Rheinische Friedrich-Wilhelms-Universität Bonn Institut für Informatik I



Torsten Baumgartner

Sensorfreie Orientierung von polygonalen Werkstücken mit dem EduBot 250

31. Januar 2006

Diplomarbeit Betreuer: Prof. Dr. Rolf Klein

Danksagung

An dieser Stelle möchte ich mehreren Personen meinen herzlichen Dank für die Unterstützung während der Bearbeitungszeit dieser Arbeit aussprechen:

Der Arbeitsgruppe Professor Dr. Rolf Klein danke ich sehr herzlich für die Überlassung des überaus interessanten Themas und der Durchführung und Korrektur dieser Diplomarbeit.

Herrn Dr. Elmar Langetepe gebührt Dank für die Betreuung und Unterstützung während der Bearbeitungszeit der vorliegenden Diplomarbeit.

Weiterhin möchte ich Herrn Cemal Ozturk von der Firma *Robotica Limited* und Herrn Zeb Dahl von der Firma *ActivMedia Robotics* danken für die stets hilfreiche Beantwortung von Fragen.

Ein besonderer Dank gilt meinen Eltern, die mir das Studium der Informatik ermöglicht und mich in jeder Hinsicht unterstützt haben. Π

Inhaltsverzeichnis

1	\mathbf{Ein}	leitung	1
	1.1	Parts-Feeder in der industriellen Fertigung	1
	1.2	Compliant Motion Planning	3
	1.3	Ziel der Arbeit	4
	1.4	Gliederung	5
2	Gru	undlagen	7
	2.1	Orientierung polygonaler Werkstücke	7
		2.1.1 Der Aktionsplan	8
		2.1.2 Annahmen	9
		2.1.3 Durchmesser- und Greiffunktion eines Polygons	10
		2.1.4 Orientierung bis auf Symmetrie	13
		2.1.5 Berechnung von Aktionsplänen	14
		2.1.6 Radius- und Schiebefunktion eines Polygons	16
	2.2	Gelenkarmroboter	20
		2.2.1 Aufbau, Teilsysteme und Terminologie	21
		2.2.2 Relative Lage von Objekten	22
		2.2.3 Vorwärtskinematik	25
		2.2.4 Inverse Kinematik	28
	2.3	Der EduBot 250M	29
		2.3.1 Eigenschaften und Aufbau	30
		2.3.2 Steuerung	32
3	Kor	nzept	35
	3.1	Annahmen	35
	3.2	Werkebene	36
	3.3	Reibungsarmer Greifer	37
	3.4	Translationsdistanz	39
	3.5	Translationsgeschwindigkeit	43
	3.6	Ein Bewegungsablauf für den EduBot	44
	3.7	Details	49
	38	Zusammenfassung	53

INHALTSVERZEICHNIS

4	Um	setzung 5	55
	4.1	Technische Konfiguration	55
	4.2	Machbarkeitstest	58
	4.3	Materialien	30
	4.4	Architektur der Software	31
		4.4.1 Das Paket jEduLib	32
		4.4.2 Das Paket appsSwingGui.JawGripper	52
	4.5	Implementierung der Steuerungssoftware	33
		4.5.1 Eigenschaften der Software	34
		4.5.2 Vorwärtskinematik	36
		4.5.3 Rückwärtskinematik	38
		4.5.4 Verfolgung linearer Bahnen	71
		4.5.5 Ergebnis	72
	4.6	Implementierung der Steuerungskomponente	72
	4.7	Tests	74
		4.7.1 Vorbereitungen \ldots	75
		4.7.2 Ergebnisse	76
		4.7.3 Bewertung	78
	4.8	Zusammenfassung	79
5	Rob	oustheit von Aktionsplänen 8	33
-	5.1	Die Bewegung eines geschobenen Werkstücks	33
	5.2	Implikationen von Unsicherheit für Goldbergs Verfahren 8	34
	5.3	Anpassung von Goldbergs Ansatz	36
		5.3.1 Rotationsrichtung des Werkstücks	36
		5.3.2 Stabilität von Werkstückkanten	39
		5.3.3 Implementierung eines modifizierten Algorithmus	91
	5.4	Tests	95
		5.4.1 Ergebnisse	95
	5.5	Zusammenfassung	97
~			
6	Fazi	it und Ausblick 9)9
\mathbf{A}	Rob	ootersteuerungssoftware jEduLib 10)1
	A.1	Funktionen)1
	A.2	Installation)2
	A.3	Schnittstelle zu der Robotersteuerung 10)2
	A.4	Grafische Bedienoberfläche)4
в	Java	a-Anwendung zur Berechnung von Aktionsplänen 10)7
	B.1	Installation)7
	B.2	Bedienung)8
		5	-

IV

Abbildungsverzeichnis

1.1	Ein Parts-Feeder orientiert einen Strom von Werkstücken $\ .$.	2
2.1	Ein Parallel-Jaw Gripper	8
2.2	Drei Durchläufe eines vierstufigen Aktionsplans	9
2.3	Ein Greifvorgang des Jaw Grippers	10
2.4	Der Durchmesser eines Polygons	11
2.5	Durchmesser- und Greiffunktion	12
2.6	Die Abarbeitung eines Aktionsplanes mit 3 Greifaktionen	15
2.7	Realer Greifvorgang des Jaw Grippers	17
2.8	Der Radius eines Polygons	19
2.9	Radius- und Schiebefunktion	20
2.10	(i) Ein Drehgelenk, (ii) ein Schubgelenk	23
2.11	Ein Gelenkarmroboter mit mehreren Koordinatensystemen	24
2.12	Zwei Denavit-Hartenberg Parameter	26
2.13	Zwei weitere Denavit-Hartenberg Parameter	27
2.14	Es existieren mehrere Lösungen der inversen Kinematik	30
2.15	Der EduBot 250M	31
3.1	Podest für den EduBot und tieferliegende Werkebene	37
3.2	Der EduBot greift ein Werkstück auf der Werkebene	37
3.3	Die linearen Backen eines reibungsarmen Greifers	38
3.4	Die Laufschiene minimiert auftretende Reibung	39
3.5	Eine lineare Backe schiebt ein polygonales Werkstück	40
3.6	Variable versus feste Translationsrichtung des Greifers, Teil 1	45
3.7	Variable versus feste Translationsrichtung des Greifers, Teil 2	46
3.8	Die relative Position eines Werkstücks während einer Trans-	
	lation	47
3.9	Werkstücke liegen innerhalb eines kreisförmigen Bereichs	47
3.10	Es existiert keine einheitliche Startposition für die Translation	48
3.11	Bewegungsfreiheit des Greifers	50
3.12	Die Wahl einer Translationsbahn	51
4 1	Des Defablanneteltell des Controllors	57
4.1	Vanformation our Stauenum der Elepte	٥ <i>١</i> ٢٥
4.2	Konnguration zur Steuerung des EduBots	99

4.3	Ein Adapterkabel für den Anschluss des Controllers an einen	
	Computer	59
4.4	Der EduBot wurde auf einen Sockel montiert	60
4.5	Links: zwei Finger des Greifers wurden durch Holzelemente	
	ersetzt. Rechts: der modifizierte Greifer	61
4.6	Eine Laufschiene für den Greifer	61
4.7	Die einzelnen Software-Pakete und ihre Abhängigkeiten	62
4.8	Klassendiagramm des Pakets jEduLib	63
4.9	Klassendiagramm des Pakets appsSwingGui.JawGripper	64
4.10	Zeitlicher Ablauf der Ansteuerung	66
4.11	Koordinatensysteme des EduBots	67
4.12	Ermittelte Denavit-Hartenberg Parameter des EduBots	68
4.13	Berechnung neuer Gelenkpositionswerte	70
4.14	Eine lokal optimale Lösung der inversen Kinematik	71
4.15	Die verwendeten Werkstücke	75
4.16	Geometriedaten der verwendeten Werkstücke	75
4.17	Testresultate	77
4.18	Instabile Orientierungen während des Translationsvorgangs .	78
4.19	Falsche Rotationsrichtung während des Translationsvorgangs	78
4.20	Der EduBot in Aktion	81
5.1	Ein Reibungskegel	84
5.2	Zwei Reibungskegel klassifizieren eine Kante als stabil	85
5.3	Eine kritische Orientierung des Werkstücks	87
5.4	Der Orientierungsvorgang im Detail	88
5.5	Instabile Orientierungen während des Translationsvorgangs .	89
5.6	Die Stabilität zweier lokaler Minima in einem t-Intervall wird	
	anhand von Abstandskriterien bewertet	91
5.7	Eine Lösung wird mittels Backtracking ermittelt	92
5.8	Testresultate der zweiten Testserie	97
A.1	Das Hauptfenster der grafischen Oberfläche	105
A.2	Dialog für die Eingabe von Gelenkpositionswerten	106
B.1	Das Hauptfenster der Anwendung	108
B.1 B.2	Das Hauptfenster der Anwendung	$\begin{array}{c} 108 \\ 109 \end{array}$
B.1 B.2 B.3	Das Hauptfenster der Anwendung	108 109 110

Kapitel 1

Einleitung

1.1 Parts-Feeder in der industriellen Fertigung

Roboter werden in vielfältigen Aufgabengebieten eingesetzt. Sie kommen z. B. dann zum Einsatz, wenn die Durchführung von Aufgaben für den Menschen unter den herrschenden Bedingungen gefährlich oder schwierig ist. Roboter unterstützen betagte Menschen in Situationen des alltäglichen Lebens, die diese Menschen aufgrund ihres hohen Alters nicht mehr selbstständig durchführen können. Roboter werden aufgrund von hohen Anforderungen an die Präzision bei Eingriffen in der Medizin eingesetzt. Mobile Serviceroboter erledigen alltägliche Arbeiten wie Staubsaugen oder Rasenmähen. Häufig werden Roboter mit Aufgaben betraut, die sich aufgrund von monoton immer wieder kehrenden Bewegungsabläufen gut dazu eignen, automatisiert zu werden.

Insbesondere die industrielle Serienfertigung erfüllt aufgrund hoher produzierter Stückzahlen in der Produktion die Voraussetzungen für den wirtschaftlich sinnvollen Betrieb von Robotern. Eine Produktionslinie besteht hier im Allgemeinen aus mehreren Stationen, welche von einem Strom von Werkstücken auf einem Transportband sequentiell durchlaufen werden. Mittels der Aufteilung der Arbeit auf die einzelnen Stationen wird ein komplexer Verarbeitungsprozess in mehrere Teilabläufe organisiert. Sogenannte Industrieroboter erfüllen an den Stationen u. a. Teilaufgaben wie Schweißen, Schneiden, Fräsen oder Lackieren. Trotz der Popularität von Industrierobotern ist nicht genau definiert, welche Merkmale einen Roboter als Industrieroboter klassifizieren. Die Möglichkeit der flexiblen Programmierung für ein weites Feld von Aufgaben wird als wesentliches Merkmal eines Industrieroboters angesehen (vgl. Craig [10]). Ist ein Industrieroboter einmal programmiert, führt er den programmierten Arbeitsablauf autonom (und je nach Ausstattung mit Sensoren mehr oder weniger statisch) aus.

Oftmals erwarten Industrieroboter die auf dem Laufband ankommenden Werkstücke in einer bestimmten Position und Orientierung. Zu diesem Zweck bringen sogenannte *Parts-Feeder* lose zugeführte Werkstücke für die nachfolgende Arbeitsstation in die gewünschte Zielorientierung. Die Orientierung von Werkstücken durch Parts-Feeder kann entweder auf eine passive oder auf eine aktive Weise geschehen (vgl. Singer [32]). Passive Parts-Feeder bestehen aus Filtern. Filter sind mechanische Bauteile, welche Werkstücke mit "falschen" Orientierungen ausmustern, sodass nur Werkstücke mit der gewünschten Orientierung an der weiteren Verarbeitung teilnehmen. Aktive Parts-Feeder üben dagegen einen aktiven mechanischen Einfluss auf die Werkstücke auf dem Laufband aus, um diese in die Zielorientierung zu bringen.



Abbildung 1.1: Ein Parts-Feeder orientiert einen Strom von Werkstücken

Ein verbreiteter Parts-Feeder ist der Vibratory Bowl Feeder. Diese Vorrichtung nutzt mechanische Vibrationen. Die Werkstücke befinden sich in einer schalenförmigen Vorrichtung und werden aufgrund der Vibrationen entlang eines Tracks in Form einer Helix befördert (vgl. Böhringer et. al. [4]). Die spezielle Beschaffenheit des Tracks mit Schikanen und Aussparungen wirkt wie ein mechanischer Filter. Werkstücke mit der falschen Orientierung fallen auf ihrem Weg zum Ausgang wieder in die Vorrichtung zurück. Am Ausgang kommen lediglich Werkstücke in der gewünschten Orientierung an. Vibratory Bowl Feeder werden speziell für ein Werkstück auf manuelle Weise entworfen. Das Design des Tracks bedarf der Intuition des Entwicklers. Zusätzlich sind Funktionstests und Verbesserungen nötig, bis der Parts-Feeder seine Aufgabe korrekt erfüllt. Ansätze für automatisiertes Track-Design entwickelten u. a. Berretty et. al. [3], Wolfsteiner und Pfeiffer [33] oder Berkowitz und Canny [2].

In der industriellen Massenfertigung ist der Einsatz von auf die Verarbeitung einzelner Produkte optimierten Anlagen gängig. Wechselt die Art der zu bearbeitenden Werkstücke, so ist oft auch die Anschaffung neuer Hardware nötig. Dies gilt u. a. auch für Parts-Feeder. Im Gegensatz dazu stehen *reprogrammierbare Parts-Feeder*. Diese können für verschiedene Arten von Werkstücken konfiguriert werden. Wird eine Produktion häufig auf die Fertigung unterschiedlicher Produkte umgestellt, können durch den Einsatz von reprogrammierbaren Parts-Feedern Anschaffungskosten teilweise auf mehrere Produkte umgelegt werden. Dadurch entstehen eventuell Kostenvorteile gegenüber *dedizierten Parts-Feedern*. Diese Vorteile verringern sich tendenziell mit der Erhöhung der produzierten Stückzahl. In der Massenfertigung kommen daher reprogrammierbare Parts-Feeder kaum zum Einsatz.

1.2 Compliant Motion Planning

Die Planung von Bewegungen und deren Ausführung durch Roboter unterliegt meist Unsicherheit. So besteht laut Erdmann und Mason [12] in der Roboterbewegungsplanung üblicherweise Unsicherheit bezüglich der relativen Lagen von Objekten zueinander. Ursachen können Abweichungen in der Steuerung und Sensorik des Roboters sein, aber auch unzutreffende Modellannahmen. Laut Peshkin [26] besteht ebenso Unsicherheit bezüglich der Maße der bearbeiteten Werkstücke, da Ungenauigkeiten während der Produktion der Werkstücke Abweichungen von den vorgesehenen Maßen ergeben können. Die Roboterbewegungsplanung muss Verfahren entwickeln, welche robust genug sind, um ihre Zielsetzung in der Praxis trotz der Existenz von Unsicherheit zu erreichen. Zwei mögliche Ansätze dazu sind die Folgenden:

- **Bewegungsstrategien** verwenden Modelle aus der Bewegungsmechanik. Durch gezielte Ausnutzung von mechanischen Gesetzen wird Unsicherheit reduziert.
- Sensor-basierte Ansätze benutzen Sensoren, um Daten betreffend der relativen Lage von Objekten zueinander zu erhalten.

Im Kontext von Parts-Feedern in der industriellen Fertigung besteht u.a. Unsicherheit über die Position und Orientierung von Werkstücken auf dem Fließband – vor, während und auch nach ihrer Orientierung. Der Einsatz von Sensoren und die korrekte Interpretation der Sensordaten vermindern die Unsicherheit und ermöglichen dem Parts-Feeder situationsgerechte Reaktionen.

Der Einsatz von Sensoren ist aus Kostengründen nicht immer realisierbar. Als Alternative können geeignete Bewegungsstrategien verwendet werden. Beispielsweise lassen sich die bei einem mechanischen Kontakt zweier Objekte auftretenden Kräfte gezielt für die Verminderung von Unsicherheit nutzen. Dieser Ansatz wird in der Fachliteratur als "*Compliant motion planning*" bezeichnet (z. B. Goldberg [15]). Der Ansatz setzt eine genaue Analyse der Aufgabe im Vorfeld voraus.

