboter verwendbar.

Joystick-Module

Die Module zum Betrieb eines Joysticks werten die translatorischen und rotatorischen Bewegungen eines Joysticks aus. Zunächst wird in einem Interpret-Modul Rotation und Geschwindigkeit aus den Joystickdaten eines Empfangsmoduls berechnet, im Anschluss werden die Daten skaliert. Nach der Skalierung erfolgt eine Berechnung der Motorsolldrehzahlen. Die Berechnungen beziehen sich auf den kinematischen Schwerpunkt des Roboters ohne den Abstand zwischen den Rädern zu berücksichtigen. Die Joystick-Module können im Rahmen dieser Arbeit nicht verwendet werden, weil für den *ProfiBot*-Roboter im Rahmen dieser Arbeit keine tele-operierende Steuerung vorgesehen ist.

²⁴ Phoenix-Contact Modellnummer: IB IL AI 2/SF

3.3.2 Firmware und Regelalgorithmus

In diesem Kapitel werden die zwei wichtigsten Funktionen der Firmware behandelt, zum einen die Integration eines kinematischen Modells, zum anderen die Reglerstruktur des TMC200.

Die Firmware des TMC200 beinhaltet ein komplettes kinematisches Modell von Robotern mit Differentialantrieb oder omnidirektionalem Antrieb [34]. Für den *ProfiBot*-Roboter ist ausschließlich das Modell des Differentialantriebes von Interesse. Dem Modell müssen roboterspezifische Werte, wie beispielsweise Radabstand l_R und Raddurchmesser d_R bekannt sein. Innerhalb der Steuersoftware müssen lediglich die translatorische und rotatorische Geschwindigkeit, v_K und ω_K , des kinematischen Schwerpunkts vorgegeben werden. Die notwendigen Geschwindigkeiten der Einzelräder werden beim *ProfiBot*-Roboter mit den Gleichungen 3.6 berechnet, die der Quelle [34] entnommen wurden.

$$v_l = v_K + \frac{l_R}{2} \cdot \omega_K$$

$$v_r = -v_K + \frac{l_R}{2} \cdot \omega_K$$
(3.6)

Weiterhin ist innerhalb der Firmware ein PID-Regler integriert, der die in 3.7 beschriebene, Laplace-transformierten Übertragungsfunktion hat.

$$G_{R(s)} = K_P + \frac{1}{T_I \cdot s} + T_D \cdot s \tag{3.7}$$

Bei der Reglerstruktur handelt es sich um einen einschleifigen Drehzahlregelkreis mit einem parallelen Stromregelkreis, welcher ausschließlich die Funktion hat, den Motorstrom beim Überschreiten einer bestimmten Motortemperatur zu begrenzen. P- und I-Anteil des Drehzahlreglers wurden empirisch ermittelt. Der D-Anteil wurde aus Stabilitätsgründen und wegen der hohen benötigten Stellamplitude der Motorspannung bei nahezu sprungförmigen Änderungen der Führungsgröße auf 0 gesetzt. Eine unterlagerte Momentregelung ist nicht realisiert worden, weil nach Paulus [34] nur der Betrag des Motorstroms gemessen wird und somit nicht die Drehrichtungen eines Motors bestimmt werden kann.

Durch die eingesetzten Encoder kann die Drehrichtung bestimmt werden. Ein Tick berechnet sich aus zwei periodischen Rechtecksignalen A und B, die um eine Phasenverschiebung von $\frac{\pi}{2}$ zueinander versetzt sind und eine Pulsweite von 50% haben. Die eine Drehrichtung wäre erkannt, wenn eine positive Flanke von A gleichzeitig mit einem Lowpegel von B erkannt werden würde. Wenn eine positive Flanke A gleichzeitig mit einem High-Pegel von B erkannt werden würde, wäre die andere Drehrichtung ermittelt (s. Abbildung 3.14).



Abb. 3.14: Drehrichtungserkennung durch Flankenauswertung

Die Reglerausgangsgröße wird im Drehzahlregelkreis mit einer Anti-Reset Windup (ARW) Maßnahme auf den Maximalwert der Stellgröße begrenzt. Bei dieser Maßnahme ist der I-Anteil null, wenn der Maximalwert der Stellgröße erreicht ist. Der Vorteil dieses einschleifigen Drehzahlregelkreises ist, dass keine hohe Rechenleistung erforderlich und somit eine schnelle Regelzykluszeit von 4ms realisierbar ist. Durch die fehlende Momentregelung besteht die Gefahr, dass der Strom bei Laständerungen im Fahrbetrieb sehr stark ansteigen kann. Das kann zu ruckartigen Bewegungen des Roboters führen, verursacht durch instabiles Regelverhalten.

Kapitel 4

Sichere Navigation

4.1 Begriffseinordnung

Das Wort Navigation¹ beschreibt nach Brockhaus [7] "das Führen eines Wasser-, Luftoder Raumfahrzeugs von einem Ausgangsort auf bestimmtem Weg zu einem Zielort, einschließlich der dazu erforderlichen Mess- und Rechenvorgänge zur Bestimmung des augenblicklichen Standortes (Ortung) und des Kurses".

Die Navigationsstrategie ist die Planung und Fahrzeugüberwachung in möglichst optimaler Weise [7]. Dazu gehören das

- Erreichen bestimmter Orte zu bestimmten Zeiten;
- Ermitteln des bereits zurückgelegten Wegs;
- Einhalten eines vorgeschriebenen Kurses;
- schnelle, treibstoffsparende und sichere Erreichen eines Ziels

Man unterscheidet die Arten der Navigation anhand ihrer messtechnischen Hilfsmittel. Hiermit sind die Hilfsmittel gemeint, mit der die Navigation durchgeführt wird (z.B. Sensoren und Methoden, mit denen die Messdaten verarbeitet werden). Nach VDI-Richtlinien [50] kann man die Navigation zusammenfassend in zwei unterschiedliche Arten aufteilen:

1. KOPPELNAVIGATION:

Bei der Koppelnavigation können sämtliche erforderlichen Meß- und Rechenvorgänge an Bord des Fahrzeuges vorgenommen werden. Die Koppelnavigation, die auch bordautonome Navigation genannt wird, bezieht sich auf die Bestimmung der Lage des

¹lat. Schifffahrt

Fahrzeuges mit Hilfe fahrzeuginterner Koppelsensorik² [50]. Die Lagebestimmung wird meistens in einem kartesischen Bezugssystem durchgeführt (s. Abb. 4.1). In kleinen Zeitintervallen werden bei der *Koppelnavigation* die Bewegungsgrößen des Fahrzeuges integriert und somit die Position und Lage des Fahrzeuges bestimmt (s. Dead-Reckoning-Verfahren, Kapitel 2.3). Allerdings besteht das Problem, dass das kartesische Bezugssystem nicht kongruent zur realen Umgebung ist [32]. Im übergeordneten Gebilde des kartesischen Bezugssystems wird bei der Koppelnavigation präzise gemessen. Wegen des Schlupfes zwischen Fahrbahn und Rädern ist die *Koppelnavigation* in der praktischen Anwendung bei Bodenfahrzeugen jedoch fehlerbehaftet, da systematische Messfehler (und somit Positionsfehler) mit der Zeit ansteigen. Der Vorteil der *Koppelnavigation* liegt in der ständigen Präsenz der Navigationsdaten unabhängig von der Umgebung.

2. LAGEPEILUNG:

Unter Lagepeilung versteht man das Navigieren durch Peilung stationärer Marken, die zur Positions- und Kursbestimmung ausgenutzt werden. Die Lage des Fahrzeuges wird durch sogenannte Peil- oder Landmarken bestimmt (s. Abb. 4.2). Bodenfreie Landmarken sind von größter Bedeutung, weil sie eine hohe Flexibilität gewährleisten. Zu den Navigationsverfahren gehören u.a. die terrestrische Navigation, die Astronavigation, die Funknavigation und die Satellitennavigation. Die terrestrische Navigation bedient sich naturgegebener Eigenschaften der Erde, beispielsweise des Erdmagnetfelds. Die Astronavigation behilft sich mit den Positionen der Sterne. Bei der Satellitennavigation wird auf künstliche Hilfsmittel zurückgegriffen (z.B. Satelliten beim GPS³). Weil sich die Lagepeilung an Punkten der realen Welt orientiert, liegt die Lage als Absolutwert vor. Somit ist das Bezugssystem für eine Navigation mit der realen Welt kongruent.



Abb. 4.1: Koppelnavigation

Abb. 4.2: Lagepeilung mit Landmarken

²Die Koppelsensorik ist eine an das Fahrzeug gekoppelte Sensorik ³global positioning system

Um die Vorteile beider Verfahren nutzen zu können, kann auch eine Kombination aus der *Koppelnavigation* und der *Lagepeilung* angewendet werden.

Weiterhin wird zwischen lokaler und globaler Navigation unterschieden[45]. Lokale Navigation beschäftigt sich mit der Navigation innerhalb der Sichtweite oder der Reichweite der Sensoren eines Fahrzeuges, globale Navigation bezieht sich auf die Navigation anhand eines Weltmodells, welches z.B. auf Karten basieren kann. Wenn zur Navigation ausschließlich eine Distanzsensorik genutzt wird, wie das auch bei dem modifizierten ProfiBot der Fall ist (s. Kapitel "Sensorik"), können in Karten nur dynamische Objekte verzeichnet sein, die sich im Erfassungsbereich der Distanzsensorik befinden. Die Hindernisvermeidung und die damit verbundene sichere Navigation kann folglich nur im Rahmen der lokalen Navigation durchgeführt werden, oder mit einer Kombination aus globaler und lokaler Navigation. Mit Hilfe der Satellitennavigation können zwar auch dynamische Objekte auf Karten erkannt werden. Die Anforderungen an die Genauigkeit der Positionsbestimmung für eine Hindernisvermeidung reichen jedoch nicht aus. Die Satellitennavigation funktioniert desweiteren nicht in geschlossenen Räumen, in denen der modifizierte ProfiBot-Roboter eingesetzt werden soll. Die wichtigsten Merkmale lokaler und globaler Navigation sind nachfolgend aufgelistet:

LOKALE NAVIGATION

- Navigation mit Distanzsensorik
- Hohe Echtzeitanforderungen
- Bezieht sich auf nähere Umgebung
- Keine optimale Routenplanung
- Reaktiv

GLOBALE NAVIGATION

- Navigation mittels Weltmodell
- Geringe Echtzeitanforderungen
- Bezieht sich auf eine vorhandene Karte der Umgebung
- Optimale Routenplanung
- Deliberativ⁴

⁴lat. deliberare: "überlegen, beschließen", im Bezug auf die *globale Navigation* ist hier ein weitläufig im Voraus planendes Verhalten des Navigationssystems gemeint

Das Zusammenspiel lokaler und globaler Navigation ist schwierig zu implementieren. Im Allgemeinen ist unklar, wie die deliberativen und reaktiven Komponenten zusammenarbeiten sollen [45]. Dieses Zusammenspiel benötigt eine umfassende Steuerungsarchitektur, die im Rahmen dieser Arbeit jedoch nicht erarbeitet werden kann.

Die Hindenisvermeidung wird im Rahmen der Anforderungen für das *ProfiBot*-System ausschließlich auf der Basis der *lokalen Navigation* realisiert.

4.2 Roboternavigation

Hinsichtlich der sicheren Navigation werden die Definitionen zur Navigation zweckmäßig für FTS und AMR, die als Sonderfälle von FTS bezeichnet werden können, näher betrachtet. Beide verbindet das gleiche Konzept, nämlich eine berührungslose Führung und Steuerung durch eine Umgebung. Die primäre Aufgabe eines FTS ist der Materialtransport in innerbetrieblichen Hallen. Ein AMR kann außerdem noch für weitere Aufgaben in anderen Umgebungen eingesetzt werden. Er kann beispielsweise zusätzlich Handhabungsfähigkeiten besitzen, die er mit Hilfe von Manipulatoren durchführt. Im Wesentlichen besitzen AMR im Unterschied zu FTS einen höheren Grad an Autonomie und Flexibilität.

Um die Navigation eines AMR steuern zu können, muss die Pose des Roboters in Bezug auf eine gewünschte Spur oder Zielposition erfasst und geregelt werden [50].

Damit ein AMR autonom navigieren kann, müssen ausreichend Informationen über

- den Standort,
- den Zielort und
- die Möglichkeiten den Zielort zu erreichen,

vorhanden sein [10]. Im Folgenden werden einige Methoden zur technischen Realisierung der Aspekte benannt. Außerdem werden die Methoden im Hinblick auf die Navigationsverfahren analysiert und entschieden, welche der Methoden im Rahmen dieser Arbeit angewendet werden kann.

Einen mobilen Roboter oder ein FTS auf ein bestimmtes Ziel hin zu bewegen kann durch das Führen auf einer fest installierten Spur erfolgen [32]. Eine solche Navigation ist mit wenig Rechenaufwand durchzuführen, man benötigt keine komplexen Algorithmen zur Routenplanung. Jedoch besitzen diese Fahrzeuge nicht die Fähigkeit sich selbständig in ihrer Umgebung zurechtzufinden, sie sind deshalb nicht autonom, sondern werden automatisch betrieben [32]. Sogenannte physische Leitlinien dienen den Fahrzeugen zur Wegorientierung. Hierzu gehören:

• Induktionsschleifen;

- Magnete im Boden;
- Markierungen auf dem Boden.

Die Navigation über eine physische Leitlinie wird oft in Fabrikhallen mit FTS genutzt. Für den Betrieb der Fahrzeuge wird eine speziell auf die Fahrzeuge abgestimmte Umgebung benötigt, diese Art der Navigation ist unflexibel. Infolge einer Änderung der Fahrzeugroute abseits der ursprünglichen Leitlinie müssen oft die Umgebungsbedingungen verändert werden, was sehr kostspielig und zeitaufwändig sein kann. Dieses Verfahren wird nicht in dieser Arbeit angewendet, denn die zu entwickelnde Roboterplattform soll in ebenen Räumen mit unbekannter Ausstattung eingesetzt werden. Des Weiteren ist es auch nicht möglich, ausschließlich mit diesem Verfahren Hindernissen auszuweichen, wenn diese zufällig ihre Positionen wechseln.

Bewegt sich ein Fahrzeug vollständig autonom in einer Umgebung, sind dem Fahrzeug anderweitige Fähigkeiten zur Navigation gegeben. In den folgenden Kapiteln wird beschrieben, wie diese Methoden in das Konzept des modifizierten *ProfiBot* einfließen können.

4.2.1 Kartographie

Die Kartographie ist ein wichtiges Verfahren in der Erkundungsrobotik. Der Grad an Autonomie eines AMR wächst mit der Fähigkeit, präzise Karten von seiner Umgebung erstellen zu können. Bei AMR werden vorwiegend 2D-Karten benutzt. Gegenüber 3D-Karten können mit 2D-Karten die Sensordaten leichter ausgewertet werden und erforderen einen geringeren Rechenaufwand für die Routenplanung. Mit 3D-Karten ist eine räumliche Darstellung der Umgebung möglich.

Unterschieden wird zwischen geographischen Karten⁵ und topologischen Karten. Geografische Karten liefern im Allgemeinen Informationen über Freiflächen und über Hindernisse, die mit Hilfe von Polygonen dargestellt werden. AMR können immer nur eine Seite eines Hindernisses betrachten, die *globale Navigation* wird dadurch kompliziert. Topologische Karten abstrahieren dagegen Hindernisse als Objekte und klassifizieren diese. Hierdurch können z.B. Sackgassen auch als diese erkannt und alternative Routen gefunden werden. In Karten werden Informationen über Objekte gespeichert, sie benötigen daher eine bestimmte Speicherkapazität im System eines AMR.

In Bezug auf die Verfahren, Algorithmen und Hilfsmittel, wird zwischen lokaler und globaler Kartographie unterschieden [22]. Lokale Kartographie bezeichnet man auch als Modellierung der Umgebung, bei der die abgetasteten Umgebungsstrukturen in einem roboterzentrischen Koordinatensystem abgebildet werden. Globale Kartographie repräsentiert die Umgebung in einem statischen Koordinatensystem. Das Navigationssystem erkennt den AMR als dynamisches Objekt auf der Karte.

⁵geographische Karten werden auch metrische Karten genannt

Zur Kartographie eines unbekannten Terrains werden die Koppelnavigation und die Lagepeilung (s. Kapitel 4.1) verwendet. Um die Verfahren zu optimieren, können beide Verfahren miteinander kombiniert werden. Im Rahmen der Koppelnavigation wird als Bezugssystem zumeist ein kartesisches Koordinatensystem verwendet. Ein Verfahren der Kartographie ist es, das Koordinatensystem in Evidenzraster aufzuteilen [32] (s. Abb. 4.3). Der Status der Evidenzraster wird anfänglich als unbekannt initialisiert. Während der AMR durch die Umgebung fährt und sie erkundet, benutzt er die Odometrie, um Informationen darüber zu erhalten, in welcher Zelle des Rasters er sich im Augenblick befindet. Mit Hilfe der Koppelnavigation können nur statische Hindernisse verzeichnet werden. Die Umgebung muss bei der Erkundung und Kartenerstellung frei von dynamischen Hindernissen sein, weil diese ansonsten auch auf der Karte als statische Hindernisse verzeichnet werden würden. Durch eine entsprechende Sensorik kann das Navigationsmodul des AMR erkennen, in welchen der Zellen Hindernisse zu erwarten sind. Hindernisse erkennt das Robotersystem über Entfernungssensoren, wie z.B. Ultraschallsensoren, Lasertriangulationssensoren, Laserscanner etc. oder durch berührungsbehaftete Sensoren, wie z.B. Schaltleisten. Das Navigationsmodul markiert alle Zellen, die der AMR bereits befahren hat, als bekannt. Die Genauigkeit der so entstehenden Karte, ist abhängig von der Präzision der Odometriesensorik, die Auflösung der Karte ist abhängig von der Feinheit des Rasters.



Abb. 4.3: Kartenerstellung in einem Evidenzraster in Anlehung an [32]

Mit der Feinheit des Rasters steigt die benötigte Rechenleistung. Im Hinblick auf die Echtzeitanforderungen, "rechtzeitig" und "gleichzeitig" auf Ereignisse zu reagieren, sind für die *lokale Navigation* schnelle und kostspielige Rechner von Nöten.

Ein AMR kann andererseits mit der Distanzsensorik eine Karte erstellen. Ein mögliches Verfahren ist SLAM.

Der modifizierte ProfiBot-Roboter muss im Hinblick auf eine sichere Navigation seine

Umwelt interpretieren können. Dafür wird eine Karte benötigt. Er benötigt sie weiterhin zur Hinderniserkennenung. Diese Karte kann sehr einfach strukturiert sein. Distanz und Winkel von Hindernissen sind ausreichende Informationen, um eine Hindernisvermeidung durchzuführen. Eine zweidimensionale Punktmenge, die in einem Bezugssystem dargestellt wird und zu jeder Zeit verfügbar ist, genügt.

4.2.2 Lokalisationsfähigkeit

Um die genaue Selbstlokalisation innerhalb einer Umgebung zu ermöglichen, sollte der AMR anhand seiner Distanzsensorik erkennen, wo er sich befindet. Die Umgebung wird als bekannt vorausgesetzt, folglich muss z.B. eine Karte vorhanden sein. Die Selbstlokalisation kann mittels Dead-Reckoning-Verfahren (s. Kapitel 2.3) erfolgen. Dieses Lokalisationsverfahren gehört zur Kategorie *Koppelnavigation*. Wegen der fehlerbehafteten Auswertung der Odometrie gibt es Verfahren, die anhand von lokalen Landmarken eine Selbstlokalisation ermöglichen. Hierzu muss der AMR in der Lage sein Landmarken zu erkennen. Es gibt zwei mögliche Verfahren zur Selbstlokalisation mittels Landmarken. Die Position wird entweder durch Distanzmessung zu bekannten Landmarken bestimmt, oder es werden die Winkel zwischen Roboter und bekannten Landmarken gemessen. Um die aktuelle Position zu bestimmen, ist es nötig zunächst die Sensordaten auszulesen, aufzubereiten und auszuwerten. Daraufhin ist es erforderlich die Sensordaten mit einer Karte zu vergleichen. Die Selbstlokalisation mittels Landmarken wird als *Lagepeilung* durchgeführt, weil es sich um prägnante Punkte in der Umgebung handelt, die den Roboter anpeilen oder von ihm angepeilt werden.

Es wird zwischen natürlichen und künstlichen Landmarken unterschieden. Natürliche Landmarken werden nicht primär für die Navigation erzeugt, sondern existieren schon im Vorfeld als Objekte in der Umgebung. Ihre Positionen sind im Weltmodell verzeichnet [45]. Künstliche Landmarken werden eigens für den Zweck der Navigation im Umfeld von autonomen, mobilen Robotern eingesetzt, um die Lokalisierung zu erleichtern [45]. Künstliche Landmarken werden weiterhin unterteilt in aktive und passive Landmarken. Aktive, künstliche Landmarken⁶ senden Signale aus, die von Fahrzeugen zur Navigationsunterstützung interpretiert werden. Das können z.B. Peilsender, Ampeln oder Leuchttürme etc. sein. Für die Navigation von AMR sind Peilsender von größter Bedeutung. Die Signale von Peilsendern können elektromagnetische Wellen, Schall oder Licht sein. Die Signale werden gepulst. AMR erkennen anhand der Laufzeitunterschiede (s. Abb. 4.4), bedingt durch unterschiedliche Positionen der Peilsender, ihre eigene Position. Durch Modulation der Signale können die Peilsender eindeutig identifiziert werden. Passive, künstliche Landmarken senden keine Signale aus und verarbeiten auch keine Informationen. Abbildung 4.5 verdeutlicht die Klassifizierung der Landmarken.

⁶auch Baken genannt



Abb. 4.4: Unterschiedliche Laufzeiten bei Peilsendern

Abb. 4.5: Klassifizierung der Landmarken

In Bezug auf den modifizierten *ProfiBot* ist eine Selbstlokalisierung ausschließlich für eine Hindernisvermeidung notwendig. Informationen über den augenblicklichen Standort des modifizierten *ProfiBot* müssen in Relation zu Hindernissen vorhanden sein.

4.2.3 Routen- und Bahnplanung

Bei der Routenplanung wird in einem vorhandenen Umgebungsmodell, z.B. in einer Karte, ein Weg gesucht, der von einem Startpunkt zu einem Zielpunkt führt [23]. Im Allgemeinen stellt der Startpunkt den aktuellen Aufenthaltsort des Roboters beim Beginn der Routenplanung dar. Die Routenplanung kann anhand von Rastermodellen oder Vektormodellen erfolgen. Auf globalen Karten ist eine Routenplanung für AMR einfach durchzuführen, für das dynamisch dargestellte Modell des Roboters muss auf Grund von feststehenden Zielkoordinaten auf der Karte lediglich ein statisches Raster- oder Vektormodell erstellt werden. Auf lokalen Karten erweist sich die Routenplanung als schwieriger und rechenaufwändiger, weil die Modelle in ein globales Koordinatensystem transformiert werden müssen.

An die Routenplanung setzt direkt die Bahnplanung an und berechnet die für die Steuerung notwendigen Parameter entlang des ermittelten Weges [23]. Die mechanische Konstruktion eines AMR spielt dabei eine große Rolle, physikalische und mechanische Eigenschaften, wie z.B. Masse, Abmaße, Lage des kinematischen Zentrums und Antriebsart des Robotersystems, fließen mit in die Bahnberechnung ein. Steht die Bahn fest, folgt ein AMR der Bahn auf der Trajektorie⁷. Trajektorien beschreiben in der Physik die Bewegung eines Teilchens und beinhalten Informationen über Ort und Geschwindigkeit [7], um einer Bahn zu folgen. Auf ein AMR bezogen bedeutet dies die Bewegung unter Berücksichtigung der Pose und der translatorischen und rotatorischen Geschwindigkeit des AMR.

Es ist zu untersuchen, ob das Benutzen einer Routen- und Bahnplanung im Rahmen der Arbeit mit in die Navigation integriert werden soll. Hier ist zu unterscheiden, ob der modifizierte ProfiBot die Routen- und Bahnplanung für das ziellose Umherfahren, für eine

⁷lat. *trajector*: Durchdringer, hier Bahnkurve

Hindernisvermeidung oder für ein Stoppen benötigt. Eine Routenplanung wird beispielsweise dann benötigt, wenn der *ProfiBot* an eine Dockingstation fährt, um die Batterien aufzuladen (s. Kapitel 8.2.3).

Wenn sich der modifizierte *ProfiBot* ziellos in der Umgebung bewegt, ist eine Routenplanung nicht notwendig, die Routenplanung würde die Vorgabe eines Zielorts beinhalten. Eine Bahnplanung beinhaltet das zufällige Generieren von Trajektorien. Das Navigationssystem sollte diese Trajektorien jedoch eine gewisse Zeit, die bedeutend länger als die Zykluszeit⁸ des *IConnect*-Signalgraphen ist, beibehalten. Zielloses Umherfahren geschieht zwar durch die Vorgabe zufälliger Trajektorien, wenn diese aber in kurzen Zeitintervallen neu vorgegeben werden, ist für den *ProfiBot* wegen seines hohen Gewichts kein Vorankommen möglich. Ist die auf den AMR wirkende Beschleunigung infolge einer sich schnell ändernden Trajektorie groß, so muss von den Motoren ein hohes Drehmoment für einen Trajektorienwechsel aufgewendet werden. Das Resultat ist ein hoher Stromverbrauch des Antriebssystems (s. dazu auch Kapitel 5.4.3). Zusätzlich können dynamische Änderungen der Trajektorie das Antriebssystem von *ProfiBot* stark beanspruchen, wodurch die Lebensdauer sinkt.

An Hindernissen kann der modifizierte *ProfiBot*-Roboter einerseits durch Planung einer Ausweichroute vorbeifahren, andererseits durch ein reflexbasiertes Verhalten, das der Routenplanung wegen des geringen Aufwands zur Integration in das Navigationssystem zunächst vorzuziehen ist. Weiterhin sind Methoden zur Bahnplanung im Navigationssystem nicht notwendig. Der *ProfiBot*-Roboter kann einerseits mit einer Trajektorie um ein Hindernis herumfahren; dafür ist es notwendig, die Trajektorie bei einer Kollisionsvermeidung mit einem dynamischen Objekt neu zu berechnen. Der Rechenzyklus hängt hier von der Geschwindigkeit des Hindernisses ab. Bei einem reflexbasierten Verhalten können sich die notwendige Orientierung und die Geschwindigkeit für die Hindernisvermeidung zyklisch aus dem Rechenalgorithmus ergeben. Hierfür sind geeignete Algorithmen zu überprüfen und gegebenenfalls zu bearbeiten. Im Kollisionsfall kann eine Routenplanung eingesetzt werden. Hier wird davon ausgegangen, dass eine lokale Karte im Stillstand eines AMR einer globalen Karte gleichzusetzen ist.

Beim Stoppen wird die Bahnplanung ausschließlich zur Wiederaufnahme des Fahrbetriebs angewendet. Vor dem Stoppen wird nur die Geschwindigkeit des Fahrzeugs verringert, eine Bahnplanung ist hier nicht nötig, weil die Orientierung für ein Bremsmanöver beim *ProfiBot*-Roboter keine Rolle spielt. Andererseits beginnt bei Wiederaufnahme des Fahrbetriebs eine Bahnplanung infolge der Routenplanung. Der *ProfiBot* folgt hier einer Trajektorie, um den Abstand zum Hindernis zu vergrößern, welches das Stoppen ausgelöst hat.

