



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Ausarbeitung Anwendungen 1 WiSe 2009/2010

Philipp Teske

Bewegungsplanung von mehrachsigen
Gelenkrobotern

Inhaltsverzeichnis

| | |
|--|-----------|
| 1 Einführung | 4 |
| 1.1 Motivation..... | 4 |
| 1.2 Steuerungsarten..... | 5 |
| 2 Bahnplanungsalgorithmen | 6 |
| 2.1 Probabilistische Straßenkarte..... | 6 |
| 2.2 Voronoi Diagramme..... | 7 |
| 2.3 Suchbäume..... | 7 |
| 2.4 Zellzerlegung..... | 8 |
| 2.5 Potentialfeldmethode..... | 9 |
| 3 Verwandte Projekte | 10 |
| 4 Fazit | 11 |
| Abbildungsverzeichnis | 12 |
| Literaturverzeichnis | 13 |

Kurzzusammenfassung

Der Einsatzbereich von Industrierobotern beschränkt sich schon lange nicht mehr nur auf die Nutzung in Fabriken. In vielen unterschiedlichen Bereichen kommen heute Gelenkarmroboter bei schweren und auch filigranen Arbeiten zum Einsatz. Einige Forschungseinrichtungen beschäftigen sich mit der Entwicklung von autonomen Haushaltsrobotern.

Zu der Steuerung der Gelenkarme in diesen autonomen Systemen gehören komplexe Algorithmen, welche die Berechnung des Bewegungsablaufes übernehmen. In diesem Dokument werden einige aktuelle Forschungsprojekte vorgestellt, sowie Verfahren zur Bewegungs- bzw. Bahnplanung erläutert.

1 Einführung

1.1 Motivation

Durch die fortschreitende Industrialisierung wurde es unabdingbar, schwere oder gefährliche Aufgaben nicht mehr von Menschen, sondern von Maschinen durchführen zu lassen. Industrieroboter wurden bereits in den 80er Jahren routinemäßig eingesetzt. Während sie anfangs in Automobilfabriken und Kraftwerken ihren Dienst verrichteten, wurden sie im Laufe der Zeit mehr und mehr weiter entwickelt und an andere Aufgaben angepasst. So unterteilt man heute die Industrieroboter u.a. in die Bereiche Gelenk- bzw. Gelenkarmroboter (Abbildung 1), Portalroboter sowie Maschinen mit paralleler Kinematik. Diese Maschinen werden seit mehreren Jahren nicht mehr nur im industriellen Umfeld eingesetzt.

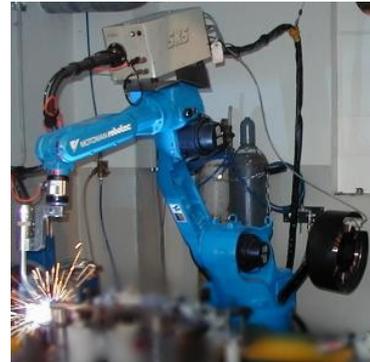


Abbildung 1: Gelenkarmroboter mit Schweißbrenner

Durch die Entwicklung immer kleinerer Materialien und leistungsfähigeren Schaltkreisen, konnten sich Industrieroboter auch in anderen Anwendungsbereichen bewähren. So werden bereits seit einigen Jahren besonders in der Medizintechnik und bei chirurgischen Eingriffen immer häufiger Gelenkarmroboter eingesetzt. Diese erlauben es aufgrund ihrer Präzision, Operationen am menschlichen Körper wesentlich schonender durchzuführen, als es Ärzten möglich ist. Während in der Medizintechnik die Roboter überwiegend noch von Menschen bedient werden, erlangen autonom agierende Roboter in anderen Aufgabenbereich stetig an Bedeutung. Speziell im Bereich des häuslichen Umfeldes werden sie zunehmend eingesetzt. Kleine Roboter, die in der Lage sind, einfache Aufgaben wie z.B. das Schneiden des Rasens zu übernehmen, werden bereits seit einigen Jahren im Fachhandel angeboten. Komplexere Roboter, welche dem Menschen bei unterschiedlichen Aufgaben im Haushalt helfen sollen, existieren jedoch bisher nur in den Laboratorien der Forschungsinstitute, dennoch zeigt sich bereits, welches Potential diese Technik enthält (Abbildung 2).