Kevin M. Lynch [18] betrachtet beispielsweise verschiedene Möglichkeiten, Werkstücke mit einem Roboter zu einem Zielort zu bewegen. Üblicherweise wird dafür der Roboter mit einem Greifer ausgestattet, um das Werkstück zu greifen und zum Zielort zu transportieren. Im Falle eines sehr schweren oder sehr großen Werkstücks ist dieser Ansatz nicht praktikabel. Die Möglichkeiten des Roboters werden erweitert, wenn Bewegungsstrategien eingesetzt werden, welche zusätzliche Arten von Kontrollkräften berücksichtigen. Zu diesem Zweck eignen sich z. B. Gravitation, Reibung, oder Objekte in der Umgebung des Roboters (z. B. die Werkebene, Wände, Hindernisse). So kann das Objekt möglicherweise unter Ausnutzung von Gravitation über die Werkebene zum Zielort geschoben werden. Über die Bedeutung von Modellen aus der Mechanik für die Roboterbewegungsplanung schreibt Kevin M. Lynch:

"A good model of the mechanics of a task is a resource for the robot system, just as actuators, sensors and computers are resources."

1.3 Ziel der Arbeit

Man stelle sich folgende Situation vor: auf einem Fließband einer industriellen Produktionslinie wird ein Strom von polygonalen Werkstücken transportiert. Für die maschinelle Verarbeitung wird eine bestimmte Orientierung der Werkstücke auf dem Fließband vorausgesetzt. Da die Werkstücke beliebig auf dem Fließband liegen, soll ein Parts-Feeders die gewünschte Orientierung der Werkstücke herstellen. Der Parts-Feeder soll ohne Sensorik auskommen und für die Bearbeitung von unterschiedlichen Typen von polygonalen Werkstücken programmierbar sein.

Kenneth Y. Goldberg [15] beschreibt einen Ansatz, welcher die Grundlage für einen derartigen, reprogrammierbaren Parts-Feeder bildet. Der Ansatz nutzt Ideen des Compliant Motion Planning, um die sensorfreie Orientierung polygonaler Werkstücke zu ermöglichen. Zu einem Werkstück wird in einem Vorverarbeitungsschritt einmalig ein Aktionsplan berechnet. Die Ausführung des Aktionsplans durch einen Greifer mit zwei parallelen Backen bewerkstelligt die Orientierung polygonaler Werkstücke des gegebenen Typs unabhängig von der anfänglichen Orientierung der Werkstücke bis auf Symmetrie.

Das Ziel dieser Arbeit ist die praktische Umsetzung dieses Verfahrens. Für die Umsetzung steht ein Gelenkarmroboter der Baureihe *EduBot 250M* des Herstellers *Robotica Limited* [24] zur Verfügung. Die Aufgabenstellung umfasst im Einzelnen die Wahl einer technischen Konfiguration für die Ansteuerung des Gelenkarmroboters. Weiterhin ist der Roboter zu befähigen, mittels der Ausführung von Aktionsplänen polygonale Werkstücke gemäß der Methode von Goldberg zu orientieren. Für die Berechnung von Aktionsplänen für den Greifer kann eine bereits bestehende Software verwendet werden.

1.4 Gliederung

Die vorliegende Arbeit ist wie folgt gegliedert:

Kapitel 1 enthält eine Einleitung in das Themengebiet.

- Kapitel 2 gibt eine Einführung in die Orientierung polygonaler Werkstücke nach Kenneth Y. Goldberg. Weiterhin werden für diese Arbeit relevante Grundlagen betreffend Gelenkarmrobotern behandelt. Anschließend wird der Gelenkarmroboter EduBot 250M vorgestellt.
- **Kapitel 3** beschreibt ein allgemeines Konzept, welches im Rahmen dieser Arbeit für die Realisierung von Goldbergs Methode mit einem Edu-Bot 250M erarbeitet wurde.
- Kapitel 4 beschreibt die Umsetzung des Konzepts aus Kapitel 3. Die Umsetzung umfasst die Wahl einer technischen Plattform für die Ansteuerung des Roboters, das Design der zu entwickelnden Software, die Entwicklung einer allgemein verwendbaren Steuerungssoftware für den EduBot 250M, sowie die Implementierung einer auf der Steuerungssoftware basierenden Steuerungskomponente für die Orientierung von polygonalen Werkstücken. Abschließend werden anhand von Tests Erkenntnisse über die Wirksamkeit der Umsetzung erlangt und vorhandene Fehlerquellen analysiert.
- Kapitel 5 beleuchtet die Robustheit von Goldbergs Methode gegenüber einer fehlerbehafteten Robotersteuerung und gegenüber Abweichungen von Werkstücksmaßen. Es werden Verbesserungsvorschläge erarbeitet, welche anhand von erneuten Tests auf ihre Wirksamkeit hin empirisch überprüft werden.

Kapitel 6 zieht abschließend ein Fazit der Arbeit.

Kapitel 2

Grundlagen

Der erste Abschnitt des Kapitels gibt eine Einführung in die sensorfreie Orientierung polygonaler Werkstücke nach Kenneth Y. Goldberg [15]. Im Hinblick auf Kapitel 5 wird das Verfahren detailliert erläutert. Im zweiten Abschnitt werden einige für die Arbeit wichtige Aspekte von Gelenkarmroboter behandelt. Der letzte Abschnitt stellt den Roboter EduBot 250M vor.

2.1 Orientierung polygonaler Werkstücke

John Canny und Kenneth Y. Goldberg haben in [7, 8] eine Klasse von sogenannten RISC-Robotern beschrieben. RISC, eine Abkürzung für *Reduced Intricacy in Sensing and Control*, zeichnet sich durch die Verwendung und geschickte Kombination simpler Hardwarekomponenten und effizienter Steuerungssoftware in der industriellen Fertigung aus. Nach Meinung der Autoren ergeben sich im Vergleich zu Allzweck-Robotern durch die Anwendung ihrer Prinzipien eine hohe Verlässlichkeit, Kosteneffizienz und eine einfachere Handhabung der Produktionsanlagen. Ein weiterer Vorteil von RISC-Systemen liegt in ihrer Anpassungsfähigkeit. Sie sind rekonfigurierbar, um neue Typen von industriellen Werkstücken bearbeiten zu können.

Als Sensoren für RISC-Systeme eignen sich aufgrund ihrer Einfachheit Beam-Sensoren¹. Für die Manipulation von Gegenständen wird der *Parallel-Jaw Gripper* vorgeschlagen. Dieser einfach konzipierte Greifer besitzt zwei parallele, lineare Backen. Abbildung 2.1 zeigt einen Parallel-Jaw Gripper.

Ein mögliches Anwendungsfeld von RISC-Systemen ist im Bereich des Parts-Feeding zu sehen. Dabei geht man von einem Strom von Werkstücken auf einem Transportband aus. Die Werkstücke sollen für nachfolgende Maschinen in eine bestimmte Zielorientierung gebracht werden.

 $^{^1\}mathrm{Ein}$ Beam-Sensor besteht le diglich aus einer Lichtquelle und einem binärem Lichtsensor.



Abbildung 2.1: Ein Parallel-Jaw Gripper

Dieses Kapitel widmet sich einem RISC-Ansatz für diese Aufgabenstellung. Ist die Beschreibung eines polygonalen² Werkstücks gegeben, berechnet der Ansatz einen entsprechenden Aktionsplan. Die Anwendung des Aktionsplans durch einen Parallel-Jaw Gripper orientiert das Werkstück unabhängig von seiner anfänglichen Orientierung bis auf Symmetrie. Der Ansatz kommt ohne den Einsatz von Sensoren aus und dient als Basis für einen rekonfigurierbaren Parts-Feeder. In den weiteren Abschnitten wird dieser Ansatz im Detail betrachtet.

2.1.1 Der Aktionsplan

Der Aktionsplan besteht aus einer Folge von Winkeln. Jeder einzelne Winkel entspricht der Ausführung einer Greifaktion durch den Jaw Gripper. Die Greifaktion wird mit Winkel α parametrisiert und sieht wie folgt aus:

- Drehe den Greifer auf den Winkel α bezüglich einer festen Normalstellung³.
- Schließe beide Backen des Greifers gleichzeitig, bis das Werkstück eingeklemmt ist.
- Öffne beide Backen des Greifers.

Der Aktionsplan ist grundsätzlich nur einmal in einem Vorverarbeitungsschritt zu berechnen. Danach können durch Anwendung des Aktionsplans beliebig viele Werkstücke des gleichen Typs orientiert werden. Daher kann der Plan als *universell* für das Werkstück bezeichnet werden. Die Konfiguration auf einen neuen Werkstückstyp umfasst lediglich die Berechnung eines neuen Aktionsplans.

 $^{^{2}}$ Ein planares Werkstück heißt *polygonal*, falls seine konvexe Hülle polygonale Gestalt hat. Mit einer Abwandlung des folgenden Verfahrens kann auch die allgemeinere Klasse der *algebraischen* Werkstücke sensorfrei orientiert werden (vgl. Rao und Goldberg, [31]).

 $^{^{3}\}mathrm{Ein}$ positiver Winkel im mathematischen Sinn entspricht einer Drehung gegen den Uhrzeigersinn.

2.1. ORIENTIERUNG POLYGONALER WERKSTÜCKE

Abbildung 2.2 zeigt die Ausführung eines vierstufigen Aktionsplans. Der Plan besteht aus den Winkeln 0, -42.73, 76.44 und 17.90 Grad. Die Backen des Greifers sind als schwarze Linien dargestellt. Eine Zeile in der Abbildung entspricht einem Durchgang des Plans. Die linke Spalte in der Abbildung entspricht der Situation vor, die rechte Spalte der Situation nach Ausführung des Plans. Das fünfeckige Werkstück befindet sich nach allen drei Durchläufen in der gleichen Orientierung.



Abbildung 2.2: Drei Durchläufe eines vierstufigen Aktionsplans.

2.1.2 Annahmen

Der Ansatz stützt sich auf verschiedene Annahmen. Diese werden im Folgenden genannt.

- **Der Greifer** Der Greifer besteht aus zwei linearen Backen, welche parallel zueinander stehen. Während des Schließvorgangs bewegen sich die Backen des Greifers orthogonal zu ihrer Orientierung aufeinander zu.
- **Die Werkstücke** Es wird die Klasse der polygonalen Werkstücke betrachtet. Die konvexe Hülle eines Werkstücks kann als festes planares Polygon betrachtet werden. Die Werkstücke werden als isoliert voneinander angesehen. Es geht nicht darum, aus einer Menge von Werkstücken auf dem Transportband ein einzelnes Stück zu isolieren.
- Das physikalische Modell Zwischen den Backen des Greifers und dem zu greifenden Werkstück existiert keine Reibung. Alle Bewegungen finden in der Ebene statt und sind derart langsam, dass Trägheitseffekte der Masse keine Rolle spielen. Dies entspricht einem *quasi-statischen Modell*, welches dynamische Effekte vernachlässigt (vgl. Lynch [18]).

Der Bewegungsablauf Das Werkstück liegt zu Anfang der Greifaktion zwischen den Backen des Greifers (siehe Abbildung 2.3(i)). Über die initiale Orientierung und Position des Werkstücks liegen keine weiteren Informationen vor. Während der Greifaktion verbleibt das Werkstück stetig zwischen den Backen. Beide Backen stellen gleichzeitig einen Kontakt zum Werkstück her (siehe Abbildung 2.3(ii)). Nachdem der Kontakt zwischen Werkstück und Greifer hergestellt wurde, bleiben die beteiligten Oberflächen bis zum Ende der Greifaktion in Kontakt. Die Schließbewegung des Greifers ist beendet, wenn eine Fortführung das Werkstück verformen würde (siehe Abbildung 2.3(iii)).



Abbildung 2.3: (i) Das Werkstück liegt zwischen den Backen, (ii) Ein Kontakt zwischen Greifer und Werkstück entsteht, (iii) der Greifvorgang ist beendet

Einige der genannten Annahmen sind vereinfachend und schließen real existierende Effekte wie z. B. die Reibung zwischen dem Werkstück und den Backen des Greifers aus. Für eine praktische Umsetzung ist interessant, wie diesen Annahmen möglichst in idealer Weise nachgekommen werden kann. Kapitel 3 beschäftigt sich genauer mit den Annahmen im Kontext einer praktischen Umsetzung.

2.1.3 Durchmesser- und Greiffunktion eines Polygons

Es sei ein polygonales Werkstück gegeben. Für die Berechnung von Aktionsplänen muss der Einfluss einer Greifaktion des Greifers auf das Werkstück bekannt sein. Für die Greifaktion gelten die Annahmen aus dem vorigen Abschnitt. Gesucht ist eine passende *Greiffunktion* $s : [0, 2\pi) \rightarrow [0, 2\pi)$, für die gilt: ist θ die Orientierung des Greifers in Bezug auf das Werkstück vor dem Schließen des Greifers, so ist $s(\theta)$ die Orientierung des Greifers in Bezug auf das Werkstück nach dem Greifvorgang. Für die Berechnung der Greiffunktion werden die folgenden Definitionen benötigt (vgl. Kamphans und Klein [17]).

Definition 1 Gegeben sei ein konvexes, planares Polygon P in der Ebene. Die Abbildung $h_P(\alpha) : [0, 2\pi) \longmapsto \mathbb{R}$ sei wie folgt definiert:

$$h_P(\alpha) := \sup \left\{ x \cos \alpha + y \sin \alpha \mid (x, y) \in P \right\}, \quad \alpha \in [0, 2\pi)$$

10

Definition 2 (Durchmesserfunktion eines Polygons) Gegeben sei ein konvexes, planares Polygon P in der Ebene und $\alpha \in [0, 2\pi)$. Die Funktion dia_P : $[0, 2\pi) \rightarrow \mathbb{R}$ heisst Durchmesserfunktion von P. Sie ermittelt den Durchmesser von P bei Winkel α und ist wie folgt definiert:

$$dia_P(\alpha) := h_P(\alpha) + h_P(\alpha + \pi), \quad \alpha \in [0, 2\pi)$$

Anschaulich kann man sich zwei parallele Geraden mit Winkel α vorstellen, welche das Polygon von der jeweils gegenüberliegenden Seite tangieren. Der Durchmesser eines Polygons ist dann gegeben durch die Distanz der beiden Geraden zueinander. Aus der Definition lässt sich leicht ersehen, dass die Durchmesserfunktion eines Polygons eine Periode von π besitzt. Weiterhin ist die Funktion stetig und besteht stückweise aus Sinuskurven (Rao und Goldberg [30]).



Abbildung 2.4: Der Durchmesser eines Polygons

Hinweis Im Folgenden soll einem polygonalen, planaren Werkstück eine Durchmesserfunktion zugeordnet werden. Es wird von der Durchmesserfunktion des Werkstücks gesprochen, wenn eigentlich die Durchmesserfunktion der konvexen Hülle des Werkstücks gemeint ist. Ist im Kontext klar, um welche Durchmesserfunktion es sich handelt, so wird der Zusatz "des Werkstücks" weggelassen.

Die Greiffunktion eines Werkstücks lässt sich aus der Durchmesserfunktion des Werkstücks ableiten. Sei (θ_0, θ_1) ein Paar von Winkeln, welches zu zwei benachbarten lokalen Maxima der Durchmesserfunktion korrespondiert. Alle Orientierungen zwischen θ_0 und θ_1 werden von der Greiffunktion *s* abgebildet auf die Orientierung, welche zu dem lokalen Minimum der Durchmesserfunktion zwischen θ_0 und θ_1 korrespondiert. Die Greiffunktion eines Werkstückes ist eine stückweise konstante Treppenfunktion. Sie besitzt Diskontinuitätsstellen an den Werten, die den lokalen Maxima der entsprechenden Durchmesserfunktion entsprechen. Aufgrund der Periodizität der Durchmesserfunktion besitzt jede Greiffunktion ebenfalls eine Periode von π . Ein Beispiel für eine Durchmesserfunktion und die daraus abgeleitete Greiffunktion ist in Abbildung 2.5 dargestellt.



Abbildung 2.5: Die Durchmesserfunktion des L-förmigen Polygons aus Abbildung 2.4, darunter die abgeleitete Greiffunktion.

Zum besseren Verständnis sei auf Folgendes hingewiesen: Orientierungen eines Werkstücks werden relativ zu zwei verschiedenen Bezugssystemen angegeben.

- 1. Die Orientierung des Werkstücks kann bezüglich eines globalen Bezugssystems angegeben werden. Dieses ist im Verlauf des Verfahrens unveränderlich. Die Werkstücke sind durch den Parts-Feeder auf dieses Bezugssystem bezogen in die gleiche Orientierung zu bringen.
- 2. Die Durchmesser- und die Greiffunktion bilden Orientierungen des Werkstücks bezüglich des Greifers auf neue Orientierungen des Werkstücks in Bezug auf den Greifer ab. In diesem Falle wird die Orientierung des Werkstücks bezüglich der Orientierung des Greifers verwendet.

Definition 3 (s-Intervall) Sei s die Greiffunktion eines Werkstücks. Ein s-Intervall ist ein halboffenes Intervall auf \mathbb{R} der Form [a, b), sodass a und b Diskontinuitäten im Wertebereich von s sind.