⁸Die Zykluszeit beträgt 20ms, s. Kapitel 3.3.1

4.2.4 Karteninterpretation

Die Möglichkeiten, eine Karte zu interpretieren, sind abhängig vom jeweils erstellten Kartentyp. Auf einer Karte, auf der eine Punktmenge mit Distanzinformationen vorhanden ist, wie auch im Rahmen dieser Arbeit, kann jeder Messpunkt einzeln interpretiert werden. Weiterhin können besondere Merkmale extrahiert werden, die eine Roboternavigation zusätzlich unterstützen. Hierfür werden Algorithmen verwendet, die eine Linienerkennung in einer Umgebungsstruktur ermöglichen [22]. Diese Algorithmen beziehen sich auf eine Punktmenge von Distanzen auf einer lokalen Karte. Ein Algorithmus zur Linienerkennung wird von Surmann und Liang [46] beschrieben. Zunächst erfolgt eine Datenreduktion, bei der alle Punkte P, die einen definierten Mindestabstand d_{step} zueinander unterschreiten, zu einem geometrischem Schwerpunkt⁹ zusammengefasst werden. Daraufhin wird eine Linie erstellt. Wenn die in 4.1 bis 4.3 dargestellten Bedingungen erfüllt sind, wird ein Punkt P_{j+1} einer Linie hinzugefügt, auf der P_i bis P_j Punkte liegen (Quelle: [22]).

$$\frac{||P_i - P_{j+1}||}{\sum_{k=i}^j ||P_{k-1} - P_k||} > 1 - \frac{0,3}{1+1,5\cdot(j+1)}$$
(4.1)

$$\frac{||P_{j-1} - P_{j+1}||}{||P_{j-1} - P_{j}|| + ||P_{j} - P_{j+1}||} > 0,8$$
(4.2)

$$||P_j - P_{j+1}|| < 3 \cdot d_{step} \tag{4.3}$$

Eine Merkmalsextraktion wurde im Rahmen dieser Arbeit nicht benutzt, deswegen wird im Weiteren nicht auf sie eingegangen. Der *ProfiBot*-Roboter interpretiert ausschließlich die Punktmenge.

4.3 Roboterverhaltensweisen

Schwerpunkte in diesem Abschnitt sind Verhaltensweisen eines AMR durch Implementierung von KI^{10} . Im Hinblick auf die KI kann zwischen den Begriffen starke KI und schwache KI unterschieden werden [54]. Das Ziel eines Systems mit einer starken KI ist es, die Intelligenz des Menschen nachzuahmen, z.B. ein Bewusstsein zu erhalten sowie Emotionen zu zeigen und kreativ zu sein. Ein System mit einer schwachen KI ist dagegen anwendungsorientiert. Im Rahmen der schwachen KI gilt es spezielle Probleme zu lösen, die ein gewisses Maß an Intelligenz erfordern. Die Einordnung eines Systems in eine der benannten Definitionen hängt folglich von der Ausprägung der Intelligenz in diesem System ab. Im Rahmen dieser Arbeit gilt es ein spezielles Problem zu lösen, nämlich die Hindernisvermeidung, daher wird im Folgenden auf Verhaltensweisen eingegangen, denen

⁹auch *Centroid* genannt

¹⁰künstliche Intelligenz

eine schwache KI zugrunde liegt.

Verhaltensweisen von AMR können auf dem *Wahrnehmen-Denken-Handeln*-Modell basieren [32] (Abb. 4.6), das ein funktionaler Ansatz für eine *schwache KI* ist. Die Funktionen eines AMR sind aufgeteilt in den Bereich des Wahrnehmens, des Denkens und des Handelns und sie werden anhand einer modularen Struktur realisiert.



Abb. 4.6: Schematische Darstellung des Wahrnehmen-Denken-Handeln-Modells in Anlehnung an Nehmzow [32]

Ziel des *Wahrnehmen-Denken-Handelns* ist es, eine möglichst geringe Diskrepanz zwischen erwünschtem und tatsächlichem Verhalten zu schaffen. Die Sensordaten werden sequenziell von den einzelnen Modulen verarbeitet. Die Steuerbefehle erreichen die Aktoren erst, wenn die Sensordaten alle Module durchlaufen haben.

Die Problematik des Ansatzes ist, dass das Verhaltensmodul des AMR in der Lage sein muss, zu entscheiden, welche Informationen über die Umgebung in das Weltmodell eingehen. Zum Beispiel braucht ein AMR keine Farbinformationen aus seiner Umgebung, wenn er eine Hindernisvermeidung mit Distanzsensoren durchführt. Weil das Navigationsmodul von *ProfiBot*, wie im Abschnitt 4.1 erwähnt, nur auf *lokaler Navigation* basiert, bleibt es von dieser Problematik unberührt, da es kein Weltmodell erstellen kann. Ein weiteres Problem entsteht, wenn die Funktionen des Wahrnehmens, des Denkens und des Handelns zu eng miteinander verknüpft sind. Das Robotersystem kann instabil werden, wenn ein Modul im Betrieb ausfällt, von dem viele andere Module abhängig sind. Die Vernetzung der einzelnen Funktionen sollte deswegen mit einer möglichst geringen Anzahl an Schnittstellen auskommen. Alle aufgeführten Nachteile schließen einen Einsatz eines Verhaltensmoduls, das auf dem Wahrnehmen-Denken-Handeln-Modell aufbaut, im Verhaltensmodul von *ProfiBot* aus.

Es ist möglich ein zuverlässigeres Steuerungssystem für einen AMR zu entwerfen, indem eine engere Verknüpfung zwischen Wahrnehmen und Handeln (Reiz-Reaktions-Schema) erfolgt (s. [32] und [8]). Ein Weltmodell wird hierbei nicht unbedingt benötigt, die einzelnen Funktionen eines AMR müssen sich klar abgrenzen von anderen Funktionen.

• Die Steuerungsarchitektur sollte nicht aus einer Einheit bestehen, die alle Verhaltensweisen beinhaltet und bei der die Sensordaten in jedem Verhaltensmodul verarbeitet werden müssen, bevor sie an das nächste Verhaltensmodul weitergeleitet werden können.

- Die Sensordaten sollten von den einzelnen Verhaltensmodulen parallel ausgewertet werden.
- Die Kommunikation zwischen den einzelnen Verhaltensmodulen sollte auf ein absolutes Minimum beschränkt werden.

Brooks [8] entwarf ein Konzept, das diesen Anforderungen entspricht. Dieses Konzept wird auch Subsumptionsarchitektur genannt und ist für das Verhaltensmodul von *ProfiBot* nutzbar.

4.3.1 Subsumptionsarchitektur

Bei der Subsumptionsarchitektur sind die Verhaltensweisen klar voneinander abgegrenzt und in einzelne Schichten unterteilt. Die Verbindungen der Verhaltensweisen sind darauf beschränkt, Verhaltensweisen mit niedrigeren Prioritäten zu unterdrücken. Zu jeder Zeit ist nur das auf das Ereignis passende Verhalten aktiv. Ein intelligentes Gesamtverhalten ergibt sich aus der geeigneten Zusammenfassung mehrerer Kompetenzen. Die Daten werden trotz der parallelen Steuerungsarchitektur innerhalb der einzelnen Schichten sequenziell verarbeitet. Das kann mit Hilfe von Zustandsautomaten geschehen oder durch Abarbeitung einfacher Sequenzen.



Abb. 4.7: Kommunikation zwischen Verhaltensmodulen in einer Subsumptionsarchitektur [8]

Die Subsumptionsarchitektur ist an einem Beispiel in Abb. 4.7 schematisch dargestellt. Die einfachen und notwendigen Verhaltensweisen befinden sich in den unteren Schichten, die komplexeren Verhaltensweisen in den oberen. Das Verhaltensmodul *Objekte identifizieren* ist das Modul mit der höchsten Priorität und unterdrückt die anderen Module, sobald es aktiv ist. Das Verhaltensmodul *Notstopp* hat die niedrigste Priorität und wird erst aktiv, wenn es durch die anderen Module nicht mehr unterdrückt wird. Die einzelnen Verhaltensweisen in Abbildung 4.7 können aus mehreren Verhaltensweisen bestehen, die ebenfalls als Verhaltensstruktur eine Subsumptionsarchitektur haben. Es gibt bereits eine Vielzahl an AMR, deren Verhaltensweisen mit einer Subsumptionsarchitektur realisiert worden sind. Als Beispiel ist das von Lörken [26] entwickelte Steuerungssystem eines AMR zu erwähnen mit den Roboterverhaltensweisen *Turn* (Drehen), *Brake* (Bremsen) und *Steer* (Lenken) für die Hindernisvermeidung.

Die Subsumptionsarchitektur soll auch im Rahmen dieser Arbeit als Steuerungsarchitektur in der autonomen Roboterplattform verwenden werden, weil sie außer den bereits genannten Vorteilen noch die nachfolgenden besitzt (s. [8]):

- Einfache Erweiterbarkeit
- Ausfallsicherheit und Robustheit
 → maßgeblich f
 ür Anforderungen an die Zuverl
 ässigkeit (s. Kap. 2.5)
- Schnelle Verarbeitungszeiten der Daten
- Gute Übersichtlichkeit

Die Verhaltensweisen "Notstopp" und "Hindernisvermeidung" sind maßgeblich für die Sicherheit eines AMR und müssen in der Steuerung der zu entwickelnden Roboterplattform vorhanden sein. Des Weiteren wird die Verhaltensweise "Ziellos umherfahren" im normalen Betrieb verwendet. Die Verhaltensweisen werden in den nachfolgenden Abschnitten näher erläutert.

4.3.2 Potentialfeldverfahren

Das Verhaltens eines AMR kann mit Hilfe von Potentialfeldern (s. Abb. 4.8) durchgeführt werden. Ein Potentialfeld ist eine spezielle Form eines Vektorfeldes. Mit Potentialfeldern werden physikalische Phänomene beschrieben wie beispielsweise elektrische und magnetische Felder. In der Robotik gehen von Objekten innerhalb des näheren Umfeldes eines AMR Potentialfelder aus.



Abb. 4.8: Potentialfeld nach [24]

Handelt es sich bei den Objekten um Hindernisse, wirken die Feldvektoren abstoßend auf den AMR. Ihre Richtung ist in diesem Fall konträr zum Objekt. Bei Zielen wirken sie anziehend. Das Steuersystem des Roboters erzeugt durch Summation der Potentialfeldvektoren einen Bewegungsvektor \vec{v} , der mit Gleichung 4.4 ausgedrückt wird.

$$\vec{v} = \sum_{i=1}^{n} \vec{v_i} \tag{4.4}$$

Der Bewegungsvektor beschreibt die optimale Bewegungsrichtung, die Geschwindigkeit und die Rotation. Problematisch bei Potentialfeldern sind lokale Minima, die in Abbildung 4.8 rot eingekreist sind. Dabei handelt es sich um Vektoren, deren Beträge gegen null konvergieren. Wenn ein AMR in ein lokales Minimum gerät, fährt er sich fest. Eine genauere Beschreibung ist bei Laue [24] gegeben.

Die Summation von Bewegungsvektoren ist ein geeignetes Verfahren für ein AMR, um ein Verhaltensmodul zu erstellen. Trotzdem gibt es demgegenüber folgende Gründe, weshalb das Verfahren nicht im Rahmen dieser Arbeit angewendet werden kann.

Das Potentialfeldverfahren wird im Zusammenhang mit globalen Karten benutzt. Auf lokalen Karten sind Transformationen der Feldvektoren notwendig, das kann bei einer Navigation mit einer lokalen Karte, wie sie bei dem modifizierten *ProfiBot* angewendet wird, zu erheblichem Rechenaufwand führen. *IConnect* ist im Zusammenhang mit dem verfügbaren Notebook dafür nicht geeignet. Die Potentialfeldmethode wird daher im Rahmen dieser Arbeit nicht weiter betrachtet. Im Übrigen kann das Navigationssystem von *ProfiBot* Hindernisse nicht als Objekte klassifizieren, da es ausschließlich Punktwolken verarbeiten soll (s. Kapitel 4.2.1).

4.3.3 Zielloses Umherfahren

Das ziellose Umherfahren wird im störungsfreien Fahrbetrieb von ProfiBot angewendet und ergibt sich aus Trajektorien, bei denen die Geschwindigkeit und die Orientierung des modifizierten ProfiBots belanglos ist. Außerdem findet mit ziellosem Umherfahren der Wechsel von zufälligen Steuerparametern innerhalb einer Mindestzeit statt. Das Verhaltensmodul sollte im Betrieb des modifizierten ProfiBots wegen der Zuverlässigkeit und der Sicherheit des Roboters aktiv sein, wenn sich kein Hindernis innerhalb eines ausreichend großen Sicherheitsradius befindet. Die Sicherheitsanforderungen sind im Ablauf der Verhaltensweise nicht zu beachten, weil es ausreicht diese über die untergeordneten Verhaltensweisen zu berücksichtigen. Nichtsdestotrotz ist eine Schnittstelle unverzichtbar, um das Verhaltenmodul Hindernisvermeidung zu aktivieren. Es ist in diesem Verhaltensmodul nicht nötig, weitere Funktionen zu integrieren, die das Erkunden der Umgebung ermöglichen oder das Erstellen einer Karte.

Auch wenn das Verhaltensmodul dem Navigationssystem keine Zielposition vorgibt, ist es dennoch erforderlich, eine Orientierung und Richtung vorzugeben, um die Steuersignale für die Aktoren zu generieren. Wie bereits in Kapitel 4.2.3 erwähnt ist es wichtig, die Trajektorien über einen längeren Zeitraum (z.B. 5s...10s) konstant zu belassen. Hierfür kann das Steuerprogramm innerhalb eines bestimmten Zeitraums konstante Steuerparameter bereitstellen.

Unter der Betrachtung der genannten Aspekte sind für die Beschaffenheit der Steuerparameter nachfolgende Rahmenbedingungen getroffen worden:

- Der Lenkwinkel sollte $\pm \frac{\pi}{2}$ nicht übersteigen.
- Die Trajektorien sollten mindestens 5s lang eine konstante Orientierung besitzen.
- Die Beschleunigung des *ProfiBot*-Roboters durch hintereinanderfolgende Trajektorien darf nicht zu hoch sein.

4.3.4 Hindernisvermeidung

Bei der Hindernisvermeidung wird, wie in Abschnitt 4.1 dargelegt, zwischen *lokaler* und *globaler Navigation* unterschieden. Hindernisse werden mit einer Distanzsensorik erkannt, und auf der Ebene der *lokalen Navigation* wird eine lokale Karte der Umgebung erstellt. Im Rahmen der Hindernisvermeidung können

1. eine lokale Karte erstellt werden, in der die Hindernisse verzeichnet und evtl. gespeichert sind, und 2. die Sensordaten nach entsprechender Verarbeitung und Aufbereitung direkt als Steuerbefehle an die Aktoren weitergeleitet werden.

Nach den Methoden wird die Umgebung (s. [39]) klassifiziert in:

- Befahrbar \rightarrow Freiraum
- Nicht befahrbar oder schlecht befahrbar \rightarrow Hindernis
- Unbekanntes Gelände (nur bei erster Methode)

Für die Hindernisvermeidung gibt es eine Vielzahl an Verfahren, die z.B. auf dem Potentialfeldverfahren oder auf reflexbasiertem Verhalten aufbauen. Das Potentialfeldverfahren wird nicht weiter betrachtet, da das Verfahren in dieser Arbeit nicht angewendet wird. Im folgenden werden zwei mögliche Verfahren zu reflexbasiertem Verhalten beschrieben.

Mit dem Verfahren nach *Braitenberg* [5] (s. Abschnitt 2.4) ist von Vorteil, dass nur zwei relativ einfache Distanzsensoren für die Hindernisvermeidung benötigt werden (z.B. Ultraschallsensoren oder Triangulationssensoren)). Somit ist auch die Auswertung simpel. Nachteilig ist die geringe Zuverlässigkeit, mit der schmale Hindernisse erkannt werden, bei der Auswertung von zwei Distanzsensoren.

Das zweite Verfahren, das auf reflexbasiertem Verhalten beruht, berücksichtigt diese Mankos und wird auch in dem AMR *KURT3D* angewendet. Dieses Verfahren ist in dem Artikel von Surmann et al. [25] (s. Kapitel 2.6.3) beschrieben. Basierend auf Fuzzy-Regeln gewichtet ein AMR die Informationen über Orientierung φ_i und Distanz r_i zu Hindernissen nach den Funktionen f_1 und f_2 . Aus den Informationen wird ein Lenkwinkel α_l für den Roboter generiert, der sich aus Gleichung 4.5 ergibt, in der *n* die Anzahl der erfassten Punkte pro Scan ist.

$$\alpha_{l} = atan2^{11} \left(\frac{\sum_{i=1}^{n} \cos\left(\varphi_{i}\right) \cdot f_{1}\left(\varphi_{i}\right) \cdot f_{2}\left(r_{i}\right)}{\sum_{i=1}^{n} \sin\left(\varphi_{i}\right) \cdot f_{1}\left(\varphi_{i}\right) \cdot f_{2}\left(r_{i}\right)} \right)$$
(4.5)

Die Gewichtsfunktionen f_1 und f_2 werden durch die Gleichungen 4.6 und 4.7 beschrieben:

$$f_1(\varphi_i) = \cos\left(\varphi_i\right) = [0...1] \tag{4.6}$$

$$f_2(r_i) = \frac{1}{2} \cdot tanh(m \cdot r_i - d_T) + 1 = [0...1]$$
(4.7)

In 4.6 wird eine Kosinusfunktion zur Gewichtung der Orientierung verwendet, d.h. je seitlicher die Hindernisse im Sichtfeld des Laserscanners liegen, desto weniger berücksichtig der AMR diese für die Berechnung des Lenkwinkels. In der Gewichtsfunktion 4.7 wird

¹¹Die atan2-Funktion ist eine arctan-Funktion bei der zusätzlich der Quadrant des erzeugten Lenkwinkels berücksichtigt wird

die Distanz zu den Hindernissen gewichtet. Der Übergangswertebereich u um einen Distanzschwellwert d_T , der in der Gewichtsfunktion 4.8 beschrieben wird, lässt sich mit der Konstanten m festlegen.

$$u = f_2 (\approx 0) < f_2 (r_i) < f_2 (\approx 1)$$
(4.8)

Unterschreiten Distanzen den Sollwert d_T , werden sie stärker gewichtet. Der Distanzschwellwert d_T hat die Gewichtung $f(r_i) = 0.5$. Die Funktionsgraphen der Gewichtungsfunktionen sind in Abbildung 4.9 dargestellt.

In jedem Zyklus des Steuerprogramms wird mit Hilfe der Gewichtungsfunktionen ein Lenkwinkel berechnet, der ohne Umrechnung direkt als rotatorische Geschwindigkeit ω_l verwendet wird. Durch ω_l wird die Orientierung des AMRs in eine Richtung geregelt, in welcher am wahrscheinlichten ein objektfreier Raum erwartet wird. Das Verhalten ist



Abb. 4.9: Die Gewichtungsfunktionen f_2 (l) und f_1 (r) (Quelle: [22])

für den modifizierten *ProfiBot* verwendbar. Die Parameter m und d_T müssen experimentell ermittelt werden. Weiterhin ist zu überprüfen, ob eine zusätzliche Verhaltensweise *Drehen* eingesetzt werden muss, damit der *ProfiBot*-Roboter auch aus Sackgassen herausfahren kann. Mit der Verhaltensweise *Drehen* sollte auch das Passieren von *gefährlichen Freiräumen*, z.B. Gängen, in denen sich der ProfiBot-Roboter nicht drehen kann, während der Navigation umgangen werden.

4.3.5 Notstopp

Die Verhaltensweisen in diesem Abschnitt beziehen sich primär auf einen AMR, der Sensoren zur Kollisionserkennung besitzt, die in diesem Zusammenhang in dem modifizierten *ProfiBot* eingesetzt werden. Die Verhaltensweise *Notstopp* verursacht ein sofortiges Stoppen der Motoren, das in Situationen geschieht, in der keine Hindernisvermeidung mehr möglich ist. Es liegen keine Quellen zu den Annahmen vor, aufgrund welcher Ereignisse ein Notstopp erfolgen sollte, daher müssen diese umso sorgfältiger zur Gewährleistung der Sicherheit in Fahrversuchen eruiert werden. Das Stoppen kann beispielsweise durch eine Kollision geschehen oder als präventive Maßnahme, wenn ein AMR nicht mehr rechtzeitig einem Hindernis ausweichen kann. Eine Kollision ist grundsätzlich zu vermeiden. Des Weiteren ist der Notstopp die sicherheitsrelevanteste Verhaltensweise in der Subsumptionsarchitektur, bei der eine hohe Ausfallsicherheit erforderlich ist.

Das Stoppen ist wichtig, da nur so gewährleistet ist, dass durch den AMR selbst kein Schaden entstehen kann. Es ist eine durch Software realisierte Unterbrechung im Fahrbetrieb des AMR möglich, die einerseits ausschließlich durch den Bediener oder die Bedienerin wieder aufgehoben werden kann. Andererseits kann nach Ablauf einer bestimmten Wartezeit der Fahrbetrieb fortgesetzt werden, in der Annahme, dass dies ungehindert geschieht. Beim Stoppen ohne Wiederaufnahme des Fahrbetriebs sollte das komplette Steuerprogramm unterbrochen werden. Das hat allerdings den Nachteil, dass sich jederzeit ein Bediener oder eine Bedienerin, der/die für die Wiederaufnahme des Fahrbetriebs zuständig ist, in der Nähe des Roboters befinden muss bzw. durch ein Funksignal über den Roboterstatus, wie bei den Robotern an dem Berliner Museum für Kommunikation, informiert wird (s. Kapitel 2.6.2). Durch Betriebsstopps sinkt auch die stationäre Verfügbarkeit (s. Kapitel 2.5). Hingegen sollte der AMR immer die passende Verhaltensweise zeigen, wenn er nach jedem Stoppvorgang den Fahrbetrieb wieder aufnimmt. Bei einer falschen Verhaltensweise können sicherheitsrelevante Mängel entstehen.

Eine optimale Lösung ist das automatische Bereitstellen der geeigneten Verhaltensweise nach dem Ereignis, welches das Stoppen verursacht hat. Hierfür sollten im Hinblick auf eine ideale Erfüllung der Sicherheitskriterien wie Verlässlichkeit, Zuverlässigkeit und stationäre Verfügbarkeit, alle Szenarien berücksichtigt werden, die einen Notstopp verursachen können. Es würde den Rahmen dieser Arbeit sprengen, wenn viele Szenarien berücksichtigt werden sollen, zumal die meisten dieser Szenarien nur durch nachhaltige Fahrversuche unter realen Einsatzbedingungen herausgefunden werden können. Es ist möglich, einige im Vorfeld bekannte Szenarien zu analysieren und die passende Verhaltensweise zu erstellen. Es wird dabei unterschieden, ob eine Kollision stattgefunden hat, oder ob der AMR vor dem Hindernis zum Stillstand gekommen ist.

Eine Auswahl an Szenarien (in Abbildungen 4.10 bis 4.13 dargestellt) zeigt, bei welcher Art von Kollisionen die jeweilige Verhaltensweise angemessen ist. Die roten Linien in den Abbildungen 4.10 bis 4.13 stellen den Fahrweg dar, bevor die Kollisionen stattgefunden haben. Die schwarzen Linien stellen die Fahrwege im Zusammenhang mit einer möglichen Verhaltensweise nach einer Kollision dar. Zum Zweck der Anschaulichkeit haben die Hindernisse in den Abbildungen einfache geometrische Formen. In diesem Zusammenhang wurde bedacht, dass bei statistisch sehr häufig auftretenden Szenarien möglichst kein fortdauerndes Stoppen aufteten sollte, um einen kontinuierlichen Betrieb des AMR zu gewährleisten. Wird diese Maßnahme auch auf das Verhalten von *ProfiBot* angewendet, erhöht dies die Zuverlässigkeit und die stationäre Verfügbarkeit im Rahmen der Sicherheitsanforderungen.



Abb. 4.10: Frontalzusammenstoß mit Hindernis 2

Abb. 4.11: Seitlicher Zusammenstoß (1)

In Abbildung 4.10 ist ein Frontalzusammenstoß mit einem statischen Hindernis (Hindernis 2) dargestellt. Das Hindernis könnte z.B. eine Wand sein oder ein Gegenstand. Nach der Kollision muss der AMR stoppen, um (weitere) Schäden zu vermeiden. Nach einer bestimmten Wartezeit sind mindestens zwei geeignete Trajektorien denkbar, die ein Ausweichen und eine Weiterfahrt erlauben, ohne erneut mit dem gleichen Hindernis zusammenzustoßen. In Bezug auf Trajektorie a fährt der AMR eine gerade Strecke zurück, die eine festgelegte Mindestlänge haben muss, damit ein anschließendes Drehen um die eigene Achse erfolgen kann. Der Drehwinkel hierfür sollte mindestens 90° betragen, damit die Plattform nicht ein weiteres Mal mit dem Hindernis kollidiert. Der AMR kann jedoch bei einer Rückwärtsfahrt gegen andere Hindernisse stoßen (s. in Abb. 4.10: Hindernis 1), wenn er keine zusätzlichen Methoden besitzt, bei denen Hindernisse in einer Rückwärtsfahrt erkannt und vermieden werden. Eine bessere Lösung ist Trajektorie b, bei der die Roboterplattform annähernd auf der gleichen Bahnkurve zurückfährt, die sie vor der Kollision abgefahren hat. Von Vorteil ist, dass sich auf der Bahnkurve b keine statischen, sondern allenfalls mobile, Objekte befinden können.

Im Szenario in Abbildung 4.11 ist ein seitlicher Zusammenstoß mit einem statischen Hindernis 2 dargestellt, der sich durch Kurvenfahrt oder Drehen des AMR ereignen kann. Auch hier kann der AMR wie im Szenario in Abb. 4.10 nach der Kollision auf einer linearen Bahn zurückfahren (Trajektorie a) oder auf der vorherigen Bahnkurve (Trajektorie b). Trajektorie b ist wegen benannter Gründe auch in diesem Szenario der bessere Fahrweg. Entsteht ein seitlicher Zusammenstoß durch ein sich auf den AMR hinzubewegendes Hindernis 2 (Personen oder Fahrzeuge), wie in Abbildung 4.12 illustriert, ist ein Stoppen notwendig, solang sich das Hindernis weiterbewegt. In einem solchen Szenario ist es sinnvoll, die Wirkungszeit der Kollisionskraft auszuwerten, um so offen zu legen, ob es sich um ein dynamisches oder ein statisches Hindernis handelt. Die Trajektorien a und b (vgl. Abb. 4.10 und 4.11) sind zwar denkbar, es ist aber nicht absehbar, ob das Hindernis 2 nach einer Kollision ebenfalls stoppt. Wenn Hindernis 2 seine Bewegung fortsetzt, so können bei einem Fahrbetrieb des AMR Schäden durch Stöße und Reibkräfte entstehen.



Abb. 4.12: Seitlicher Zusammenstoß (2)

Abb. 4.13: Worst Case

Auf die Motoren von *ProfiBot* wirkt im Stillstand bei aktivem Steuerprogramm ein Bremsmoment. Der modifizierte *ProfiBot* beharrt bei einer Kollision auf seiner Stillstandsposition, der größte Teil der kinetischen Energie beim Aufprall wird in Verformungsenergie umgewandelt. Weil bei einem Frontalzusammenstoß mit einem dynamischen Hindernis die Kollisionskräfte in Fahrtrichtung auf den *ProfiBot*-Roboter wirken, könnte eine Unterbrechung der Steuerbefehle erfolgen, die durch das Stoppen der *IConnect*-Signalgraphen erfolgt. Der *ProfiBot*-Roboter kann dem Aufprall nachgeben, indem er zurückrollt, und somit den Schaden begrenzen. Es ist in Bezug auf den *ProfiBot*-Roboter zu untersuchen, ob die Reaktionszeit des gesamten Steuersystems ausreicht, die Unterbrechung zweckgemäß auszuführen. In dem Szenario in Abb. 4.12 wirken die Kollisionskräfte quer zur Fahrtrichtung, ein Zurückrollen ist nicht möglich. Somit ist es nicht sinnvoll die Steuerbefehle zu unterbrechen. Wenn im Rahmen der Sensordatenauswertung erkannt wird, dass keine Kollisionskräfte mehr auf *ProfiBot* wirken, kann der Fahrbetrieb fortgesetzt werden.