Abbildung 2: Care-O-bot® des Fraunhofer IPA beim Greifen einer Dose
(Quelle: [1])

Diese neue Technik verbindet mehrere Bereiche der Robotik miteinander. So ist es nicht mehr nur wichtig, dass der Roboter Objekte nehmen und bearbeiten kann, es ist zudem nötig, dass er in der Lage ist, sich zu bewegen. Diese neue Art von Robotern muss also in der Lage sein, sowohl autonom zu agieren, als auch mit einer Person zu interagieren.

Um die Möglichkeiten, die diese Technik mit sich bringt, besser einschätzen zu können, soll für das von der Wirtschaftsbehörde unterstützten *Living Place Hamburg*, welches zur Zeit am Campus Berliner Tor der HAW entsteht, eine mobile elektronische Haushaltshilfe entwickelt werden.

Eine besondere Herausforderung stellt dabei die Steuerung des Gelenkarms dar. Um Gegenstände fassen zu können, muss er in der Lage sein, um Hindernisse herum zu greifen. Zudem darf er Menschen durch unerwartete Bewegungen nicht gefährden.

1.2 Steuerungsarten

Es gibt zwei Verfahrensarten, die Bewegung eines Roboters festzulegen. Zum einen die Online-Verfahren, welche eine direkte Interaktion mit dem Roboter benötigen und zum anderen die Offline Verfahren, welche unabhängig vom Roboter entwickelt werden können. Zu den Online-Verfahren gehört unter anderem das „*Teach-In*“- und „*Playback*“-Verfahren. Beim *Teach-In*-Verfahren wird der Roboterarm per Hand oder mit Hilfe einer Steuerkonsole zu signifikanten Koordinaten, welche sich auf seiner späteren Bewegungsbahn befinden, gefahren und diese gespeichert. Zusätzlich werden für jeden Punkt spezielle Parameter, wie z.B. die Bewegungsgeschwindigkeit oder die Anfahrbeschleunigung, festgelegt. Ebenfalls wird festgelegt, ob die einzelnen Gelenke ihre Position auf dem geometrisch günstigsten Weg anfahren sollen (Point-To-Point), oder ob das Arbeitswerkzeug einem bestimmten Pfad folgen soll (Continuous Path) [2].

Bei dem *Playback*-Verfahren hingegen wird dem Roboter der komplette Bewegungsablauf vorgegeben. Dies geschieht, indem per Hand oder über eine Steuerkonsole die gesamte Bewegung einmal ausgeführt und in kurzen Zeitabständen die Position und Winkel der einzelnen Gelenke gespeichert wird.

Da bei Online-Verfahren die Maschinen jedoch je nach Programmieraufwand lange still stehen, werden bevorzugt die Offline-Verfahren verwendet. Hierbei wird der Bewegungsablauf des Roboters unabhängig von diesem am Computer erstellt. Dies hat den Vorteil, dass die Bewegungen simuliert werden können und somit Fehler an der Maschine und der Arbeitsumgebung reduziert werden. Des Weiteren werden durch dieses Verfahren die Stillstandszeiten der Maschine erheblich verkürzt, da diese lediglich zum Einspielen des Programms angehalten werden muss [2].

Alle diese Verfahren haben jedoch den Nachteil, dass die Bewegungen fest vorgegeben sind und der Roboter somit nicht eigenständig auf zufällig eintretende Änderungen in seiner Arbeitsumgebung reagieren kann. Das bedeutet, wenn eine Person oder ein Gegenstand in den Bewegungsbereich des Roboters gerät, muss dieser von einem Sicherungsmechanismus angehalten werden um eine Kollision zu verhindern.

2 Bahnplanungsalgorithmen

Das Ziel von Bahnplanungsalgorithmen ist es, Robotern die Fähigkeit zu geben, ihre Bewegung selbständig zu planen und der aktuellen Situation anzupassen.