Definition 4 (s-Image) Set s die Greiffunktion eines Werkstücks. Set Θ ein halboffenes Intervall auf \mathbb{R} der Form [a, b) mit $0 \le a < b < 2\pi$. Das s-Image $s(\Theta)$ von Θ ist das kleinste Intervall, welches die Menge

$$\{s(\theta) \mid \theta \in \Theta\}$$

vollständig enthält.

Sei n die Anzahl der Ecken eines polygonalen Werkstücks. Dann besitzt die Greiffunktion des Werkstücks O(n) viele Diskontinuitätsstellen und es existieren $O(n^2)$ verschiedene s-Intervalle.

2.1.4 Orientierung bis auf Symmetrie

Bevor nun mit Hilfe der Greiffunktion die Berechnung eines Aktionsplans erläutert wird, erklärt dieser Abschnitt kurz, was es heißt, ein Werkstück "bis auf Symmetrie" zu orientieren.

Sprechweise 1 Sei s die Greiffunktion eines polygonalen Werkstücks. Die Greiffunktion besitzt eine Periode von T, wenn für alle $\theta \in [0, 2\pi)$ gilt:

$$s(\theta + T) = s(\theta) + T \tag{2.1}$$

Sei s nun eine Greiffunktion mit Periode T. Jeder Aktionsplan basierend auf der Greiffunktion s, welcher Werkstücke mit der Orientierung θ in die Orientierung θ überführt, wird Werkstücke mit der Orientierung $\theta+T$ in die Orientierung $\theta'+T$ bringen. Es kann also kein Plan existieren, welcher zwei Werkstücke mit den Orientierungen θ und $\theta+T$ in eine gemeinsame Endorientierung ausrichtet. Eine Periode in der Greiffunktion s bedeutet daher eine Mehrdeutigkeit in der Endorientierung des Werkstücks nach Ausführung eines Aktionsplans.

Wie entsteht Periodizität in der Greiffunktion eines Werkstücks? Die Greiffunktion wird aus der Durchmesserfunktion des Werkstücks abgeleitet. Die Durchmesserfunktion impliziert dabei Struktur in die Greiffunktion. Eine Periode in der Durchmesserfunktion überträgt sich in die Greiffunktion. Besitzt die konvexe Hülle eines Werkstücks eine r-fache Rotationssymmetrie, so hat die korrespondierende Greiffunktion die Periode

$$T_r = \frac{2\pi}{r \left(1 + r \mod 2\right)}, \quad r > 0.$$
(2.2)

Aus Gleichung 2.2 folgt für die Periode ${\cal T}_r$ der Greiffunktion eines polygonalen Werkstücks

$$T_r \le \pi, \quad r > 0. \tag{2.3}$$

Es existieren folglich aufgrund von Symmetrie in der Greiffunktion immer mindestens zwei Endorientierungen für ein polygonales Werkstück. **Sprechweise 2** Sei T die kleinste Periode in der Greiffunktion eines Werkstücks. Ein Aktionsplan orientiert das Werkstück bis auf Symmetrie, wenn die Menge der möglichen Endorientierungen des Werkstücks genau aus $\frac{2\pi}{T}$ Orientierungen besteht, welche gleichmäßig verteilt im Intervall $[0, 2\pi)$ liegen.

Das Verfahren basierend auf der Greiffunktion eines Werkstücks orientiert polygonale Werkstücke bis auf Symmetrie. Werkstücke mit einer vierfachen Rotationssymmetrie beispielsweise besitzen vier mögliche Endorientierungen. Diese haben einen Abstand von $\frac{\pi}{2}$ zueinander. Die Greiffunktion eines unsymmetrischen Werkstücks (r = 1) besitzt eine Periode von π . Dies bedeutet, dass sich für unsymmetrische Werkstücke immer zwei mögliche Endorientierungen ergeben.

2.1.5 Berechnung von Aktionsplänen

Die Generierung eines Aktionsplans beruht auf Algorithmus 2.1.1. Die Eingabe des Algorithmus besteht aus der konvexen Hülle eines polygonalen Werkstücks. Im ersten Schritt wird die Greiffunktion *s* berechnet. Anschließend wird eine Folge von größer werdenden s-Intervallen berechnet. Die while-Schleife terminiert, wenn $|\Theta_i| = T$ gilt, wobei *T* eine Periode in der Greiffunktion ist. Für die Intervalle $(\Theta_1, \Theta_2, \ldots, \Theta_i)$ gilt die folgende Ungleichung:

$$|s(\Theta_{a+1})| < |\Theta_a|, \quad \forall \ 1 \le a < i \tag{2.4}$$

Algorithmus 2.1.1 Berechnung einer Folge von s-Intervallen

Eingabe: planares, konvexes Polygon P

- 1: Berechne die Greiffunktion s von P
- 2: Finde das größte Intervall Θ_1 , auf dem s konstant ist
- $3: i \leftarrow 1$
- 4: while es existiert ein s-Intervall Θ mit $|s(\Theta)| < |\Theta_i|$ do
- 5: Θ_{i+1} sei das größte dieser s-Intervalle
- 6: $i \leftarrow i+1$
- 7: end while
- 8: Gebe die Liste $(\Theta_1, \Theta_2, \ldots, \Theta_i)$ zurück

Wie helfen diese Intervalle nun bei der Berechnung eines Aktionsplans? Die Ideen dafür sollen kurz erläutert werden.

Uber die Orientierung des Werkstücks liegen zu Beginn der Ausführung eines Aktionsplans keine Informationen vor. Sie liegt also o. B. d. A. beliebig im Intervall Θ_i bezüglich des globalen Bezugssystems. Nach Ausführung der ersten Greifaktion bei einem beliebigem Winkel liegt die Orientierung des Werkstücks per Definition innerhalb des s-Image $s(\Theta_i)$. Aufgrund von Ungleichung -++2.4 gilt $|s(\Theta_i)| < |\Theta_{i-1}|$. Wird der Greifer unabhängig vom Werkstück gedreht, ändert sich die relative Lage des Werkstücks zum Greifer, nicht aber die eigentliche Orientierung des Werkstücks zum globalen Bezugssystem. Durch eine Drehung des Greifers ohne das Werkstück ist es möglich, die beiden Intervalle $s(\Theta_i)$ und Θ_{i-1} aneinander auszurichten. Ist der Greifer in diesem Sinne korrekt gedreht, enthält das Inverval Θ_{i-1} bezüglich des Greifers das Intervall $s(\Theta_i)$ bezüglich des globalen Bezugssystems und insbesondere die aktuelle Orientierung des Werkstücks. Durch eine weitere Greifaktion werden alle Orientierungen im Intervall Θ_{i-1} bezüglich des Greifers per Definition in das Intervall $s(\Theta_{i-1})$ bezüglich des Greifers überführt. Damit liegt auch die aktuelle Orientierung des Werkstücks im Intervall $s(\Theta_{i-1})$ bezüglich des Greifers. Im nächsten Schritt werden die Intervalle $s(\Theta_{i-1})$ und Θ_{i-2} durch eine weitere Drehung des Greifers aneinander ausgerichtet. Die nächste Greifaktion überführt alle Orientierungen im Intervall Θ_{i-2} bezüglich des Greifers (welches die aktuelle Orientierung des Werkstücks beinhaltet) in das Intervall $s(\Theta_{i-2})$ bezüglich des Greifers.

Durch sukzessive Anwendung dieses Verfahrens beschränken sich die tatsächlich möglichen Orientierungen des Werkstücks im weiteren Verlauf auf immer kleinere Intervalle. Dabei ist in jeder Iteration eine Drehung und ein Schließvorgang des Greifers nötig. Die Drehwinkel können aus der Folge von s-Intervallen berechnet werden. Die Folge dieser Drehwinkel bildet den Aktionsplan. Nach der letzten Iteration liegt die Orientierung des Werkstücks in dem zu einem Punkt entarteten s-Image $s(\Theta_1)$. Abbildung 2.6 verdeutlicht dieses Vorgehen.



Abbildung 2.6: Während der Abarbeitung eines Aktionsplanes werden die Intervalle, welche die tatsächliche Orientierung des Werkstückes enthalten, immer weiter eingeschränkt.

Eine Berechnungsvorschrift für die Winkel des Aktionsplans aus der Folge von s-Intervallen ($\Theta_1, \Theta_2, \ldots, \Theta_i$) wurde bisher noch nicht gegeben. Seien $s(\Theta_{a+1})$ und Θ_a zwei Intervalle, welche aneinander auszurichten sind. Seien θ_{a+1} und θ_a die linken Intervallgrenzen von Θ_{a+1} und Θ_a . Die Ausrichtung der Intervalle erfolgt durch eine Drehung des Greifers um den Winkel $\beta = s(\theta_{a+1}) - \theta_a$. Diese Bewegung ist relativ von der aktuellen Orientierung des Greifers auszuführen, welche dem zuletzt angefahrenen Planwinkel entspricht. Für die erste Greifaktion eines Aktionsplans kann ein beliebiger Winkel gewählt werden. Es sei o. B. d. A. $\alpha_i = 0$. Ein Planwinkel α_a kann also durch die folgende Vorschrift berechnet werden:

$$\alpha_a = \begin{cases} 0, & \text{falls } a = i; \\ \underbrace{s(\theta_{a+1}) - \theta_a}_{\beta} + \alpha_{a+1}, & \text{falls } 1 \le a < i. \end{cases}$$
(2.5)

Alternativ zur Ausrichtung der beiden Intervallgrenzen aneinander besteht die Möglichkeit, das kleinere Intervall bezüglich des größeren Intervalls zu zentrieren. Dies macht die praktische Ausführung des Aktionsplans mit einem Roboter robust gegenüber Abweichungen zwischen dem theoretisch erlangten und praktisch erzieltem Winkel des Greifers. Aus Ungleichung 2.4 folgt die Positivität von $\epsilon_a = \frac{1}{2} (|\Theta_a| - |s(\Theta_{a+1})|)$. Durch die Addition der Fehlerkonstante ϵ_a zu dem Drehwinkel α_a wird das kleinere Intervall $s(\Theta_{a+1})$ in Bezug auf das Intervall Θ_a zentriert. Planwinkel werden aus diesem Grunde wie folgt berechnet:

$$\alpha_a = \begin{cases} 0, & \text{falls } a = i; \\ \underbrace{s(\theta_{a+1}) - \theta_a}_{\beta} + \alpha_{a+1} - \epsilon_a, & \text{falls } 1 \le a < i. \end{cases}$$
(2.6)

2.1.6 Radius- und Schiebefunktion eines Polygons

Bisher wurde angenommen, dass beide Backen des Greifers während des Schließvorgangs gleichzeitig einen Kontakt mit dem Werkstück herstellen. Dies ist in der Praxis im Allgemeinen nicht zu erwarten. In diesem Abschnitt wird diese unrealistische Annahme beseitigt.

Zu Beginn des Greifvorgangs ist die Distanz des Werkstückes zu den beiden Backen des Greifers im Allgemeinen unterschiedlich. Unter diesen Umständen ergibt sich der folgende Ablauf des Greifvorgangs (vgl. Abbildung 2.7). Eine der Backen des Greifers stellt vor der anderen Backe einen Kontakt mit dem Werkstück her. Im weiteren Verlauf des Schließvorgangs wird das Werkstück von der kontaktierenden Backe geschoben, bis auch die zweite Backe einen Kontakt mit dem Werkstück hergestellt hat. Die Schließbewegung des Greifers ist beendet, wenn die Fortführung der Bewegung das Werkstück verformen würde. PSfrag replacements



Abbildung 2.7: (i) Das Werkstück liegt zwischen den Backen, (ii) ein Kontakt zwischen Greifer und Werkstück entsteht, (iii) das Werkstück wird geschoben, (iv) der Greifvorgang ist beendet

Die weiteren Betrachtungen gehen im Gegensatz zum bisherigen Teil des Kapitels davon aus, dass eine der Backen des Greifers den Kontakt mit dem Werkstück vor der zweiten Backe herstellt. Um der geänderten Situation zu begegnen, wird der Greifvorgang aus Kapitel 2.1.1 um zwei Translatio nen^4 erweitert. Die erste Translation wird vor dem Schließen des Greifers ausgeführt. Der Greifer bewegt sich dabei orthogonal zu seiner Orientierung. Nachdem ein Kontakt zwischen einer Backe des Greifers und dem Werkstück entstanden ist, ändert sich im weiteren Verlauf der Translation die Orientierung des Werkstücks. Unter den gegebenen physikalischen Modellannahmen kann die Bewegung eines Werkstücks, welches von einer linearen Backe geschoben wird, durch eine Rotation beschrieben werden (vgl. Peshkin und Sanderson [28]). Nach der Translation schließt der Greifer die Backen und greift das Werkstück. Eine zweite Translation bringt den Greifer und das Werkstück anschließend wieder in die Ausgangsposition zurück. Da der Greifer dabei geschlossen ist, kann sich die Orientierung des Werkstücks nicht ändern. Der modifizierte Bewegungsablauf sieht wie folgt aus:

- Drehe den Greifer auf den Winkel α bezüglich einer festen Normalstellung.
- Führe eine Translation in Richtung $\alpha + \frac{\pi}{2}$ um die Distanz δ aus.
- Schließe beide Backen gleichzeitig, bis das Werkstück eingeklemmt ist.
- Führe eine Translation in Richtung $-(\alpha + \frac{\pi}{2})$ um die Distanz δ aus.
- Öffne beide Backen.

Für die Vorhersage der Bewegung eines Werkstücks während der Ausführung des modifizierten Bewegungsablaufs soll nun eine Funktion gefunden werden. Die Bewegung des Werkstücks besteht aus zwei Phasen. Die Änderung der Orientierung des Werkstücks während des Greifvorgangs kann mit der Greiffunktion aus Kapitel 2.1.3 beschrieben werden. Für die Beschreibung des Effekts der ersten Translation wird eine Schiebefunktion $p: [0, 2\pi) \rightarrow [0, 2\pi)$

 $^{^4\}rm Eine$ Translation ist die geradlinige Bewegung eines Körpers, bei der alle seine Punkte eine parallele Bahn durchlaufen.

benötigt, für die gilt: ist θ die Orientierung des Greifers in Bezug auf das Werkstück vor der ersten Translation, so ist $p(\theta)$ die Orientierung des Greifers in Bezug auf das Werkstück nach der ersten Translation. Für das weitere Vorgehen wird der Begriff der *stabilen Orientierung* eines Werkstücks benötigt, welcher von Rao und Goldberg [31] wie folgt eingeführt wird:

Sprechweise 3 Es sei ein polygonales Werkstück und ein Greifer mit zwei parallelen Backen gegeben. Die Funktion s sei die im Abschnitt 2.1.3 beschriebene Greiffunktion. Eine Orientierung θ des Werkstücks bezüglich des Greifers heißt stabil, falls $s(\theta \pm \delta) = \theta$ gilt. Eine Orientierung θ des Werkstücks heißt instabil, wenn sie nicht stabil ist.

Nach Goldberg [15] zeichnen sich stabile Orientierungen dadurch aus, dass mindestens eine Kante der konvexen Hülle des Werkstücks einen Kontakt mit einer Backe des Greifers besitzt. Ersetzt man die Greiffunktion s durch die Schiebefunktion p, so lässt sich der Begriff der stabilen Orientierung auf Situationen übertragen, in denen das Werkstück von einer linearen Backe geschoben wird.

Um eine genaue Aussage über die Bewegung eines Werkstücks während der Translationsbewegung des Greifers machen zu können, werden neue Annahmen eingeführt:

- Die Translationsdistanz δ ist groß genug, um das Werkstück in eine stabile Orientierung zu überführen, bevor die zweite Backe des Greifers einen Kontakt mit dem Werkstück herstellt.
- Der *Massenschwerpunkt⁵* des Werkstücks ist bekannt. Die Reibung zwischen dem Werkstück und der Werkebene ist unabhängig von der Position und der Geschwindigkeit des Werkstücks.

Die Lösung für die Schiebefunktion p ähnelt der Greiffunktion aus Abschnitt 2.1.3. Beide Funktionen sind stückweise konstante Treppenfunktionen. Während die Greiffunktion aus den lokalen Extrema der Durchmesserfunktion eines Polygons abgeleitet wird, kann die Schiebefunktion p in gleicher Weise aus der sogenannten *Radiusfunktion* eines Polygons gewonnen werden.