Im Szenario in Abbildung 4.13 ist es trivial, dass der Roboterbetrieb nicht wieder aufzunehmen ist. Der Fall wäre in einem Raum denkbar, in dem sich eine Vielzahl von Personen oder anderen dynamischen Hindernissen befindet.

Kommt ein AMR vor einem Hindernis zum Stillstand, so kann analog zu Trajektorie b in den Abbildungen 4.10 bis 4.12 die vorherige Bahnkurve um eine bestimmte Länge zurückgefahren werden, um die Fahrt anschliessend fortzusetzen.

In einigen Szenarien wird eine Rückwärtsfahrt des *ProfiBot*-Roboters durchgeführt. Bei einer Rückwärtsfahrt muss ebenfalls gewährleistet werden, dass der *ProfiBot*-Roboter bei einer Kollision stoppt. Das sollte durch die Montage und Auswertung der gleichen Sensorik für die Kollisionserkennung realisiert werden, die auch an der Frontseite vorhanden ist. Weil diese Konstruktion zunächst nicht vorgesehen ist, sollte der ProfiBot-Roboter nicht rückwärts fahren. Für die Trajektorien b in den Szenarien ist ein Ringspeicher notwendig, in welchem die Odometriedaten des *ProfiBot*-Roboters gespeichert werden. Die einfachste und sicherste Lösung, die auch in dieser Arbeit realisiert wurde, ist ein Stoppen bei Kollisionen.

4.4 Entscheidungsfindung

Für die sichere Navigation sind die beschriebenen Verhaltensweisen in einer Subsumptionsarchitektur ausgewählt worden. Diese Verhaltensweisen, die jeder AMR besitzen muss, sind grundlegend. Sie sollten immer die Basis der gesamten Verhaltensarchitektur eines AMR bilden.

Navigationsverfahren für das ziellose Umherfahren:

Das ziellose Umherfahren ist ein Verhalten, für das weder eine Karte, noch eine Sensorik erforderlich ist. Für die lokale Navigation werden zufällige Steuerparameter erzeugt, die den in Kapitel 4.3.3 festgelegten Rahmenbedingungen unterliegen. Aus diesem Grund ist hierbei eine Navigation mit *Lagepeilung* oder *Koppelnavigation* nicht notwendig.

Navigationsverfahren für die Hindernisvermeidung:

In Kapitel 4.2 ist bereits definiert worden, dass die Hindernisvermeidung ausschließlich auf der Basis der *lokalen Navigation* durchgeführt wird. Nachfolgend werden nur die Verfahren berücksichtigt, die im Zusammenhang mit einer *lokalen Navigation* stehen. Es ist weiterhin vorgegeben, dass eine lokale Karte für die Hindernisvermeidung eingesetzt wird.

Um die Navigationsverfahren genauer definieren zu können, muss zunächst unterschieden werden, ob als Navigationsverfahren hier eine *Koppelnavigation* oder eine *Lagepeilung* eingesetzt werden soll. Mit der Koppelsensorik des ProfiBot-Roboters (Encoder) können statische Hindernisse nicht direkt erfasst werden. Es ist nur möglich die Positionen von statischen Hindernissen indirekt aus einer globalen Karte zu entnehmen, wenn die Positionen im Vorfeld bekannt sind. Dynamische Hindernisse, die ihre Pose verändern, können weder direkt noch indirekt erfasst werden. Folglich wurde die Koppelnavigation für die Hindernisvermeidung nicht in Betracht gezogen.

Die eingesetzten Distanzsensoren ermöglichen eine direkte Erfassung statischer und dynamischer Hindernisse mittels *Lagepeilung*. Durch den 2D-Laserscanner und die analogen Triangulationssensoren ist der modifizierte *ProfiBot*-Roboter befähigt, viele angepeilte Messpunkte innerhalb eines Rechenzyklus zu erfassen und eine entsprechende Pose innerhalb der Hindernisvermeidung anzunehmen. Für diese Pose sollte der *ProfiBot*-Roboter eine Orientierung und eine Geschwindigkeit haben, die nicht zu einer Kollision führen. Dabei muss die Pose präventiv berechnet werden. Durch den Einsatz der *Lagepeilung* und der direkten Erfassung von Objekten innerhalb weniger Rechenzyklen ist dies möglich. Hierfür stellen alle Hindernisse *passive Landmarken* dar. Planende Verhaltensmodule müssen nicht eingesetzt werden, wie in Kapitel 4.2.3 beschrieben.

Navigationsverfahren für den Notstopp:

Für den Notstopp wird im Rahmen dieser Arbeit keine Karte benötigt, weil der Roboter in diesem Verhalten seine Pose nicht ändert. Die Informationen aus den Karten, die den Notstopp veranlassen, werden von dem Verhalten *Hindernisvermeidung* interpretiert. Eine in Kapitel 4.2.3 beschriebene Routen- und Bahnplanung, die auf den Notstopp folgt, wird aus Sicherheitsgründen nicht realisiert, weil der ProfiBot-Roboter während der Ausführungszeit nicht auf Ereignisse reagiert.

Kapitel 5

Auswahl der Sensorik

Dieses Kapitel behandelt die Auswahl der Sensorik, die im *ProfiBot*-Roboter für eine Erkennung von Hindernissen sowie für eine Detektierung einer Kollision eingesetzt wurde. Die Sensorik muss hinsichtlich ihres Einsatzes bestimmte Anforderungen erfüllen, die zunächst benannt werden. Anschliessend wird analysiert, ob im Hinblick auf eine höhere Sicherheit und Zuverlässigkeit zusätzliche Redundanz der Messwertaufnahme durch eine Sensorfusion ermöglicht werden kann. Weiterhin werden verschiedene Sensortypen im Hinblick auf die ensprechenden Anforderungen geprüft. Es wird untersucht, welche dieser Sensortypen beim *ProfiBot*-Roboter eingesetzt werden können. Die Sensorentypen werden im Anschluss bezüglich ihrer Eigenschaften tabellarisch miteinander verglichen. Abschließend werden die Entscheidungsfindung, die ausgewählten Sensoren und das Gesamtsystem erläutert. Unter diesem Aspekt Sensoren verschiedener Hersteller miteinander verglichen.

5.1 Anforderungen an die Sensorik

Die Sensorik, die im modifizierten *ProfiBot*-Roboter eingesetzt werden soll, muss Hindernisse und Kollisionen erkennen können. Nachfolgend werden diese beiden Aufgaben und die Anforderungen an die entsprechende Sensorik separat betrachtet.

Um Hindernisse berührungslos und zuverlässig erfassen zu können, sollte die im *Profi-Bot*-Roboter einzusetzende Sensorik in der Lage sein, die Hindernisse zu orten. Das Navigationssystem kann ausschließlich die Orientierung, Richtung und Entfernung vom AMR zum Hindernis ermitteln. Diese Informationen sind für die Erzeugung der Steuerparameter wichtig, um den Hindernissen auszuweichen. Um Objekte grundsätzlich zu erfassen, können geometrische Messgrößen (Distanz und Winkel) mit einer Distanzsensorik für eine Auswertung herangezogen werden.

Durch die Messung von Temperaturunterschieden beispielsweise mit Wärmebildkame-

ras oder Infrarotsensoren oder durch die Messung von Helligkeit und Farbtönen, z.B. mit CCD¹-Kameras oder Phototransistoren, lassen sich Objekte nicht direkt lokalisieren, weil diese Sensoren keine Informationen über die Distanz zu einem Objekt liefern. Aufgrund dieses Nachteils werden im Folgenden lediglich Sensoren betrachtet, mit denen eine direkte Distanz- und Winkelmessung zu Objekten möglich ist. Der Vollständigkeit halber wird aber darauf hingewiesen, dass mit der Erfassung der Temperatur von Objekten eine Unterscheidung zwischen Personen und anderen Gegenständen direkt möglich ist. Im Rahmen des modifizierten *ProfiBot*-Roboters ist jedoch eine Klassifizierung nicht notwendig, weil alle Objekte potentielle Hindernisse sind.

Die Distanzsensorik sollte weiterhin mindestens eine zweidimensionale Punktmenge² in der euklidischen Ebene so aufnehmen können, dass alle Hindernisse auf ihr verzeichnet und die Distanzen zu ihnen bekannt sind. Je mehr Punkte erfasst werden, desto größer ist die geometrische Auflösung der registrierten Umgebung, und somit ist es möglich, mehrere Hindernisse bei gleichbleibender Pose des *ProfiBot*-Roboters zu erkennen. Einige AMR, wie beispielsweise *KURT3D*, sind sogar in der Lage, eine dreidimensionale Punktwolke aufzunehmen, die ausführliche Erläuterung des Funktionsprinzips ist in Abschnitt 2.6.3 zu finden. Mit eindimensional messenden Distanzsensoren, wie z.B. Trinagulationssensoren und Ultraschallsensoren, ist für die Aufnahme einer mehrdimensionalen Punktwolke eine bestimmte Anzahl dieser Sensoren notwendig. Eine Möglichkeit der Sensorpositionierung in einem AMR ist in Abb. 5.1 dargestellt.



Abb. 5.1: Anordung von eindimensional messenden Sensoren in einem AMR

Die beschriebenen Muss- und Wunschkriteríen für die Auswahl der Distanzsensorik sind in Tabelle 5.1 zusammengefasst.

Die Distanzauflösung der Sensorik spielt eine untergeordnete Rolle. Diesbezüglich wird ein Mindestabstand zu einem Hindernis, den der *ProfiBot*-Roboter bei einem Ausweichmanöver haben muss, addiert.

Eine Kollision kann nur berührungsbehaftet erfasst werden. Unter anderem können folgende physikalische Messgrößen bei einer Kollision zur Weiterverarbeitung herangezogen werden.

¹charge-coupled devices

²Mengen von kartesischen Koordinaten

Musskriterien	Bemerkung
Hohe Reichweite	mind. $2m$
Einfache Datenerfassung/Auswertung	
Sender und Empfänger in räumlicher Nähe	
Schneller Messzyklus	Echtzeitfähig
2D oder 3D Umgebungserfassung	
Geringer Energieverbrauch	lange Betriebsdauer
Hohe Zuverlässigkeit	
Geringe Abmessungen	Einfache Montage
WUNSCHKRITERIEN	Bemerkung
hohe Robustheit	Ggf. durch Schutzvorrichtung
Preisgünstig	Wirtschaftlichkeit
Niedrige Störempfindlichkeit	

Tabelle 5.1: Anforderungen an die Distanzsensorik

- Kraft (z.B. DMS³, induktive Kraftaufnehmer, etc.)
- Druck (z.B. DMS, oder piezoelektrische Drucksensoren, etc.)
- Beschleunigung (z.B. Beschleunigungssensoren, etc.)

Die aufgeführten Messgrößen für eine Kollisionserkennung werden bei einem AMR im Allgemeinen durch Sensoren in Schaltleisten oder Kontaktpuffer erfasst. Die Sensoren müssen für den Einsatz in dem *ProfiBot*-Roboter die in 5.2 benannten Kriterien erfüllen.

MUSSKRITERIEN	Bemerkung
Hohe Zuverlässigkeit	
Preisgünstig	
Einfache Messwerterfassung	Schaltsignal
Einfache Montage	

Tabelle 5.2: Anforderungen an Sensorik zur Kollisionserkennung

Mit der ausgewählten Sensorik gilt es, den *ProfiBot*-Roboter in seinem Fahrverhalten abzusichern. Hinsichtlich dieser Anforderungen werden einige Typen von Distanzsensoren im Folgenden untersucht. Das Funktionsprinzip sowie die Vor- und Nachteile dieses Sensortyps und die Eignung für den Einsatz der *ProfiBot*-Plattform wird in den folgenden Abschnitten beschrieben.

 $^{^{3}{\}rm Dehnungsmess streifen}$

5.2 Redundanz durch Sensorfusion

Die Sensorfusion ist das Kombinieren und Zusammenfügen verschiedener Sensorsignale [59]. Damit ist das Erfassen von Messwerten der gleichen physikalischen Messgröße gemeint, das durch unterschiedliche Sensoren geschieht. Durch die Sensorfusion kann die Genauigkeit der Messdaten gesteigert werden. Des Weiteren können Messfehler kompensiert werden. Die Beobachtung und Schätzung von Systemzuständen fällt ebenso unter den Begriff der Sensorfusion. Die Sensorfusion liefert einen höheren Informationsgehalt als die einzelnen Sensoren. Sogenannte Fusionsfilter können Fehler, die in den Messungen entstehen, erkennen und korrigieren. Sie werden beispielsweise eingesetzt, um den nächsten Systemzustand zu schätzen und den darauf folgenden Messfehler zu korrigieren.

Der *ProfiBot*-Roboter sollte, um Ausfällen in der Sensorik vorzubeugen, mindestens zwei unterschiedliche Distanzsensoren besitzen, damit eine höhere Sicherheit und Zuverlässigkeit bezüglich der Messwertaufnahme gegeben ist. Weiterhin sollte er Verfahren besitzen, mit denen die Prioritäten der einzelnen Sensoren bezüglich ihrer Messdatenverwertung festgelegt werden können.

5.3 Sensoren für die Hinderniserkennung

Im folgenden Abschnitt werden verschiedene Distanzsensoren beschrieben, die für einen Einsatz in einem *ProfiBot*-Roboter denkbar wären. Es wurde untersucht, ob die Sensoren den gestellten Anforderungen genügen. Distanzsensoren gehören zu der Klasse der Sensoren, die Objekte aus der Entfernung erfassen. Durch das berührungsfreie Arbeitsprinzip werden Distanzsensoren in ihrer Funktion und Genauigkeit nicht durch Abnutzung beeinflusst. Folgende Sensortypen wurden untersucht: Ultraschallsensoren, Triangulationssensoren, kapazitive und induktive Sensoren, Laserscanner und 3D-Kameras.

5.3.1 Ultraschallsensoren

Ultraschallsensoren sind eindimensional messende Sensoren und werden beispielsweise für Geschwindigkeitsmessungen, Füllstandsmessungen und Messungen der Dichte [49] eingesetzt. Ein weiteres Aufgabengebiet ist die Abstandsmessung und das Erfassen von Objekten und Objektsituationen. Ultraschallsensoren finden auch in vielen AMR Verwendung, wie z.B. im Serviceroboter *Scitos C5* (s. Kapitel 2.6.1), der ähnlich wie der modifizierte *ProfiBot*-Roboter eine Hindernisvermeidung durchführen kann. Aus diesem Grund wurden Ultraschallsensoren auch für den Einsatz beim *ProfiBot*-Roboter in Erwägung gezogen.

Ultraschallsensoren sind aktive Sensoren. Sie senden Ultraschallwellen aus und empfangen diese wieder. Die Abstandsmessung ist eine häufige Anwendung von Ultraschallsensoren aufgrund der einfachen Bestimmbarkeit der Laufzeit eines von einem Objekt reflektierten Ultraschallimpulses. Vom Ultraschallwandler werden zu vorgegebenen Zeiten periodische Bursts⁴ ausgesendet, die nach einer Laufzeit den Empfangswandler erreichen und in ein elektrisches Empfangssignal umgewandelt werden. Dieses Prinzip wird auch *Impuls-Echo-Prinzip* genannt. Viele Ultraschallsensoren haben als Sender und Empfänger ein und denselben Ultraschallwandler [49]. Weil die Schwingungsamplitude am Sender des Ultraschallempfängers erst abklingen muss, bevor Empfangssignale verarbeitet werden können, gibt es bei dieser Art von Sensoren immer einen gewissen Mindestabstand, der eingehalten werden muss, damit der Sensor eine Messung erfassen kann. Der Bereich zwischen Sensor und Mindestabstand wird auch Totbereich genannt.

Der Bereich, in dem Objekte durch Ultraschallsensoren registriert werden können, wird auch Schallkeule genannt und ist eine sensorspezifische Eigenschaft. Bei einer Reihe von Ultraschallsensoren ist es möglich, die Schallkeule zu variieren. Vergrößert man die Schallkeule, können auch ferne Objekte erfasst werden. Proportional zur Vergrößerung der Schallkeule vergrößert sich allerdings auch der Totbereich eines Ultraschallsensors. Ein Beispiel für eine typische Schallkeule ist in Abbildung 5.2 dargestellt. Es handelt sich hier um die Schallkeule des Ultraschallsensors UB6000-30M [35] von Pepperl+Fuchs. Der ensprechende Sensor ist in Abbildung 5.3 dargestellt.



Abb. 5.2: Schallkeule *UB6000-30M* (Quelle: [35])



Abb. 5.3: UB6000-30M (Quelle: [35])

Der Erfassungsbereich eines Ultraschallsensors hängt von der Intensität der Schallwellen ab. Durch die Schalldivergenzdämpfung⁵, verringert sich die Intensität der Schallwelle mit zunehmender Enfernung vom Schallsender. Mit der Abnahme der Intensität eines empfangenen Signals wird die Reichweite einer Messung begrenzt. Dennoch ist die Reichweite gegenüber anderen Distanzsensoren im Allgemeinen relativ hoch (bis zu 12m je nach Sensorausführung). Bei einer Intensität, die unter der dreifachen Maximalamplitude von Störgeräuschen liegt, ist keine genaue Messwertaufnahme mehr möglich. Eine nähere Be-

⁴engl. *burst*: Stoß, hier Wellenpaket

⁵die entfernungsabhängige Aufweitung der Schallwelle

schreibung des Funktionsprinzips ist bei Tränkler und Obermeier [49] zu finden.

Die Ausgangssignale von Ultraschallsensoren können je nach Ausführung des Sensors unterschiedlich sein. Einerseits kann mit Hilfe von Schaltausgängen ein binäres Signal ausgegeben werden, bei dem sich der Ausgangspegel ab einem variablen Distanzschwellwert ändert. Für die genaue Distanzbestimmung eines Objekts ist diese Ausführung nicht geeignet und wird hier nicht weiter erläutert. Andererseits kann eine analoge Spannung ausgegeben werden, die zur Distanz proportional ist; diese Sensoren eignen sich gut für eine Distanzbestimmung.

Eine genaue Lokalisation von Hindernissen ist jedoch mit dem Impuls-Echo-Verfahren nicht möglich, wenn sich mehrere Hindernisse in der Schallkeule eines Ultraschallsensors befinden. Für eine Lokalisation unterschiedlicher Hindernisse ist ein frequenzmoduliertes Ultraschallsignal notwendig, das im Anschluss nur mit einer FFT⁶-Analyse ausgewertet werden kann, eine genaue Beschreibung der Frequenzmodulation ist in [2] zu finden. Der hierzu benötigte Rechenaufwand ist enorm, so dass Informationen nicht schnell genug weiterverarbeitet werden können, was sich negativ auf die Echtzeitanforderungen auswirkt.

Im Bezug auf den Einsatz in einem AMR haben Ultraschallsensoren die in Tabelle 5.3 dargestellten Vor- und Nachteile [49].

VORTEILE	NACHTEILE
Reproduzierbare Ergebnisse	Eindimensionale Abstandsmessung
Variabler Messbereich	Totbereich
Robustheit, Langzeitstabilität	Hohe Anschaffungskosten
Einfache Auswertung bei Distanzmessung	Hörbare Betriebsgeräusche
Hohe Reichweite	

Tabelle 5.3: Vor-/Nachteile von Ultraschallsensoren

Nachteilig ist weiterhin, dass die Intensität der reflektierten Schallwelle abhängig ist von der Oberflächenbeschaffenheit des Reflektors.

Die gezeigten Vor- und Nachteile von Ultraschallsensoren lassen einen Einsatz für die Erfassung einer Punktwolke beim *ProfiBot*-Roboter nicht zu. Die Hauptgründe dafür sind das schlechte Preis-/Leistungsverhältnis bei der Applikation von mehreren Sensoren und die Abhängigkeit der Intensität von der Oberfächenbeschaffenheit. Letzteres wirkt sich z.B. negativ auf die Erkennung von Kleidung aus und demzufolge auch auf die Erkennung von Personen. Des Weiteren ist das Betriebsgeräusch eines AMR beim Einsatz mehrerer Ultraschallsensoren als Sensorik für eine Punktwolkenerfassung sehr laut. Zwei Ultraschallsensoren sind jedoch für die Messwerterfassung geeignet. Der Hauptgrund hierfür ist ihre hohe Reichweite.

⁶fast fourier transformation

5.3.2 Triangulationssensoren

Triangulationssensoren gehören zur Klasse der optischen Aufnehmer. Triangulationssensoren werden für Aufgaben wie z.B. Füllstandsmessungen, Steuerung von Roboterarmen, Durchhang von Papier- und Kunststoffbahnen, Messungen des Einfederwegs bei Kraftfahrzeugen und in der mobilen Robotik zur Distanzmessung eingesetzt. Triangulationssensoren werden bei dem von J. und U. Altenburg entwickelten AMR *ROBOTECH 1* [2] eingesetzt, ebenso wie in einigen namenlosen AMR, die in dem Buch *Embedded Robotics* [12] beschrieben sind. In Abbildung 5.4 ist ein Lasertriangulationssensor der Firma *MEL Mikroelektronik* dargestellt.





Abb. 5.4: Triangulationssensor *MEL M1* (Quelle: [29])

Abb. 5.5: Doppeltriangulationssensor MEL M72 (Quelle: [29])

Triangulationssensoren funktionieren nach dem Licht-Triangulationsprinzip, das in Abbildung 5.6 dargestellt ist und im Folgenden erläutert wird.



Abb. 5.6: Triangulationsprinzip nach [49]

Ein Objekt, bei dem die Distanz zum Sensor gemessen werden soll, wird von einer Licht-

quelle aus über eine Fokussierlinse punktförmig beleuchtet. Auf dem Detektor im Sensor wird dieser Lichtfleck unter einem Winkel α mit Hilfe einer Abbildungslinse abgebildet. Die Position des Lichtflecks auf dem Detektor ist ein Maß für die Entfernung zum Objekt [49]. Der Detektor kann aufgrund seiner Lage innerhalb des Triangulationssensors ausschließlich Streulicht erfassen, aus diesem Grund muss immer ein diffuser Anteil vom zu messenden Objekt reflektiert werden. Glatte und spiegelnde Oberflächen können mit speziellen Doppeltriangulationssensoren erfasst werden [29] (s. Abb. 5.5).

Die Erfassung des diffusen Anteils kann mit positionsempfindlichen Detektoren (PSD), mittels CCD-Zeilensensoren⁷, oder mit Hilfe von Differential-Photodioden erfolgen. Der Lichtfleck auf dem Objekt ist bei den meisten Triangulationssensoren sehr klein, weil er scharf im Detektor abgebildet wird und somit eine sehr hohe Distanzauflösung möglich ist. Zur Lichterzeugung eignen sich u.a. Laserdioden und Leuchtdioden, wobei Laserdioden eine bessere Abstrahleigenschaft besitzen.

Handelsübliche Triangulationsensoren haben einen Arbeitsbereich, der bei 50mm...1500mm liegt, und sie werden meistens für Distanzmessungen unter 1m eingesetzt. Triangulationssensoren haben, wie bereits bei den Ultraschallsensoren erwähnt, einen Totbereich, der sich aus den geometrischen Beziehungen zwischen Detektor und Lichtquelle ableiten lässt, eine ausführliche Erklärung hierzu ist in [49] zu finden. Vor- und Nachteile sind in Tabelle 5.4 aufgeführt.

VORTEILE	NACHTEILE
Einfache Messwerterfassung/-auswertung	Empfindliche Optik
Hohe Zuverlässigkeit	Totbereich
Preisgünstig (LED)	Hohe Kosten (Laser)
Einfache Auswertung bei Distanzmessung	Fester Messbereich
Hohe Genauigkeit	Eindimensionale Abstandsmessung
Niedriger Stromverbrauch	

Tabelle 5.4: Vor-/Nachteile von Triangulationssensoren

Nachteilig ist, dass die Sensoroptik staub- und schmutzempfindlich ist und dadurch fehlerhafte Messungen entstehen können. Zusätzlich zu den benannten Nachteilen werden in Lasertriangulationssensoren Dioden der Laserschutzklasse 2 oder höher eingesetzt. Laserstrahlen dieser Klasse können bei Augenkontakt Schäden der Netzhaut verursachen. Ein Einsatz beim *ProfiBot*-Roboter zwecks Sensorik für die Punktwolkenerfassung wurde damit ausgeschlossen. Hinsichtlich einer erhöhten Sicherheit und eines größeren Erfassungsbereichs des *ProfiBot*-Roboters sind Triangulationssensoren ebenso wie Ultraschallsensoren als zusätzliche Sensorik in einer Sensorfusion einsetzbar. Preisgünstige Sensoren, wie beispielsweise LED-Triangulationssensoren, reichen für diese Anwendung vollkommen aus.

⁷licht- oder strahlungsempfindliche Sensoren, die aus einem zweidimensionalen Feld von Fotodetektoren bestehen

5.3.3 Kapazitive und induktive Wegaufnehmer

Kapazitive Wegaufnehmer werden in Anwendungen, in denen eine hohe Auflösung der Distanz erforderlich ist, z.B. in der Dickenmessung bei Papier- und Folienbändern, eingesetzt. Eine genaue Darstellung des Messprinzips ist bei Tränkler und Obermeier [49] aufgeführt. Die Reichweite dieser Sensoren beträgt maximal 50mm, so dass diese den in Kapitel 5.1 aufgeführten Anforderungen nicht entsprechen. Induktive Wegaufnehmer können nur Distanzen zu ferromagnetischen Objekten messen. Daher kommen sie für den Einsatz beim *ProfiBot*-Roboter nicht in Frage.

5.3.4 Laserscanner

Die Laserscanner gehören zur Gruppe der optoelektronischen Sensoren. Es wird zwischen 2D-Laserscannern und 3D-Laserscannern unterschieden. 3D-Laserscanner werden in der Geodäsie, in der Medizin und im Maschinenbau eingesetzt, vornehmlich um die geometrischen Daten von Objekten zu erfassen und aus den Daten CAD-Modelle zu erstellen. Mit Hilfe von 3D-Laserscannern kann eine dreidimensionale Punktwolke erfasst werden, was in der mobilen Robotik ein großer Vorteil hinsichtlich des Erfassens von Objekten ist. Ein Anwendungsbeispiel ist das Sensorsystem AIS 3D-Laserscanner⁸ des Roboters KURT3D (s. Kapitel 2.6.3), der sich einer Laserscanner-Nickvorrichtung für einen 2D-Laserscanner der Firma Sick zur dreidimensionalen Punktwolkenerfassung behilft. Dieses Sensorsystem ist gegenüber den übrigen gängigen Modellen von 3D-Laserscannern sehr kompakt. Ein hoher Installationsaufwand und eine komplexe Einbindung in die ProfiBot-Programmierumgebung sind Nachteile dieses Systems und der Grund, weshalb der AIS 3D-Laserscanner im Rahmen dieser Arbeit nicht verwendet wurde. Eine ausführliche Beschreibung des AIS 3D-Laserscanners findet sich bei Holz [22].



Abb. 5.7: 2D-Laserscanner LS4 (Quelle: [43])



Abb. 5.8: Schutzfelder eines Industrieroboters (Quelle: [43])

2D-Laserscanner werden vorwiegend zur berührungslosen Überwachung eines Schutzbereichs in Industrieanlagen, z.B. bei Maschinen, Industrierobotern und Fahrerlosen Trans-

⁸der AIS 3D-Laserscanner wurde vom Fraunhofer Institut für autonome intelligente Systeme entwickelt
portsystemen (FTS), eingesetzt. Sobald sich eine Person in einem definierbaren Schutzfeld befindet, wird ein Betriebstopp der Anlage ausgelöst. Als Beispiel ist der Laserscanner LS4 von Siemens in Abbildung 5.7 dargestellt. Als Anwendungsbeispiel sind in Abb. 5.8 die Schutzfelder eines 2D-Laserscanners in einer Roboterzelle dargestellt.