Während die in Kapitel 1.2 beschriebenen Verfahren es lediglich erlauben, bekannte statische Hindernisse zu umfahren, ist es dem Roboter mit Hilfe von dynamischen Bahnplanungsalgorithmen möglich, eigenständig auf flexible Änderungen in ihrem Arbeitsbereich zu reagieren. Damit diese Art der Algorithmen jedoch funktionieren kann, muss sichergestellt werden, dass im Arbeitsbereich auftretende Hindernisse zuverlässig und schnell erkannt werden. Dies wird im Regelfall entweder durch fest installierte Kameras oder mit Hilfe von Laserscannern erreicht.

Die existierenden Algorithmen lassen sich in zwei Kategorien unterteilen. Die globalen Algorithmen erfassen und analysieren den kompletten Arbeitsbereich des Roboters, bevor sie den endgültigen Bewegungsablauf berechnen. Hierzu zählen alle Verfahren, die Bewegungen auf Basis von Graphen berechnen. Die andere Kategorie umfasst die „lokalen Verfahren“. Diese teilen den Bewegungsablauf in viele kleine Schritte auf und verwenden zur Berechnung des nächsten Bewegungsschrittes lediglich die aktuelle Umgebung des Roboters. Sie ermöglichen bei dynamischen Änderungen des Arbeitsraumes eine schnellere und effektivere Berechnung eines alternativen Pfades, als es bei globalen Bahnplanungsalgorithmen möglich ist.

2.1 Probabilistische Straßenkarte

Dieses Verfahren gehört zu der Gruppe der globalen Planungsalgorithmen. Es setzt sich aus den zwei Phasen Konstruktion und Planung zusammen. In der Konstruktionsphase werden aus dem Arbeitsbereich des Roboters willkürlich Koordinaten gewählt (Abbildung 3b). Für jeden gewählten Punkt wird geprüft, ob dieser mit einem Hindernis kollidiert. Alle Punkte die sich im freien Raum befinden werden gespeichert und mit ihren Nachbarn verbunden. Hierbei wird der Punkt in der Regel entweder mit einer bestimmten Anzahl an Nachbarpunkten verbunden oder mit allen Punkten innerhalb eines gewissen Abstandes. Dies geschieht so lange, bis eine ausreichend gute Abdeckung des Raumes erfolgt ist. Somit entsteht ein Graph, welcher den Bereich, in dem sich der Roboter bewegen kann, abbildet (Abbildung 3c). In der Planungsphase wird anschließend die Start- und Zielkoordinaten mit dem Graphen verbunden und mit Hilfe von Algorithmen zur Graphensuche, wie z.B. Dijkstra's „*shortest path query*“, der kürzeste Pfad zwischen den Koordinaten gefunden (Abbildung 3d) [4], [5], [6].

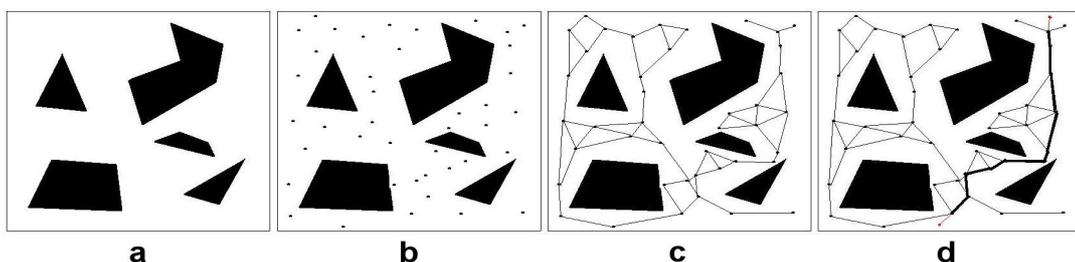


Abbildung 3: Probabilistic Roadmap Algorithm
(Quelle: [3])

2.2 Voronoi Diagramme

Das im vorherigen Kapitel beschriebene Verfahren hat den Nachteil, dass der Roboter bei inkorrekt planung den Hindernissen sehr nahe kommen und es bei einer zusätzlichen steuerungsbedingten Abweichung von der Bahn zu einer Kollision kommen kann.