Die Radiusfunktion $rad_P : [0, 2\pi) \to [0, 2\pi)$ eines Polygons besteht analog zur Durchmesserfunktion eines Polygons stückweise aus Sinuskurven. Sie bildet eine Orientierung $\alpha \in [0, 2\pi)$ auf den Radius von P bei Winkel α ab. Der Radius des Polygons lässt sich wie folgt anschaulich erklären: sei cmder Massenschwerpunkt des Polygons. Man stelle sich eine zur y-Achse eines festen Koordinatensystems parallele Gerade "rechts" von P vor, welche

 $^{^5 {\}rm Im}$ Sinne der klassischen Mechanik ist der Massenschwerpunkt der Punkt, an dem die Masse des Körpers die gleiche Wirkung auf andere Körper hätte, wenn sie in diesem Punkt vereint wäre.

den Rand des Polygons berührt, nicht aber das Innere von P. Dann ist der Abstand dieser Geraden zum Massenschwerpunkt cm des Polygons der Radius von P bei Winkel 0. Die Radiusfunktion beschreibt die Änderung des Radius, wenn die Gerade um das Polygon gedreht wird. Die Abbildung 2.8 verdeutlicht den Radius eines Polygons. Abbildung 2.9 zeigt die Graphen einer Radiusfunktion und der abgeleiteten Schiebefunktion.



Abbildung 2.8: Der Radius eines Polygons

Es besteht also die Möglichkeit, den Effekt einer Translationsbewegung des Greifers auf das Werkstück durch die Schiebefunktion auszudrücken. Der Effekt der vollständigen Ausführung des modifizierten Bewegungsablaufs kann durch die Konkatenation der Schiebefunktion p mit der Greiffunktion saus Abschnitt 2.1.3 beschrieben werden. Es ergibt sich die *Transferfunktion* des Werkstücks $t : [0, 2\pi) \rightarrow [0, 2\pi)$ mit $t = s \circ p$. Die Transferfunktion ist eine stückweise konstante Treppenfunktion.

Mit Hilfe der Transferfunktion können Aktionspläne für den Parallel-Jaw Gripper berechnet werden. In Algorithmus 2.1.1 ist anstelle der Greiffunktion die Transferfunktion zu berechnen und zu verwenden. Der Rest des Algorithmus sowie die Gleichung 2.6 zur Transformation der resultierenden Intervalle in Drehwinkel bleiben unberührt. Die so berechneten Aktionspläne orientieren polygonale Werkstücke unter Verwendung des erweiterten Bewegungsablaufs aus diesem Abschnitt. Auf die vereinfachende Annahme, dass beide Backen den Kontakt mit dem Werkstück gleichzeitig herstellen, kann verzichtet werden.

Ein weiterer positiver Aspekt der Resultate dieses Abschnitts sei kurz erwähnt. Die Durchmesser- und die Greiffunktion eines Polygons besitzen aufgrund der Symmetrie des Greifers immer eine Periode von π . Dies impliziert eine Mehrdeutigkeit in der Endorientierung des Werkstücks (siehe Abschnitt 2.1.4). Die Radius-, Schiebe- und Transferfunktionen eines Polygons besitzen im Allgemeinen keine Periode von π . Aufgrund der Verwendung der Transferfunktion anstelle der Greiffunktion in Algorithmus 2.1.1 entfällt eine Mehrdeutigkeit in der Endorientierung des Werkstücks. Besitzt



Abbildung 2.9: Die Radiusfunktion des L-förmigen Polygons aus Abbildung 2.8, darunter die abgeleitete Schiebefunktion.

die Transferfunktion lediglich eine Periode von 2π , so existiert eine *eindeu*tige Endorientierung des Werkstücks. Dieser Abschnitt wird abgeschlossen mit dem folgenden

Theorem 5 (Goldberg, 1993) Zu einem Werkstück mit n Ecken kann in Zeit $O(n^2 \log n)$ ein Aktionsplan der Komplexität $O(n^2)$ berechnet werden. Die Anwendung des Aktionsplans durch einen Parallel-Jaw Gripper orientiert das Werkstück in beliebiger anfänglicher Orientierung bis auf Symmetrie. Es existiert kein kürzerer Plan mit dieser Eigenschaft.

2.2 Gelenkarmroboter

Als eine Gattung aus der Familie der industriellen Roboter sind Gelenkarmroboter weit verbreitet und werden für diverse Aufgaben eingesetzt. Dieser Abschnitt 2.2 behandelt einige für diese Arbeit wichtige Aspekte dieser Roboter. Der Inhalt beruht, soweit nicht anders angegeben, auf dem Buch Introduction To Robotics: Mechanics and Control von John J. Craig [10].

2.2.1 Aufbau, Teilsysteme und Terminologie

Ein Gelenkarmroboter besteht aus mehreren Teilsystemen. Diese sollen hier neben einigen wichtigen Begriffen kurz vorgestellt werden.

- Manipulator Äußerlich besteht ein Gelenkarmroboter aus einer Menge von starren Gliedern, welche durch Gelenke (joints) kettenartig miteinander verbunden sind. Die Gelenke erlauben relative Bewegungen zwischen benachbarten Gliedern. Übliche Arten von Gelenken sind Drehgelenke (revolute joints), welche eine Drehung um eine Gelenkachse ermöglichen und Schubgelenke (prismatic joints), welche die Translation eines Gliedes erlauben (siehe Abbildung 2.10). Typische Manipulatoren besitzen fünf oder sechs Gelenke. Als Basis des Roboters wird das mechanische Teil bezeichnet, welches der Befestigung des Gelenkarmroboters dient. Das Gesamtgebilde aus Gliedern und Gelenken heißt Manipulator. Als Anzahl der Freiheitsgrade (degrees of freedom) eines Manipulators wird die Anzahl der unabhängigen Positionsvariablen bezeichnet, welche für die vollständige Beschreibung der Position und Orientierung des Manipulators notwendig wären.
- **Endeffektor** Am Ende der Kette von Gliedern des Manipulators kann ein *Endeffektor* befestigt sein. Ein Endeffektor ermöglicht dem Manipulator die Interaktion mit seiner Umwelt. Typische Beispiele sind Greifer, Schraubendreher und Messvorrichtungen.
- Antrieb Der Antrieb erlaubt es dem Gelenkarmroboter, die relative Lage der einzelnen Glieder zueinander zu halten oder auch zu verändern. Die große Mehrzahl von Gelenkarmrobotern besitzt einen Antrieb, welcher eine Bewegung durch die Ausübung von Kraft oder durch die Übertragung eines Drehmoments auf die Glieder erzielt.
- Sensoren Für die Erfassung von internen Zuständen sowie von Eigenschaften seiner Umwelt besitzt ein Gelenkarmroboter Sensoren. Beispielsweise erlauben Sensoren die Messung der relativen Lage von benachbarten Gliedern oder der Geschwindigkeit der Bewegung von Gliedern. Je nach Aufgabenstellung ist auch die Messung von externen Zuständen des Roboters sinnvoll. Bei einem Kontakt mit Gegenständen in seiner Umgebung übt ein Roboter Kräfte auf die Gegenstände aus. Die Messung dieser Kräfte durch Kraftsensoren befähigt den Manipulator zu präzisen und behutsamen Interaktionen mit seiner Umwelt.
- **Programmiersystem** Für die Programmierung eines Roboters wird ein Programmiersystem benötigt. Eine einfache Art der Programmierung geschieht über das manuelle Anfahren von Positionen, dem *Teachen*. Die Parameter des Roboters in der jeweiligen Position werden für einen

späteren Abruf gespeichert. Komplexere und flexiblere Programmierung von Robotern wird durch die Verwendung von speziellen Roboterprogrammiersprachen möglich. Die Programmierung erfolgt in Form von reinem Text, dem Programm. Dieses erlaubt die Beschreibung von Aktionen des Roboters sowie die Interaktion mit Sensoren und anderem Zubehör. Ein drittes Konzept zur Programmierung von Robotern erlaubt dem Benutzer die gewünschte Aufgabe direkt zu beschreiben, ohne dabei genauer auf die Details der Ausführung einzugehen.

Um Roboterprogramme unabhängig von der Verfügbarkeit eines Roboters entwerfen zu können, wurden sogenannte *offline Programmiersysteme* entwickelt. Diese besitzen in der Regel grafische Fähigkeiten zur Darstellung des Roboters und anderer Objekte. Durch den Einsatz von offline Programmiersystemen ergibt sich der Vorteil, dass die Roboter länger in der Produktion zur Verfügung stehen und damit Ausfallzeiten minimiert werden.

Wiederholungsgenauigkeit und Präzision In vielen Situationen werden Positionen von Robotern angefahren, welche durch vorgegebene Gelenkpositionswerte definiert sind. Dies ist beispielsweise dann der Fall, wenn ein Bewegungsablauf mittels Teachen programmiert wurde und zu einem späteren Zeitpunkt ausgeführt werden soll. In diesem Fall müssen alle Gelenke die jeweils gespeicherten Positionswerte einnehmen. Mit der *Wiederholungsgenauigkeit (repeatability)* des Roboters gibt der Hersteller an, welche Abweichungen beim wiederholten Anfahren von gelernten Positionen auftreten können.

Wurde die Zielposition und -orientierung für den Greifer des Roboters in kartesischen Koordinaten formuliert, so müssen diese Koordinaten in entsprechende Gelenkwinkel umgerechnet werden (vgl. Abschnitt 2.2.4 auf Seite 28). Die Genauigkeit, mit welcher der Roboter die Zielposition und -orientierung in Form der berechneten Gelenkwinkel einnehmen kann, wird durch die *Präzision (accuracy)* des Roboters beschrieben. Die Präzision eines Manipulators wird durch seine Wiederholungsgenauigkeit nach unten beschränkt.

2.2.2 Relative Lage von Objekten

Während der Arbeit mit Robotern ist es oft nötig, die Lage von verschiedenen Objekten (zueinander) zu beschreiben. Die Lage eines Objekts im dreidimensionalen Raum \Re^3 lässt sich eindeutig durch die Angabe der Position und Orientierung des Objekts darstellen.

Für die mathematische Repräsentation der Lage eines Objektes sind kartesische Koordinatensysteme hilfreich. Mit Hilfe eines Koordinatensystems



Abbildung 2.10: (i) Ein Drehgelenk, (ii) ein Schubgelenk

kann die Position eines Punktes im \Re^3 in Form eines dreidimensionalen Ortsvektors dargestellt werden. Der Vektor enthält die Koordinaten des Punktes bezüglich des als Referenz gewählten Koordinatensystems. Neben der Position eines Punktes im \Re^3 ist oft nach der Orientierung eines Objekts im Raum gefragt. Wird dem entsprechenden Objekt ein Koordinatensystem zugeordnet, lässt sich die Orientierung des Objekts durch die Beschreibung des zugeordneten Koordinatensystems relativ zu einem Referenzkoordinatensystem ausdrücken. Diese Beschreibung resultiert in einer 3x3-Matrix. Als Referenzkoordinatensystem kann ein beliebiges Koordinatensystem gewählt werden.

Die vollständige Beschreibung der Lage eines Objekts bezüglich eines Referenzkoordinatensystems, bestehend aus Position *und* Orientierung des Objekts, ist mit einem dreidimensionalen Ortsvektor und einer 3x3-Matrix möglich. Eine Beschreibung in dieser Form wird als *Frame* bezeichnet. Ein Frame entspricht gerade einem Koordinatensystem, denn neben einer Orientierung besitzt ein Koordinatensystem einen Ursprung als Positionsangabe. Zusammenfassend sei festgehalten, dass ein Frame die relative Beschreibung eines Koordinatensystems bezüglich eines Referenzkoordinatensystems im Raum \Re^3 erlaubt.

Ein Manipulator besitzt in der Regel einige standardisierte Referenzkoordinatensysteme, auf welche sich Lageangaben von Objekten beziehen können. Zu jedem Gelenk des Manipulators wird ein lokales *Gelenkkoordinatensystem* definiert, dessen Ursprung im Zentrum des Gelenks liegt. Das lokale *Werkzeugkoordinatensystem* hingegen bezieht sich auf das Ende des Endeffektors, mit welchem der Manipulator bestückt ist. Die Position des Ursprungs der bisher beschriebenen Koordinatensysteme ist fest an die veränderliche Position bestimmter Objekte gebunden. Um einen globalen Bezug zu ermöglichen wird ein beliebig, aber fest gewähltes *Weltkoordinatensystem* definiert. Abbildung 2.11 zeigt einen Manipulator mit den Gelenkkoordinatensystemen O_0 , O_1 und O_2 sowie einem Werkzeugkoordinatensystem O_3 .

Zu dem Zweck der einheitlichen mathematischen Darstellung von Lagebeziehungen zwischen Koordinatensystemen sollen homogene Transformationsmatrizen eingeführt werden (vgl. Pfalzgraf, [29]). Punkte des \mathbb{R}^3 seien



Abbildung 2.11: Ein Gelenkarmroboter mit mehreren Koordinatensystemen

wie folgt in den \mathbb{R}^4 eingebettet:

$$\mathbb{R}^3 \ni a = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \longmapsto \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ 1 \end{pmatrix} = a' \in \mathbb{R}^4.$$

Die Translation $\tau_t(a)$ eines Vektors $a = (a_1, a_2, a_3)^T$ um einen Vektor $t = (t_1, t_2, t_3)^T$ im dreidimensionalen Raum kann durch eine Matrizenmultiplikation beschrieben werden:

$$\begin{pmatrix} 1 & 0 & 0 & t_1 \\ 0 & 1 & 0 & t_2 \\ 0 & 0 & 1 & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ 1 \end{pmatrix} = \begin{pmatrix} a_1 + t_1 \\ a_2 + t_2 \\ a_3 + t_3 \\ 1 \end{pmatrix} = \tau_t(a).$$

Die Rotation eines Vektors $a = (a_1, a_2, a_3)^T$ im dreidimensionalen Raum um die x-, y-, oder z-Achse⁶ um den Winkel α im mathematisch positiven Sinn lässt sich ebenso als homogene 4x4-Matrix schreiben. Das folgende Beispiel zeigt die Rotation eines Vektors um die y-Achse:

$$\begin{pmatrix} \cos(\alpha) & 0 & -\sin(\alpha) & 0\\ 0 & 1 & 0 & 0\\ \sin(\alpha) & 0 & \cos(\alpha) & 0\\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_1\\ a_2\\ a_3\\ 1 \end{pmatrix} = Rot_y^{\alpha}(a).$$

Nach der gerade beschriebenen Sichtweise wird eine homogene Matrix als Operator interpretiert (Craig, [10]). Die Anwendung des Matrixoperators auf einen Vektor bewerkstelligt eine lineare Abbildung des Vektors. Der ursprüngliche und der abgebildete Vektor sind in demselben Koordinatensystem definiert.

 $^{^6 \}rm Rotationen$ um beliebige Achsen im ${\rm I\!R}^3$ lassen sich ebenfalls durch geeignete homogene Matrizen beschreiben.

Neben der Sichtweise als Operator ist eine weitere Interpretation möglich: die Anwendung einer homogenen Matrix auf einen Vektor beeinflusst nach dieser Sichtweise nicht den Vektor, sondern sein Koordinatensystem. In der einen Sichtweise z. B. wird ein Vektor von einer Transformation entlang der x-Achse verschoben, in der alternativen Sichtweise hingegen wird das Koordinatensystem des Vektors entgegengesetzt der x-Achse verschoben. Der resultierende Vektor besteht aus den Koordinaten bezüglich des transformierten Koordinatensystems. Die Anwendung einer homogenen Matrix auf einen Vektor kann also als ein Wechsel des Koordinatensystems des Vektors interpretiert werden. Die mehrfache Anwendung von homogenen Matrizen auf einen Vektor stimmt mit einem mehrfachen Wechsel des Koordinatensystems des Vektors überein.

Die den beiden Sichtweisen zugrunde liegende Mathematik ist identisch, lediglich die Interpretation des Ergebnisses unterscheidet sich. Die letztere Sichtweise homogener 4x4-Matrizen ist für die Angabe einer Transformationsbeziehung zwischen verschiedenen Koordinatensystemen hilfreich. In diesem Sinne werden homogene Matrizen im nächsten Abschnitt zur Lösung der Vorwärtskinematik eines Manipulators verwendet.

2.2.3 Vorwärtskinematik

Die Kinematik ist die Lehre von der Bewegung, welche nicht auf die verursachenden Kräfte eingeht. Die Kinematik beschäftigt sich mit der Position, Geschwindigkeit, Beschleunigung sowie mit höheren Ableitungen der Position nach der Zeit.

Dieser Abschnitt beschäftigt sich mit der sogenannten Vorwärtskinematik eines Gelenkarmroboters. Das Ziel ist die Berechnung der Position und Orientierung des Endeffektors relativ zur Basis des Manipulators als eine Funktion der Gelenkwinkel. Betrachtet wird eine statische Situation, Aspekte wie Geschwindigkeit und Beschleunigung werden nicht behandelt. Weiterhin wird angenommen, dass jedes Gelenk des Manipulators über genau einen Freiheitsgrad verfügt. Sollte ein Gelenk n > 1 Freiheitsgrade besitzen, kann es durch n verschiedene Gelenke mit jeweils einem Freiheitsgrad modelliert werden, welche von n-1 Gliedern der Länge 0 verbunden werden. Die Glieder eines Gelenkarmroboters werden im Rahmen der vorliegenden Arbeit durchgehend nummeriert. Die Basis des Roboters erhält die Nummer 0. Der erste bewegliche Körper erhält die Nummer 1 und so weiter. Die Gelenke sind von 1 bis n nummeriert. Das Gelenki rotiert um die Gelenkachse i. Die Glieder des Manipulators werden als feste Körper angesehen, welche Beziehungen zwischen benachbarten Gelenkachsen definieren. Die Gelenkachsen werden als Geraden im Raum betrachtet.