In der mobilen Robotik werden bevorzugt kompakte 2D-Laserscanner für die Objekterfassung eingesetzt, wie beispielsweise im Roboter *Scitos C5*.

Das Funktionsprinzip von Laserscannern ist sowohl bei 3D-Laserscannern, als auch bei 2D-Laserscannern das gleiche und wird nachfolgend erläutert. Laserscanner senden gepulstes Laserlicht aus, um Distanzen zu Objekten zu ermitteln. Hierzu wird ein Laserstrahl über einen rotierenden Scannerspiegel abgelenkt, von den zu erfassenden Objekten reflektiert und trifft im Empfänger (z.B. APD-Empfänger⁹) wieder auf. Durch die Rotation des Laserstrahls ist eine zweidimensionale Punktmengenerfassung möglich, ein sogenannter Scan mit n_s Einzelmessungen. Die scannerspezifische, geometrische Auflösung der Umgebung hängt von der Geschwindigkeit des Scannerspiegels ab und der Aufnahme- und Verarbeitungszeit der Empfängerelektronik. Zur Distanzmessung werden zwei Verfahren angewendet, die Impulslaufzeitmessung und die Phasendifferenzmessung.

Impulslaufzeitmessung

Bei der Impulslaufzeitmessung, auch TOF^{10} -Prinzip genannt, wird die Zeitspanne Δt zwischen der Emission eines Laserlichtimpulses und der Detektion im Empfänger gemessen und die Distanz r nach Gleichung 5.1 bestimmt.

$$r = \frac{c \cdot \Delta t}{2}, \ c \approx 300 \cdot 10^6 \frac{m}{s} \tag{5.1}$$

Phasendifferenzverfahren

Beim Phasendifferenzverfahren wird das ausgesendete, amplitudenmodulierte Laserlicht in der Empfängerelektronik als Referenzwert benutzt und mit dem Empfangssignal verglichen. Die Phasenverschiebung der beiden Signale ist proportional zur Distanz (s. 5.2).

$$d \sim \varphi_{Empf} - \varphi_{Send}, \ \{0 < \varphi_E < 2\pi\}$$

$$(5.2)$$

⁹avalanche photo diode

¹⁰time-of-flight

Die Wellenlänge n des Laserlichts kann je nach Laserscanner im Bereich $380nm^{11} < n < 1500nm^{12}$ liegen. Wellenlängen im nicht sichtbaren Bereich $n > \approx 900nm$ kommen in Anwendung mit Personenkontakt nicht in Frage. Der natürliche Lidschlusseffekt der Augen, der beim Blick in den Laserstrahl Schäden an der Netzhaut verhindern kann, ist bei diesen Wellenlängen nicht gegeben.



Abb. 5.9: Messprinzip eines 2D-Laserscanners mit 180° Erfassungswinkel

Mit 2D-Laserscannern, die zur berührungslosen Überwachung geeignet sind, lassen sich Schutzfelder lediglich über Anwendungsprogramme definieren. Mit diesen Programmen ist die Ermittlung von Distanzinformationen nicht möglich. Diese Informationen können nur direkt aus den Übertragungsdaten ausgelesen werden. Um die Distanzmesswerte abzufragen, müssen die Schnittstellen der 2D-Laserscanner ausgelesen werden.

2D-Laserscanner haben eine hohe Reichweite, die je nach Scanner zwischen 4m und 80m betragen kann. Abbildung 5.9 verdeutlicht das Messprinzip von Laserscannern. Die Messdaten von 2D-Laserscannern werden zyklisch nach jedem Scan als Array mit n_S Distanzinformationen ausgegeben, der dazugehörige Winkel φ_{scan} des Einzelscans lässt sich durch den Index k des Arrays und den Differenzwinkel $\Delta \varphi_{scan}$ zweier Scans nach der Gleichung $\varphi_{scan} = k \cdot \Delta \varphi_{scan}$ ermitteln. Die Vor und Nachteile von 2D-Laserscannern sind in Tabelle 5.5 aufgelistet.

2D-Laserscanner sind sehr zuverlässig und haben infolge ihrer weitläufigen Marktpräsenz einen hohen technologischen Standard. Dem hohen Anschaffungspreis der 2D-Laserscanner ab ca. 2000 Euro steht die Tatsache gegenüber, dass 2D-Laserscanner durch die benannten Verfahren Konturen erfassen können. Somit kann ein Scanner eine Vielzahl eindimensional messender Sensoren ersetzen. 2D-Laserscanner sind folglich für den Einsatz mit dem *ProfiBot*-Roboter gut geeignet.

¹¹blaues Laserlicht

 $^{^{12} {\}rm Infrarot-Laserlicht}$

Vorteile	NACHTEILE
Mehrdimensionale Messung	Empfindliche Optik
Hohe Zuverlässigkeit	Hohe Anschaffungskosten
Hohe Reichweite	

Tabelle 5.5: Vor-/Nachteile von Laserscannern

5.3.5 3D - Kameras

Für die Objekterfassung können 3D-Kameras eingesetzt werden. Zu den 3D-Kameras gehören Stereokameras oder Kameras, die mit dem Prinzip der Impulslaufzeitmessung arbeiten. Für diese Sensortypen gibt es zur Zeit noch keine spezielle Anwendung in der Industrie, weil sie relativ neue Sensoren sind. Stereokameras werden in dem Teilgebiet der Robotik eingesetzt, das sich mit humanoiden Robotern beschäftigt, wie z.B. bei dem Roboter Justin, der vom DLR^{13} entwickelt wurde (s. Abb. 5.10).



Abb. 5.10: Humanoider Roboter Justin (Quelle: [33])

Mit Hilfe von Stereokameras lassen sich Tiefenbilder von der Umgebung erzeugen. Bei einem Tiefenbild lassen sich Informationen über die Distanz zu Objekten anhand verschiedener Farbtöne (s. Abb. 5.11) oder ähnlicher Kriterien (z.B. Grautöne) feststellen. Eine Stereokamera besteht aus mindestens zwei Einzelkameras, die versetzt zueinander angeordnet sind. Das Beispiel *Bumblebee* vom Hersteller *Point Grey* in Abbildung 5.12 zeigt eine Stereokamera mit drei Einzelkameras.

Mit diesen Kameras wird die Distanz von Objekten in einem dreidimensionalen Raum detektiert, die aus den Aufnahmen der Einzelkameras mittels Querdisparität¹⁴ bestimmt wird, wie dies auch beim menschlichen Auge der Fall ist. Der große Vorteil von Stereokameras ist die dreidimensionale Erfassung der Umgebung. Die Auswertung des Tiefenbilds von Stereokameras benötigt einen hohen Rechenaufwand, der mit dem Steuersystem des *ProfiBot*-Roboters nicht in der geforderten Zykluszeit realisierbar ist. Außerdem ist für die Installation einer Stereokamera im *ProfiBot*-Roboter ein erheblicher Aufwand notwendig. Deshalb wurde ein Einsatz beim *ProfiBot*-Roboter ausgeschlossen.

 $^{^{13}\}mathrm{Deutsches}$ Zentrum für Luft- und Raumfahrt

¹⁴Verschiedenheit von Aufnahmen eines Objektes durch Versatz



Abb. 5.11: Tiefenbild (Quelle: [44])



Abb. 5.12: *Bumblebee XB3* (Quelle: [38])

Die Time Of Flight (TOF)-Kamera *Swissranger* vom Hersteller *CSEM* kann in der mobilen Robotik laut Herstellerangabe zur Hindernisvermeidung, Kartographie, Selbstlokalisation und Routenplanung eingesetzt werden [55]. Das aktuelle Modell ist in Abbildung 5.13 dargestellt. Das Funktionsprinzip der Kamera wird im folgenden erläutert:

von der LED-Matrix wird eine modulierte Infrarotstrahlung ausgesendet und an den Objekten in der Umgebung reflektiert. Die Strahlung wird von dem CMOS/CCD Bildsensor registriert und ausgewertet.



No contest

Abb. 5.13: *Swissranger SR-3000* (Quelle: [55])

Abb. 5.14: Punktwolke Styroporkugel (Quelle: [15])

Je nach Phasenverschiebung der Strahlung können Informationen über die Umgebung mittels Distanz, Intensität und Amplitude des erstellten Umgebungsbildes ausgewertet und eine dreidimensionale Punktwolke erstellt werden. Eine umfangreichere Darstellung des Funktionsprinzips ist bei Gut [15], bei May et. al. [27] und bei Holz [22] zu finden. Durch diese Art der Umgebungserfassung können Fehler entstehen, die sich auf die Genauigkeit der Punktwolke auswirken, wie beispielsweise Fehlereinflüsse durch Temperaturschwankungen oder Ungenauigkeiten bei der Infrarotlichtmodulation. Eine nähere Beschreibung der genannten Fehlereinflüsse, die sich auf ein Vorgängermodell der Kamera beziehen, ist bei Gut [15] zu finden. Es sind keine Quellen von Untersuchungen bekannt, die belegen, dass die bei Gut [15] benannten Fehler im aktuellen Modell behoben wurden. Diese Tatsache hat zu der Entscheidung beigetragen, diese Kamera nicht im Zusammenhang mit dem *ProfiBot*-Roboter anzuwenden. Die Kosten von 3D-Kameras sind etwas höher als die von Laserscannern.

5.4 Sensoren für Kollisionserkennung

Für die Kollisionserkennung werden in FTS^{15} und mobilen Robotern Schaltleisten und Kontaktpuffer eingesetzt. Schaltleisten oder Kontaktpuffer sind für eine Kollisionsvermeidung nicht geeignet, sie können jedoch die Folgeschäden einer Kollision minimieren. Der folgende Abschnitt behandelt das Funktionsprinzip dieser Sensoren sowie die Eignung für den Einsatz beim *ProfiBot*-Roboter.

5.4.1 Schaltleisten

Schaltleisten bestehen aus einem Gummiprofil (s. Abb. 5.15) und Komponenten, die eine Deformierung dieses Gummiprofils durch den Kontakt mit Objekten registriert. Für die Kontakterkennung werden unterschiedliche Verfahren eingesetzt, z.B. die Auswertung des Drucks im Inneren des Gummiprofils oder Elemente, wie beispielsweise Bandschalter, die bei Kontakt schalten. Bandschalter bestehen aus zwei Kupferbändern, die voneinander isoliert sind, an den Seiten sind sie durch Kunststoff isoliert, in der Mitte durch Luft.



Abb. 5.15: Aufbau einer Schaltleiste

Nach einer Kollision werden die beiden Kupferbänder in der Mitte zusammengedrückt, wodurch ein Kurzschluss entsteht. Der resultierende höhere Strom bei einem Kurzschluss wird von der Auswertelektronik gemessen und ein binäres Steuersignal erzeugt. Durch eine entsprechende Dimensionierung des Gummiprofils kann eine Schwellkraft vorgegeben werden, bei der die Bandschalter den Schaltvorgang ausführen. Schaltleisten mit Drucksensoren haben den Vorteil, dass analoge Signalwerte ausgewertet werden können, die proportional zum Druck $(\frac{\vec{F}}{A})$ sind. Weil das *ProfiBot*-Basissystem bereits Schaltleisten mit Bandschaltern besitzt, wurden diese auch weiterhin verwendet.

 $^{^{15}{\}rm fahrerlose}$ Transport systeme

5.4.2 Kontaktpuffer

Kontaktpuffer (s. Abb. 5.16), die auch *Safety Bumper* genannt werden, funktionieren nach dem gleichen Prinzip wie Schaltleisten. Kontaktpuffer bestehen aus einem Schaumstoffkern, der mit einer Behautung (meistens aus Polyurethan) versehen ist, um die Robustheit zu erhöhen. Sie sind wesentlich größer dimensioniert als Schaltleisten. Aus diesem Grund ist mehr Platz für die Montage erforderlich.



Abb. 5.16: Beispiel eines Kontaktpuffers (Quelle: [58])

Kontaktpuffer werden bei FTS eingesetzt, weil diese aufgrund der relativ hohen Fahrgeschwindigkeiten und dem höheren Gewicht einen längeren Anhalteweg als AMR haben. Anfragen an Hersteller haben ergeben, dass diese Kontaktpuffer im Zusammenhang mit einer dazugehörigen Auswertelektronik betrieben werden. Dadurch ist zusätzlicher Installationsaufwand nötig. Aufgrund der geringen Fahrgeschwindigkeiten des *ProfiBot*-Roboters ist der Einsatz von Kontaktpuffern nicht unbedingt notwendig.

5.4.3 Auswertung des Motorlaststroms

Für die Kollisionserkennung mit einem AMR ist eine Auswertung des Motorlaststroms $i_{L(t)}$ möglich. Das Prinzip wird im nachfolgend erläutert.

Innerhalb des TMC wird der Betrag des Motorstroms $|i_{M(t)}|$ über einen Shunt¹⁶ gemessen. Der Schaltplan dieser Baugruppe ist in Abbildung 5.17 dargestellt.

Zusätzlich zu dem gemessenen Strom kann der Strom auch durch ein Modell ermittelt werden. Mit dem Bemessungsstrom $i_{B(t)}$ im lastfreien Fahrbetrieb¹⁷, ergibt sich der in Gleichung 5.3 gezeigte Zusammenhang zwischen den Motorströmen.

$$i_{M(t)} = i_{B(t)} + i_{L(t)} \tag{5.3}$$

¹⁶Messwiderstand

¹⁷unter lastfreiem Fahrbetrieb wird in diesem Zusammenhang ein Fahrbetrieb der *ProfiBot*-Roboterplattform ohne zusätzliche Beladung oder Krafteinflüsse durch eine Kollision verstanden, bei zusätzlicher Beladung würde der Motorstrom in den Beschleunigungphasen proportional zur Beladung ansteigen.



Abb. 5.17: Schaltplan der Baugruppe zur Messung des Motorstrombetrags (Quelle: [9])

Bei lastfreiem Fahrbetrieb ist $i_L = 0$, wenn Reibkräfte vernachlässigt werden, somit ist $i_{M(t)} = i_{B(t)}$. Unter dieser Voraussetzung ist $i_{L(t)}$ proportional zur Beschleunigung a des AMR. Der Motorstrom kann mit einem Modell aus der Beschleunigung berechnet und mit dem realen Motorstrom verglichen werden. Eine Störfall ist erkannt, wenn der reale Motorstrom vom Modellstrom abweicht. Hierzu müssen die Reibungsverluste in das Modell einfliessen.

Im Kollisionsfall wirkt ein zusätzliches Lastmoment M_L auf die Motoren eines AMR. Proportional zu diesem Lastmoment ist der Betrag des Motorlaststroms, wie aus der zweiten Maschinengrundgleichung 5.4 zu erkennen ist. Die Herleitung der Maschinengrundgleichungen ist bei Müller und Ponick [31] zu finden. Beim *ProfiBot*-Roboter in der Basisversion kann die für das Modell benötigte Beschleunigung ausschließlich aus den Odometriedaten ermittelt werden. Hierzu wird zunächst die Geschwindigkeit $v \sim \frac{\Delta T icks}{T}$ berechnet, T ist die Zykluszeit des *IConnect*-Signalgraphen und $\Delta T icks$ die Encoderticks (relativ). Die Beschleunigung a ist $\frac{\Delta v}{T}$. Bei einer Zykluszeit von 20ms, die auch im modifizierten *ProfiBot*-Roboter beibehalten werden soll, werden für die Berechnungen nur einige Ticks gemessen. Fehlmessungen der Ticks wirken sich gravierend auf die Berechnungen von vund somit auf a aus.

$$M_{M(t)} = c_r \Psi_d \cdot i_{q(t)}, \ i_q = i_M, \ \Psi_d = const$$

$$(5.4)$$

Versuche, in denen die relativen Ecoderticks unter einer konstanten Sollbeschleunigung von $\frac{200Ticks}{s^2}$ im lastfreien Motorbetrieb gemessen wurden, ergaben, dass die aus den Odometriewerten ermittelten Geschwindigkeiten und Beschleunigungen Peaks aufweisen (s. Abb. 5.18). Deswegen wurde von dem Vergleich von i_L und *a* abgesehen. Dessen ungeachtet kann eine einfache Strombegrenzung integriert werden, bei der festgelegt wird, dass i_M nicht über einen gewissen Betrag ansteigen darf. Hier ist darauf zu achten, dass der festzulegende Betrag von i_M nicht im normalen Betrieb erreicht wird.



Abb. 5.18: $v_{(t)}$, $a_{(t)}$ und $i_{M(t)}$ mit $a_{soll} = \frac{200Ticks}{s^2}$

Dieses Verfahren ist einfach und benötigt für die Ausführung kein Modell, es ist dennoch sehr unzuverlässig.

5.5 Entscheidungsfindung

Distanzmessung

Für die Distanzmessung wurde ein 2D-Laserscanner ausgewählt. In Tabelle 5.6 werden drei ausgewählte 2D-Laserscanner gegenübergestellt.

Die Wahl fiel auf den 2D-Laserscanner URG-04LX des Herstellers Hokuyo, der in Abbildung 5.19 dargestellt ist, weil er die geringste Leistungsaufnahme sowie die geringsten Abmessungen hat und somit wenig Platz für eine Montage benötigt. Zusätzlich ist er der preisgünstigste Laserscanner (ca. 2000 Euro).

Modellbezeichnung	Leuze RS4-4	Sick S300	Hokuyo URG-04LX
Reichweite $[m]$	4	8	4
Zykluszeit $[ms]$	40	80	100
Preis [Euro]	ca. 2500	ca. 2500	ca. 1850
Erfassungswinkel	190	270	240
Versorgungsspannung $[V]$	24	24	5
Leistungsaufnahme $[W]$	ca. 8	8	2, 5
INSTALLATIONSAUFWAND	hoch	mittel	gering
Abmessungen	hoch	mittel	gering

Tabelle 5.6: Vergleich von drei zur Auswahl stehenden Scannertypen

Die Reichweite von 4m reicht für eine Hindernisvermeidung bei geringen Fahrgeschwindigkeiten aus. Der URG-04LX verfügt über die in Tabelle 5.7 aufgeführten Spezifikationen.



Abb. 5.19: Der im *ProfiBot*-Roboter verwendete 2D-Laserscanner *Hokuyo URG-04LX* (Quelle: [20])

Aus Tabelle 5.7 ist ersichtlich, dass der URG-04LX in die Laserschutzklasse 1 nach internationaler Norm eingestuft worden ist, d.h. die Laserstrahlung ist bei vorhersehbaren Bedingungen für den Menschen ungefährlich. Die USB- und RS232-Schnittstelle des URG-04LX ist zur Kommunikation zwischen Laserscanner und Rechner vorgesehen. Das Kommunikations-Protokoll wird über die RS232- und die USB-Schnittstelle ausgegeben. Die Übertragung erfolgt aufgrund der hohen Geschwindigkeit über die USB-Schnittstelle. Der URG-04LX kann direkt an dem USB-Port des Notebooks angeschlossen werden. Für den Betrieb ist ein Treiber vom Hersteller notwendig. Für die Ausgabe der Distanzwerte in der Programmierumgebung IConnect ist ein eigenes Modul notwendig, mit dem die Daten des USB-Ports ausgelesen und interpretiert werden können. Für das Auslesen des Kommunikations-Protokolls und die Datenkonvertierung sind bereits C++-Klassen vorhanden. Hierfür ist es lediglich erforderlich, diese Klassen in ein IConnect-Modul einzubinden. Aus dem Kommunikations-Protokoll können die Daten in 768 Schritten pro Scan ausgelesen werden, bei denen die Schritte 44 bis 725 die Daten der Distanzmessungen enthalten. Daraus folgt, dass der URG-04LX 681 Distanzessungen pro Scan durchführt. Übertragungsfehler können durch eine Paritätsprüfung registriert werden.

Sensortyp	2D-Laserscanner
Modell	URG-04LX
Laserwellenlänge	785 <i>nm</i>
Laserschutzklasse	1 (IEC60852-1)
Versorgungsspannung	$5V \pm 5\%$
Stromverbrauch	500mA
Reichweite	4000mm
Genauigkeit	$20mm \dots 1000mm: \pm 10mm$
	$1000mm$ $4000mm:\pm10\%$
Distanzauflösung	1mm
Erfassungswinkel	240
$\Delta \varphi_{scan}$	0,36
Schnittstellen	USB 2.0 $(12\frac{Mb}{s})$
	RS-232C $(115, 2\frac{kb}{s})$
Zykluszeit pro Scan T_{scan}	100 <i>ms</i>
Abmessungen	$50mm \ge 50mm \ge 70mm$
Scanrichtung	v.r.n.l.

Tabelle 5.7: Spezifikationen Hokuyo URG-04LX

Der weitere Aufbau des Kommunikations-Protokolls ist im Datenblatt des Herstellers [19] beschrieben und wird hier nicht weiter erläutert.

Für die Spannungsversorgung des URG-04LX ist ein DC-Spannungswandler notwendig, der die Batteriespannung auf konstante 5V reduziert. Ein einfacher Spannungsteiler kommt hierfür nicht in Frage, weil die Batteriespannung um ca. 5V schwanken kann und somit auch die Ausgangsspannung des Spannungsteilers um ca. 1V. Dies ist nicht zulässig, weil die Elektronik des URG-04LX ausschließlich für Unstetigkeiten von maximal 0,25V(s. Tab. 5.7) in der Versorgungsspannung ausgelegt ist. Ein Festspannungsregler kommt ebenfalls nicht in Frage. Grund ist die hohe Verlustleistung von ca. 10W, die aus der Differenzspannung von 24V - 5V = 19V und der Stromaufnahme des Scanners von 500mAhervorgeht.

Die Reichweite von 4000mm reicht bei den gringen Fahrgeschwindigkeiten des modifizierten *ProfiBot*-Roboters für eine Hinderniserkennung aus. Ein weiterer Vorteil des URG-04LX ist der relativ geringe Stromverbrauch von 500mA.

Ein Nachteil des URG-04LX ist die relativ hohe Zykluszeit von 100ms. Es muss in Fahrversuchen herausgefunden werden, ob sich die hohe Zykluszeit nachteilig auf die Echtzeitfähigkeit auswirkt. Weil der URG-04LX nur eine zweidimensionale Messwerterfassung ermöglicht, muss er so angebracht werden, dass auf der Scanebene möglichst viele Hindernisse erfasst werden können. Die beste Lösung ist eine Scanebene, die parallel zur Untergrundebene liegt. Mit dieser Lösung können alle Konturen der Umgebung in der Reichweite des URG-04LX erfasst, und sehr einfach auf das zweidimensionale, roboterzentrische Koordinatensystem umgerechnet werden.

Absicherung

Für eine komplette Absicherung des *ProfiBot*-Roboters in seiner gesamten Größe wurden zusätzliche Sensoren ausgewählt. Mit "Absichern" ist ein Erfassen von Hindernissen gemeint, die nicht auf der Fahrspur des *ProfiBot*-Roboters liegen. Dies ist beispielsweise bei einer Kurvenfahrt notwendig. Weil der *ProfiBot*-Roboter große Abmessungen hat (s. Kapitel 2.5) werden mehrere Sensoren eingesetzt.

Tabelle 5.8 zeigt die Kennwerte zweier Sensoren, die für den Einsatz am ProfiBot-Roboter näher betrachtet wurden.

Modellbezeichnung	Baumer UNAM 30U9101	Sharp GP2Y0
FUNKTIONSPRINZIP	Ultraschall	Triangulation (Infrarot-LED)
REICHWEITE $[m]$	2,5	1, 5
Preis [Euro]	ca. 270	ca. 20
Versorgungsspannung $[V]$	24	5
Leistungsaufnahme $[W]$	1,4	0, 15
INSTALLATIONSAUFWAND	gering	mittel
Abmessungen	mittel	gering

Tabelle 5.8: Sensortypen zur redundanten Messwertaufnahme

Die Auswahl ist auf Sensoren *GP2Y0XXX* von *Sharp* gefallen. Weil mehrere Sensoren eingesetzt werden sollen, ist der günstige Preis von Vorteil. Die Sensoren nehmen vergleichsweise eine geringere Leistung auf, als der *UNAM 30U9101*, was sich positiv auf die Betriebsdauer und somit auf die stationäre Verfügbarkeit auswirkt. Die geringere Reichweite ist im Vergleich, wie bereits erwähnt, bedeutungslos.

Für die Absicherung des *ProfiBot*-Roboters bei Kurvenfahrten werden vier Triangulationssenoren GP2Y0D02 (s. Abbildung 5.20) von *Sharp* eingesetzt.



Abb. 5.20: Die im *ProfiBot*-Roboter verwendeten Triangulationssensoren *Sharp GP2Y0D02* (Quelle: [53])

Der GP2Y0D02 gibt einen "High-Pegel^{"18} aus, sobald sich ein Objekt innerhalb einer Distanz von 800mm befindet. Für die Signalausgabe hat der GP2Y0D02 einen Open-Collector-Ausgang¹⁹ (s. Tietze und Schenk [48]). Zur Messwertaufnahme bieten sich hier die digitalen Eingänge des Buskopplers an (s. Kapitel 3.3.1). Weil der Open-Kollektor-

¹⁸in der Digitaltechnik ist ein "High-Pegel" eine logische 1, ein "Low-Pegel" eine logische 0

 $^{^{19} {\}rm offener}$ Kollektor-Ausgang eines n
pn-Transistors, dessen Emitter an Masse liegt

Ausgang im geschalteten Zustand ("High-Pegel") für eine Spannung $U_H < 5, 3V$ ausgelegt ist, muss eine geeignete Schaltung gefunden werden, welche die Messspannung so verstärkt, dass eine Messwerterfassung an den digitalen Eingängen des Buskopplers möglich ist. Diese erkennen erst ab einer Messspannung $U_H > 15V$ einen "High-Pegel". Weil diese Sensoren nur zur Erkennung von Hindernissen und nicht zur Distanzbestimmung eingesetzt werden, ist das binäre Schaltsignal von großem Vorteil. Es kann im Unterschied zu einem analogen Signal einfacher ausgewertet werden. Die Sensoren werden zur präventiven Unterstützung der Sensorik für die Kollisionserkennung und somit nicht zur Hindernisvermeidung eingesetzt. Das Schaltsignal löst einen Notstopp des *ProfiBot*-Roboters aus.

Sensorfusion

Zur Unterstützung des URG-04LX bei der Distanzbestimmung werden zusätzlich zwei Triangulationssensoren GP2Y0A02, ebenfalls von Sharp, in einer Sensorfusion eingesetzt. Es handelt sich hierbei um eine Sensorfusion, weil die Messwerte der beiden eingesetzten GP2Y0A02 und des URG-04LX sich in ihrer Messwertaufnahme gegenseitig ergänzen. Die sinnvolle Montage der Sensoren ist in Kapitel 6.1.1 beschrieben. Diese Sensoren sind baugleich mit dem GP2Y0D02, sie haben allerdings Analogausgänge und, wie in Tabelle 5.8 dargestellt, eine Reichweite von 1500mm. Weil die Ausgangsspannung nichtlinear zur Distanz ist (s. Abbildung 5.21), ist der Messbereich linearisiert worden (s. Kapitel 6.1.2).



Abb. 5.21: Ausgangsspannung/Distanz des Sensors GP2Y0A02 von Sharp (Quelle: [41])

Weil die Sensorvarianten GP2Y0D02 und GP2Y0A02 eine Versorgungsspannung von $V_{CC} = 5V$ benötigen, sind sie an den gleichen Spannungswandler angeschlossen, an den auch der URG-04LX angeschlossen ist.