Um dies zu vermeiden, sollte ein Pfad gefunden werden, der immer den größtmöglichen Abstand zu Hindernissen und den Rändern des Arbeitsbereiches einhält. Eine Möglichkeit, dies zu erreichen, ist mit Hilfe von Voronoi Diagrammen.

Voronoi Diagramme, benannt nach dem russischen Mathematiker Georgi Feodosjewitsch Woronoi, werden verwendet, um einen Bereich zwischen Objekten in mehrere Regionen aufzuteilen, deren Grenzen zu allen benachbarten Objekten den gleichen Abstand haben.

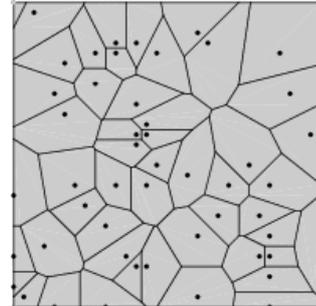


Abbildung 4: Voronoi Diagramm
(Quelle:

mathworld.wolfram.com/VoronoiDiagram.html)

Diese Grenzen werden auch als Voronoi-Kanten bezeichnet.

Der Arbeitsraum C des Roboters wird hierfür, in Abhängigkeit von Hindernissen, in mehrere Regionen, sogenannte Voronoi-Regionen V unterteilt. Alle auf den Kanten der Hindernisse und des Arbeitsraumes liegenden Punkte werden als Mittelpunkte für Voronoi-Regionen angesehen und als Menge $P = \{p_1, p_2, p_3, \dots, p_n\}$ zusammengefasst. Jeder Punkt im Arbeitsraum wird nun einem Element aus P zugeordnet, woraus sich folgende Formel ergibt:

$$V(p_i) = \{ x \in C : | p_i - x | \leq | p_j - x | \text{ für alle } j \neq i \}$$

Diese besagt, dass ein Punkt x , welcher dichter an dem Mittelpunkt p_i als an allen anderen Punkten aus P liegt, teil der Voronoi-Region $V(p_i)$ ist. Die Menge aller Punkte, welche in mehr als einer Region liegen, werden als Voronoi Diagramm bezeichnet. In der Praxis kommen jedoch meist Näherungsverfahren, wie das Plane-Sweep- oder Divide-and-Conquer-Verfahren, zum Einsatz [7],[8].

2.3 Suchbäume

Während die bisher beschriebenen Verfahren darauf abzielen, eine gewisse Auswahl an unterschiedlichen Wegen zur Verfügung zu stellen, zielt der Baum-basierte Ansatz auf das Finden eines einzigen Weges.

Für die Bahnplanung werden drei Grundarten von Suchbäumen eingesetzt, die alle nach einem ähnlichen Verfahren arbeiten. Ausgehend von der Startposition wachsen sie schrittweise in Richtung Ziel. Da diese Verfahren nicht den kompletten Arbeitsraum des Roboters abdecken, sondern nur den Teil, der sich zwischen den Start- und Zielkoordinaten befindet, benötigen sie wesentlich weniger Zeit.

Bei den sogenannten *Rapidly-Exploring Random Trees* (RRTs) wird in jedem

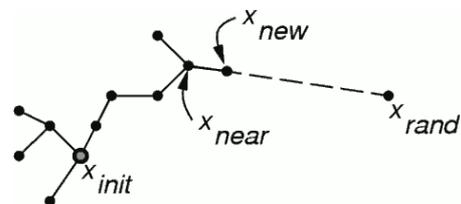


Abbildung 5: Aufbau eines Rapidly-Exploring Radom Tree
(Quelle:

aigamedev.com/open/highlights/rapidly-exploring-random-trees)

Iterationsschritt ein Punkt in der näheren Umgebung der Zielkoordinate ausgewählt. Anschließend wird der Knoten des Baumes, welcher dem Punkt am nächsten liegt um eine bestimmte Strecke in dessen Richtung erweitert. Zuletzt wird am Ende der neu erstellen Strecke ein neuer Knoten erstellt.