Die Beschreibung des Gelenkarmroboters erfolgt durch die Angabe von vier Werten je Glied des Manipulators. Zwei der Werte beschreiben das Glied selbst, zwei weitere beschreiben die Verbindung zu einem benachbarten Glied. Diese den Roboter beschreibenden Werte werden gemäß ihrer geistigen Väter als **Denavit-Hartenberg Parameter** bezeichnet.

- 1. Die Distanz zweier Achsen im Raum ist definiert über die eindeutig festgelegte Gerade, welche zu beiden Achsen einen rechten Winkel bildet. Im Spezialfall zweier paralleler Achsen existieren unendlich viele solcher Geraden. Der Denavit-Hartenberg Parameter a_{i-1} von Glied i-1 misst die Distanz zwischen der Gelenkachse i-1 und der Gelenkachse i entlang dieser Geraden (siehe Abbildung 2.12).
- 2. Der rechtwinkligen Geraden zwischen zwei Achsen kann eindeutig eine Ebene im Raum zugeordnet werden, sodass diese Gerade als Normale der Ebene auftritt. Der Denavit-Hartenberg Parameter α_{i-1} von Glied i-1 ergibt sich durch eine Projektion der Gelenkachsen i-1 und i auf diese Ebene. Der Parameter α_{i-1} entspricht dem Winkel zwischen den projektiven Bildern der beiden Gelenkachsen auf der Ebene. Im Falle sich schneidender Gelenkachsen wird α_{i-1} in der durch beide Achsen definierten Ebene gemessen.
- 3. Benachbarte Glieder des Manipulators besitzen jeweils eine gemeinsame Gelenkachse. Der Denavit-Hartenberg Parameter d_i beschreibt die Distanz von Glied i - 1 zu Glied i entlang der Gelenkachse i (siehe Abbildung 2.13).
- 4. Der Denavit-Hartenberg Parameter θ_i gibt den Drehwinkel zwischen den Gliedern i-1 und i um die Gelenkachse i an.



Abbildung 2.12: Zwei Denavit-Hartenberg Parameter beschreiben ein Glied des Manipulators



Abbildung 2.13: Zwei weitere Denavit-Hartenberg Parameter beschreiben die Verbindung zum benachbarten Glied des Manipulators

Um die Position eines Gliedes des Manipulators relativ zu einem benachbarten Glied beschreiben zu können, wird nun jedem Glied i jeweils ein Frame i zugeordnet. Dies geschieht gemäß der **Denavit-Hartenberg Konvention** auf die folgende Art und Weise:

- 1. Identifiziere die Gelenkachsen. Für die Schritte 2 bis 5 betrachte die Gelenkachsen i und i + 1.
- 2. Stelle Dir den Schnittpunkt bzw. die rechtwinklige Gerade der beiden Achsen vor, falls die beiden Achsen keinen Schnittpunkt besitzen. Der Ursprung des Frames *i* liegt im Schnittpunkt der beiden Achsen bzw. im Schnittpunkt der rechtwinkligen Gerade mit der Gelenkachse *i*.
- 3. Die z-Achse von Frame i hat die Richtung der Gelenkachse i
- 4. Die x-Achse von Frame i läuft in Richtung der rechtwinkligen Geraden der Gelenkachsen i und i + 1. Falls die Gelenkachsen sich schneiden, steht die x-Achse lotrecht auf der Ebene, welche durch die beiden Gelenkachsen aufgespannt wird.
- 5. Die y-Achse von Frame *i* bildet mit der z- und x-Achse ein rechtshändiges System.
- 6. Der Ursprung und die Richtung der x-Achse für Frame n kann beliebig gewählt werden.

Sind die Koordinatensysteme für den Gelenkarmroboter wie beschrieben festgelegt, helfen die Denavit-Hartenberg Parameter bei der Ableitung einer Transformationsbeziehung zwischen zwei benachbarten Frames. Die Transformation T_i^{i-1} besteht aus den folgenden vier Schritten:

- 1. Translation entlang der z-Achse von Frame i um d_i
- 2. Rotation um die z-Achse von Frame ium den Winkel θ_i
- 3. Translation entlang der x-Achse von Frame i-1 um a_{i-1}
- 4. Rotation um die x-Achse von Frame i-1 um den Winkel α_{i-1}

In Abschnitt 2.2.2 wurde gezeigt, dass Rotationen und Translationen in Form von homogenen 4x4-Matrizen angegeben werden können. Die Transformation T_i^{i-1} kann daher als Produkt von vier Matrizen geschrieben werden. Man erhält eine T_i^{i-1} entsprechende homogene 4x4-Matrix. Sei v ein Vektor, welcher in Frame i gegeben ist. Die Anwendung von T_i^{i-1} auf den Vektor v kann gemäß Abschnitt 2.2.2 als Wechsel des Koordinatensystems des Vektors interpretiert werden. Der resultierende Vektor besteht aus den Koordinaten bezüglich des Frames i-1.

Die Verkettung mehrerer Transformationen T_i^{i-1} entspricht dem mehrfachen Wechsel des Koordinatensystems. Die folgende Gleichung ergibt eine Transformation, welche Frame n zu Frame 0 in Beziehung setzt:

$$T_n^0 = T_1^0 \circ T_2^1 \circ T_3^2 \circ \cdots \circ T_n^{n-1}$$

Eine mathematische Darstellung von T_n^0 kann durch die Multiplikation der homogenen Matrizen erreicht werden, welche den auf der rechten Seite der Gleichung vorkommenden Transformationen T entsprechen. Es ergibt sich eine homogene 4x4-Matrix zur Transformation der Darstellung von Vektoren bezüglich des Frames n in eine Darstellung bezüglich des Frames 0.

Allgemein ist die Transformation T_n^0 eine Funktion der 4n Denavit-Hartenberg Parameter eines Gelenkarmroboters. Aufgrund des mechanischen Designs von Gelenkarmrobotern ist je Gelenk ein Parameter variabel, während die restlichen drei Parameter fest sind. Die der Transformation T_n^0 entsprechende Matrix besteht daher bis auf die Gelenkwinkel des Roboters aus festen Einträgen. Die homogene Matrix zu T_n^0 beschreibt in Abhängigkeit von den Gelenkwinkeln die Position und die Orientierung des Endeffektors bezüglich der Basis des Gelenkarmroboters. Damit ist das Problem der Vorwärtskinematik eines Gelenkarmroboters gelöst. Im nächsten Abschnitt wird das inverse Problem der Vorwärtskinematik vorgestellt.

2.2.4 Inverse Kinematik

Die inverse Kinematik geht von einer gewünschten Position und Orientierung des Werkzeugs eines Gelenkarmroboters relativ zu einem festen Weltkoordinatensystem aus. Die Aufgabe besteht darin, eine Menge von Gelenkwinkeln zu berechnen, welche das Werkzeug in die gewünschte Lage bringen.
Grundsätzlich sind bei der Lösung der inversen Kinematik zwei Dinge zu beachten:

- **Existenz einer Lösung** Kann die Zielposition in der gewünschten Orientierung nicht erreicht werden, so existiert keine Lösung der inversen Kinematik. In diesem Fall stellt sich die interessante Frage nach der bestmöglichen Näherungslösung. Alle erreichbaren Paare bestehend aus Position und Orientierung bilden den sogenannten Arbeitsraum des Roboters.
- Eindeutigkeit der Lösung Für einen Manipulator mit mehreren Freiheitsgraden und Drehgelenken bieten sich eventuell mehrere unterschiedliche Möglichkeiten, die gewünschte Position und Orientierung des Werkzeugs zu erreichen. In diesem Fall ist die Lösung der inversen Kinematik nicht eindeutig. Anhand von Kriterien sind die verschiedenen Lösungen zu bewerten, um eine "gute" Lösung zu wählen. Ein naheliegendes Kriterium könnte die für das Anfahren der Lösung benötigte Variation der Gelenkvariablen sein, welche zu minimieren wäre. Abbildung 2.14 zeigt einen Gelenkarmroboter mit zwei Gliedern. Die getrichelte Linie weist auf eine alternative Lösung der inversen Kinematik hin.

Bei der Lösung der inversen Kinematik treten nichtlineare Gleichungen auf, für deren Lösung kein allgemein gültiger Algorithmus existiert. Lösen heißt im Kontext der inversen Kinematik, dass im Falle der Mehrdeutigkeit alle Lösungen berechnet werden müssen. Verfahren der inversen Kinematik können in zwei Gattungen klassifiziert werden: *numerische Verfahren* basieren auf iterativen und im Allgemeinen zeitaufwendigen Berechnungen der Lösung(en), während *analytische Verfahren* einen Ausdruck in geschlossener Form erzeugen. Die Parameter der gewünschten Position und Orientierung des Manipulators werden in den Ausdruck eingesetzt, um durch eine Auswertung die Lösung zu erhalten. Nur in Spezialfällen sind analytische Verfahren für die inverse Kinematik von Manipulatoren mit sechs Freiheitsgraden geeignet. In diesen Spezialfällen besitzt der Manipulator entweder mehrere sich schneidende Gelenkachsen und/oder viele der Denavit-Hartenberg Parameter des Manipulators besitzen die Werte 0 Grad oder \pm 90 Grad.

Für weitergehende Informationen zu diesem Thema siehe beispielsweise Craig [10]. Konkrete Ansätze für die inverse Kinematik werden z. B. in [19, 34, 9] vorgestellt.

2.3 Der EduBot 250M

Das Hauptgeschäftsfeld der schottischen Firma *Robotica Limited* liegt nach Angaben auf den Firmenwebseiten [24] in der Entwicklung und Fertigung



Abbildung 2.14: Es existieren mehrere Lösungen der inversen Kinematik

von Produkten aus dem Bereich Robotik und Bewegungskontrolle. Die Produktpalette umfasst u. a. einen Gelenkarmroboter namens EduBot, welcher in mehreren Varianten erhältlich ist. Dem ersten Modell EduBot 100 folgte die Weiterentwicklung EduBot 250. Der EduBot 250 existiert in einer Ausführung für den stationären Betrieb (EduBot 250S) und in einer mobilen Version (EduBot 250M).

Ebenfalls vertrieben wird der EduBot von der Firma ActivMedia Robotics. Diese Firma bietet auf ihrer Internetpräsenz [25] ein breites Spektrum an Roboterplattformen, Roboterzubehör und -steuerungssoftware an. Unter den Roboterplattformen befindet sich auch die Familie der mobilen, geländegängigen *Pioneer* Roboter. Die Firma vertreibt den Roboter Edu-Bot unter dem Namen *Pioneer Arm* als Zubehör zu ihren mobilen Robotern Pioneer 2 und Pioneer 3. Der EduBot kann auf einem Roboter aus der Pioneer-Familie als Werkzeug aufgeschraubt werden und erweitert dessen Fähigkeiten.

Der für diese Arbeit benutzte Roboter vom Typ EduBot 250M wurde zusammen mit einem mobilen Roboter ActivMedia Pioneer 3-AT als Forschungsroboter von der Universität Bonn angeschafft und für die Durchführung dieser Diplomarbeit bereitgestellt. Abschnitt 2.3 beschreibt den EduBot 250M, so wie er von der Firma ActivMedia Robotics geliefert wird. Weitere Details zum EduBot 250M finden sich im offiziellen Datenblatt des Herstellers [22], in der Bedienungsanleitung des Pioneer Arms [20], sowie auf den Internetseiten der Firmen Robotica Limited [24] und ActivMedia Robotics [25].

2.3.1 Eigenschaften und Aufbau

Bei dem EduBot 250M handelt es sich um einen seriellen Gelenkarmroboter mit fünf Drehgelenken. Zur Ausstattung gehört ein Endeffektor in der Form eines Greifers mit zwei Fingern (siehe Abbildung 2.15). Die maximale Nutzlast des Roboters beträgt 250 Gramm bei einem Gewicht von drei Kilogramm. Die Wiederholungsgenauigkeit wird mit \pm 5 mm angegeben. Das erste Gelenk in der Basis des Roboters rotiert um eine Achse, welche vertikal durch die Basis verläuft. Das zweite Gelenk dreht sich um eine waagerecht im Schulterstück liegende Achse. Das dritte Gelenk ist ebenfalls ein Rotationsgelenk. Dessen Rotationsachse verläuft parallel zu der Drehachse des zweiten Gelenks. Zwei weitere Rotationsgelenke üben die Funktion eines Handgelenks aus. Jedes der fünf Gelenke verfügt über einen Gleichstrom-Servomotor, welcher für die Bewegungen des jeweiligen Gelenks zuständig ist. Ein sechster Servomotor bewerkstelligt die Schließbewegungen des Greifers. Der Bewegungsspielraum der Servomotoren ist beschränkt. Im Lieferumfang des Roboters befindet sich eine Tabelle, welche für jeden Servomotor einen Minimal- und einen Maximalpositionswert angibt. Die Servomotoren ermöglichen den Rotationsgelenken einen Bewegungsspielraum von 180 Grad. Im ausgestreckten Zustand beträgt die Reichweite des Manipulators von der Basis aus 48 cm in horizontale und 56 cm in vertikale Richtung.



Abbildung 2.15: Der EduBot 250M

Die Stromzufuhr und Steuerung der Motoren geschieht über den mitgelieferten "Robotica R/C Servo Controller II". Diese elektronische Schaltung ist auf einer Platine in einem externen Gehäuse untergebracht. Neben einer seriellen Schnittstelle nach dem RS-232 Standard zur Ansteuerung des Controllers von einem Computer aus besitzt das Gehäuse einen Anschluss für die eigene Stromzufuhr und eine Anschlussmöglichkeit für die Verbindung mit dem EduBot. Über diese Verbindung werden die Servomotoren des Roboters angesteuert und mit Strom versorgt.

2.3.2 Steuerung

Es existieren mehrere technische Konfigurationen zur Ansteuerung des Roboters EduBot 250M. Grundsätzlich kann der Controller des EduBots mit einem seriellen Kabel direkt an einen Steuerrechner angeschlossen werden. Alternativ existiert die Möglichkeit, den Controller mit einem mobilen Pioneer Roboter der Firma ActivMedia Robotics zu verbinden. Der Pioneer Roboter wiederum besitzt in dieser Konfiguration eine Verbindung zu einem Steuerrechner.

Neben der verwendeten Hardware bildet der Ansatz für die Ansteuerung des EduBots seitens der Steuerungssoftware einen wesentlichen Bestandteil der Konfiguration. Die Steuerungssoftware wird auf dem Steuerrechner ausgeführt und erzielt die gewünschten Bewegungen des Roboters. In einigen Konfigurationen besteht die Möglichkeit, eine Steuerungssoftware gemäß den eigenen Anforderungen in einer Programmiersprache selbst zu entwickeln. Die Kommunikation mit dem Roboter kann beispielsweise auf der offenen Schnittstelle einer Softwarebibliothek oder auf dem Befehlsprotokoll des Controllers des EduBots basieren. Alternativ zur Entwicklung einer eigenen Software bietet sich die Verwendung der Robotersteuerungssoftware eines Drittanbieters an. Der gewünschte Bewegungsablauf des Roboters wird dann mit den Mitteln der Software des Drittanbieters erzielt, z. B. als Programm in einer Roboterprogrammiersprache, welches von der Software des Drittanbieters ausgeführt wird. Es folgt eine Beschreibung der einzelnen Konfigurationen.

Direkter Anschluss In dieser Konfiguration wird der Controller des Edu-Bots mittels einem seriellen Kabel direkt an die serielle Schnittstelle eines Steuerrechners angeschlossen. Der Controller unterstützt ein simples Befehlsprotokoll, welches in der Bedienungsanleitung zum Pioneer Arm [20] beschrieben wird. Das Protokoll ermöglicht Softwareentwicklern die Erstellung einer eigenen Steuerungssoftware. Die Low-Level-Befehle des Protokolls werden vom Steuerrechner mittels serieller Kommunikation an den Controller übermittelt. Die Befehle ermöglichen es, den Roboter an- und auszuschalten, einzelne Servomotoren durch die Vorgabe von Gelenkpositionswerten anzusteuern und die Softwareversion des Controllers zu erfragen. Die Bewegungen einzelner Gelenke des Manipulators erfolgen immer mit der höchstmöglichen Geschwindigkeit. Die Abfrage der aktuellen Positionswerte der Gelenke des Manipulators ist nicht möglich. Nach dem Anschalten stellen alle Gelenke automatisch die im Controller gespeicherte, letzte bekannte Position her. Diese Bewegung geschieht ebenfalls mit der maximal

möglichen Geschwindigkeit. Nach einem Abschalten erschlaffen die Gelenke des EduBots und geben dem Einfluss der Schwerkraft nach. Im ausgeschalteten Zustand des Roboters sind die Glieder flexibel und können manuell bewegt werden.