Kollisionserkennung

Für die Kollisionserkennung werden Schaltleisten *TS16S* mit den Bandschaltern *121BP* vom Hersteller *Tapeswitch* benutzt. Es sind die gleichen, die bereits am *ProfiBot*-Roboter in einer ungünstigen Position montiert waren, wie in Kapitel 3.2.5 erwähnt. Diese Position muss verändert werden. Die Schaltleisten können in der ursprünglichen Montageposition Kollisionen an der Front- und an der Heckseite registrieren. Um in Kurvenfahrten Kollisionen erfassen zu können, müssen Schaltleisten an den Seiten montiert werden. Weil der modifizierte ProfiBot-Roboter nicht rückwärts fahren soll, wird die Schaltleiste am Heck nicht benötigt.

Die für das Sicherheitskonzept ausgewählten Sensoren in Tabelle 5.9 aufgeführt.

Sensor (Modellbezeichnung)	Anzahl
Hokuyo URG-04LX	1
Sharp GP2Y0D02	4
Sharp GP2Y0A02	2
Tapeswitch TS16S & 121BP	je 3

Tabelle 5.9: Übersicht der für das Sicherheitskonzept ausgewählten Sensoren

Kapitel 6

Technische Realisierung

In diesem Kapitel wird die Integration der in Kapitel 5 ausgewählten Sensoren in das *ProfiBot*-System beschrieben. Außerdem ist die Einbindung der in Kapitel 4 ausgewählten Navigationsverfahren und Roboterverhaltensweisen erläutert. Die Modifikationen am *ProfiBot*-Chassis werden beschrieben, die für die Integration der Sensorik, sowie für eine sichere Konstruktion des ProfiBot-Roboters notwendig gewesen sind. Zur Demontration der Einsatzmöglichkeiten ist auf dem Chassis eine Mülltonne montiert worden.

6.1 Physikalische Ebene

Dieses Kapitel dient der Darstellung des Hardwareaufbaus von dem modifizierten *ProfiBot*-Roboter. Nachfolgend werden die Hardwareeinbindung der Sensorik, die Softwareeinbindung und die notwendigen Modifikationen am Chassis des *ProfiBot*-Roboters beschrieben.

6.1.1 Hardwareintegration der Sensorik

Für die ausgewählte Distanzsensorik sind mehrere Einbaupositionen denkbar. Dabei muss beachtet werden, dass möglichst viele der in Kapitel 1.2.3 definierten Objekte in der untersuchten Einbauposition erfasst werden. Der Einbauposition des 2D-Laserscanners URG-04LX wurde eine höhere Priorität gegenüber den anderen Sensoren gegeben, weil der URG-04LX Sensor eine höhere Reichweite als die anderen Sensoren sowie einen größeren Erfassungsbereich besitzt. Es ist festgelegt worden, dass die Position des Sensorzentrums kongruent zum Mittelpunkt des URG-04LX ist. Weil Konturen erfasst werden sollen (s. Kapitel 5.5), muss die Scanebene des URG-04LX parallel zur Bodenfläche liegen. Die Position des URG-04LX wurde auf die Abszisse des roboterzentrischen Koordinatensystems gelegt. So ist eine Umrechnung der Distanzwerte in das kinematischen Zentrum des *ProfiBot*-Roboters einfacher durchzuführen, weil das kinematische Zentrum ebenfalls auf der Abszisse liegt. Der URG-04LX sollte an der Frontseite montiert werden, und zwar so, dass der halbe Erfassungswinkel des Laserscanners ungefähr 0 im roboterzentrischen Koordinatensystem beträgt. Vor diesem Hintergrund sind drei sinnvolle Einbaupositionen des URG-04LX am ProfiBot-Roboter in Abbildung 6.1 dargestellt.



Abb. 6.1: Mögliche Einbaupositionen (1 bis 3) des URG-04LX

Einbauposition 1 zeigt den URG-04LX auf dem Sensoraufbaurahmen des Chassis. Diese Position hat den Vorteil, dass der volle Erfassungswinkel des URG-04LX von $\gamma = 240$ ausgenutzt werden kann. Hierbei können Objekte verzeichnet werden, die mindestens so hoch sind wie der *ProfiBot*-Roboter.

Einbauposition 2 stellt den Scanner mittig auf der Frontseite dar. In dieser Position kann der volle Erfassungswinkel nicht ausgenutzt werden, sondern ein Winkel von höchstens $\gamma = 180$.

 γ ist in Einbauposition 3, in welcher der URG-04LX verkehrt herum am Chassis montiert ist, ebenfalls 180. In dieser Einbauposition können alle relevanten Objekte erfasst werden, die sich auf dem Boden befinden. Aus diesem Grund wurde Einbauposition 3 ausgewählt.

Da der URG-04LX sich in dieser Position im Trittbereich von Personen befindet, konnte er durch Tritte mechanisch beschädigt werden. Daher wurde ein Schutzblech konstruiert, das den Scanner schützt. Das Schutzblech wurde auf dem Bodenrahmen montiert, der in Kapitel 6.1.3 beschrieben ist.

Es sind Einbaupositionen denkbar, bei denen der URG-04LX geneigt wird. Eine Distanzbestimmung würde hierbei wegen des zusätzlichen Rechenaufwandes erschwert werden. Eine Neigung der Scanebene nach unten kann sich negativ auf die Reichweite auswirken, wenn der Neigewinkel zu groß ist. Bei einer Neigung nach oben könnten Personen in den Laserstrahl blicken. Trotz der als ungefährlich eingestuften Laserstrahlung (s. Kapitel 5.5) der Laserschutzklasse 1 ist ein Augenkontakt zu vermeiden. Von einer Neigung in die eine oder andere Richtung wurde daher abgesehen.

Die Triangulationssensoren sind so montiert worden, dass sie den Bereich um den *ProfiBot*-Roboter abdecken, den der Laserscanner nicht erfassen kann. Hierzu gehören die Seiten, das Heck und der obere Bereich der Front. Als Halterung für die Triangulationssensoren ist der Mülltonnenaufsatz gewählt worden, der in Kapitel 6.1.4 beschrieben wird. Die Wahl der Einbauposition wird ebenfalls in Kapitel 6.1.4 beschrieben.

Die analogen Ausgänge der zwei Triangulationssensoren GP2Y0A02 sind direkt an die Analogeingänge der Inline-Klemme *IB IL AI 2/SF* angeschlossen worden. Die vier digitalen Triangulationssensoren GP2Y0D02 sind mit einer Verstärkerschaltung an die digitalen Sensoreingänge des Buskopplers angeschlossen worden.

Um die gesamte Distanzsensorik mit Spannung zu versorgen, ist ein Spannungswandler der Leistungsklasse 5W mit der Ausgangsspannung 5V benutzt worden. Der Spannungswandler wurde in einem Gehäuse untergebracht und ausgangsseitig mit einer Drei-Stock-Klemme verbunden, an welche die Sensoren angeschlossen sind. Die Sensoren haben insgesamt eine Stromaufnahme von $500mA + 6 \cdot 33mA = 698mA$, somit wird der maximale Ausgangsstrom des Spannungswandlers von 1A nicht überschritten.

Für die Messwerterfassung der Triangulationssensoren wurde ein *Sensorboard* und ein *Verstärkerboard* erstellt. Sowohl die vier *GP2Y0D02* als auch die zwei *GP2Y0A02* sind an dem *Sensorboard* angeschlossen, das sich in räumlicher Nähe der Sensoren befindet. Der Schaltplan ist in Abbildung 6.2 dargestellt.



Abb. 6.2: Schaltplan des Sensorboards

Das Sensorboard ist mit dem Verstärkerboard über ein 10-poliges Flachbandkabel SB verbunden. Die Ausgangssignale der Sensoren, die an den Pins XX-1 anliegen, werden über dieses Flachbandkabel zum Verstärkerboard weitergeleitet. Die Zusammenführung der sechs Ausgangssignale macht den Aufbau übersichtlich. Nur ein gemeinsames Kabel ist notwendig, das zum Verstärkerboard geführt wird. Die Pins 5V und GND sind über eine separate Versorgungsleitung angeschlossen.



Abb. 6.3: Verstärkerschaltungen zur Schaltsignal-Auswertung der Triangulationssensoren GP2Y0D02

Das Verstärkerboard, über welches die vier digitalen Distanzsensoren GP2Y0D02 angeschlossen sind, besteht aus vier invertierenden Verstärkerschaltungen, wie in Abbildung 6.3 dargestellt. Die PNP-BJTs¹ (Q1, Q2, Q3 und Q4) sind in einer Emitterschaltung betrieben, die NPN-BJTs (Q5, Q6, Q7 und Q8) in einer Basisschaltung. Die Reihenschaltungen der Widerstände R5, R8, R11 und R14 mit den Widerständen R4, R7, R10 und R13 stellen die Arbeitswiderstände für die NPN-BJTs. Im ungeschalteten Zustand der Distanzsensoren sperren alle Transistoren. Schalten Q1 ... Q4 durch, sind die Emitterspannungen größer als der "High-Pegel" der digitalen Sensoreingänge vom Buskopplers. Bei einem "High-Pegel" am entsprechenden Sensoreingang des Buskopplers befindet sich kein Objekt innerhalb der Reichweite der GP2Y0D02. Bei einem Schaltvorgang wird der

¹bipolar junction transistor

Basistrom unterbrochen und die Eingangsspannungen am Buskoppler sind kleiner als der "Low-Pegel". Die Schaltsignale werden an den Boardausgängen *Out4*, *Out6*, *Out8* und *Out10* mit dem zehnpoligen Flachbandkabel *OUT* zu den digitalen Eingängen 1, 2, 3 und 4 des Buskopplers geführt. Im ungeschalteten Zustand liegt an den Boardausgängen bzw. an dem Buskopplereingängen eine Spannung von ca. 22V an. Im geschalteten Zustand liegt an den Sensorausgängen eine Schaltspannung $U_H \approx 2V$ an, die unterhalb der maximal zulässigen Spannung der Sensorausgänge liegt. Diese Spannung wird mit dem Spannungsteiler R2 und R3 erzeugt. Die Kondensatoren C1 und C2 glätten Spannungsspitzen bei den Schaltvorgängen. Die Ausgänge der analogen Distanzsensoren SB1 und SB2 werden unverändert durch das Verstärkerboard geschleift und über das Flachbandkabel *OUT* auf die Inline-Klemme geführt. Durch die Invertierung der Signale bei den *GP2Y0D02* liegt im Fall eines Leitungsbruchs ein "Low-Pegel" (Objekt innerhalb der Reichweite der *GP2Y0D02*) ein Notstopp ist, stoppt der ProfiBot-Roboter auch bei einem Leitungsbruch. Aufgrund dieser Maßnahme kann eine höhere Fahrsicherheit gewährleistet werden.

Die Layouts für die Platinen beider Boards wurden mit dem Programm *Eagle* erstellt und sind im Anhang zu finden.



Abb. 6.4: Spannungseinbrüche der GP2Y0D02 (Basisspannung von Q3 ohne R3)

Im ungeschalteten Zustand sinkt die Ausgangsspannung der *GP2Y0D02* in zufälligen Zeitintervallen unter den "High-Pegel". Die gemessene Dauer eines solchen Spannungseinbruchs beträgt ca. 40ms (s. Abbildung 6.4), das entspricht der Dauer eines Messzyklusses der Triangulationssensoren (s. Datenblatt [41]). Nach Überprüfung des *Verstärker*- und *Sensorboards*, bei der keine Fehler festgestellt wurden, können die Spannungseinbrüche nur aufgrund von Störspannungen in den Leitungen oder von einer Baugruppe der Sensoren selbst verursacht werden. Fehlerbehebung wäre, anstelle der verwendeten Litzen dreiadrige, geschirmte Leitungen zu verwenden. Die andere Möglichkeit ist die durch die Spannungseinbrüche verursachten Schaltsignale durch ein entsprechendes *IConnect*-Modul herauszufiltern. Grundsätzlich ist immer eine Hardwarelösung vorzuziehen, weil jedoch die Fehlerursache nicht eindeutig bekannt ist, ist eine Softwarelösung gefunden worden. Mit Hilfe eines Interpret-Moduls wird ein "High-Pegel" erst dann als logische 1 erkannt und ausgegeben, wenn fünf Zyklen hintereinander eine logische 1 am Eingang des Moduls anliegt. Für die Schaltleisten ist eine Einbauposition gewählt worden, die 20mm über dem Boden liegt. Somit hat der *ProfiBot*-Roboter noch eine genügend große Bodenfreiheit, um über Teppiche oder Fußmatten zu fahren. Bei der gewählten Bodenfreiheit ist die Wahrscheinlichkeit am größten, dass alle Kollisionen mit den definierten Hindernissen (s. Kapitel 1.2.3) erkannt werden. Für die Auswertung des Schaltsignals liegt es nahe, die digitalen Eingänge des Buskopplers zu benutzen. Bei einem "High-Pegel" ist eine Kollision erkannt, bei einem "Low-Pegel" liegt ein störungsfreier Fahrbetrieb vor. Durch den Anschluss an den Buskoppler wird die Auswertelektronik eingespart.

Die Schaltleisten hatten einen Widerstand, über den der Strom zur Leitungsbrucherkennung (s. Kapitel 3.2.5) geflossen ist. Dieser Widerstand ist entfernt worden, weil ein Strom von 3, 3mA, der im ungeschalteten Zustand und somit im störungsfreiem Betrieb fließt, die Erkennung eines "High-Pegels" verursacht (vgl. Abbildung 3.9). Ein Leitungsbruch sollte im idealen Fahrbetrieb (keine Kollisionen) nicht vorkommen, weil die Leitungen der Schaltleisten, wie auch die der GP2Y0D02, gering dynamisch beansprucht werden. Um die Funktionsweise der Schaltleisten zu überprüfen, können sie vor dem Start des ProfiBot-Roboters betätigt werden. Bei einer intakten Leitung müssen die LEDs über den ensprechenden Sensoreingängen am Buskoppler aufleuchten. Die Schaltleisten wurden an den digitalen Eingängen 5, 6 und 8 des Buskopplers angeschlossen. An dieser Stelle sei darauf hingewiesen, dass mit der Sensorauswertung über den Buskoppler ausschließlich ein softwareseitiger Systemstopp erfolgen kann. In der Basisversion erfolgte die Auswertung der Schaltleisten ebenfalls softwareseitig über den TMC. Für eine erhöhte Sicherheit sollte ein hardwareseitiger Systemstopp der Kategorie 0 erfolgen, bei dem die Energieversorgung der Motoren unterbrochen wird. Hierdurch sinkt die stationäre Verfügbarkeit, weil ein selbständiger Wiederanlauf nicht möglich ist. Deswegen ist ein Systemstopp der Kategorie 0 nur durch den Notaus-Schalter realisiert worden. Durch die Schaltleisten erfolgt ein Systemstopp der Kategorie 2.

6.1.2 Softwareintegration der Sensorik

Die Integration der Schaltleisten und der Triangulationssensoren in die Programmierumgebung *IConnect* beruht auf dem vorhandenen *IConnect*-Modul *ILBETH*, das die an den Eingängen des Buskopplers anliegenden Messdaten ausgibt.

Für die Einbindung des 2D-Laserscanners URG-04LX wurde ein IConnect-Modul, mit der Bezeichnung URG04LX erstellt. Das Modul liest den USB-Port am Notebook aus, an dem der URG-04LX angeschlossen ist. Für die Einbindung wurden die in der FAIRLib² enthaltenden Klassen IDeviceAdapter, CComAdapter und CHokuyoScanner benutzt. Mit Hilfe der Klasse CHokuyoScanner wird eine Instanz des URG-04LX erstellt, die ande-

²Fraunhofer Autonomous Intelligent Robotic Libary. Die FAIRlib ist eine vom Fraunhofer Institut für Intelligente Analyse- und Informationssysteme erstellte Programmbibliothek, welche die Entwicklung von Steuerprogrammen und Verhaltensweisen eines AMR unterstützt.

ren beiden Klassen initialisieren den USB-Ports. Ein Scan des URG-04LX wird mit der Methode getScan(double[]) eingelesen, die zu der Klasse CHokuyoScanner gehört. Diese Methode liefert als Rückgabewert ein Double-Array mit den 681 Messwerten pro Scan (s. Kapitel 5.3.4). Die Distanzinformationen werden in ein Integer-Array kopiert und über den Modulausgang *Data* zur Weiterverarbeitung in *IConnect* ausgegeben. Der C++-Quellcode dieser Rechenoperationen ist in Abbildung 6.5 dargestellt.

```
///
// !!!The code of the Hokuyo-URG 04LX module is beginning here.!!!
//
int i = 0;
int nRange = myScanner->getValuesPerScan(); //get the number of measures of one scan
m data.SetSize(RANGE); //sets the size of m_data to 511
double points[681];
myScanner->getScan(points); //get a scan
for (i=0;i < (nRange-170) ;i++) //copies the data to m_data
{
    m_data[i] = points[(i+85)];
}
if (IsOutputConnected(m_data)) //Writes m_data to Output
ret = Write(m_data);
if (ret < 0) return ret;
return 0;</pre>
```

Abb. 6.5: Quellcodeauszug aus dem IConnect-Modul URG04LX

Durch die Konvertierung des Double-Arrays zu einem Integer-Array wird der Rechenaufwand für eine Weiterverarbeitung reduziert. Es entsteht kein Datenverlust, weil das eingelesene Double-Array aus Ganzzahlen besteht.

Durch die gewählte Einbauposition sind die Distanzwerte innerhalb des Scanwinkels $\varphi_{scan} = 0 \dots 30$ und $\varphi_{scan} = 210 \dots 240$ nicht verwendbar. Die entsprechenden Indizes des Double-Arrays von 0 \dots 85 und von 596 \dots 681 werden aus dem Double-Array herausgefiltert. Das Integer-Array hat aus diesem Grund eine Größe von 511.

Die annähernd hyperbolischen Kennlinien der Triangulationssensoren GP2Y0A02 werden in *IConnect* approximiert. Es ist möglich, diese Kennlinie durch eine Funktion $f_{(U_s)}$ zu beschreiben und die Distanz $d_s = f_{(U_s)}^{-1}$ aus den A/D-Ausgabewerten des Buskopplers U_s des Sensors zu berechnen. Eine solche Funktion zu finden, die zufriedenstellende Ergebnisse liefert, würde den Rahmen dieser Arbeit sprengen. Aus diesem Grund ist eine einfachere Methode gewählt worden.

Die Kennlinien beider GP2Y0A02 wurden durch Messungen erfasst (s. Abb. 6.6). Für die Durchführung von n Messungen wurde ein Pappkarton in Distanzen

$$\sum_{k=1}^{n} d_{s}^{*} = \left\{ d_{s,1}^{*}, d_{s,2}^{*}, d_{s,3}^{*}, \dots, d_{s,n}^{*} \right\}$$

mit $d_{s,1}^* = 20mm$, $d_{s,2}^* = 50mm$ und $d_{s,3}^* = 100mm$ und anschliessend in Distanzintervallen $\Delta d_s^* = 100mm$ zum jeweiligen Sensor aufgestellt und die entsprechenden AD- Ausgabewerte des Buskopplers

$$\sum_{k=1}^{n} U_{s}^{*} = \left\{ U_{s,1}^{*}, U_{s,2}^{*}, U_{s,3}^{*}, \dots, U_{s,n}^{*} \right\}$$

aufgenommen.



Abb. 6.6: Aufgenommene Messpunkte der beiden GP2Y0A02

Es wurde für jeden Messpunkt ein Mittelwert $\bar{U}_{s,k}^*$ aus ca. 500 Messwerten gebildet. Erkennbar ist, dass die beiden hyperbolischen Kennlinien einen unterschiedlichen Offset haben und bei einer Distanzerfassung von 1300mm und im Teil des Totbereich 100mm $< d_s <$ 200mm auseinanderdriften. Die Kennlinie von *GP2Y0A02* 2 weist ab ca. 1400mm einen positiven Anstieg auf.

Die Kennlinien wurden durch die Verbindung von zwei benachbarten Messpunkten mit Geraden $g_{(U_{s,k})}$ approximiert. Zunächst wird entschieden, in welchem Intervall von $U_{s,k}^*...U_{s,k+1}^*$ sich der gemessene Wert von U_s befindet. Im Anschluss wird d_s mit der jeweiligen Geradengleichung 6.1 interpoliert.

$$d_{s} = \frac{\Delta U_{s}^{*}}{\Delta d_{s}^{*}} \cdot \left(U_{s} - U_{s,k}^{*}\right) + d_{s,k}^{*} = m_{s,k} \cdot U_{s,k} + b_{s,k}$$

$$\Delta U_{s}^{*} = U_{s,k+1}^{*} - U_{s,k}^{*}$$

$$\Delta d_{s}^{*} = d_{s,k+1}^{*} - d_{s,k}^{*}$$
(6.1)

Als Anschauungsbeispiel ist die approximierte Kennlinie von GP2Y0A02 1 in Abbildung 6.7 dargestellt. Aus der Abbildung ist zu entnehmen, dass bei einer linearisierten Kennlinie

die Abweichung zu den realen Messpunkten unzulässig hoch ist. Bei der Berechnung von d_s muss ausgeschlossen werden, dass sich ein Objekt im Totbereich der Sensoren befinden kann, weil ansonsten die Messwerte doppelt interpretiert werden. Durch die Montagepositionen der Sensoren ist dies bei $0mm < d_s \leq 100mm$ der Fall (s. Kapitel 6.1.4). Die Approximation wird mit einem *IConnect*-Macro mit Hilfe von aufgenommenen Messwerten durchgeführt, die in Tabellen gespeichert sind. In Abbildung 6.8 ist der Signalgraph des *IConnect*-Macros dargestellt. Die Tabellen werden mit dem Modul *LoadTable* in *IConnect* eingebunden und als Double-Arrays weiterverarbeitet. In den *Interpret*-Modulen *linearisation* werden die Steigungen $m_{s,k}$ und die Konstanten $b_{s,k}$ von $g_{(U_{s,k})}$ bestimmt und als Double-Array ausgegeben.



Abb. 6.7: approximierte Kennlinie von GP2Y0A02 1

Die Module *compare* bestimmen das Intervall von U_s und haben als Ausgabewert den Index k. Mit Hilfe von k können im Modul $GP2Y_X_Out$ die jeweiligen Werte von $m_{s,k}$ und $b_{s,k}$ zur Erstellung von $g_{(U_{s,k})}$ aus dem Double-Array herausgefiltert werden.



Abb. 6.8: IConnect-Macro zur Approximation der Kennlinien

6.1.3 Modifikationen am Chassis

Die Modifikationen am Chassis sind primär für die Sicherheit im Kollisionsfall geeignet. Alle Objekte, die einer zu definierenden Mindesthöhe entsprechen, sollen im Kollisionsfall detektiert werden können. Objekte, die dieser Mindesthöhe nicht entsprechen, sollten so klein sein, dass die Roboterplattform über sie hinwegfahren kann. Unter dem Aspekt der Mindesthöhe und unter Berücksichtigung der Personensicherheit ist die Anatomie eines menschlichen Fußes³ beim Design des Bodenrahmens mit einkalkuliert worden.

- 1. Die Roboterplattform soll nach der Modifikation Füße nicht mehr überfahren (s. Kapitel 3.2.5).
- 2. Es dürfen keine harten Komponenten der Plattform mit einem Fuß, insbesondere mit dem Knöchel kollidieren.

Es sind weiterhin folgende Anforderungen an den Rahmen gestellt:

- Keine Ecken und Kanten \Rightarrow verringerte Verletzungsgefahr
- Rahmendesign unter Berücksichtigung der Sicherheitsanforderungen \Rightarrow Produktreife
- Einfache Montage \Rightarrow evtl. Kleinserienfertigung
- Integration des 2D-Laserscanners URG-04LX
- Niedrige Kosten

Unter Berücksichtigung der genannten Kriterien wurde ein Bodenrahmen entworfen, der sich aus den in Tabelle 6.1 aufgeführten Komponenten zusammensetzt.

³In der definierten Umgebung ist vorauszusehen, dass die Personen Schuhe tragen

ANZAHL	Түр	Länge
1	Halterung für Schaumstoffleisten	-
1	Scannerschutzblech	-
3	Schaltleisten	-
3	Schaltleistenhalterungen	-
2	Schaumstoffprofile	-
4	Aluminiumprofilstangen	490mm
2	Aluminiumprofilstangen	90mm
2	Aluminiumprofilstangen	80mm
2	Aluminiumprofilstangen	60 <i>mm</i>
6	Winkelsätze	-
8	Automatikverbinder	-

Tabelle 6.1: Komponenten für den Bodenrahmen

Die Positionierung des 2D-Laserscanners URG-04LX ist so gewählt worden, dass die Sicherheit im Fahrbetrieb nicht zusätzlich beeinträchtigt und eine breite Erfassung der Umgebung ermöglicht wird. Da sich die Plattform in Vorwärtsrichtung bewegt, ist die Positionierung des Scanners an der Vorderseite gewählt worden. Des Weiteren ist der URG-04LXkopfüber montiert, weil sich das Objektiv auf der Oberseite befindet. Durch diesen Einbau entsteht im Betrieb ein bodennaher Scan, der auch niedrigere Objekte erfasst. Der korrekten Position des 2D-Laserscanners wurde höchte Priorität gegeben.

Die Schaltleisten sind ebenfalls in Bodennähe befestigt, als Träger wird der Bodenrahmen (s. Abb. 6.9) verwendet. Der Abstand zwischen Bodenrahmen und Boden beträgt 20mm. Die Position des Laserscanner ist in Abbildung 6.10 erkennbar.

Die beiden Schaumstoffleisten, die sich über den Schaltleisten befinden, sind zur Absicherung gegen Stöße im Bereich oberhalb von 40mm über Grund. Als Material der Schaumstoffleisten ist ein Verbundschaumstoff der Firma Nanotech gewählt worden. Die Auswahl des Verbundschaumstoffs begründet sich auf folgenden Aspekten:

- Dämpfende Wirkung bei einem Aufprall
- Günstiger Preis
- Schaumstoff mit sehr hoher Festigkeit

Trotz dieser Vorteile weist der Verbundschaumstoff einige Nachteile auf.

- Schlechte Bearbeitbarkeit
- Unpassende Optik durch den Verbund von Schaumstoff in verschiedenen Farben



Abb. 6.9: Bodenrahmen, 1 Halteblech, 2 Schaumstoffleisten, 3 Schaltleisten, 4 Scannerschutzblech, 5 Scannerhalter

Anfragen an verschiedene Firmen, die alternative Lösungen anbieten, haben gezeigt, dass diese Lösungen kostspielig sind. Die Kontaktpuffer der Firmen sind immer mit einer zusätzlichen Sensorik versehen, die Stöße detektiert. Zur Integration in das Gesamtsystem der Roboterplattform ist zusätzlich meist eine Auswertelektronik vom gleichen Hersteller notwendig, die die Kosten des Systems weiterhin ansteigen lässt. Die zusätzliche Integration einer separaten Sensorik ist beim Design des Bodenrahmens nicht vorgesehen. Des Weiteren besteht die Möglichkeit einen Kantenschutz einzusetzen. Der wird nur in bestimmten Größen hergestellt. Das Design des Bodenrahmens muss nach den jeweiligen Abmaßen des Kantenschutzes angepasst werden.

Genaue Berechnungen der Tiefe der verwendeten Schaumstoffleisten sind nicht vorgenommen worden, da keine genaue Angabe über die Festigkeit des Schaumstoffmaterials vorhanden ist. Um die Robustheit des Schaumstoffs zu steigern, wurden Vefahren für die Fertigung einen nachträglichen Schaumstoffüberzugs untersucht. Folgende Alternativen wurden verglichen:

1. Überzug aus Schrumpfschlauch

Schrumpfschlauchüberzüge sind wegen geringer Maßhaltigkeit nach dem Schrumpfvorgang nicht geeignet. Der Schaumstoffbumper würde unförmig werden und sich nur noch schwer am Halteblech montieren lassen. Es ist außerdem schwierig, Schrumpfschläuche in einer Größe zu finden, die für einen Überzug geeignet ist.