Die *Expansive Space Trees* (ESTs) hingegen wählen in jedem Iterationsschritt einen Knoten des Baumes aus, der anschließend in eine beliebige Richtung erweitert wird. Die Priorität hierbei liegt jedoch auf einer gleichmäßigen Verteilung der Knoten, so dass Regionen, die vom EST noch nicht abgedeckt sind, bevorzugt werden.

Die Dritte Kategorie der Baum-basierten Verfahren enthält die *Path Directed Subdivision Trees* (PDSTs). Diese Arbeiten ähnlich dem Zellzerlegungsverfahren, welche in den nachfolgenden Kapiteln näher beschrieben werden [4],[3].

2.4 Zellzerlegung

Bei der Bahnplanung mit Hilfe von Zellzerlegungsverfahren wird der Arbeitsraum des Roboters in mehrere Zellen zerlegt.

Eine Variante dieses Verfahren ist die *Vertical Cell Decomposition*. Hierbei wird der Arbeitsraum in mehrere vertikale Bereiche unterteilt. Diese können entweder eine feste Breite besitzen, oder sind wie in Abbildung 6 dargestellt, an den Kanten der Hindernisse und des Arbeitsbereichs ausgerichtet. Anschließend wird für jede Zelle und jede Linie der Mittelpunkt bestimmt und mit Hilfe dieser Punkte ein Graph erstellt. Jeder Knoten im Graphen enthält somit die Zelle zu der er gehört und die Positionsangabe, also den Mittelpunkt der Zelle oder der Linie im realen Raum. Diese Informationen dienen dem Roboter später als Anlaufpunkte und helfen, einen größtmöglichen Abstand zu Hindernissen einzuhalten.

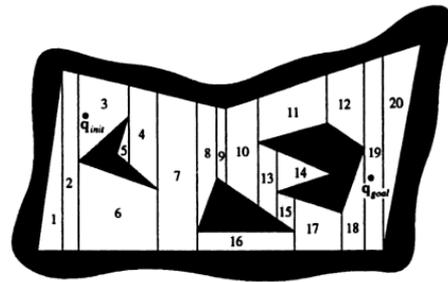


Abbildung 6: Vertical Cell Decomposition
(Quelle: [4])

Damit nun ein Pfad durch den Graphen gefunden werden kann, wird die Zelle, in der sich der Startpunkt befindet als Startknoten des Graphen angenommen und die Zelle, in der sich die Zielkoordinate befindet, als Endknoten. Somit kann nun über einen einfachen Suchalgorithmus der kürzeste Weg zwischen den beiden Knoten gewählt werden [9].

Eine weitere Art der Zellzerlegungsverfahren sind die *Quadtrees* und deren Erweiterung, die *Octrees*. Beide Verfahren laufen in mehreren Schritten ab.

Der Arbeitsbereich des Roboters wird in Vier, bzw. 8, gleich große Zellen unterteilt. Diese Zellen werden anschließend nach den Kategorien *frei*, *teilweise belegt* und *vollständig belegt* bewertet. Im nächsten Schritt werden alle Felder, welche als teilweise belegt markiert wurden, erneut aufgeteilt und die so

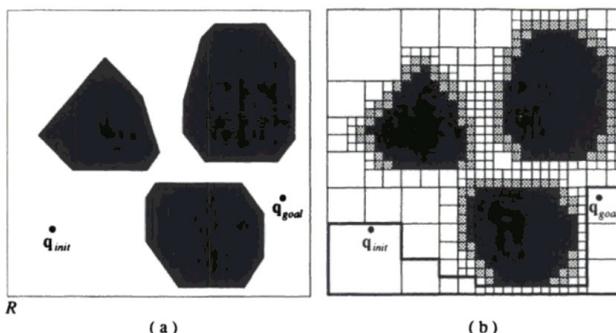


Abbildung 7: Quadtree

entstandenen Felder bewertet. Je öfter die Schritte aufteilen und Bewerten durchgeführt werden, desto genauer feinstufiger werden Hindernisse abgegrenzt. Zum Abschluss wird, z.B. mit Hilfe des A*-Algorithmus, ein Pfad über alle als frei markierten Zellen berechnet (Abbildung 7b) [4],[9].