Als Alternative zur Entwicklung einer auf dem Protokoll des Controllers basierenden Software bietet die Firma Robotica Limited auf ihrer Webseite [24] den Download einer umfassenden Steuerungssoftware für den EduBot 250M frei zugänglich an. Die Software steht nach einer Installation unter dem Betriebssystem Windows zur Verfügung. Die grafische Bedienoberfläche erlaubt die intuitive Bedienung und Programmierung des Roboters. Geschwindigkeiten der Gelenke sowie Minimalund Maximalpositionswerte können separat für jedes Gelenk passend zur jeweiligen Anwendung gesetzt werden. Diese Parameter können zusammen mit anderen Einstellmöglichkeiten als Konfiguration zur späteren Wiederherstellung gespeichert werden. Die Software erlaubt die einzelne oder simultane Ansteuerung der Servomotoren des Edu-Bots. Die Anwendung unterstützt verschiedene Modi. Im sogenannten Designmodus wird der Roboter manuell über die Kontrollelemente des Hauptfensters angesteuert. Im Recordmodus hingegen werden sämtliche Benutzeraktionen automatisch aufgezeichnet und als Befehle eines Steuerungsprogramms in einem Editorfenster angezeigt. Steuerungsprogramme können manuell editiert, gespeichert und ausgeführt werden. Die Software bietet keine offene Schnittstelle an. Softwareentwickler haben daher keine Möglichkeit, die vielfältigen Funktionen des Programms in eigenen Anwendungen zu nutzen.

Indirekter Anschluss In der für diese Arbeit vorliegenden Version ist der EduBot als Werkzeugaufsatz auf einem mobilen Pioneer Roboter von der Firma ActivMedia Robotics gedacht (vgl. [20]). Für diese Konfiguration wird der EduBot fest auf einen dafür vorgesehenen Sockel am Pioneer Roboter geschraubt. Die vom Controller des Edu-Bots ausgehenden Strom- und Signalleitungen werden an entsprechende Anschlüsse des Pioneer Roboters angeschlossen. Die Ansteuerung des EduBots erfolgt integriert mit der Steuerung des mobilen Roboters über eine Client-Server-Verbindung zwischen einem Steuerrechner und dem Pioneer Roboter. Auf dem Pioneer Roboter wird ein Betriebssystem ausgeführt, welches Funktionen zur Steuerung des mobilen Roboters und des EduBots über eine offene Softwareschnittstelle bereitstellt. Die Funktionen ermöglichen u. a. das An- und Ausschalten des EduBots, das Bewegen einzelner Glieder mit variablen Geschwindigkeiten und das Abfragen von Status- und Positionswerten des Manipulators. Weiterhin ist je Gelenk des Manipulators ein Minimal- und ein Maximal positions wert, eine Center- und eine Home-Position definiert. Nach dem Anschalten sucht der Arm automatisch die konfigurierte Homeposition auf. Vor dem Ausschalten sollte der EduBot über die entsprechende Funktion des Betriebssystems in seine Homeposition gefahren werden, um willkürliche Bewegungen des EduBots nach dem Anschalten zu vermeiden. Im Betriebssystem des Pioneer Arms sind einige Sicherheitsvorkehrungen vorhanden, welche die Servomotoren des EduBots vor einer Überlastung schützen sollen. Ist der EduBot beispielsweise für längere Zeit inaktiv, so wird er vom Betriebssystem des Pioneer Roboters automatisch ausgeschaltet.

Alternativ zu den erweiterten Funktionen des Betriebssystems kann in der indirekten Konfiguration auch das Befehlsprotokoll des Controllers zur Steuerung des Roboters verwendet werden. Das Betriebssystem des Pioneer Roboters unterstützt hierfür die Weiterleitung von Low-Level-Befehlen an den EduBot.

Weiterhin kann in dieser Konfiguration die ARIA-Bibliothek [21] zur Steuerung des Pioneer Roboters verwendet werden. ARIA (ActivMedia Robotics Interface Application) bietet eine komplette Schnittstelle zur Steuerung von Robotern der Firma ActivMedia Robotics. Diese ist für die Programmierung von Steuerungssoftware in der Programmiersprache C++ unter den Betriebssystemen GNU Linux und Microsoft Windows gedacht. ARIA steht für Kunden der Firma ActivMedia Robotics auf deren Webseite zum Download bereit. Auf ARIA basierende Software kommuniziert über eine Client-Server Verbindung mit dem Betriebssystem des Pioneer Roboters. Die Bibliothek unterstützt neben der Ansteuerung der Roboter der Firma ActivMedia Robotics die Interaktion mit eventuell vorhandenen Sensoren und anderem Zubehör wie dem EduBot.

Kapitel 3

Konzept

Dieses Kapitel beschreibt ein allgemeines Konzept, welches im Rahmen dieser Diplomarbeit für die Realisierung von Goldbergs Methode zur Orientierung von polygonalen Werkstücken mit einem EduBot 250M erarbeitet wurde. Das Ziel dieses Kapitels ist die Beschreibung von geeigneten Bewegungen des Manipulators, um den in Abschnitt 2.1.6 vorgegebenen Bewegungsablauf zu bewerkstelligen. Während der Erarbeitung des Konzepts ergaben sich für die Durchführung einiger der Bewegungen mehrere plausible Haltungen des Roboters. Um verschiedene Alternativen auf ihre Realisierbarkeit und Eignung zu überprüfen, war ein guter Eindruck von der Bewegungsfähigkeit des Roboters nötig. In diesem Sinne erwies sich die manuelle Durchführung der jeweiligen Bewegungen mit dem ausgeschalteten EduBot 250M als förderlich.

Die Korrektheit von Goldbergs Methode basiert auf einigen Annahmen. Manche dieser Annahmen wie beispielsweise das quasi-statische Modell sind von vereinfachender Natur und können in der Praxis bei unzureichender Berücksichtigung zu Problemen bis hin zu Fehlschlägen führen. Neben der Erarbeitung des Bewegungsablaufs galt es durch geeignete Maßnahmen sicherstellen, dass die Gültigkeit der Annahmen im Kontext des Konzepts erhalten bleibt. In diesem Sinne betrachten einige der folgenden Abschnitte einzelne Modellannahmen und erläutern deren Implikationen für die praktische Umsetzung.

3.1 Annahmen

Als theoretische Grundlage für die Methode von Goldberg dienen neben den in Abschnitt 2.1.2 auf Seite 9 eingeführten Annahmen zusätzlich deren in Abschnitt 2.1.6 auf Seite 16 beschriebenen Modifikationen und Erweiterungen. Die Annahmen werden an dieser Stelle nochmals kollektiv aufgeführt:

Der Greifer Der Greifer besteht aus zwei linearen Backen, welche parallel zueinander stehen. Während des Schließvorgangs bewegen sich die Backen des Greifers orthogonal zu ihrer Orientierung aufeinander zu.

- Die Werkstücke Es wird die Klasse der polygonalen Werkstücke betrachtet. Die konvexe Hülle eines Werkstücks kann als festes planares Polygon betrachtet werden. Die Werkstücke werden als isoliert voneinander angesehen. Es geht nicht darum, aus einer Menge von Werkstücken auf dem Transportband ein einzelnes Stück zu isolieren. Der Massenschwerpunkt des Werkstücks ist bekannt.
- Das physikalische Modell Zwischen den Backen des Greifers und dem zu greifenden Werkstück existiert keine Reibung. Alle Bewegungen finden in der Ebene statt und sind derart langsam, dass Trägheitseffekte der Masse keine Rolle spielen. Die Reibung zwischen dem Werkstück und der Werkebene ist unabhängig von der Position und der Geschwindigkeit des Werkstücks.
- Die Translation des Greifers Das Werkstück liegt zu Beginn der Translation zwischen den Backen des Greifers. Über die initiale Position und Orientierung des Werkstücks liegen keine weiteren Informationen vor. Während der Translation verbleibt das Werkstück stetig zwischen den Backen. Nachdem der Kontakt zwischen dem Werkstück und einer der Backen hergestellt wurde, bleiben die beteiligten Oberflächen bis zum Ende der anschließenden Greifaktion in Kontakt. Die Translationsdistanz ist groß genug, um das Werkstück in eine stabile Orientierung zu überführen, bevor die zweite Backe des Greifers einen Kontakt mit dem Werkstück herstellt.
- Der Greifvorgang Das Werkstück liegt zu Beginn der Greifaktion zwischen den Backen des Greifers und verbleibt dort während der Greifaktion. Während des Greifvorgangs trifft die zweite Backe auf das Werkstück. Von diesem Zeitpunkt an bleiben alle an dem neuen Kontakt beteiligten Oberflächen bis zum Ende der Greifaktion in Kontakt. Die Schließbewegung des Greifers ist beendet, wenn eine Fortführung das Werkstück verformen würde.

3.2 Werkebene

In dieser Diplomarbeit wird eine Konstellation von zwei benachbarten Ebenen vorgeschlagen (siehe Abbildung 3.1). Die erste Ebene dient der Befestigung des Roboters. Zu diesem Zweck wird die Basis des Manipulators auf der ersten Plattform fest angeschraubt. Die zweite Ebene, im Folgenden Werkebene genannt, liegt tiefer als die erste Plattform und dient der Manipulation von Werkstücken. Die Werkebene verfügt über eine glatte Oberfläche, damit die Reibung zwischen dem Werkstück und der Werkebene möglichst positionsunabhängig und konstant ist. Die Höhendifferenz zwischen den beiden Ebenen soll etwa 25 cm betragen. Diese Konstellation erlaubt dem Greifer des EduBots die Durchführung von flexiblen Bewegungen über der Werkebene. Für den Greifer empfiehlt sich während des Greifvorgangs eine lotrechte Orientierung über dem auf der Werkebene liegenden Werkstück. Der Greifer kann in dieser Position über das im Handgelenk des EduBots befindliche Rotationsgelenk auf Orientierungen in einem Intervall von der Größe π gedreht werden. Damit sind die wesentlichen Anforderungen an die Beweglichkeit des Greifers erfüllt, wie die folgenden Abschnitte mittels der Angabe eines Bewegungsablaufs darlegen werden. Um Werkstücke greifen zu können, müssen an den beiden Fingern des Greifers zwei lineare Backen parallel zueinander montiert werden. Die Backen sollten lang genug sein, damit die Werkstücke während der Translation des Greifers nicht zur Seite wegrutschen und den Kontakt mit der Backe verlieren. Abbildung 3.2 verdeutlicht die Greifsituation.



Abbildung 3.1: Podest für den EduBot und tieferliegende Werkebene



Abbildung 3.2: Der EduBot greift ein Werkstück auf der Werkebene

3.3 Reibungsarmer Greifer

Während der Ausführung des Bewegungsablaufs zur Orientierung polygonaler Werkstücke ist neben weiteren Teilaufgaben u. a. das Werkstück zu greifen. Aus der Reibung zwischen dem Werkstück und den Backen des Greifers können Verklemmungen des Werkstücks in einer nicht stabilen Orientierung zwischen den Backen des Greifers resultieren (Canny und Goldberg, [8]). Eine Annahme von Goldbergs Verfahren setzt vereinfachend voraus, dass zwischen den Backen des Greifers und dem Werkstück keine Reibung existiert. Während der Berechnung von Aktionsplänen wird die Problematik von Verklemmungen des Werkstücks also nicht berücksichtigt. Das Auftreten einer Verklemmung in der Praxis hat zur Folge, dass die tatsächliche Orientierung des Werkstücks nach dem Greifvorgang nicht der geplanten Orientierung entspricht. Verklemmungen des Werkstücks können daher einen Fehlschlag des Orientierungsvorgangs verursachen. Um dieses Risiko zu minimieren, soll ein möglichst reibungsarmer Greifer verwendet werden.

Goldberg stellt in [14] einen Greifer vor, welcher von Canny und Goldberg [8] für die praktische Umsetzung des Verfahrens zur Orientierung polygonaler Werkstücke vorgeschlagen wird. Der Greifer besitzt eine niedrige Reibung in Richtung der parallelen Backen. Dies minimiert das Risiko von Verklemmungen des Werkstücks während des Greifvorgangs. Gleichzeitig besitzt der Greifer eine hohe Reibung in eine zweite Richtung, um dem Effekt der Schwerkraft auf das Werkstück entgegenzuwirken, während sich der Greifer in der Luft befindet. Dieser Greifer soll auch im vorliegenden Konzept für die Orientierung von Werkstücken verwendet werden. Die zwei an den Fingern des EduBots zu montierenden linearen Backen (siehe Abschnitt 3.2) müssen der folgenden Beschreibung von Goldbergs reibungsarmen Greifer genügen.

Der Greifer besteht aus zwei linearen, parallelen Backen. Die Backen können orthogonal zu ihrer Richtung aufeinander zu und voneinander weg bewegt werden. An der Innenseite einer der Backen ist parallel zu der Richtung der Backen eine Laufschiene befestigt (siehe Abbildung 3.3). Die Laufschiene ermöglicht einem sogenannten Wagen eine gleitende Bewegung parallel zu den Backen. Der Wagen weist in die Richtung der Innenseite des Greifers eine rauhe Oberfläche auf. Die zweite Backe besitzt in die Richtung der Innenseite des Greifers ebenfalls eine rauhe Kontaktoberfläche.



Abbildung 3.3: Die linearen Backen eines reibungsarmen Greifers

Abbildung 3.4 illustriert in der linken Darstellung ein zwischen den Backen eines Greifers verklemmtes Werkstück, während das Werkstück in

der mittleren Darstellung eine Drehbewegung ausführt. In der rechten Darstellung hat das Werkstück eine stabile Orientierung eingenommen.



Abbildung 3.4: Die Laufschiene minimiert auftretende Reibung

3.4 Translationsdistanz

Der Bewegungsablauf zur Orientierung von polygonalen Werkstücken umfasst u. a. eine lineare Translation des Greifers. Der Greifer bewegt sich dabei orthogonal zu seiner Orientierung und beeinflusst die Orientierung des Werkstücks. Der weitere Bewegungsablauf steht unter der Prämisse, dass die Translationsdistanz π des Greifers groß genug ist, um das Werkstück in eine stabile Orientierung zu überführen (vgl. Abschnitt 2.1.6). Im Folgenden werden Ergebnisse von Peshkin und Sanderson [28] dargestellt, welche eine Vorstellung von der Bewegung des Werkstücks während der Translation des Greifers vermitteln. Weiterhin werden Erkenntnisse bezüglich der Berechnung einer geeigneten Translationsdistanz π erläutert.

Es sei ein polygonales Werkstück mit seinem Massenschwerpunkt *cm* (*center of mass*) sowie eine lineare Backe gegeben. Weiterhin habe die lineare Backe eine Kontaktstelle mit einer Ecke des Werkstücks. Die Backe führe eine lineare Translation aus. Es existiere keine Reibung zwischen der linearen Backe und dem Werkstück. Die Reibung zwischen dem Werkstück und der Werkebene sei unabhängig von der Position und der Geschwindigkeit des Werkstücks. Peshkin und Sanderson gehen von einem quasi-statischen Modell aus. Dies bedeutet, dass durch die Reibung zwischen dem Werkstück und der Werkebene jegliche Bewegungsenergie des Werkstücks schnell neutralisiert wird. Unter diesen Annahmen kann die Bewegung des Werkstücks während der linearen Translation der Backe als eine Rotation um ein Rotationszentrum (*center of rotation*) beschrieben werden.