2. Behautung aus Polyurethan

Die Behautung aus Polyurethan ist die von Herstellern am häufigsten angewendete Methode, um eine Schutzschicht über Schaumstoffbumper zu ziehen. Nachteilig sind die hohen Kosten.

3. Überzug aus Stoff oder Leder

Bei einem Überzug aus Stoff oder Leder ist der Zeitaufwand aufgrund der langen Fertigungszeiten einer Maßfertigung zu hoch. Weiterhin ist robuster Stoff bzw. Leder sehr teuer.

Der Schaumstoffbumper wurde folglich im Rahmen dieser Arbeit mit keinem der genannten Verfahren behandelt, die Behautung aus Polyurethan wäre jedoch für ein Redesign die beste Alternative, um den Schaumstoffbumper robuster zu gestalten.

Der Grundrahmen des Bodenrahmens ist aus Aluminiumprofilstangen erstellt, um das Konzept der Erweiterbarkeit von *ProfiBot* zu bewahren. Die Konstruktion ist außerdem günstig, da die Profile aus Meterware hergestellt sind. Dieser Prozess spart Zeitkosten gegenüber einer Lösung, bei der die Teile für den Grundrahmen noch anderweitig bearbeitet werden müssen. Der Grundrahmen wird mit Hilfe von sechs Haltern am Chassis montiert, wie in Abbildung 6.11 zu sehen ist. Eine einfache Montage und Demontage ist somit gewährleistet.

Die Schaltleisten befinden sich am Träger des Grundrahmens. Zur Befestigung der Schaumstoffbumper am Bodenrahmen, wird ein Halteblech verwendet. Das Halteblech ist direkt am Bodenrahmen befestigt. Das Halteblech ist in Abbildung 6.10 erkennbar.





Abb. 6.11: Grundrahmen

Das Scannerschutzblech schützt den URG-04LX vor Schäden bei einer Kollision. Die Schaltleiste unter dem Scannerschutzblech wurde gebogen, damit sie vor dem Scanner verlaufen kann. Dadurch kann eine Kollision im Bereich vor dem Scanners detektiert werden. Eine weitere Möglichkeit besteht darin, die Schaltleiste trapezförmig um das Scannerschutzblech zu verlegen. Dadurch entstehen jedoch Kanten an den Außenflächen des

Bodenrahmens, weswegen diese Möglichkeit nicht weiter betrachtet wurde.

Durch die gebogene Schaltleiste an der Vorderseite haben die beiden Schaumstoffleisten dort kaum eine Funktion. Deshalb ist noch ein anderer Designvorschlag in Betracht gezogen worden, bei dem die Schaumstoffleisten bündig mit der gebogenen Schaltleiste sind. Entweder könnten die Schaumstoffleisten im Frontbereich tiefer werden und das Halteblech in seiner ursprünglichen Version belassen werden oder das Halteblech müsste gebogen werden und die Schaumstoffleiste in ursprünglicher Ausführung eingelegt werden. Beide Möglichkeiten hätten den Vorteil der besseren Pufferung bei einer Kollision. Schwierig ist es, die Schaumstoffleisten mit einer variablen Tiefe zu beziehen, weil es sich um Sonderformen handelt. Der Schaumstoff dafür müsste in eigener Produktion manuell bearbeitet werden. Weiterhin ist es auch nicht möglich, die Laschen an der Ober- und Unterseite des Halteblechs ganzteilig in dem geforderten Radius zu biegen. Um eine Biegung herzustellen müssten die Laschen durch lamellenartige Einschnitte zerteilt werden. Es muss bei beiden Möglichkeiten darauf geachtet werden, dass eine Aussparung von Schaumstoffleisten und Halteblech in der Scanebene des URG-04LX vorhanden ist. Diese Möglichkeiten sind nicht im Rahmen dieser Arbeit realisiert worden.

Nachfolgend sind eine Verbesserungsvorschläge für ein Redesign des Bodenrahmens augezählt.

- 1. Veränderung der Montageposition beider Halterungen an der Frontseite
 - $\bullet \rightarrow$ Bearbeitung der Schaumstoffleiste an der Frontseite entfällt
 - $\bullet~\rightarrow$ Bearbeitung des Halteblechs an der Frontseite entfällt
- 2. Montage der Seitenhalter an der gegenüberliegenden Seite des Halteblechs
 - $\bullet \rightarrow$ geringere Anzahl an Bearbeitungsschritten des Halteblechs an den Seiten
 - $\bullet\,\to$ Bearbeitung der Schaumstoffleiste an der Seiten entfällt

6.1.4 Mülltonnenaufsatz

Der Mülltonnenaufsatz (s. Abbildung 1.1 in Kapitel 1.2) des *ProfiBot*-Roboters ist für Demonstrationszwecke der Einsatzmöglichkeiten und als Halterung für die Triangulationssensoren konstruiert, die im Innenraum des oberen Mülltonnenrands befestigt wurden. Auf dem Deckel des Mülltonnenaufsatzes ist desweiteren ein Notausschalter vorhanden. Weil der Mülltonnenaufsatz keine Funktion bezüglich der sicheren Navigation hat, wird die Konstruktion an dieser Stelle nicht weiter erläutert, sondern nur die Montagepositionen der Triangulationssensoren.



Abb. 6.12: Montagepositionen der GP2Y0D02

Die Einbaupositionen der digitalen Triangulationssensoren GP2Y0D02 sind in dem CAD-Modell 6.12 mit roten Pfeilen markiert. Ein CAD-Modell der Mülltonne ist sehr komplex und würde die Leistung des CAD-Programms senken, aus diesem Grund sind die Halterungen der Sensoren provisorisch mit Aluminiumprofilen modelliert worden. Die Sensorachsen sind mit der Länge des Erfassungsbereichs der GP2Y0D02 durch schwarze Linien dargestellt. Um einen möglichst großen Messbereich abzudecken, sind die Sensorachsen der GP2Y0D02 in einem Winkel von $\gamma_{mess} = 30$ zur yz-Ebene bzw. zur xz-Ebene montiert worden, so dass sie diagonal zu den Außenflächen der Mülltonne messen. Somit werden die Außenflächen in ihrer gesamten Breite erfasst.



Abb. 6.13: Haltewinkel für die Montage der GP2Y0D02

Die Triangulationssensoren wurden an Haltewinkeln montiert. Als Beispiel ist ein Haltewinkel mit *GP2Y0D02* in Abbildung 6.13 dargestellt. Die Haltewinkel an der linken und rechten Seite der Mülltonne, sowie der Haltewinkel an der Rückseite sind anstelle eines rechten Winkels um einen Biegewinkel γ_b von 85° gebogen, der Wert von γ_b des Haltewinkels an der Fronseite beträgt 81°. Das hat den Vorteil, dass der Messbereich der *GP2Y0D02* in Bodennähe bei einer bestimmten Distanz d_{mess} zum Bodenrahmen endet (s. Abb. 6.14).



Abb. 6.14: schematische Darstellung von d_{mess}

Bei den gewählten Werten von γ_b kann der *ProfiBot*-Roboter präventiv auf Kollisionen reagieren. Der Wert von γ_b sollte 80° nicht unterschreiten. Der Messbereich endet ansonsten in einer zu hohen Distanz über dem Boden. Der Biegewinkel wurde sukzessiv anhand des CAD-Modells in Abbildung 6.12 bestimmt, indem Biegewinkel im Bereich $\gamma_b = 90...$ 80 untersucht wurden, unter der Vorgabe von $d_{mess} = 80mm$ an der linken und rechten Seite sowie an der Rückseite und $d_{mess} = 25mm$ an der Frontseite. Der Messbereich der *GP2Y0D02*, die an den rechten und linken Außenseiten befestigt wurden, endet an der Rückseite des *ProfiBot*-Roboters. Der Grund hierfür ist, dass bei einer Drehung um das kinematische Zentrum des *ProfiBot*-Roboters die Rückseite eine höhere rotatorische Geschwindigkeit hat, als die Frontseite. Der Anhalteweg der Rückseite ist somit größer. Durch den optimalen Wert von d_{mess} wird eine bevorstehende Kollision allerdings rechtzeitig erkannt.

Die analogen Triangulationssensoren GP2Y0A02 unterstützen die Navigation, indem sie Objekte erkennen, die der URG-04LX nicht erfassen kann. Die Position dieser Sensoren ist an der Fronsteite, die Sensorachsen sind jeweils um 10° zur xz-Ebene geneigt. Bei den Haltewinkeln ist $\gamma_b = 45$. Die Montageposition ist im CAD-Modell 6.15 dargestellt.

Der Infrarotlichtstrahl der *GP2Y0A02* trifft in einer Distanz $\Delta x = 750mm$ zur Roboterfront auf dem Boden auf. Die gemessenene Entfernung im Fahrbetrieb ohne Hinderniserkennung ist demnach ca. 1100mm, die Reichweite der Sensoren von 1500mm wird nicht komplett ausgenutzt. Aus diesem Grund kann ein Umkippen des *ProfiBot*-Roboters nach hinten registriert und ein Stoppen des Steuerprogramms veranlasst werden. Vertiefungen im Boden, wie beispielsweise Gruben oder abwärts führende Treppenstufen, die allerdings nicht in den definierten Umgebungen enthalten sind, können in Fahrtrichtung erkannt werden.



Abb. 6.15: Montagepositionen der GP2Y0A02

6.2 Logische Ebene

In diesem Kapitel ist die Realisierung der Verhaltensweisen beschrieben, die der *Profi-Bot*-Roboter für eine autonome Fahrweise benötigt. Weiterhin werden der Aufbau und die Funktionen der Software erläutert, die für den Betrieb und die Verhaltensweisen des *ProfiBot*-Roboters erstellt wurden. Die Struktur des Steuerprogramms ist in einer Subsumptionsarchitektur realisiert worden, welche die Verhalten *zufälliges Fahren, Hindernis-vermeidung* und *Notstopp* beinhaltet. Für das Erstellen der Programme wurde die Programmierumgebung *IConnect* benutzt.

6.2.1 Eingesetzte Verhaltensweisen und Drehzahlberechnung

In diesem Kapitel wird die Realisierung der Verhaltensweisen beschrieben, die Grundlage der sicheren Navigation sind. Weiterhin wird die Berechnung der Solldrehzahlen für die DC-Motoren beschrieben, um die eingesetzten Verhaltensweisen ausführen zu können.

Die Solldrehzahlen sind Parameter für den einschleifigen Drehzahlregelkreis. Um einen bedienerfreundlichen Einsatz zu gewährleisten sind die berechneten Solldrehzahlen alle abhängig von einer festzulegenden Solldrehzahl n_{max} pro DC-Motor, die von dem Bediener oder der Bedienerin an die Umgebung angepasst werden kann. Je mehr Freiraum in einer Umgebung vorhanden ist, desto größer kann n_{max} sein. Im Rahmen dieser Arbeit unter der Vorgabe, dass dynamische Hindernisse eine Geschwindigkeit von bis zu 7km/h haben können, ist die maximale Fahrgeschwindigkeit des ProfiBot-Roboters in Fahrversuchen

empirisch ermittelt worden. Der Wert von n_{max} ist auf $800min^{-1}$ voreingestellt und kann erhöht werden, wenn dynamische Hindernisse eine geringere Geschwindigkeit haben als 7km/h.

Zielloses Umherfahren:

Das Zielloses Umherfahren wird in Fahrsituationen aktiv, in denen der ProfiBot-Roboter keine Hindernisvermeidung durchführen muss. Diese Situationen sind vorhanden, wenn sich kein Objekt innerhalb des Erfassungsradius von 3000mm befindet. Durch das Ziellose Umherfahren fährt der Roboter nicht gerade durch einen Raum, sondern nach Trajektorien, die den in Kapitel 4.3.3 beschriebenen Rahmenbedingungen unterliegen. Die relativ einfache Generierung der Trajektorien ergibt sich aus der Zufallszahl f, und der Randbedingung 0 > f > 1. Die Solldrehzahlen der Motoren werden alle 5s nach den Gleichungen 6.2 bestimmt.

$$n_{soll,r} = n_{max} \cdot \left(1 - \frac{f}{2}\right)$$

$$n_{soll,l} = n_{max} \cdot \left(1 + \frac{f}{2}\right)$$
(6.2)

Hindernisvermeidung:

Für die Hindernisvermeidung ist der in Kapitel 4.3.4 beschriebene Algorithmus zur Generierung des Lenkwinkels α_l , bei dem sich der *ProfiBot*-Roboter befahrbaren Freiraum sucht, eingesetzt worden, mit $d_T = 1500mm$ und m = 0,002. Weiterhin wurde definiert, dass $\alpha_l = \omega_l$ ist. Dieser Algorithmus reicht für eine sichere Fahrweise nicht aus. Bei geringen Distanzen zu Hindernissen sollte die Fahrgeschwindigkeit des Roboters gering sein, damit bei einer Kollision die kinetische Energie des Aufpralls gering ist. Aus diesem Grund sind weitere Algorithmen integriert worden, welche die aktuelle translatorische Geschwindigkeit v im kinematischen Zentrum an die Entfernung des nächsten Messpunktes d_{min} zur Roboterfront anpassen. Eine weitere Verhaltensweise *Drehen* veranlasst den *ProfiBot*-Roboter, vor Hindernissen zu drehen, wenn er nicht vorbeifahren kann.

Welche der Verhaltensweisen Drehen und Freiraum suchen ausgeprägter ist, wird durch den Gewichtungsfaktor β bestimmt. Der Betrag der translatorischen Robotergeschindigkeit v ergibt sich nach dem Gewichtungsfaktor ϵ , die rotatorische Geschwindigkeit ω der Verhaltensweise Drehen ergibt sich nach dem Gewichtungsfaktor λ . Die Gewichtungsfaktoren werden in den nachfolgenden Abchnitten erläutert.

 d_{min} wird nach einer Fallunterscheidung entweder vom 2D-Laserscanner URG-04LXbestimmt $(d_{min,URG})$, oder durch den rechten und linken Triangulationssensor GP2Y0A02 $(d_{min,GP2Y_R}$ bzw. $d_{min,GP2Y_L})$, bei denen die gemessene Entfernung aufgrund der Einbauposition in $d_{min,GP2Y_{R,L}} = d_{min} \cdot cos(45) \cdot cos(10)$ umgerechnet wird und $d_{min,GP2Y_{R,L}}$ maximal 750mm betragen kann. Auf eine Umrechnung der Distanzwerte d_s in das Sensorzentrum wurde verzichtet, weil bei den entsprechenden Verhaltensweisen nur das Vorzeichen der x-Komponete von Bedeutung ist.



Abb. 6.16: Virtuelle Fahrspur und nächster Messpunkt

Der nächste Messpunkt d_{min} liegt innerhalb einer virtuellen Spur (s. Abb. 6.16) mit der Breite b_v , die sich aus der Roboterbreite b_{rob} und einer Sicherheitszugabe b_s nach der Gleichung $b_v = b_{rob} + 2 \cdot b_s$ zusammensetzt. Aufgrund der in Kapitel 6.1.4 beschriebenen Einbaupositionen der *GP2Y0D02* muss $b_s > 80mm$ sein, andererseits würde durch Hindernisse, die außerhalb der Fahrspur liegen und deren gemessene Distanz nicht in die Berechnung der Drehzahlen mit einfliesst, ein Notstopp verursacht werden. Für die nachfolgenden Berechnungen der Solldrehzahlen von den Motoren wird anstelle der Sollgeschwindigkeit v_{soll} der Einfachheit halber eine Solldrehzahl n_{soll} eingeführt, welche die Drehzahl eines gedachten Rades im kinematischen Schwerpunkt des Roboters beschreibt. Die Berechnung der Solldrehzahl n_{soll} erfolgt nach den Gleichungen 6.3 und 6.4. ϵ ist ein Gewichtungsfaktor. Bei einem kleinen Wert von ϵ ist auch die Geschwindigkeit gering. ϵ und somit n_{soll} werden größer (nach der tanh-Funktion in 6.4), je größer die gemessene Distanz zu einem Objekt auf der Fahrspur ist.

$$n_{soll} = n_{max} \cdot \epsilon \tag{6.3}$$

$$\epsilon = \frac{1}{2} \cdot tanh(0,002 \frac{1}{mm} \cdot (d_{min} - 800mm) + 1)$$
(6.4)

Die Geschwindigkeitsanpassung nach einer linearen Funktion, wie beispielsweise nach der Gleichung $n_{soll} = m_s \cdot d_{min}$, benötigt weniger Rechenaufwand, führt jedoch zu einer ungünstigeren Fahrweise des *ProfiBot*-Roboters, wie in Fahrversuchen festgestellt wurde.

Damit der *ProfiBot*-Roboter aus Sackgassen heraus fahren kann, ist ein Verhalten integriert worden, bei dem sich der *ProfiBot*-Roboter auf der Stelle dreht, d.h. ausschließlich seine Orientierung ändert, wenn d_{min} unter einem Distanzschwellwert $d_{min}^* = 700mm$ liegt. Das Drehen wird durch jeweils unterschiedliche Drehrichtungen mit gleichen Geschwindigkeiten beider Reifen vom *ProfiBot*-Roboter realisiert. Das Verhalten *Drehen* wird durch eine Lenkwinkelvorgabe von α_{max} erreicht, aus der sich direkt die Geschwindigkeit ω_{max} berechnet. Das Vorzeichen von α_{max} und somit die Drehrichtung vom *ProfiBot*-Roboter ergibt sich aus der Fallunterscheidung 6.5 mit dem vom *URG-04LX* erfassten minimalen Abstand $d_{min,URG}$ und den von den analogen Triangulationssensoren erfassten Abständen $d_{min,GP2Y_{(R,L)}}$. Aus dem Koeffizienten β (s. Fallunterscheidung 6.8) errechnet sich der Betrag von λ nach Gleichung 6.6.

$$\alpha_{max} \begin{cases} -\lambda, & \text{falls } \angle d_{min,URG} > 0 \text{ oder } d_{min,GP2Y_{(L)}} < \frac{2}{3} d_{min,GP2Y_{(R)}} \\ = 0, & \text{falls } d_{min} > d_{min}^{*} \\ +\lambda, & \text{falls } \angle d_{min,URG} < 0 \text{ oder } d_{min,GP2Y_{(R)}} < \frac{2}{3} d_{min,GP2Y_{(L)}} \\ \lambda = (1 - \beta) \end{cases}$$

$$(6.5)$$

Es können Situationen vorhanden sein, in denen der Winkel zwischen d_{min} und der Ordinate des roboterzentrischen Koordinatensystems das Vorzeichen wechselt, während die Verhaltensweise *Drehen* ausgeführt wird. In der Folge würde der Roboter die Drehrichtung so lange wechseln, bis er von dem Bediener oder der Bedienerin aus dieser Situation befreit wird. Zwei Beispiele solcher Situationen sind in den Abbildungen 6.17 und 6.18 dargestellt.



Abb. 6.17: Drehen vor einer Wand



Um diese unerwünsche Fahrweise zu umgehen sind zwei Verfahren untersucht worden.

1. Zusätzlich zu der in Fallunterscheidung 6.5 beschriebenen Abfrage von $\angle d_{min,URG}$ wird der Betrag von $d_{min,URG}$ in der Variablen d'_{min} gespeichert, wenn $d_{min,URG} < d^*_{min}$ ist. Der Roboter wechselt nur dann wieder die Drehrichtung, wenn der aktuelle Wert von $|d_{min,URG}| < d'_{min} - 100mm$ ist. Ist das der Fall, so wird der aktuelle Wert wiederum in d'_{min} gespeichert. Sobald $d_{min,URG} > d^*_{min}$ wird d'_{min} gelöscht. Messfahrten ergaben, dass der Roboter mit diesem Verfahren aus der Situation in Abbildung 6.18 nicht herausfinden kann, wenn die Entfernung zu Hindernis 3 größer als d_{min}^* ist.

2. Im zweiten Verfahren wird ein Zähler eingesetzt, der die Drehrichtungswechsel zählt. Wenn der *ProfiBot*-Roboter zweimal die Drehrichtung gewechselt hat, wird ein Zustand aktiviert, bei dem der *ProfiBot*-Roboter während des Drehens den dritten Drehrichtungswechsel solange ignoriert, bis sich das Vorzeichen von $\angle d_{min,URG}$ umkehrt und der *ProfiBot*-Roboter sich ohnehin in diese Richtung drehen würde. Der Zähler wird nach Verlassen des Zustands wieder zurückgesetzt. Damit sich der benannte Zustand nicht in anderen Situationen aktiviert, wird der Zähler nach dem Ablauf einer bestimmten Zeit ebenfalls zurückgesetzt. Diese Zeit ist größer als die Zeit, die der ProfiBot-Roboter benötigt, um aus den benannten Situationen herauszufinden und beträgt 5s. Das Verfahren kann in beiden dargestellten Situationen angewendet werden und wird daher im *ProfiBot*-Roboter benutzt.

 α_l und α_{max} sind, abhängig von d_{min} , mit dem Koeffizient β nach der Fallunterscheidung 6.8 gewichtet. Die Fallunterscheidung ist graphisch in Abbildung 6.19 dargestellt. Durch die Gewichtung gehen α_l und α_{max} fließend ineinander über. Im Gegensatz zu einem unstetigen Übergang ist hierbei die Fahrweise sanfter.

$$\alpha_{soll} = \alpha_l \cdot \beta + \alpha_{max} \cdot (1 - \beta) \tag{6.7}$$

$$\beta = \begin{cases} 0, & \text{falls } d_{min} \le d_{min}^* \le 500mm \\ \frac{1}{2} \cdot tanh(0,004 \frac{1}{mm} \cdot (d_{min} - 800mm) + 1), & \text{falls } d_{min}^* < d_{min} < 3000mm \\ 1, & \text{falls } d_{min} \ge 3000mm \end{cases}$$
(6.8)



Abb. 6.19: Gewichtung von α_l und α_{max} nach Fallunterscheidung von β

Der ProfiBot-Roboter fährt nur dann in Passagen hinein, wenn diese mindestens so breit

sind wie der Radius r_w seines Wendekreises (s. Abb. 6.16). r_w wird nach Gleichung 6.9 berechnet. Im Rahmen dieser Arbeit wurde $b_s = r_w - b_{rob}$ gewählt. Eine Nichtbeachtung dieser Tatsache hätte zur Folge, dass der *ProfiBot*-Roboter beim Drehen gegen Objekte stoßen könnte oder die Drehung nicht komplett ausführen würde, was einen Betriebsstopp zur Folge hätte.

$$r_w \approx \sqrt{\left(\frac{l_{rob}}{2} + \overline{MK}\right)^2 + \left(\frac{b_{rob}}{2}\right)^2} \tag{6.9}$$

Mit diesen Algorithmen ist die Verhaltensweisen Hindernisvermeidung abgedeckt.

Die Sollgeschwindigkeiten des linken und rechten Motors werden durch die Gleichungen 6.10 und 6.11 generiert.

$$n_{soll,l} = n_{max} \cdot (\epsilon + \alpha_{soll}) \tag{6.10}$$

$$n_{soll,r} = n_{max} \cdot (\epsilon - \alpha_{soll}) \tag{6.11}$$

Notstopp:

Für den Notstopp werden die vier Schaltsignale D_F , D_B , D_L und D_R^4 der digitalen Triangulationssensoren *GP2Y0D02* und die Schalsignale S_F , S_L und S_R der Schaltleisten in einer logischen Oder-Schaltung ausgewertet. Das erzeugte Signal Schaltsignal *B* ist 1, wenn eines der benannten Schaltsignale 1 ist. Bei B = 1 wird der *ProfiBot*-Roboter in den Zustand *Stop* überführt, in welchem n_r und n_l auf $0min^{-1}$ gesetzt werden. Wenn B = 0 wird, führt das Steuerprogramm nach einer Wartezeit t = 2s das Verhalten *Hindernisvermeidung* oder *Zielloses Umherfahren* wieder aus. Das Steuerprogramm wird ausgeschaltet, sobald die Frontschaltleiste länger als *500ms* betätigt wurde. In diesem Fall kann der ProfiBot-Roboter in Störsituationen nicht weiterfahren.

 n_r und n_l wird weiterhin auf $0min^{-1}$ gesetzt, wenn $d_{min,URG} < 200mm$ ist und $sin(\angle d_{min,URG}) \cdot d_{min,URG} > \frac{b_{rob}}{2}$. Mit diesem Verfahren wird zusätzlich ein Schutzfeld definiert. Der ProfiBot-Roboter stoppt, wenn sich ein Hindernis innerhalb dieses Schutzfelds befindet.

In Abbildung 6.20 sind zusammenfassend die jeweils stärker ausgeprägten Verhaltensweisen unabhängig von den Gewichtungsfaktoren qualitativ dargestellt, wenn Hindernisse in den entsprechenden Feldern erkannt werden.

 $^{{}^{4}}D_{F}$: Schaltsignal vom *GP2Y0D02* an der Frontseite, D_{B} : Rückseite, D_{L} : linke Seite, D_{R} : rechte Seite



Abb. 6.20: Die stärker ausgeprägten Verhaltensweisen des Steuerprogramms

6.2.2 IConnect Signalgraphen

Die erstellten IConnect Programme sind die Signalgraphen Scanner und AutonomeFahrt.

Signalgraph Scanner:

Scanner beinhaltet das in Kapitel 6.1.2 beschriebene *IConnect*-Modul *URG04LX* und das *IConnect*-Modul *O_Avoidance*. In Abbildung 6.21 ist das Blockschaltbild des Signalgraphen Scanner dargestellt.

Die Zykluszeit des Signalgraphen Scanner wurde an die Zykluszeit des URG04LX angepasst und beträgt somit 100ms. Das Scan-Array des Ausgangs Data des Modul URG04LXwird an das Modul $O_Avoidance$ weitergegeben. Dieses Modul wurde mit dem Template-Modul erstellt und in der Programmiersprache C++ geschrieben. Das Modul $O_Avoidance$ hat die nachfolgend aufgelisteten Funktionen:

- Transformieren der Distanzwerte vom Sensorzentrum in das kinematische Zentrum.
- Generieren eines Lenkwinkels nach den in Kapitel 4.3.4 beschriebenen Algorithmen.
- Generieren einer virtuellen Fahrspur.
- Herausfiltern des Messwerts, der die geringste Distanz zum Roboter innerhalb dieser Fahrspur hat.


Abb. 6.21: Blockschaltbild des Signalgraphen Scanner

In der Tabelle 6.2 sind die Ausgänge des Moduls *O_Avoidance* aufgelistet, deren Werte über das Modul *MultiComm* an den Signalgraphen *AutonomeFahrt* übergeben werden.

Ausgang	FUNKTION
X	Array mit den X-Koordinaten eines Scans
Y	Array mit den Y-Koordinaten eines Scans
$steering_angle$	Lenkwinkel α_l
cp	X-Koordinate von d_{min}
$cp_{-}angle$	Winkel von d_{min}

Tabelle 6.2: Ausgänge des Moduls O_Avoidance

In dem Interpret-Modul *Robot-Info* können alle für die Hindernisvermeidung relevanten Parameter eines beliebigen Roboters eingestellt werden, wie beispielsweise b_{rob} , b_s , \overline{KS} . Als Ausblick wird an dieser Stelle vorgeschlagen, eine Bedienoberfläche für alle Signalgraphen zu gestalten, in der alle Parameter in Stringeingabefeldern verändert werden können.