2.5 Potentialfeldmethode

Anders als bei den bisher beschriebenen Methoden, wird bei der *Potentialfeldmethode* (PFM) nicht der komplette Bewegungspfad des Roboters erstellt. Vielmehr wird nur die nähere Umgebung analysiert und aufbauend darauf der nächste Streckenabschnitt berechnet. Dies erlaubt es, schnell auf dynamische Änderungen in der Umgebung reagieren zu können.

Möglich wird dieses Verfahren durch die Simulation von abstoßenden und anziehenden Kräften. Der Roboter wird hierbei von seinem Ziel angezogen und von Hindernissen abgestoßen. Je mehr er sich einem Hindernis nähert, desto stärker stoßen die Kräfte ihn zurück. Dies stellt sicher, dass es nicht möglich ist, mit einem Hindernis zu kollidieren. In der

Praxis werden die abstoßenden Potentialfelder der Hindernisse und das anziehende Feld des Zieles häufig getrennt berechnet und anschließend zu einem einzigen Kraftvektor verrechnet (Abbildung 8 oben/mitte). Dies bewirkt, dass lokale Minima entstehen können, welche den Roboter vor dem Ziel einfangen würden [7],[10],[11]. Um dieses Problem zu umgehen, werden verschiedene Verfahren angewendet. So ist es möglich, komplexere Felder zu erzeugen, welche keine lokalen Minima enthalten, wie z.B. harmonische Potentialfelder. Eine andere Möglichkeit ist, es die PFM um ein globales Verfahren zu erweitern. Der Roboter kann, wenn er sich in einem lokalen Minimum befindet, auf das globale Verfahren zurückgreifen, um einen alternativen Pfad zu finden. Ein weitere Punkt der bei der Erstellung eines Potentialfeldes beachtet werden muss, ist, dass die Bewegung des Roboters in der Nähe von Hindernissen oder zwischen zwei Hindernissen anfangen kann zu schwingen. Die geschieht, wenn sich das Ziel hinter dem Hindernis befindet. Der Roboter wird von dem Ziel angezogen, gleichzeitig jedoch vom Hindernis abgestoßen (Abbildung 9).

Bei dicht aneinander liegenden Hindernissen, wird er von einem Hindernis zum jeweils anderen gestoßen (vgl. [12]).

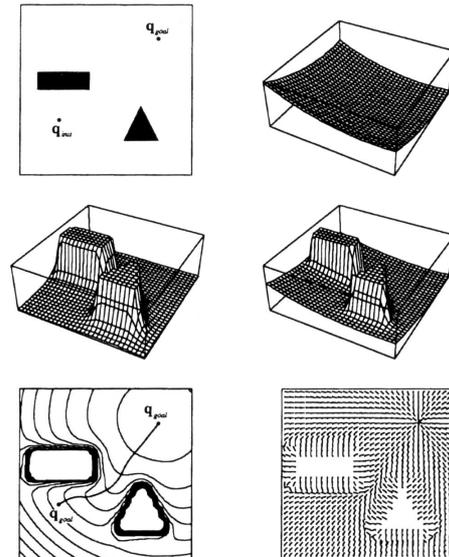


Abbildung 8: Potentialfeldmethode
(Quelle: [9])

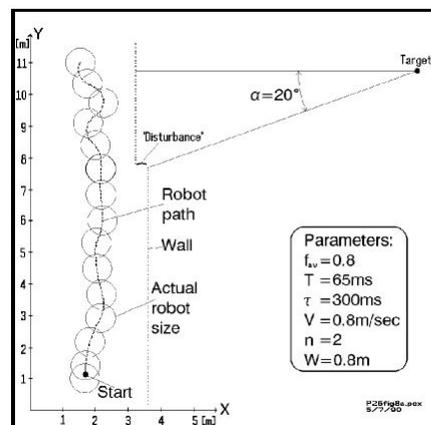


Abbildung 9: Potentialfeldmethode -
Oszillation an Wand
(Quelle: [12])