Die Position des Rotationszentrums genau zu berechnen ist schwierig, da sie bereits bei geringsten Unebenheiten der Werkebene oder der aufliegenden Fläche des Werkstücks stark variieren kann. Die Geschwindigkeit der Rotation ist unter den gegebenen Annahmen allerdings beschränkt. Sei α der Winkel zwischen der linearen Backe und der Translationsrichtung der Backe (siehe Abbildung 3.5). Sei *c* der Abstand der Kontaktstelle der Backe und des Werkstücks zum Massenschwerpunkt des Werkstücks. Sei *a* der Radius des kleinsten Kreises, dessen Mittelpunkt der Massenschwerpunkt des Werkstücks ist und der das Werkstück vollständig beinhaltet. Betrachte die Gerade, welche durch den Massenschwerpunkt des Polygons und durch die Kontaktstelle verläuft. Sei β der positive Winkel zwischen dieser Geraden und der Translationsrichtung der linearen Backe. Im Verlauf der Rotation des Werkstücks ändere sich der Winkel β von β^{start} zu β^{end} . Gleichung 3.1 leitet eine obere Schranke h für die Länge der Translationsdistanz der linearen Backe ab. Diese obere Schranke garantiert die Änderung des Winkels β von β^{start} zu β^{end} im Verlauf der Translation der linearen Backe.

$$f(\beta) := -c\cos\left(\beta\right) - \frac{a^{2}\sin\left(\alpha\right)}{2c}\log\left|\frac{1+\sin\left(\alpha+\beta\right)}{1-\sin\left(\alpha+\beta\right)}\right|$$
$$g(\beta) := \frac{-c^{3}}{a^{2}}\left(1-\sin\left(\alpha\right)\right)\left(\sin\left(\alpha+\beta\right) - \frac{\sin^{3}\left(\alpha+\beta\right)}{3}\right)$$
$$h(\beta^{start}, \beta^{end}) := f(\beta^{end}) + g(\beta^{end}) - f(\beta^{start}) - g(\beta^{start})$$
(3.1)

Abbildung 3.5: Eine lineare Backe schiebt ein polygonales Werkstück

Die beschriebenen Erkenntnisse von Peshkin und Sanderson und das Verfahren zur Orientierung polygonaler Werkstücke basieren im Wesentlichen auf den gleichen Annahmen. Die Gleichung 3.1 kann allerdings nicht zur Berechnung der oberen Schranke der Translationsdistanz für die praktische Umsetzung verwendet werden, da die Werte β^{start} , β^{end} und c während der Ausführung eines Aktionsplans nicht bekannt sind. Immerhin können anhand der Formeln einige allgemeine Aussagen getroffen werden. So schreibt

Goldberg [15], dass die obere Schranke für die Translationsdistanz in ungünstigen Fällen unendlich große Werte annehmen kann. In einem solchen Fall kann nicht garantiert werden, dass sich das Werkstück nach der Translation des Greifers in einer stabilen Orientierung befindet. Dies birgt gleichzeitig die Gefahr eines Fehlschlags des Verfahrens. Laut Goldberg lässt sich nach der Abarbeitung des ersten Planwinkels und im weiteren Verlauf der Durchführung ein endlich großer Wert der oberen Schranke h für die Translationsdistanz garantieren.

Sei μ der Reibungskoeffizient, welcher die Reibung zwischen der linearen Backe und dem Werkstück angibt. Die bisherigen Ergebnisse sind unter der idealisierten Annahme gültig, dass zwischen dem Werkstück und der linearen Backe keine Reibung existiert ($\mu = 0$). Für die praktische Umsetzung mit dem reibungsarmen Greifer aus dem vorigen Abschnitt ist Reibung zwischen dem Werkstück und dem Greifer zu erwarten. Aufgrund dieses Umstands werden nachfolgend weitere Ergebnisse von Peshkin und Sanderson [27] betrachtet. Diese beschreiben die Bewegung eines Werkstücks im Fall $\mu > 0$, aber mit ansonsten unveränderten Annahmen.

Der Kontakt zwischen dem Werkstück und der linearen Backe besteht aus einer Ecke des Werkstücks und einem Punkt auf der Oberfläche der linearen Backe. Bleibt das Werkstück während der Translation der Backe mit genau diesem Punkt der Backe in Kontakt, spricht man von einer haftenden Bewegung des Werkstücks. Ändert sich der am Kontakt beteiligte Punkt auf der Oberfläche der linearen Backe, wird die Bewegung des Werkstücks als gleitend klassifiziert. Beide Arten der Bewegung können während der Translation der linearen Backe auftreten. Der Winkel β wird während der Translation der linearen Backe stumpfer, und es kann infolgedessen zu einem Wechsel von haftender zu gleitender Bewegung des Werkstücks kommen. Sei $v := tan^{-1}(\mu)$. Der Winkel, bei welchem der Wechsel stattfindet, kann nach folgender Gleichung berechnet werden:

$$\beta^{trans} := \frac{c^2 \tan\left(\alpha \pm \upsilon\right) \pm \left[c^4 \tan^2\left(\alpha \pm \upsilon\right) - 4a^2 \left(a^2 + c^2\right)\right]^{\frac{1}{2}}}{2a^2} \tag{3.2}$$

Die obere Schranke der Translationsdistanz berechnet sich wie folgt:

$$f(\beta) := \frac{-c\sin(\alpha - v + \beta)}{\sin(\alpha)} - \frac{a^2}{2c\sin(\alpha)} \log \left| \frac{1 + \sin(\alpha - v + \beta)}{1 - \sin(\alpha - v + \beta)} \right|$$

$$g(\beta) := \frac{c^2 + a^2}{2c} \log \left| \frac{1 - \cos\left(\beta\right)}{1 + \cos\left(\beta\right)} \right|$$

$$h(\beta^{start}, \beta^{trans}, \beta^{end}) := f(\beta^{end}) - f(\beta^{trans}) + g(\beta^{trans}) - g(\beta^{start}) \quad (3.3)$$

Der Reibungskoeffizient wird im Intervall (0, 1] angenommen. Damit gelten die folgenden Aussagen:

$$\mu \in (0, 1] \Rightarrow v \in (\tan^{-1}(0), \tan^{-1}(1)] \Rightarrow v \in (0, 0.786]$$
(3.4)

Goldbergs Methode sieht eine Translationsrichtung der Backe orthogonal zu ihrer Orientierung vor, daher gilt $\alpha = \frac{\pi}{2}$. Die Werte von v sind gemäß Gleichung 3.4 beschränkt und es ist der Fall $\beta^{trans} \approx v$ möglich. In dieser Situation gilt:

$$\lim_{\beta^{trans} \to \upsilon} 1 - \sin\left(\alpha - \upsilon + \beta^{trans}\right) = 0$$

$$\Rightarrow \lim_{\beta^{trans} \to \upsilon} f(\beta^{trans}) = -\infty$$

$$\Rightarrow \lim_{\beta^{trans} \to \upsilon} h(\beta^{start}, \beta^{trans}, \beta^{end}) = +\infty$$
(3.5)

Die obere Schranke h konvergiert also für $\beta^{trans} \rightarrow v$ gegen $+\infty$. Nach diesen Erkenntnissen kann Folgendes festgehalten werden: unabhängig davon, ob Reibung zwischen dem Greifer und dem Werkstück angenommen wird oder nicht, kann die obere Schranke h der Translationsdistanz unendlich große Werte annehmen. Aufgrund der Unkenntnis der Werte β^{start} , β^{end} und c während der Ausführung eines Aktionsplans lässt sich die Schranke mit den gegebenen Formeln nicht berechnen. Die von Goldberg betreffend der Translationsdistanz des Greifers aufgestellte Annahme ist in der Praxis aus den genannten Gründen nicht mit Gewissheit erfüllbar. Als Resultat dieses Abschnitts wird die Empfehlung gegeben, die Translationsdistanz des Greifers im Kontext des vorliegenden Konzepts so groß wie möglich zu wählen. Dadurch soll das Risiko, dass das Werkstück aufgrund einer zu kurz gewählten

Translationsdistanz des Greifers in einer nicht stabilen Orientierung endet, soweit wie möglich minimiert werden.

3.5 Translationsgeschwindigkeit

Goldbergs Verfahren zur Orientierung polygonaler Werkstücke geht von einem quasi-statischen Modell aus. Dies bedeutet, dass die Bewegungen des Werkstücks und des Greifers derart langsam sind, dass die Bewegungsenergie des Werkstücks von der auftretenden Reibung zwischen dem Werkstück und der Werkebene dominiert wird (Peshkin [27]). Die quasi-statische Annahme ist nur in trivialen Fällen exakt, aber als Annäherung der Realität in vielen Fällen nützlich. Peshkin untersucht geeignete Geschwindigkeiten, für welche die Annahmen des quasi-statischen Modells in der Praxis vertretbar sind. Seine Erkenntnisse sollen in der praktischen Umsetzung Anwendung finden, um eine geeignete Geschwindigkeit für die Translationsbewegung des Greifers zu bestimmen.

Als Ausgangspunkt für seine Betrachtungen legt Peshkin eine Schranke für den maximalen Fehler fest, welcher durch die Vernachlässigung der Bewegungsenergie des Werkstücks auftreten darf. Die Fehlerschranke besteht aus zwei Komponenten: dem rotatorischen Fehler Θ (beschrieben im Bogenmaß) und dem translatorischen Fehler X (beschrieben in Zentimetern). Bei gegebener Fehlerschranke kann eine charakteristische Geschwindigkeit angegeben werden, bei der sich die kinetischen und die quasi-statischen Effekte annähernd ausgleichen, sodass der auftretende Fehler die Fehlerschranke nicht übersteigt. Diese Geschwindigkeit wird von Peshkin als obere Schranke für die zu verwendende Geschwindigkeit empfohlen. Die charakteristische Geschwindigkeit ist abhängig von den Details des betrachteten Kontakts. Peshkin analysiert zwei verschiedene Situationen und entwickelt die folgende Formel für die charakteristische Geschwindigkeit v, welche einen maximalen translatorischen Fehler X zulässt:

$$v^2 = 2g\mu X \tag{3.6}$$

Dabei ist $g \approx \frac{9,81m}{s^2}$ die Konstante der Fallbeschleunigung. Weiterhin gibt der Reibungskoeffizient μ die Reibung zwischen dem Werkstück und der Werkebene an. Peshkin ermittelt μ , indem er die Oberfläche der Werkebene kippt, bis ein manuelles Anstoßen des Werkstücks zu einer kontinuierlichen Gleitbewegung des Objekts auf der Werkebene führt. Der Reibungskoeffizient ergibt sich als $\mu = \tan^{-1}(\theta)$, wobei θ derjenige Neigungswinkel der Werkebene ist, bei dem die kontinuierliche Gleitbewegung des Werkstücks zu einer kontinuierliche Gleitbewegung des Werkstücks zu einer kontinuierliche Gleitbewegung des Werkstücks zu einer kontinuierlichen Kerkebene ist, bei dem die kontinuierliche Gleitbewegung des Werkstücks zu einer kontinuierliche Kerkebene kingt.

Im Zuge der praktischen Umsetzung des vorliegenden Konzepts soll der Reibungskoeffizient zwischen den zu verwendenden Werkstücken und der Werkebene auf die beschriebene Weise bestimmt werden. Nach der Festlegung einer Fehlerschranke gibt die Formel 3.6 Aufschluss über eine geeignete Geschwindigkeit des Greifers, sodass die Fehlerschranke durch den auftretenden translatorischen Fehler nicht überschritten wird. Diese Geschwindigkeit soll während der Ausführung von Aktionsplänen, insbesondere während der Translation des Greifers, beachtet werden. Als Konsequenz aus diesen Planungen ergibt sich eine funktionale Anforderung an die Steuerungssoftware des EduBots: die Software muss die Möglichkeit bieten, die Geschwindigkeit der Bewegungen des Manipulators während der Ausführung von Aktionsplänen gemäß den erlangten Vorgaben festzulegen.

3.6 Ein Bewegungsablauf für den EduBot

Dieser Abschnitt entwickelt einen Bewegungsablauf für die Durchführung von Aktionsplänen mit dem EduBot 250M. Der von Goldberg beschriebene Bewegungsablauf für den Greifer dient dabei als Ausgangspunkt und wird hier zur Erinnerung nochmal aufgeführt:

- Drehe den Greifer auf den Winkel α bezüglich einer festen Normalstellung.
- Führe eine Translation in Richtung $\alpha + \frac{\pi}{2}$ um die Distanz δ aus.
- Schließe beide Backen gleichzeitig, bis das Werkstück eingeklemmt ist.
- Führe eine Translation in Richtung $-(\alpha + \frac{\pi}{2})$ um die Distanz δ aus.
- Öffne beide Backen.

Der EduBot besitzt als stationär betriebener Roboter einen kleinen Arbeitsraum. Der vom Greifer erreichbare Bereich auf der Werkebene beschränkt die maximale Länge der Translationsdistanz δ in obigem Bewegungsablauf. Die erste Translation des Ablaufs startet immer an der festen Ausgangsposition des Werkstücks, die zweite Translation bringt das Werkstück wieder an den Ausgangspunkt zurück. Die Richtung der ersten Translation wird durch einen Winkel des Aktionsplans bestimmt, welcher im Intervall $(-2\pi, +2\pi)$ liegt. Die möglichen Richtungen der ersten Translation decken daher alle Winkel zwischen 0 und 2π ab. Um eine lange Translationsdistanz des Greifers in alle Richtungen von 0 bis 2π zu garantieren, ist der Ausgangspunkt der Translation im Zentrum des durch den Greifer auf der Werkebene erreichbaren Bereichs zu wählen (siehe Abbildung 3.6(i)).

Das vorliegende Konzept schlägt eine Modifikation des Bewegungsablaufs vor, um eine längere Translationsdistanz zu ermöglichen. Die wesentliche Idee dabei ist, die Translationsbahn des Greifers fest in der Richtung



Abbildung 3.6: Der Greifer führt Translationen aus in (i) variable Richtungen, (ii) eine feste Richtung.

orthogonal zu der unveränderlichen Normalstellung zu wählen (siehe Abbildung 3.6(ii)). Vor Beginn der ersten Translation nimmt der Greifer unabhängig vom aktuellen Winkel α des Aktionsplans seine Normalstellung ein und folgt der festen Translationsbahn. Nach der Beendigung der Translation wird das Werkstück gegriffen. Um den Effekt der Reorientierung des Greifers relativ zum Werkstück zu erzielen wird nicht der Greifer, sondern das Werkstück gedreht. Hierfür sind geeignete Drehwinkel α' zu berechnen. Die Drehung erfolgt im Anschluß an die Greifaktion am Ende des Bewegungsablaufs. Somit wird die Ausführung der Reorientierung vom Anfang des (n + 1)-ten auf das Ende des *n*-ten Bewegungsablaufs vorgezogen. Im letzten Durchgang des Bewegungsablaufs ist zu beachten, dass keine Drehung mehr nötig ist und die Rotation entfällt. Es ergibt sich der folgende Bewegungsablauf, welcher mit dem neu berechneten Drehwinkel α' parametrisiert wird:

- Drehe den Greifer in die Orientierung der festen Normalstellung.
- Führe eine Translation in der Richtung $\frac{\pi}{2}$ um die Distanz δ aus.
- Schließe beide Backen gleichzeitig, bis das Werkstück eingeklemmt ist.
- Führe eine Translation in der Richtung $-\frac{\pi}{2}$ um die Distanz δ aus.
- Drehe den Greifer auf den Winkel α' .
- Öffne den Greifer.

Anhand eines Beispiels soll illustriert werden, wie die neuen Drehwinkel α' für das Werkstück zu berechnen sind. Abbildung 3.7 vergleicht die Ausführung des originalen Bewegungsablaufs mit der Durchführung des modifizierten Bewegungsablaufs. Der Aktionsplan besteht aus den Winkeln 0 Grad, 60 Grad und 10 Grad. Die Winkel sind im mathematischen Sinn als Drehung gegen den Uhrzeigersinn zu verstehen. Die beiden Ausführungen des Aktionsplans werden von links nach rechts dargestellt. Der erste Drehwinkel des

Aktionsplans ist wie immer 0 Grad. Die Situation ist unabhängig vom verwendeten Bewegungsablauf vor und nach der ersten Translation des Greifers gleich. Im nächsten Schritt ist der Greifer nach dem originalen Ablauf um 60 Grad gegen den Uhrzeigersinn zu drehen. Im modifizierten Ablauf hingegen muss das Werkstück offensichtlich um 60 Grad in die entgegengesetzte Richtung gedreht werden, um es bezüglich des Greifers in die gleiche Orientierung zu bringen. Dies ist in der dritten Spalte der Abbildung dargestellt. Die Orientierungen des Werkstücks bezüglich des Greifers sind auch nach der nun folgenden Translation in beiden Fällen gleich, wie in der vierten Spalte zu sehen ist. Nun ist der Greifer nach dem originalen Bewegungsablauf auf den dritten Winkel des Aktionsplans, also auf 10 Grad bezüglich der Normalstellung zu drehen. Dies entspricht einer relativen Drehung des Greifers um 50 Grad im Uhrzeigersinn. Für den modifizierten Ablauf ergibt sich eine Drehung des Werkstücks um 50 Grad gegen den Uhrzeigersinn. Aus diesem Beispiel ergibt sich, dass für die Berechnung der neuen Drehwinkel α' die relativen Bewegungen des Greifers aus zwei aufeinander folgenden Winkeln des Aktionsplans ermittelt werden müssen. Die relativen Bewegungen des Greifers müssen jeweils negiert werden, da nicht der Greifer, sondern das Werkstück zu drehen ist:



Abbildung 3.7: Der Greifer führt Translationen aus (i) in variable Richtungen, (ii) in eine feste Richtung.

In Abschnitt 3.4 wurde auf die Bewegung des Werkstücks während der Translation des Greifers genauer eingegangen. Diese Bewegung entspricht einer Rotation um einen bestimmten Punkt. Dabei wird das Werkstück neben seiner relativen Orientierung auch seine relative Position zum Greifer verändern. In Abbildung 3.8 befindet sich das Werkstück anfangs mittig zur Backe des Greifers. Nachdem es eine stabile Orientierung erlangt hat, ist es am dagegen äußeren Rand des Greifers positioniert.