Signalgraph AutonomeFahrt:

Der Signalgraphen AutonomeFahrt hat folgende Funktionen:

- Einlesen der Sensordaten des Buskopplers
- Verknüpfung der Verhaltensweisen in einer Subsumptionsarchitektur

Es ist eine Zykluszeit von 20*ms* gewählt worden, weil diese Zykluszeit der Standart-Zykluszeit der Signalgraphen für das *ProfiBot*-Basissystem entspricht. Die Subsumptionsarchitektur wird mit Hilfe von drei *IConnect*-Macros für die jeweiligen Verhaltensweisen realisiert.

- Z_Umherfahren
- *H_vermeidung*
- NotStopp

Das Blockschaltbild des Signalgraphen von *AutonomesFahren* ist in Abbildung 6.22 dargestellt.



Abb. 6.22: Blockschaltbild von AutonomesFahren

Das Modul PolyMUX ist ein Zweikanal-Multiplexer, der entweder die Drehzahlen von $H_Vermeidung$ oder die von $Z_Umherfahren$ auf die Ausgänge durchschaltet, abhängig von einem Schaltsignal, das in $Z_Umherfahren$ generiert wird. Das Macro NotStopp erzeugt ebenfalls ein Schaltsignal, welches am Relais-Modul Notstopp anliegt und entweder der Drehzahlvorgabe $0min^{-1}$ bei einer logischen 0 oder die generierten Drehzahlen aus $Z_Umherfahren$ oder $H_Vermeidung$ bei einer logischen 1 zu den Ausgängen durchlässt. Die Drehzahlvorgaben werden an das Modul TMC weitergeleitet. Um die Übersichtlichkeit zu erhöhen sind einige Daten des Moduls ILBETH und ScannerData mit Hilfe des Moduls Scanner_BUS in jeweils einer Übertragungsleitung zusammengefasst.

Das Macro $H_vermeidung$ (s. Abb. 6.23) erzeugt die Drehzahlen der Motoren für das Verhalten *Hindernisvermeidung* nach den in 6.2.1 beschriebenen Gleichungen und Gewichtungsfaktoren.

In dem Macro NotStopp werden die sieben Schaltsignale der digitalen Sensoren ausgewertet. Bei B = 1 befindet sich der ProfiBot-Roboter in einem Störzustand.



Abb. 6.23: Blockschaltbild von $H_-Vermeidung$

Für die digitalen Triangulationssensoren ist ein eigenes Interpret-Modul je Sensor vorhanden, dass eine logische 1 erst ausgibt, wenn eine logische 1 fünf Zyklen lang am Eingang anliegt. Somit werden die in Kapitel 6.1.1 beschriebenen Spannungseinbrüche herausgefiltert. Die Schaltsignale werden in einer Oder-Schaltung mit Hilfe eines Gate-Moduls ausgewertet. Zusätzlich wird mit Hilfe des IConnect-Modul Stop der Signalgraph ausgeschaltet, wenn die Frontschaltleiste länger als 500ms gedrückt ist. Der Signalgraph des Macros NotStopp ist in Abbildung 6.24 dargestellt.



Abb. 6.24: Blockschaltbild von NotStopp

Das Modul *BinaryCoder* wandelt die binären Schaltsignale in einen Dezimalcode um. In dem Modul *Case* kann für jeden beliebigen Code definiert werden, ob das Modul *Stopp* ausgeführt werden soll. Im Rahmen dieser Arbeit ist das der Fall, wenn alle drei Schaltleistensignale eine logische 1 aufweisen.

Zur graphischen Darstellung der erfassten Kontur eines Scans und des Lenkwinkels ist das Macro *Display* erstellt worden (s. Abb. 6.25). Dieses Modul ist eine Anzeigeoberfläche, in der alle Messpunkte eines Scans in einem kartesischen Koordinatensystem dargestellt sind. Der Lenkwinkel ist mit Hilfe eines Polarzeigers im Ursprung dargestellt. Die Schalt-



signale aller sieben digitalen Sensoren sind mit Hilfe von LEDs dargestellt, die bei einer logischen 1 leuchten.

Abb. 6.25: Anzeigeoberfläche von Display

Durch die benötigte Rechenleistung für die graphische Darstellung der Konturen wird die Zykluszeit des *IConnect*-Signalgraphen drastisch erhöht, was an einer stockenden Fahrweise des *ProfiBot*-Roboters zu beobachten ist. Hier ist die Zykluszeit von *Display* in einigen Zyklen höher als die Timeout-Zeit des TMC200, die 200ms beträgt (s. Abb. 6.26). Die Zykluszeiten der verwendeten Module sind bei laufendem Signalgraphen in einer Logdatei gespeichert worden. Im Diagramm 6.25 ist die gemittelte Zykluszeit mit anderen Modulen verglichen.



Abb. 6.26: Zykluszeiten (gemittelte Zykluszeit über 10s)

Aus diesem Grund ist das Macro *Display* nicht verwendet worden. Für Abhilfe könnte u.a. ein leistungsfähigeres Notebook im *ProfiBot*-Roboter sorgen.

Kapitel 7

Messungen und Ergebnisse

Neben den bereits in Kapitel 6 beschriebenen Messungen, wie beispielsweise die Aufnahme der Sensorkennlinien, sind zusätzlich Messungen durchgeführt worden, die in diesem Kapitel beschrieben werden. Anhand einiger Messungen wurden wichtige Funktionen überprüft. Außerdem werden Messfahrten beschrieben, in denen unerwünschtes Fahrverhalten untersucht wurde.

7.1 Funktionsprüfung der Konturenmessung

Die Funktion des 2D-Laserscanner URG-04LX wurde mittels neun Versuchsanordnungen überprüft. In Versuchsanordnung 1-3 wurde überprüft, wie die Messwertaufnahme beeinflusst wird, wenn der Laserstrahl auf transparente Objekte oder Objekte, die eine spiegelnde Oberfläche haben, trifft. Für diese Objekte wurden Alltagsgegenstände zur Aufnahme in das Areal hineingestellt. Eine exakte Positionierung war bedeutungslos, weil nur die Reaktion auf die Materialien der Objekte untersucht werden sollte.

In Versuchsanordnung 4-8 wurde ein Röhrchen mit einem Durchmesser von 20mm in verschiedenen Distanzen zum ProfiBot-Roboter platziert. Durch diese Distanzmessungen sollte überprüft werden, bis zu welcher Entfernung schmale Objekte erfasst werden können.

In Versuchsanordnung 9 ist der Roboter auf einem Korridor postitioniert worden. Hierbei sollte die Reaktion auf Distanzen oberhalb der Reichweite des URG-04LX überprüft werden.

Die Distanzwerte sind in kartesische Koordinaten umgerechnet worden. Die x- und y-Koordinaten der aufgenommenen Punkte wurden mit Hilfe des *IConnect*-Moduls *Save-Table* in einer Datei gespeichert. Durch die Aufnahme mehrerer Scans pro Versuchsanordnung konnte die Reproduzierbarkeit einer Messung überprüft werden. Nach zehn Scans wurde die Messwertaufnahme einer Versuchsanordnung abgebrochen.

Versuchsanordung 1:

Der *ProfiBot*-Roboter wurde in einem annähernd rechteckigen Versuchsareal positioniert, das mit planen Gegenständen abgegrenzt war. Bei dieser Versuchsanordnung befanden sich in dem Areal keine Gegenstände. Die Versuchsanordnung ist in Abbildung 7.1 dargestellt. Die Kontur des Versuchsareals aus den aufgenommen Koordinaten ist in Abbildung 7.2 dargestellt.





Abb. 7.1: Versuchsanordnung 1



In Abbildung 7.2 ist zu erkennen, dass das komplette Areal aufgenommen wurde. Die Konturen des Areals sind leicht verzerrt, was durch Rundungsfehler in der Koordinatentransformationen oder durch eine fehlerhafte Optik des URG-04LX hervorgerufen werden könnte.

Versuchsanordung 2:







Abb. 7.4: Konturendiagramm zu Versuchsanordnung 2



sche, eine transparente Kunststofffolie und eine Lochrasterplatte in einem Abstand von ca. 1000mm zum ProfiBot-Roboter positioniert worden, wie in Abbildung 7.3 dargestellt. Es wurde hierbei untersucht, wie transparente Objekte die Messung beeinflussen.

Aus dem Konturdiagramm, das in Abbildung 7.4 dargestellt ist, ist ersichtlich, dass die Weißglasflasche und die Lochrasterplatte vollständig erfasst wurden. Bei der Grünglasflasche und bei der Folie sind Fehler entstanden. Die Position der Grünglasflasche ist auf dem Konturendiagramm nicht zu erkennen. Die umrandeten Messpunkte sind im Areal nicht vorhanden, aus den y-Koordinaten dieser Messpunkte ist ersichtlich, dass an dieser Stelle die Kunststofffolie positioniert wurde. Die Position der Messpunkte wird auf der Ordinate mit ca. 4250mm angegeben. Eine mögliche Fehlerursache könnte eine durch die Kunststofffolie hervorgerufene Phasenverschiebung des Laserstrahls sein.

Versuchsanordung 3:

Bei dieser Versuchsanordnung sollte die Aufnahme von transparenten und spiegelnden Objekten überprüft werden. Als Versuchsobjekt ist die Rückseite einer handelsüblichen CD benutzt worden, die in der entsprechenden CD-Hülle vor dem *ProfiBot*-Roboter aufgestellt wurde, wie in Abbildung 7.5 dargestellt. Rechts daneben ist der Deckel einer 50-er CD-Spindel aus transparentem Kunststoff positioniert worden.



Abb. 7.5: Versuchsanordnung 3



Abb. 7.6: Konturendiagramm zu Versuchsanordnung 3

Sowohl die Aufnahme der CD mit der CD-Hülle, als auch die Aufnahme des Spindeldeckels weisen Fehler auf. Im Konturdiagramm ist zu erkennen, dass durch die Erfassung der CD-Hülle Messpunkte vorhanden sind, die nicht innerhalb des Versuchsareals liegen. Weiterhin ist die aufgenommene Kontur der CD fehlerhaft. Die eigentliche Anzahl an Messpunkten müsste bei dieser Aufnahme deutlich höher sein. Der Spindeldeckel wird im Konturendiagramm an der oberen rechten Ecke des Areals abgebildet. CDs haben eine spiegelnde Oberfläche, auf der eine Totalreflexion von dem Großteil des Laserlichts stattfindet. Ist der Einfallswinkel des Laserlichts auf der CD-Oberfläche zu groß, kann der Empfänger nicht mehr genügend Laserlicht für eine korrekte Messwerterfassung aufnehmen. Die Erzeugung der fehlerhaften Messpunkte innerhalb des Areals könnte deswegen durch einen zu geringen Anteil des empfangenen Laserlichts stattgefunden haben. Abhilfe ist evtl. ein anderer Laserscannertyp.

Versuchsanordungen 4-8:

In diesen Versuchsreihen ist ein Röhrchen mit einem Durchmesser von 20mm in Distanzen (x-Richtung) von 200mm (Versuchsreihe 4), 400mm (Versuchsreihe 5), 600mm (Versuchsreihe 6), 800mm (Versuchsreihe 7) und 1000mm (Versuchsreihe 8) vor dem *ProfiBot*-Roboter platziert worden (s. Abbildung 7.7). Bis zu einer Entfernung von 600mm wird das Röhrchen erfasst, ab 800mm Entfernung ist die Distanz benachbarter Messpunkte der Einzelmessungen für eine Erfassung des Röhrchen zu groß. Die Konturendiagramme der Versuchsreihen 6 und 7 sind in den Abbildungen 7.8 und 7.9 dargestellt. Die Konturendiagramme der Versuchsreihen 4, 5 und 8 sind im Anhang zu finden.



Abb. 7.7: Versuchsanordnungen 4 bis 8





Abb. 7.8: Konturendiagramm des Röhrchens in 600mm Entfernung

Abb. 7.9: Konturendiagramm des Röhrchens in 800mm Entfernung

In den Versuchsanordungen 4, 5 und 6 wurde das Röhrchen erfasst, in der Versuchsanordnung 7 und 8 ist es im Konturendiagramm (s. Abbildung 7.9) nicht zu erkennen. Mit diesen Versuchsreihen ist dargestellt, dass die Erfassung von schmalen Gegenständen durch die horizontale Distanzauflösung des URG-04LX begrenzt ist (vgl. Abbildung 5.9 in Kapitel 5.3.4). Gegenstände, die schmaler als 20mm sind, werden ab einer Schwelldistanz, die ungefähr bei 600mm liegt, nicht mehr erfasst.

Versuchsanordung 9:

In diesem Versuch wurde die Kontur vom Korridor C2-1 des Fraunhofer Instituts für Intelligente Analyse- und Informationssysteme aufgenommen. Die Versuchsanordnung ist in Abbildung 7.10 dargestellt.



4000 (U) 2000 -2000 y-Distanz[mm]

Abb. 7.10: Versuchsanordnung 9

Abb. 7.11: Konturendiagramm zu Versuchsanordnung 9

Das Konturendiagramm in Abbildung 7.11 zeigt, dass alle Messpunkte erfasst wurden, ausschließlich der Punkte, die ausserhalb der Reichweite des URG-04LX lagen.

Fazit:

Die Versuchsanordungen 1-3 haben gezeigt, dass der URG-04LX Gegenstände, die aus einem transparenten oder spiegelnden Material bestehen, nicht zuverlässig erfassen kann. Der Laserstahl wird von transparenten Materialien verschluckt oder nur zum Teil mit einer zusätzlichen Phasenverschiebung reflektiert. Bei spiegelnden Materialien ist durch das geringe Streulicht die Intensität des reflektierten Laserlichts so gering, dass diese Messungen ebenfalls fehlerhaft sind. In der Folge dürfen beim Fahrbetrieb des ProfiBot-Roboters keine derartigen Materialien im Erfassungsbereich des URG-04LX liegen, weil ansonsten die Sicherheit beeinträchtigt werden könnte. Der *ProfiBot*-Roboter könnte anhand der fehlerhaften Distanzinformationen einer unvorhersehbaren Trajektorie folgen.

Aus den Versuchsanordnungen 4-8 geht hervor, dass sich im Einsatzort des *ProfiBot*-Roboters keine Gegenstände befinden sollten, deren Durchmesser oder Breite kleiner als

20mmist. Diese Gegenstände könnten für ein Ausweichmanöver nicht rechtzeitig erkannt werden.

Sobald die Distanz zu Objekten mehr als 4000mm ist, nimmt der URG-04LX in dieser Richtung keine Messwerte auf und gibt den Distanzwert 0mm aus.

7.2 Lenkwinkelgenerierung

Um die Funktion der Solllenkwinkelberechnung α_{soll} grafisch zu überprüfen (s. Kapitel 6.2.1) wurden Messungen in zwei Versuchsanordnungen durchgeführt, bei denen der modifizierte *ProfiBot*-Roboter in ein Versuchsareal gestellt wurde, das ähnlich aufgebaut ist, wie das in Kapitel 7.1 beschriebene (s. Abbildung 7.12).



Abb. 7.12: Versuchsareal mit Hindernis

In das Versuchsareal ist ein weiterer *ProfiBot*-Roboter hineingestellt worden, der die Funktion eines Hindernisses übernimmt. Der modifizierte *ProfiBot*-Roboter ist in der ersten Versuchsanordnung in einer Distanz von 1100mm zum Hindernis positioniert worden. Der erwartete Wert $\alpha_{soll} \approx 0$ ist in dem Konturendiagramm in Abbildung 7.13 erkennbar, in der α_{soll} als Winkel eines Polarzeigers mit dem Betrag 1000mm dargestellt wird. Das Hindernis ist in dieser Grafik rot umrandet. Bei einer Distanz zum Hindernis von 700mm würde die Vorgabe durch α_{soll} den modifizierten *ProfiBot*-Roboter, wie erwartet, am Hindernis vorbeiführen (s. Abb. 7.13).



Abb. 7.13: Lenkwinkel bei einer Hindernisdistanz von 1100*mm*

Abb. 7.14: Lenkwinkel bei einer Hindernisdistanz von 700mm

Durch Modifikation von Gleichung 4.5 in Kapitel 4.3.4 zu Gleichung 7.1 erhält der *ProfiBot* eine Fahrweise, bei der er, statt sich Freiraum zu suchen, auf nächstgelegene Objekte hinzufährt. Durch dieses Verhalten in Kombination mit dem Verhalten *Drehen* ist ein sehr günstiges Gesamtverhalten beobachtet worden, um aus Situationen herauszufinden, die in der Abbildung 6.18 in Kapitel 6.2.1 dargestellt sind.

$$\alpha_{l} = atan2 \left(\frac{\sum_{i=1}^{n} \sin\left(\varphi_{i}\right) \cdot f_{1}\left(\varphi_{i}\right) \cdot f_{2}\left(r_{i}\right)}{\sum_{i=1}^{n} \cos\left(\varphi_{i}\right) \cdot f_{1}\left(\varphi_{i}\right) \cdot f_{2}\left(r_{i}\right)} \right) - \frac{\pi}{2}$$
(7.1)

Die Ursache hierfür ist, dass das Verhalten Drehen für die Ausführung eben ein solches nahes Objekt benötigt. Durch das Verhalten Objekt suchen wird auch teilweise das Passieren von gefährlichem Freiraum vermieden. Eine unerwünschte Fahrweise kann beobachtet werden, wenn der ProfiBot-Roboter an einem Objekt vorbeifährt, dass sich links oder rechts von ihm befindet. In einer solchen Situation fährt der ProfiBot-Roboter zunächst zu dem Objekt hin und dreht sich im Anschluss wieder von ihm weg, bis das Objekt eine Entfernung und Richtung hat, in der die Verhaltensweise Objekt suchen wieder aktiv ist. In der Folge fährt der ProfiBot-Roboter Schlangenlinien, bis das Objekt ausserhalb des Erfassungsbereichs liegt. Eine Kombination der beiden Verhalten Objekt suchen und Freiraum suchen und eine Aktivierung des passenden Verhaltens in den ensprechenden Situationen, wäre eine gute Lösung für eine sichere Navigation, würde aber den Rahmen dieser Arbeit sprengen und ist somit als Ausblick dargelegt.

7.3 Überprüfung der Robustheit

Das eingesetzte Sicherheitskonzept des ProfiBot-Roboters muss robust sein. Damit ist gemeint, dass sich ändernde Parameter während des Fahrbetretriebs keinen Einfluss auf die Funktionen des Sicherheitssystems und die damit verbundenen Verhalten haben. Die wesentlichen Parameter, die sich während des Fahrbetriebs ändern bzw. ändern können sind in Tabelle 7.1 aufgelistet.

PARAMETER	Minimalbedingung	Maximalbedingung
Batteriespannung [V]	23,75	29
Lichtintensität	abgedunkelter Raum	tageslichtdurchfluteter Raum
Raumtemperatur [°C]	10	30

Tabelle 7.1: Variable Parameter und Extrembedingungen

Durchführung:

Zur Überprüfung der Fahrweise auf Parameteränderungen sind Messfahrten der *ProfiBot*-Roboter unter den entsprechenden Extrembedingungen durchgeführt worden.

Für die Ermittlung der Robustheit bei Änderung der Batteriespannung wurde der *Pro-fiBot*-Roboter zum mit voll aufgeladenen Batterien und mit fast leeren Batterien so lange betrieben, bis der Motorcontroller automatisch ausgeschaltet wurde (s. Kapitel 3.2.2).

Die max. zulässige Beladung der Mülltonne ist wurde im Rahmen einer Messfahrt ermittelt. Hierbei ist der Roboter sukzessiv mit Gewichten beladen worden, bis sich eine sicherheitsrelevante Änderung der Fahrweise herausstellte.

Ergebnisse:

PARAMETER	REAKTION UNTER MINIMALBED.	REAKTION UNTER MAXIMALBED.
Batteriespannung	keine	keine
Lichtintensität	keine	keine
Raumtemperatur	keine	keine

Die Ergebnisse der Messfahrten sind in Tabelle 7.2 beschrieben.

Tabelle 7.2: Variable Parameter und Extrembedingungen

Aus Tabelle 7.2 ist zu entnehmen, dass es keine Beeinflussung bei Parameteränderungen gibt. Eine Beladung von 10kg sollte allerdings nicht überschritten werden.

7.4 Fahrweise

Die Umsetzung der einzelnen Verhaltensweisen während des Fahrbetriebs wurden in den nachfolgend aufgezählten, geschlossenen Räumen mit verschiedenartigem Inventar durchgeführt.

- 1. Foyer im Gebäudetrakt C2 des Fraunhofer-Institutszentrums Birlinghoven
- 2. Korridor C2-1
- 3. Raum C2-120

Zusätzlich ist ein Versuchsareal aufgebaut worden, in dem die Fahrweise erprobt wurde, welches im nachfolgenden Abschnitt beschrieben wird. Die Subverhalten von *Hindernisvermeidung (Freiraum suchen und Drehen)*, sowie die Verhalten Zufälliges Umherfahren und NotStopp funktionierten im Foyer und im Korridor bei zehnminütigen Fahrversuchen einwandfrei, es sind keine Kollisionen beobachtet worden. Die Tische im Raum C2-120 werden erst zu spät erkannt, so dass der ProfiBot-Roboter mit ihnen kollidiert (s. Kap. 7.4.1).

Es gibt einige Schwierigkeiten beim Fahrbetrieb innerhalb geschlossener Räume, die Objekte beinhalten, welche unterhalb der Scanebene viel breiter sind, als oberhalb. Dieses Problem lässt sich ausschließlich dadurch beheben, in dem der Wert von d_{min}^* erhöht wird und β durch Modifikation der Gleichung 6.8 nach einer höheren Schwelldistanz gegen 0 strebt.

Beim Drehen des ProfiBot-Roboters sind Fahrsituationen vorhanden, bei denen ein Sprung von einem nächsten Messpunkt mit geringer Distanz $d_{min,k}$ auf einen hohen Wert von $d_{min,k+1}$ erfolgen kann. Durch die Berechnung der Sollgeschwindigkeit nach Gleichung 6.3 erfolgt in einer solchen Fahrsituation ein sprunghafter Anstieg von der aktuellen Solldrehzahl der verwendeten DC-Motoren $n_{soll,k}$ auf eine hohe Solldrehzahl $n_{soll,k+1}$. Durch die fehlende Drehmomentregelung wird der ProfiBot-Roboter maximal beschleunigt. Die Folge ist ein hoher Ruck j^1 , der zu einer unerwünschten, unruhigen Fahrweise des *ProfiBot*-Roboters führt. Sie ist zwar keine sicherheitsrelevante Fahrweise, aber dennoch unerwünscht, weil durch sie die mechanischen Komponenten im *ProfiBot*-Roboter stärker dynamisch beansprucht werden. Personen, die sich innerhalb der Spurbreite befinden und anschliessend aus ihr heraustreten, können z.B. eine mögliche Ursache für diese Fahrweise sein.

Bei Messfahrten in einem Versuchsareal zeigte sich dieses Verhalten beim Durchqueren einer schmalen Passage. Der *ProfiBot*-Roboter dreht sich während der Fahrt so lange in Richtung des hinter der Passage liegenden Freiraums, bis das Verhalten *Drehen* durch Erkennen einer Passagenbegrenzung den *ProfiBot*-Roboter in die andere Richtung drehen lässt. Die Ursache für das wechselnde Drehen sind Unstetigkeiten im Verlauf des Gewichtungsfaktoren β und ϵ . Auch in diesem Fall versucht der *ProfiBot*-Roboter bei einer Richtungswechselvorgabe mit der maximalen möglichen Beschleunigung der neu ermittelten Richtung zu folgen, das wiederum zu einem hohen Wert von j führt. Eine Möglichkeit, die Unstetigkeiten von β und ϵ zu beseitigen, ist eine zusätzliche Gewichtung der Messpunkte in der Nähe des Fahrspurrands. Das könnte ebenfalls durch einen weiteren Gewichtungsfaktor geschehen, mit dem β und ϵ in Abhängigkeit zur Entfernung eines Messpunktes fließend ineinander übergehen. Mit dem Gewichtungsfaktor ξ , der sich in dem festzulegendem Distanzintervall $d_{\xi 1} \dots d_{\xi 2}$ ändert, kann innerhalb dieses Intervalls die Gewichtung nach Fallunterscheidung 7.2 erfolgen.

$$\xi = \begin{cases} 0, & \text{falls } d < d_{\xi 1} \\ \beta \cdot \left(1 - \frac{d}{d_{\xi 2} - d_{\xi 1}}\right) + \epsilon \cdot \frac{d}{d_{\xi 2} - d_{\xi 1}}, & \text{falls } d_{\xi 1} < d < d_{\xi 2} \\ 1, & \text{falls } d \ge d_{\xi 2} \end{cases}$$
(7.2)

Der tatsächliche Fahrweg des ProfiBot-Roboters in diesem Versuchsareal ist wegen den fehlerhaften Odometriedaten, die bereits in Kapitel 5.4.3 beschrieben wurden, nicht aufgezeichnet. Die Solldrehzahlen der beiden Motoren und die Gewichtungsfaktoren β und ϵ wurden während der Fahrt aufgezeichnet und der Signalverlauf in Abbildung 7.15 dargestellt. Es ist zu erkennen, dass die Motordrehzahlen, β und ϵ Unstetigkeiten aufweisen (rote Pfeile).



Abb. 7.15: Drehzahlen und Gewichtungsfaktoren während der Messfahrt

7.4.1 Korrekturen

In den gewählten Einbaupositionen der analogen Triangulationssensoren konnten einige Tischkanten nicht erkannt werden, weil die Position dieser Tischkanten über den Einbaupositionen der *GP2Y0A02* liegt. Tischkanten, die knapp unterhalb der Position der Triangulationssensoren liegen, werden erst in einer solchen Distanz zur Tischkante erkannt, in der ein Ausweichen nicht mehr möglich ist. Deswegen ist eine alternative Einbauposition gewählt worden, bei der die analogen Triangulationssensoren am Bodenrahmen des ProfiBot-Roboters befestigt wurden und die Sensorachsen unter den Winkeln $\gamma_b = 20$ und $\gamma_{mess} = 50$ schräg nach oben geneigt sind. Die Baugruppe ist in Abbildung 7.16 dargestellt. Die alternative Montageposition ist in der Komplettansicht des modifizierten ProfiBot-Roboters im Anhang in Abbildung 8 dargestellt. Die Einbauposition bietet jedoch keinen Schutz davor, dass sich ein Objekt im Totbereich befinden kann.



Abb. 7.16: GP2Y0A02 mit Montagewinkeln

Kapitel 8

Diskussion und Ausblicke

8.1 Diskussion

Das entwickelte Konzept ist für eine autonome Fahrweise des ProfiBot-Roboter geeignet. Die in Kapitel 1.2.2 benannte Zielsetzung wurden realisiert, die Übertragbarkeit auf andere Robotertypen muss allerdings noch überprüft werden. Es ist gezeigt, dass mit dem gewählten Konzept eine Hindernisvermeidung durchgeführt werden kann. Der Einsatz der Subsumtionsarchitektur und die eindeutige Prioritätenvergabe der Verhaltensweisen im Steuerprogramm des ProfiBot-Roboters führt zu einem robusten Betriebsablauf. Durch die neue Position der Schaltleisten und der Montageposition 2D-Laserscanner und analogen Triangulationssensoren werden fast alle Objekte in den definierten Umgebungen erkannt. Weil ausführliche Messungen und Fahrversuche den Rahmen dieser Arbeit gesprengt hätten, ist keine präzise Aussage möglich, ob das Konzept bzw. der Demonstrator ein absolut sicheres Produkt ist und gefahrenlos in den definierten Umgebungen eingesetzt werden kann. Um einen genaueren Standpunkt vertreten zu können, kann mit dem ProfiBot-Roboter zunächst ein umfassender Dauerbetriebstest in geeigneten Räumen des Fraunhofer-Institutzentrums Birlinghoven durchgeführt werden, um ggf. Korrekturmaßnahmen vorzunehmen. Zur Umsetzung dieses Tests empfiehlt es sich den ProfiBot-Roboter vorerst in einem menschenleeren Raum zu betreiben und nach dem erfolgreichen Abschluss des Tests einen weiteren Test durchzuführen, bei dem sich Personen im Raum befinden. Darauf hinauslaufend ist ein Dauerbetriebstest unter realen Einsatzbedingungen möglich. der z.B. in einer Empfangshalle eines Museums verwirklicht werden kann. Für die Marktreife fehlen Konformitätstests, in denen die elektromagnetische Störfestigkeit und die Sicherheit im Dauerbetrieb überprüft werden muss. Die CE-Kennzeichnung, welche geschaffen wurde, um dem Endverbraucher sichere Produkte zu garantieren, fehlt ebenfalls und kann von einer Benannten Stelle ausgestellt werden. Für die Vergabe der CE-Kennzeichnung ist eine Risikobeurteilung des Konzepts notwendig.