3 Verwandte Projekte

Bereits 1998 wurde am Fraunhofer Institut für Prozesstechnik und Automatisierung (Fraunhofer IPA) der erste Prototyp eines Haushaltsroboters entwickelt. Der auf die Bezeichnung Care-O-bot[®] I getaufte Roboter bestand aus einer mobilen Basisplattform sowie einem Touchscreen für die Kommunikation mit Menschen. Er war in der Lage, sich sicher in einem von Menschen frequentierten Bereich zu bewegen, was er auf zahlreichen Messen- und Ausstellungen unter Beweis gestellt wurde. 2002 folgte Care-O-bot[®] II. Er war mit einem Greifarm ausgestattet und in der Lage, typische Haushaltsgegenstände wie Teller, Tassen und Flaschen zu greifen. Zur Orientierung und Identifikation der Gegenstände besaß er zusätzlich einen beweglichen Sensorkopf, der zwei Kameras und einen Laserscanner enthielt. Zudem enthielt er die bereits im Care-O-bot[®] I eingesetzten Fähigkeiten zur autonomen Navigation und konnte darüber hinaus von älteren Menschen als intelligente Gehilfe eingesetzt werden. Die aktuelle Weiterentwicklung dieser Serie, der Care-O-bot[®] III, basiert auf einer omnidirektionalen Plattform, durch welche er in der Lage ist, seitwärts zu fahren, wie auch sich auf der Stelle zu drehen. Dies ermöglicht ihm, auch enge Bereiche sicher zu passieren. Sein hochflexibler Greifarm wurde um eine 3-Finger-Hand erweitert, welche sein Arbeitsspektrum noch einmal erweitert. Somit ist er nun in der Lage, seine Greifkraft genau zu dosieren und Alltagsgegenstände nicht nur zu greifen, sondern auch zu bedienen. Auch die Ausstattung an Sensoren wurde erweitert. Mit Hilfe der neu hinzugekommenen Tiefenbildkamera, ist eine noch präzisere Steuerung und Überwachung des Greifarms möglich [1].

Doch nicht nur die Fraunhofer Gesellschaft entwickelt Haushaltsroboter. Auch im Jouhou System Kougaku Laboratory der Universität von Tokio wird an dieser Art von Robotern geforscht. Hier entstanden mittlerweile mehr als sieben verschiedene Arten von Assistenzrobotern. Einige von ihnen sind auf spezielle Aufgaben im Haushalt ausgerichtet, während andere hingegen einen weiten Bereich an Einsatzmöglichkeiten abdecken. So entstand dort u.a. der Kitchen Assistant Robot (KAR). Dieser stationäre Roboterarm ist fähig, eine Spülmaschine eigenständig zu befüllen, zu bedienen und auch wieder auszuräumen. Während KAR fest in die Küche eingebaut wird, nähert sich HRP-2W mehr den Entwicklungen des Fraunhofer IPA an. Der obere Körper dieses mobilen Roboters ähnelt dem eines Menschen und besitzt neben einer beweglichen Sensoreinheit im Kopf, zwei Gelenkarme mit jeweils sieben Freiheitsgraden. Der untere Teil des Körpers besteht aus der Antriebseinheit, welche es ihm ermöglicht, sich sowohl drinnen als auch draußen fortzubewegen. HRP-2W besitzt neben der Fähigkeit, einzelne Gegenstände zu erkennen, auch die Möglichkeit Gesten, und Bewegungen zu erfassen und zu identifizieren [13].