8.2 Ausblicke (Perspektiven)

8.2.1 Änderung des Regelalgorithmus

In dieser Arbeit ist mehrmals erwähnt worden, dass eine Drehmomentregelung der DC-Motoren im Zusammenhang mit einer autonomen Navigation einige Vorteile bieten würde. Vorteile sind u.a.:

- Begrenzung des Anzugsmoments und somit der Roboterbeschleunigung
- konstante, lastunabhängige Fahrgeschwindigkeit
- hardwareseitige Begrenzung des Stroms auf einen Maximalwert

Für die Parameter der Drehmomentregelung können die nötigen Regelparameter einerseits anhand eines kinetischen Modells des *ProfiBot*-Roboters bestimmt werden, andererseits können die Parameter auch experimentell ermittelt werden. Die Drehmomentregelung kann als ein zur Drehzahlregelung unterlagerter Regelkreis in einer Kaskadenregelung realisiert werden. Der Wirkungsplan einer möglichen Struktur des Regelkreises ist in Abbildung 8.1 dargestellt.



Abb. 8.1: Wirkungsplan einer Drehzahlregelung mit unterlagerter Drehmomentregelung

Für die Realisierung muss die Ankerinduktivität L_a , der Ankerwiderstand R_a und die Maschinenkonstante $km = c_r \Psi_d$ der DC-Motoren bestimmt werden, sowie das gesamte Massenträgheitsmoment J aller rotierenden Komponenten. Der Drehmomentregler wird, wie auch der Drehzahlregler, als PI-Regler ausgeführt. Weil eine Baugruppe zur Messung des

Motorstroms für die Drehmomentermittlung bereits vorhanden ist und die Drehrichtung der Motoren durch Auswertung der Encodersignale bestimmt werden kann (s. dazu auch Kapitel 3.3.2), muss die Hardware des TMC für diese Struktur nicht verändert werden. Es ist ausschließlich der Regelalgoritmus in der Firmware zu verändern. Die Begrenzung der Roboterbeschleunigung kann mit einer zusätzlichen Vorsteuerung realisiert werden, welcher die notwendigen Führungsgrößen durch *IConnect* übergeben werden können.

Eine andere Möglichkeit wäre, die Reglerstruktur als Zustandsregelung auszuführen. Bei einer Zustandsregelung ist im Allgemeinen von Vorteil, dass die Regeldynamik für Antriebssysteme höher ist, als bei einer Kaskadenregelung, weil die Regler nur einen P-Anteil haben. Für die stationäre Genauigkeit der Zustandsregelung muss ein Beobachter integriert werden, der die Zustandsgrößen schätzt. Der Zeitaufwand für die Entwicklung eines Zustandsreglers ist allerdings hoch und es ist abzusehen, dass die Dynamik einer Kaskadenregelung für den *ProfiBot*-Roboter ausreichen würde.

8.2.2 Senkung des Rucks

Um die Beschleunigung zu begrenzen und den Ruck dadurch zu senken wäre eine weiterhin eine andere Möglichkeit denkbar, bei der keine Drehmomentregelung benötigt wird. In der Fallunterscheidung 8.1 mit den Definitionen 8.2 bis 8.4 wird eine hohe Drehzahldifferenz $\Delta n >> 0$ erkannt und das Erreichen der Sollgröße $n_{soll,k+1}$ verzögert.

$$n_{soll(t)} = \begin{cases} n_{soll(t=0)} + \left(\Delta n_{soll} \cdot \frac{1}{2} \cdot tanh\left(\frac{t}{k_v} - 3\right) + 1\right), & \text{falls } \frac{\Delta n_{soll}}{n_{max}} > 0, 4\\ n_{soll(t)}, & \text{falls } \frac{\Delta n_{soll}}{n_{max}} \le 0, 4 \end{cases}$$
(8.1)

$$k_v = \frac{\Delta n_{soll}}{n_{max}} \cdot \tau \tag{8.2}$$

$$n_{soll(t=\infty)} = n_{soll,k+1} \tag{8.3}$$

$$\Delta n_{soll} = n_{soll(t=\infty)} - n_{soll(t=0)} \tag{8.4}$$

Mit der Zeitkonstante τ lässt sich die Verzögerungszeit anpassen. Der Funktionsgraph in Abbildung 8.2 zeigt einen Sprung von $\frac{n_{soll}}{n_{max}} = 0,35$ auf $\frac{n_{soll}}{n_{max}} = 0,8$ und die entsprechenden Sollgeschwindigkeiten mit Verzögerung bei verschiedenen Werten von τ . Aus den Funktionsgraphen der Beschleunigungen $a_{soll(\tau=5)}$ und $a_{soll(\tau=80)}$ ist abzuleiten, dass der Ruck niedriger wird und das Erreichen von $n_{soll(t=\infty)}$ länger dauert, je größer τ ist. τ sollte in Fahrversuchen ermittelt werden. Bei einem zu kleinen τ ist die Wirkung der Rucksenkung unzureichendend, bei einem großen τ kann die Funktion der Verhaltensweisen beeinträchtigt werden, in dem die Ausführung verzögert wird.



Abb. 8.2: Verzögerung des Erreichens der Sollgeschwindigkeit

Die Verzögerung ist in Fahrsituationen unerwünscht, bei denen nach einem hohen Wert von d_{min} ein kleiner Wert von d_{min} folgt. In diesen Situationen, die sich z.B. durch plötzliches Auftauchen von Personen in der Fahrspur des Roboters ereignen, sollte der *ProfiBot*-Roboter seine Geschwindigkeit so schnell wie möglich drosseln, weil sonst eine Kollision verursacht werden könnte. Aus diesem Grund muss die Verzögerung des Stoppvorgangs in solchen Situationen umgangen werden.

8.2.3 Dockingstation

Wenn der im Rahmen dieser Arbeit entwickelte *ProfiBot*-Roboter im kontinuierlichen Betrieb eingesetzt wird, ist es von Vorteil, autonom an eine Dockingstation zu fahren, um die Batterien aufzuladen. Der Wartungsaufwand ist somit geringer und die stationäre Verfügbarkeit erhöht sich. Brauers stellt in seiner Diplomarbeit [6] diesbezüglich ein geeignetes Konzept vor, welches sich aus einer Dockingstation, der erforderlichen Sensorik und dem Steuerprogramm zusammensetzt. Als Navigationsverfahren wird die Lagepeilung verwendet. Weil das Konzept bereits mit der Basisversion des *ProfiBot*-Roboters erprobt wurde, ist der zusätzliche Installationsaufwand gering.

Anhang



Abb. 3: Platinenlayout des Sensorboards



Abb. 4: Platinenlayout des Verstärkerboards



Abb. 5: Konturendiagramm des Röhrchens in 200mm Entfernung



Abb. 6: Konturendiagramm des Röhrchens in 400mm Entfernung



Abb. 7: Konturendiagramm des Röhrchens in 1000mmEntfernung



Abb. 8: CAD-Modell des Mülltonnenroboters (Komplettansicht)

Inhalt der CD

- Signalgraphen AutonomeFahrt und Scanner für IConnect 6.1
- Quellcode der Module URG04LX und O_Avoidance für Microsoft Visual C++ 6.0
- CAD-Modell des modifizierten ProfiBot-Roboters für Solid Works 2007 SP 3.1
- Messreihen im Microsoft Excel Format
- Verwendete Bilder und Graphiken
- pdf-Version der Arbeit

Installationshinweise befinden sich auf der CD in der Datei readme.txt.

Formelzeichenverzeichnis

Zeichen	Bedeutung	Einheit
α_l	Lenkwinkel für Hindernisvermeidung	rad
α_{max}	Maximaler Lenkwinkel	rad
α_{soll}	Solllenkwinkel	rad
β	Gewichtungsfaktor	-
γ_s, γ_s'	Erfassungswinkel	rad
γ_{mess}	Winkel zwischen Sensorachse und xz-Ebene	rad
γ_b	Biegewinkel	rad
ϵ	Gewichtungsfaktor zu v_{soll}	
λ	Faktor	
φ	Winkel zwischen Ordinate und Hindernissen	rad
φ	Winkel zwischen Ordinate und Hindernissen	rad
φ_{scan}	Winkel eines Einzelscans	rad
$\Delta \varphi_{scan}$	Differenz benachbarter φ_{scan}	rad
φ_{Empf}	Phasenwinkel vom Empfängersignal	rad
φ_K	Winkel zwischen Ordinate und r_K	rad
φ_S	Winkel zwischen Ordinate und r_S	rad
φ_{Send}	Phasenwinkel vom Empfängersignal	rad
$ heta_y$	Winkel zwischen Orientierung und y-Achse	rad
au	Zeitkonstante	$\frac{1}{s}$
ω_K	Rotatorische Geschwindigkeit von K	$\frac{1}{s}$
ω_l	Winkelgeschwindigkeit zu α_l	$\frac{1}{s}$
ω_{mech}	Mechanische Winkelgeschwindigkeit	$\frac{1}{s}$
ω_M	Winkelgeschwindigkeit eines Motors	$\frac{1}{s}$
Ψ_d	Verkettungsfluß in der d-Achse einer el. Maschine	$\frac{Vs}{A}$
a	Beschleunigung	$\frac{m}{s^2}$
a _{soll}	Sollbeschleunigung	$\frac{m}{s^2}$
b,b'	Breite der erfassten Spur	mm
b_{rob}	Roboterbreite	mm
b_s	Ordinatenschnittpunkt	-

b_v	Breite der Fahrspur	mm
с	Lichtgeschwindigkeit	$\frac{m}{s}$
c_r	Maschinenkonstante	Vs
C_e	Encoderauflösung	Ticks
d_{step}	Mindestabstand zwischen zwei Puntken	mm
d_R	Raddurchmesser	m
d_T	Distanzschwellwert für r	mm
$\vec{d_S}$	Distanzvektor von S zum S'	mm
d_{mess}	Entfernung vom Sensorachsenende zum Roboter	mm
d_{min}	Entfernung zum nächsten Messpunkt	mm
d^*_{min}	Distanzschwellwert	mm
$d_{min,URG}$	Entfernung zum nächsten Messpunkt gemessen vom $URG-04LX$	mm
$d_{min,GP2Y}$	Entfernung zum nächsten Messpunkt gemessen vom $GP2Y0A02$	mm
d_s	Berechnete Distanz eines GP2Y0A02	mm
d_s^*	Gemessene Distanz eines GP2Y0A02	mm
D	Schaltsignal eines GP2Y0D02	-
Ι	Untersetzungsverhältnis	-
i_B	Bemessungsstrom	A
i_L	Laststrom	A
i_M	Motorstrom	A
i_d	Flußbildender Strom in der q-Achse	A
i_q	Momentbildender Strom in der q-Achse	A
j	Ruck	$\frac{m}{s^3}$
k	Index	-
k_n	Verstärkungsfaktor	-
K	kinematische Zentrum	-
K_P	Verstärkung des P-Anteils	-
l_R	Radabstand	m
m	Konstante	$\frac{1}{m}$
m_s	Steigung	-
M	Geometrischer Mittelpunkt eines Roboters	-
M_L	Lastmoment	Nm
M_M	Motormoment	Nm
n	Wellenlänge	nm
n_l	Drehzahl des linken Motors	$\frac{1}{min}$
n_r	Drehzahl des rechten Motors	$\frac{1}{min}$
n_S	Anzahl der Messungen in einem Scan	-
N_e	Aktuelle Ticks	Ticks
P	Pose des Roboters	-
r	Distanz von S zu Hindernissen	mm

r_w	Wendekreisradius	mm
r_K	Distanz vom K zum erfassten Punkt	mm
r_R	Radius eines Luftrades	m
r_S	Distanz vom S zum erfassten Punkt	mm
$\vec{r_S}$	Distanzvektor vom S zum erfassten Punkt	mm
$\vec{r_S}'$	Distanzvektor von S' zum erfassten Punkt	mm
r_{Smax}	Maximale Reichweite der Sensorik	mm
s_l	Messwert des linken Sensors	-
s_r	Messwert des rechten Sensors	-
s_R	Gefahrene Strecke vom Luftrad	m
S,S'	Sensorzenrum	-
t	Zeit	s
Т	Zykluszeit	s
T_b	Mittlere Betriebsdauer zwischen zwei Ausfällen	s
T_s	Mittlere Störungsdauer	s
T_D	Zeitkonstante des D-Anteils	ms
T_I	Zeitkonstante des I-Anteils	ms
u	Übergangswertebereich	-
U_H	Schaltspannung des "High-Pegels"	V
U_L	Schaltspannung des "Low-Pegels"	V
U_s, U_s^*	Ausgabewerte des Buskopplers	$\frac{V}{0,333mV}$
v	Geschwindigkeit	$\frac{m}{s}$
\vec{v}	Bewegungsvektor	$\frac{m}{s}$
v_l	Geschwindigkeit vom linken Rad	$\frac{m}{s}$
v_{max}	Maximalgeschwindigkeit	$\frac{m}{s}$
v_r	Geschwindigkeit vom rechten Rad	$\frac{m}{s}$
v_{soll}	Sollgeschwindigkeit des Roboters	$\frac{m}{s}$
v_K	Translatorische Geschwindigkeit von K	$\frac{m}{s}$
v_R	Geschwindigkeit von einem Luftrad	$\frac{m}{s}$
V_{CC}	Kollektor-Kollektor-Spannung	V
Δx	X-Komponente von $\vec{d_S}$	mm
Δx_S	X-Komponente von $\vec{r_S}$	mm
Δy	Y-Komponente von $\vec{d_S}$	mm

Abkürzungsverzeichnis

Abkürzung	Bedeutung
2D	2-Dimensional
3D	3-Dimensional
A/D	\mathbf{A} nalog/ \mathbf{D} igital
AMR	Autonomer, mobiler Roboter
APD	Avalance Photo Diode
API	Application Programming Interface
ARW	Anti-Reset Windup
BJT	Bipolar Junction Transistor
CAD	Computer Aided Design
CAN	Controller Area Network
CCD	Charge-Coupled Devices
CD	Compact Disc
CMOS	Complementary Metal Oxide Semiconductor
DC	Direct Current
DMS	\mathbf{D} ehnungs- \mathbf{M} ess- \mathbf{S} treifen
FAIRLib	\mathbf{F} raunhofer Autonomous Intelligent Robotic Libary
FFT	Fast Fourier Transformation
FTS	\mathbf{F} ahrerloses \mathbf{T} ransport- \mathbf{S} ystem
GND	Ground
GPS	Global Positioning System
Fraunhofer AIS	Fraunhofer Institut für Autonome Intelligente \mathbf{S} ysteme
Fraunhofer IPA	Fraunhofer Institut für Produktionstechnik und Automatisierung
Fraunhofer IAIS	Fraunhofer Institut für Intelligente Analyse- und Informations systeme
KI	\mathbf{K} ünstliche Intelligenz
KURT	${\bf K} anal {\bf \cdot} {\bf U} ntersuchung {\bf s} {\bf \cdot} {\bf R} oboter {\bf \cdot} {\bf T} est plattform$
LASER	Light Amplification by Stimulated Emission of Radiation
LED	Light Emitting Diode
PC	Personal Computer
PID	Proportional-Integral-Derivative

PSD	Position Sensitive Devices
PWM	Pulse Width Modulation
SLAM	Simultaneous Localisation And Mapping
TMC200	Triple Motor Controller 200
TOF	Time-Of-Flight
USB	Universal Serial Bus
VDI	Verein Deutscher Ingenieure

Literaturverzeichnis

- [1] ADAMY, J.; BECHTEL, P.: Sicherheit mobiler Roboter. Manuskript, Technische Universität Darmstadt, Darmstadt 2003.
- [2] ALTENBURG, J.; ALTENBURG, U.: Mobile Roboter. Hanser Verlag, München 1999.
- [3] BERTRAM, T.: Odometrie und Probabilistisches Bewegungsmodell. Manuskript, Universität Dortmund, Dortmund 2007.
- [4] BLUM, H.: Notaus oder Nothalt. http://www.elektrofachkraft.de (verifiziert 13.12.2007).
- [5] BRAITENBERG, V.: Vehikel Experimente mit künstlichen Wesen. Lit-Verlag, Münster 2004.
- [6] BRAUERS, C.: Navigation eines autonomen mobilen Roboters anhand von Landmarken zum Andocken an einer Ladestation. Diplomarbeit, FH Bonn-Rhein-Sieg - Fachbereich Elektrotechnik, Maschinenbau und Technikjournalismus, Bonn 2007.
- [7] BROCKHAUS: Die Enzyklopädie in 30 Bänden. F.A. Brockhaus, Mannheim 2005-2007.
- [8] BROOKS, R. A.: A robust layered system for a mobile robot. MIT AI Lab Memo 864, Cambridge MA 1985.
- [9] CWIENTZEK, A.: Weiterentwicklung eines Motorcontrollers zur Produktreife. Diplomarbeit, FH Wedel, Wedel 2005.
- [10] DRESSLER, J.: Navigation mobiler Systeme in Indoor-Umgebungen. Diplomarbeit, FH Brandenburg - FB Informatik und Medien, Brandenburg 2002.
- [11] GARBRECHT, F. W.; SCHAAD, H.-J.; LEHMANN, R.: Workshop der professionellen Antriebstechnik. Franzis Verlag GmbH, Poing 1996.
- [12] GOCKEL, T.; DILLMANN, R.; BIERBAUM, A.; PIASECZKI, A.; SCHRÖDER, J.; AZARD, P.: Embedded Robotics. Elektor Verlag, Aachen 2005.

- [13] GRAF, B.: Autonome mobile Roboter. Manuskript, Fraunhofer IPA, Stuttgart 2000.
- [14] GRAF, B.; BAUM, W.; TRAUB, A.; SCHRAFT, R.-D.: Konzeption dreier Roboter zur Unterhaltung der Besucher eines Museums. Manuskript, Fraunhofer IPA, Stuttgart 2000.
- [15] GUT, O.: Untersuchung des 3D-Sensors Swissranger. http://www.geometh-data.ethz.ch/publicat/diploma/gut2004/index.html (verifiziert 13.12.2007).
- [16] GUTMANN, J.-S.; NEBEL, B.: Navigation mobiler Roboter mit Laserscans. Manusskript, Universität Freiburg - Institut für Informatik, Freiburg 2005.
- [17] HANS, M.: Eine modulare Kontrollarchitektur für den Hol- und Bringdienst von Roboterassistenten. Dissertation, Institut für Industrielle Fertigung und Fabrikbetrieb der Universität Stuttgart, Stuttgart 2005.
- [18] HOKUYO AUTOMATIC CO., LTD: Datenblatt Hokuyo URG-04LX. http://www.hokuyo-aut.jp/02sensor/07scanner/urg.html (verifiziert 13.12.2007).
- [19] HOKUYO AUTOMATIC CO., LTD: URG-Series Communication Protocol Specification SCIP-Version 2.0. http://www.hokuyo-aut.jp/02sensor/07scanner/urg.html (verifiziert 13.12.2007).
- [20] HOKUYO AUTOMATIC CO., LTD: Katalog Hokuyo URG-04LX. http://www.hokuyo-aut.jp/02sensor/07scanner/urg.html (verifiziert 13.12.2007).
- [21] KAWATA, H.; OHYA, A.; YUTA, S.; WAGLE, S.; MORI, T.: Development of ultra-small lightweight optical range sensor system. http://www.hokuyo-aut.jp/02sensor/07scanner/urg.html (verifiziert 13.12.2007).
- [22] HOLZ, D.: Kontinuierliche Umgebungskartographie mittels 3D-Laserscanner auf autonomen mobilen Robotern. Diplomarbeit, FH Köln - Fakultät für Informatik und Ingenieurwissenschaften, Köln 2006.
- [23] KRASCHL, G.: Routenplanung f
 ür mobile Roboter. Magisterarbeit, TU Graz -Institut f
 ür Navigation und Satellitengeod
 äsie, Graz 2005.
- [24] LAUE, T.: Eine Verhaltenssteuerung für autonome mobile Roboter auf der Basis von Potentialfeldern. Diplomarbeit, Universität Bremen - Fachbereich Mathematik und Informatik, Bremen 2004.
- [25] LINGEMANN, K.; NÜCHTER, A.; HERZBERG, J.; SURMANN, H.: About The Control of High Speed Mobile Indoor Robots. In:Proceedings Second European Conf. on Mobile Robotics, Acona 2005, s. 218-223.
- [26] LÖRKEN, C.: Introducing Affordances into Robot Task Execution. Publications of the Institute of Cognitive Science, Volume 2, Osnabrück 2007.

- [27] MAY, S.; WERNER, B.; SURMANN, H.; PERVÖLZ, K.: 3D time-of-flight cameras for mobile robotics. Manuskript, Fraunhofer Institut AIS, Sankt Augustin 2006.
- [28] MAYSER SCHAUMSTOFFTECHNIK: Katalog Safety Bumper. http://www.mayser.de/content/sicherheitstechnik/informationen/downloads /einzelkatalog/Kat-SB_dew.pdf (verifiziert 13.12.2007).
- [29] MEL MIKROELEKTRONIK GMBH: Katalog Optische Sensoren. http://www.melsensor.de/ams_down.php?File=261 (verifiziert 13.12.2007).
- [30] MICRO-EPSILON MESSTECHNIK GMBH & CO. KG: ICONNECT in 60 Minuten. Handbuch, http://www.micro-epsilon.de, Ortenburg 2007.
- [31] MÜLLER, G.; PONICK, B.: Grundlagen elektrischer Maschinen. Wiley-VCH, Weinheim 2005.
- [32] NEHMZOW, U.: Mobile Robotik. Springer Verlag, Berlin 2002.
- [33] OTT, C.; WIMBÖCK, T.: Impedanzregelung zur beidhändigen Manipulation. Begleitfolien, DLR - Institut für Robotik und Mechatronik, München 2006.
- [34] PAULUS, J.: Entwicklung eines Motortreibers für autonome Roboter. Diplomarbeit, FH Aachen - Fachbereich Maschinenbau und Mechatronik, Aachen 2006.
- [35] PEPPERL+FUCHS GMBH: Datenblatt UB6000-30M. http://www.pepperl-fuchs.com/selector/navi/productInfo/45/450047.pdf (verifiziert 13.12.2007).
- [36] PHOENIX CONTACT GMBH & CO. KG: Handbuch Buskoppler IL ETH BK DI8 DO4 2TX-PAC. http://select.phoenixcontact.com/phoenix/dwl/dwl03c.jsp?fct=show&UID=2703981
 &qprodid=IL ETH BK (verifiziert 13.12.2007).
- [37] PHOENIX CONTACT GMBH & CO. KG: Handbuch Inline-Klemme IB IL AI 2/SF (-PAC).
 http://select.phoenixcontact.com/phoenix/dwl/dwl03c.jsp?fct=show&UID=9423374
 &qprodid=IB IL AI (verifiziert 13.12.2007).
- [38] POINT GREY RESEARCH: Datenblatt Bumblebee XB3. http://www.ptgrey.com/products/bbxb3/index.asp (verifiziert 13.12.2007).
- [39] PUTTKAMER, E. VON: Autonome mobile Roboter. Vorlesungsskript, Universität Kaiserslautern - Robotik und Process-Control, Kaiserslautern 2000.
- [40] SCHÖLL, P.: Modellbasierte Softwareentwicklung für die Generierung von Kontrollprogrammen auf Basis der Eclipse RCP. Master Thesis, FH Bonn-Rhein-Sieg - Fachbereich Informatik, Bonn 2005.

- [41] SHARP ELECTRONICS CO.: Datenblatt Sharp GP2Y0A02. http://www.sharpsme.com/Page.aspx/europe/en/part/GP2Y0A02YK/ (verifiziert 13.12.2007).
- [42] SCHRAFT, R.-D.; HÄGELE, M.; WEGENER, K.: Service Roboter Visionen. Hanser Verlag, München 2004.
- [43] SIEMENS AG AUTOMATION AND DRIVES: Katalog Sicherheitssensoren. http://www.siemens.de/simatic-sensors/fs (verifiziert 13.12.2007).
- [44] STEINBAUER, G.: Konstruktion Mobiler Roboter Einführung Hardware. http://www.ist.tugraz.at/staff/steinbauer/kmr07_hw.pdf (verifiziert 13.12.2007).
- [45] STENZEL, R.: Steuerungsarchitekturen f
 ür autonome mobile Roboter. Dissertation, TU Aachen - Fakult
 ät f
 ür Mathematik, Informatik und Naturwissenschaften, Aachen 2002.
- [46] SURMANN, H.; PERVÖLZ, K.; NÜCHTER, A.; LINGEMANN, K.; HERTZBERG, J.; HENNIG, M.: Aufbau eines 3D-Laserscanners für autonome mobile Roboter. Manuskript, GMD Forschungszentrum Informationstechnik GmbH, Sankt Augustin 2001.
- [47] TAPESWITCH GMBH: Datenblatt TS16S Schaltleiste. http://www.tapeswitch.de/pdf/Infoblaetter/Schaltleiste%20TS16S%2029.10.01.pdf (verifiziert 13.12.2007).
- [48] TIETZE, U.; SCHENK, C.: Halbleiter-Schaltungstechnik. Springer-Verlag, Berlin 1990.
- [49] TRÄNKLER, H.-R.; OBERMEIER, E.: Sensortechnik Handbuch für Praxis und Wissenschaft. Springer Verlag, Berlin 1998.
- [50] VDI RICHTLINIEN 4451 (BLATT 6): Kompatibilität von Fahrerlosen Transportsystemen (FTS) - Sensorik für Navigation und Steuerung. Verein deutscher Ingenieure, Düsseldorf 2003.
- [51] VDI RICHTLINIEN 4452: Abnahmeregeln für Fahrerlose Transportsysteme (FTS). Verein deutscher Ingenieure, Düsseldorf 2004.
- [52] WORST, R.; SURMANN, H.; CHRISTALLER, T.: Erkundung unzugänglicher Bereiche mit mobilen Robotern. In:Proceedings 5. Altbergbau-Kolloquium, Clausthal-Zellerfeld 2005.
- [53] http://www.conrad.de (verifiziert 13.12.2007). Conrad Elektronik Shop.
- [54] http://de.wikipedia.org/wiki/K%C3%BCnstliche_Intelligenz (verifiziert 13.12.2007).
 Wikipedia: Definition von künstlicher Intelligenz.

- [55] http://www.mesa-imaging.ch/prodviews.php (verifiziert 13.12.2007). MESA Swissranger SR-3000 Produktseite.
- [56] http://www.metralabs.com/downloads/MetraLabs_SCITOS_G5_DE.pdf (verifiziert 13.12.2007). Artikel der Firma MetraLabs zu Scitos G5.
- [57] http://www.profiBot.de (verifiziert 13.12.2007). ProfiBot Website.
- [58] http://www.ssz-gmbh.de/frameset.htm (verifiziert 13.12.2007). SSZ Sicherheitstechnik Firmenseite.
- [59] http://www.teg.fraunhofer.de/arbeitsgebiete/medizintechnik/systeme /sensorfusion/ (verifiziert 13.12.2007).
 Fraunhofer TEG - Sensorfusion.