Während die meisten Forschungseinrichtungen die Roboter für spezielle oder allgemeine Aufgaben im Haushalt auslegen, lag der Schwerpunkt des Projekts Armsteuerung der Forschungsgruppe Autonome Robotik der Ruhr-Universität Bochum vielmehr in der Steuerung von Gelenkarmrobotern. Kernstück des Projektes ist der Cooperative Robotic Assistant (Cora), ein autonomer stationärer Gelenkarmroboter. Dieser ist mit einer Stereokamera zur visuellen Navigation, zwei Mikrofonen, mit denen Sprachbefehle entgegen genommen werden sowie einer berührungsempfindlichen künstlichen Haut ausgestattet. Er ist in der Lage Gesten

zu erkennen, sich bewegende Objekte oder Gegenstände von Menschen entgegen zu nehmen oder ihnen diese zu reichen. Möglich wird dies durch ein komplexes Neuronales Netzwerk und die Verwendung von mehreren, miteinander verbunden PCs [14].

4 Fazit

In dieser Arbeit wurden die gängigen Verfahrensweisen zur Steuerung von Industrierobotern, speziell Gelenkarmrobotern, vorgestellt. Da sich ausschließlich die in Kapitel 2 beschriebenen Verfahren für die Steuerung autonomer Roboter eignen, wird der Fokus im weiteren Projektverlauf auf diesen liegen. Welcher Algorithmus sich für den Assistenzroboter der HAW eignen wird, soll sich im nächsten Schritt des Projektes herausstellen. Hierfür ist eine Implementation der vorgestellten Algorithmen in eine Simulationsumgebung geplant. Die Recherche hat gezeigt, dass es bereits vergleichbare Arbeiten und Projekte in dem Bereich der dynamischen Bewegungsplanung gibt. Somit ist es möglich, diese als Anregungen heranzuziehen und auf ihnen aufzubauen.

Abbildungsverzeichnis

| | |
|--|---|
| Abbildung 1: Gelenkarmroboter mit Schweißbrenner..... | 4 |
| Abbildung 2: Care-O-bot® des Fraunhofer IPA beim Greifen einer Dose. . | 4 |
| Abbildung 3: Probabilistic Roadmap Algorithm..... | 6 |
| Abbildung 4: Voronoi Diagramm..... | 7 |
| Abbildung 5: Aufbau eines Rapidly-Exploring Radom Tree..... | 7 |
| Abbildung 6: Vertical Cell Decomposition..... | 8 |
| Abbildung 7: Quadtree..... | 8 |
| Abbildung 8: Potentialfeldmethode..... | 9 |
| Abbildung 9: Potentialfeldmethode - Oszillation an Wand..... | 9 |

Literaturverzeichnis

- [1] Care-O-bot® - Fraunhofer IPA. - URL www.care-o-bot.de/. - Zugriffsdatum: 2010-02-26
- [2] Bernd Jakoby: Industrieroboter, 2001
- [3] Robotic Path Planning and Protein Modeling. - URL cnx.org/content/m11457/latest/. - Zugriffsdatum: 2010-02-26
- [4] Jean-Claude Latombe: Robot motion planning, 1991. - ISBN: 0792391292
- [5] Kavraki, L.E., Svestka, P., Latombe, J.-C., and Overmars, M.H.: Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces, 1996
- [6] Robert Bohlin: Motion Planning for Industrial Robots, Chalmers University of Technology and Göteborg University
- [7] Joachim Schröder: Adaptive Verhaltensentscheidung und Bahnplanung für kognitive Automobile , 2009. - ISBN: 3866444060
- [8] Franz Aurenhammer, Rolf Klein: Voronoi Diagrams, FernUniversität in Hagen
- [9] Marco Loh: Proseminar: Bahnplanung in der Robotik, WS99/00
- [10] Local Path Planning Using Virtual Potential Field. - URL www.cs.mcgill.ca/~hsafad/robotics/index.html. - Zugriffsdatum: 2010-02-26
- [11] Steven M. LaValle: Planning Algorithms, 2006. - ISBN: 0521862051
- [12] Y. Koren, J. Borenstein: Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation, IEEE Conference on Robotics and Automation, 1991
- [13] JSK-Research. - URL www.jsk.t.u-tokyo.ac.jp/research.html. - Zugriffsdatum: 2010-02-26
- [14] Institut für Neuroinformatik [Autonome Robotik]. - URL www.neuroinformatik.rub.de/thbio/group/robotic/project/arm/index_d.html. - Zugriffsdatum: 2010-02-26