Nach der Translation wird das Werkstück gegriffen und an einem geeigneten Ort in durch eine Rotation die nächste Orientierung α' gebracht. Aufgrund der Unkenntnis der genauen Position des Werkstücks zwischen den Backen des Greifers ist der Aufenthaltsort des Werkstücks nach diesem Schritt nicht bekannt. Die Position des Werkstücks kann bei Kenntnis des Drehwinkels lediglich auf einen rechteckigen Bereich eingeschränkt werden (siehe Abbildung 3.9). Dies wäre allerdings schwierig, da nicht bekannt ist, wie weit der Greifer geschlossen ist. Allgemein liegen die Werkstücke, dies gilt unabhängig vom Drehwinkel und Öffnungsgrad des Greifers, innerhalb eines kreisförmigen Bereichs. Sei r der Radius dieses kreisförmigen Bereichs.



Abbildung 3.8: Die relative Position eines Werkstücks bezüglich der Backe des Greifers ändert sich während einer Translation



Abbildung 3.9: Die Position von Werkstücken nach der Rotation liegt innerhalb eines kreisförmigen Bereichs

Der Greifer muss nach der Rotation des Werkstücks in die Startposition der Translation manövriert werden. Weiterhin müssen die folgenden drei Bedingungen erfüllt werden:

- 1. Der Greifer muss in der Startposition der Translation die Orientierung der Normalstellung einnehmen.
- 2. Während der Bewegung in die Startposition darf der Greifer zu keinem Zeitpunkt mit dem Werkstück kollidieren.
- 3. Das Werkstück muss an der Startposition der Translation zwischen den Backen des Greifers liegen.

Bedingung 2 soll verhindern, dass die Orientierung des Werkstücks durch einen unbeabsichtigten Kontakt verändert wird. Bedauerlicherweise existiert keine Startposition, welche die dritte Bedingung unabhängig von der Position des Werkstücks immer erfüllt. Dies wird in Abbildung 3.10 illustriert. Die Abbildung zeigt zwei mögliche Positionen des Werkstücks, nachdem der Greifer auf eine Orientierung von -40 Grad gedreht wurde. Nimmt der Greifer die durch die blauen Striche symbolisierte Position ein, so liegt lediglich eines der beiden gedachten Werkstücke zwischen den Backen. Das Gleiche gilt für die Position des durch die schwarzen Striche symbolisierten Greifers.



Abbildung 3.10: Es existiert keine einheitliche Startposition des Greifers für die Translation

Um das Problem zu lösen, soll die Annahme aufgehoben werden, dass das Werkstück während der Translation zwischen den Backen des Greifers liegt. Der Greifer wird vor der Translation geschlossen und nimmt eine Position außerhalb des kreisförmigen Bereichs ein, welcher das Werkstück enthält. Von dort beginnt die Ausführung der Translation. Im Verlauf der Translation entsteht ein Kontakt zwischen der Außenseite einer Backe des Greifers und dem Werkstück. Nach Beendigung der Translation wird der Greifer geöffnet und in eine Position gebracht, in der das Werkstück wieder zwischen den Backen liegt. Dies ist auch bei Unkenntnis der genauen Position des Werkstücks ohne Probleme möglich, da der Greifer zwischenzeitlich nicht gedreht wird. Es folgt die ausführliche Beschreibung des endgültigen Bewegungsablaufs:

- Manövriere den Greifer an den Startpunkt der Translation.
- Schließe den Greifer.
- Drehe den Greifer in die Orientierung der festen Normalstellung.
- Führe eine Translation in der Richtung $\frac{\pi}{2}$ um die Distanz δ aus. Halte dabei die Orientierung der Backen konstant.
- Manövriere den Greifer aufwärts.
- Öffne den Greifer.
- Bringe den Greifer in eine Position, in der das Werkstück zwischen beiden Backen liegt.
- Schließe beide Backen gleichzeitig, bis das Werkstück eingeklemmt ist.

3.7. DETAILS

- Führe eine Translation in der Richtung $-\frac{\pi}{2}$ um die Distanz $\delta -\frac{3}{2}r$ aus.
- Drehe den Greifer und das Werkstück auf den Winkel α' .
- Öffne den Greifer.
- Manövriere den Greifer nach oben.

Während der Erarbeitung des Bewegungsablaufs wurde darauf geachtet, dass die Orientierung des Werkstücks bezüglich des Greifers während der Translation und während des Greifvorgangs von den Modifikationen nicht verändert wird. Da sich in diesem Sinne im Vergleich zum originalen Ablauf keine Abweichungen ergeben, endet das Werkstück nach der Ausführung des Aktionsplans analog zum originalen Bewegungsablauf in einer von eventuell mehreren Endorientierungen.

3.7 Details

Dieser Abschnitt beschreibt die Details der Durchführung des im vorigen Abschnitts erarbeiteten Bewegungsablaufs.

Rotation des Werkstücks Die Rotation des Werkstücks ist in lotrechter Orientierung des Greifers über der Werkebene mit einer Drehung von Gelenk vier des Manipulators zu bewerkstelligen. Die Rotation ist genau auf der Bahn zu bewerkstelligen, welcher der Greifer des Edu-Bots bei der Drehung um das erste Gelenk folgt, wenn der Greifer eine lotrechte Position über der Werkebene eingenommen hat. Gleichzeitig muss die für die Rotation des Werkstücks gewählte Position σ_{rot} möglichst weit entfernt vom Endpunkt der Translation liegen.

Sei $\omega := 0$ die Orientierung der Normalstellung des Greifers. Das Intervall (β_{min}, β_{max}) beschränke die möglichen Orientierungen des Greifers aufgrund der Rotationsmöglichkeit von Gelenk vier um 180 Grad. Es gilt $\beta_{max} - \beta_{min} = \pi$ und somit $-\pi \leq \beta_{min} \leq \omega = 0 \leq \beta_{max} \leq \pi$ (vgl. Abbildung 3.11). Die neu berechneten Drehwinkel α' für den Greifer können durchaus größer als π , bzw. kleiner als $-\pi$ sein. Die Drehung des Greifers auf eine solche Orientierung kann nicht in einem Schritt erfolgen, da der hierfür nötige Positionswert von Gelenk vier außerhalb des zulässigen Bereichs liegen würde. Die Rotation muss in diesem Fall in mehreren Teilschritten bewerkstelligt werden.

Algorithmus 3.7.1 illustriert die Vorgehensweise für eine Drehung. Sei $\alpha' > \beta_{max}$. Der Greifer wird zu Beginn einer while-Schleife auf die Orientierung β_{max} gedreht. Anschließend wird der Greifer geöffnet und aufwärts bewegt, bis er sich in einem Sicherheitsabstand zum

Werkstück befindet. Nun kann er um 180 Grad gedreht werden. Anschließend wird der Greifer abwärts bewegt, bis die Backen einen Kontakt mit der Werkebene hergestellt haben. Jetzt liegt auch das Werkstück wieder zwischen den Backen. Der Greifer wird geschlossen und kann zu Beginn des nächsten Durchlaufs der while-Schleife um 180 Grad auf die Orientierung β_{max} gedreht werden. Jeder Durchlauf der Schleife vermindert bzw. erhöht den Drehwinkel α' um 180 Grad, bis eine einzige Drehung von Gelenk vier ausreicht, um das Werkstück in

PSfrag replaced in Grientierung α' zu überführen. Für den Fall $\alpha' < \beta_{min}$ ergibt sich eine ähnliche Vorgehensweise.



Abbildung 3.11: Bewegungsfreiheit des Greifers

Algorithmus 3.7.1	Rotation	eines	Werkstücks
-------------------	----------	-------	------------

Eingabe: Zielorientierung α' des Greifers, β_{min} und β_{max}

- 1: while $\alpha' > \beta_{max}$ do
- 2: Drehe den Greifer auf die Orientierung β_{max}
- 3: Öffne den Greifer
- 4: Führe den Greifer aufwärts, bis er genügend Abstand zum Werkstück besitzt
- 5: Drehe den Greifer um 180 Grad auf die Orientierung β_{min}
- 6: $\alpha' \leftarrow \alpha' \pi$
- 7: Führe den Greifer abwärts, bis die Backen einen Kontakt mit der Werkebene herstellen
- 8: Schließe den Greifer
- 9: end while

10: while $\alpha' < \beta_{min}$ do

- 11: Drehe den Greifer auf die Orientierung β_{min}
- 12: Offne den Greifer
- Führe den Greifer aufwärts, bis er genügend Abstand zum Werkstück besitzt
- 14: Drehe den Greifer um 180 Grad auf die Orientierung β_{max}
- 15: $\alpha' \leftarrow \alpha' + \pi$
- 16: Führe den Greifer abwärts, bis die Backen einen Kontakt mit der Werkebene haben
- 17: Schließe den Greifer
- 18: end while
- 19: Drehe den Greifer auf die Orientierung α'

Translation des Greifers Aus Abschnitt 3.4 ging die Forderung einer

50

3.7. DETAILS

möglichst großen Translationsdistanz des Greifers hervor. In diesem Sinne sollen die Start- und Endpunkte der Translation nahe an gegenüberliegenden Rändern des vom Greifer auf der Werkebene erreichbaren Bereichs gewählt werden. Doch wo genau sind die Start- und Endpunkte zu platzieren?

Befindet sich der Endpunkt τ_{end} der Translation nahe am Rand des vom Greifer auf der Werkebene erreichbaren Bereichs, so ergibt sich aufgrund der Bauart des Manipulators in dieser Position eine Schräglage des Greifers über der Werkebene. Nach dem Greifen in dieser Position befindet sich das Werkstück in einer Schräglage zwischen den Backen des Greifers. Beim nächsten Öffnen des Greifers könnte das Werkstück aus dem Greifer fallen und seine Orientierung verändern. Der Endpunkt τ_{end} der Translation ist daher nahe der bereits beschriebenen Bahn zu platzieren, welcher der Greifer des EduBots bei der Rotation um das erste Gelenk folgt, wenn der Greifer eine lotrechte Position über der Werkebene eingenommen hat. Hierdurch werden Schräglagen des Greifers während des Greifvorgangs und die daraus eventuell entstehenden Komplikationen vermieden.

Für die Wahl des Startpunkts τ_{start} der Translation ist nun die gedachte Linie zwischen dem Endpunkt der Translation τ_{end} und der Position σ_{rot} der Rotation des Werkstücks um $\frac{3}{2}r$ zu verlängern. Abbildung 3.12 skizziert die Bahn eines Greifers während der Rotation um die Basis des zugehörigen Manipulators. Ebenfalls dargestellt ist die mögliche Wahl der Endpunkte einer Translationsbahn.



Abbildung 3.12: Die Wahl einer Translationsbahn

Für die Durchführung der Translation muss die Steuerungssoftware die Möglichkeit bieten, den Greifer entlang einer linearen, kartesischen Bahn zu manövrieren und gleichzeitig die Orientierung der Backen konstant zu halten.

Greifvorgang Während des Greifvorgangs bewegen sich die Backen auf-

grund der Bauart des Greifers auf Kreisbahnen, wodurch das Höhenniveau des Greifers sinkt bzw. über die Backen eine Kraft auf die Werkebene ausgeübt wird. Um einen möglichst reibungslosen Kontakt zwischen der Werkebene und den Backen aufrecht zu erhalten, soll das Höhenniveau des Greifers während des Greifvorgangs konstant gehalten werden. Simultan zum Schließen des Greifers sorgt eine Anpassung der Positionswerte der Gelenke zwei und drei des Manipulators für eine ausgleichende Bewegung.

Um den Greifer zu schließen, wird der Positionswert des entsprechenden Servomotors von der Steuerungssoftware iterativ variiert. Ist das Werkstück fest zwischen den Backen des Greifers eingeklemmt, so ist dem Greifer eine Fortsetzung des Schließvorgangs aus mechanischen Gründen unmöglich. In einer solchen Situation der Verklemmung des Werkstücks bleiben weitere Steuersignale zum Schließen des Greifers wirkungslos. Es ist technisch nicht möglich, die Fixierung des Werkstücks zwischen den Backen des Greifers zu Erkennen und mit der Beendigung des Greifvorgangs zu reagieren, denn der Controller des EduBots kommuniziert keine Rückmeldung über die tatsächlich eingenommenen Positionswerte des Servomotors an den Steuerungsrechner. Um Werkstücke von beliebigem Durchmesser greifen zu können, muss die Steuerungssoftware das Senden der Steuersignale zum Schließen des Greifers, so weit wie es technisch möglich, ist fortführen.

Von dem Moment der mechanischen Verklemmung des Werkstücks zwischen den Backen des Greifers an wird der Greifvorgang wie beschrieben weitergeführt. Der bewerkstelligte Höhenausgleich ergibt eine Bewegung des Greifers von der Werkebene weg, welche aufgrund der verhinderten Schließbewegung der Backen aber überflüssig ist. Nach der Beendigung des Greifvorgangs befindet sich der Greifer mit dem Werkstück aufgrund des zu lange realisierten Höhenausgleichs über dem anfänglichen Höhenniveau. Da der Durchmesser des Werkstücks beliebig sein kann, ist der Zeitpunkt des Eintretens der Verklemmung und folglich das genaue Höhenniveau des Greifers nach dem Greifvorgang unbekannt.

Öffnen des Greifers Der Greifer wird durch die schrittweise Variation der Positionswerte des entsprechenden Servomotors geöffnet. Die Unkenntnis des genauen Höhenniveaus des Greifers hat zur Folge, dass das Werkstück während des Öffnens des Greifers auf die Werkebene fallen könnte. Folglich muss zeitgleich zum Öffnen des Greifers über die Gelenke zwei und drei des Manipulators ein Höhenausgleich in die Richtung der Werkebene stattfinden. Dies garantiert einen Kontakt zwischen dem Greifer und der Werkebene, rechtzeitig bevor das Werkstück abgesetzt wird.

3.8 Zusammenfassung

In diesem Kapitel wurden einige Aspekte im Zusammenhang mit der praktischen Umsetzung von Goldbergs Ansatz erläutert. Unter Berücksichtigung der getroffenen Feststellungen wurde ein Konzept für die Umsetzung vorgestellt.

Das Konzept sieht lineare Backen vor, welche an dem Greifer des Edu-Bots befestigt werden. Um eine Reduzierung der Reibung zwischen den Backen und den Werkstücken zu erreichen, ist an der Innenseite einer der beiden Backen eine bewegliche Laufschiene zu befestigen. Eine geeignete Translationsdistanz des Greifers kann mittels einer mathematischen Formel bestimmt werden, welche angegeben wurde. Weiterhin wurde festgestellt, dass sich die für eine sichere Orientierung von Werkstücken notwendige Länge der Translationsdistanz des Greifers nicht von oben beschränken lässt. Dieser Umstand motivierte die Zielsetzung, im Rahmen der Möglichkeiten des EduBots eine möglichst lange Translationsdistanz zu realisieren. Es wurde ein modifizierter Bewegungsablauf angegeben, welcher einem stationär befestigten Gelenkarmroboter im Kontext der Orientierung von polygonalen Werkstücken verglichen mit dem originalen Ablauf eine längere Translationsdistanz ermöglicht. Das Kapitel schließt mit einer Ausarbeitung einiger Details des modifizierten Bewegungsablaufs ab. 54

Kapitel 4

Umsetzung

Im Rahmen dieser Arbeit wurde das in Kapitel 3 beschriebene Konzept mit dem Gelenkarmroboter EduBot 250M umgesetzt. Im Zuge der Umsetzung wurde ein technischer Ansatz zur Ansteuerung des Roboters gewählt. Weiterhin wurde eine Steuerungssoftware implementiert, welche den Roboter dazu veranlasst, polygonale Werkstücke unter Verwendung des Bewegungsablauf aus Abschnitt 3.6 zu orientieren. Diese Steuerungssoftware besteht aus einem allgemein verwendbaren Softwarepaket zur Ansteuerung des Edu-Bots, sowie aus einer Klasse, welche den Bewegungsablauf zur Orientierung von polygonalen Werkstücken aufbauend auf der allgemeinen Software implementiert. Weiterhin erhielt die Java-Anwendung zur Berechnung von Aktionsplänen eine Anbindung an die implementierte Steuerungssoftware, um dem Benutzer die direkte Ausführung von berechneten Aktionsplänen mit dem EduBot zu ermöglichen. Abschließend wurde die entstandene Lösung getestet. Die folgenden Abschnitte beschreiben einige Aspekte der Umsetzung.

4.1 Technische Konfiguration

Dieser Abschnitt beschreibt die Wahl einer technischen Konfiguration für die Ansteuerung des EduBots im Kontext der praktischen Umsetzung von Goldbergs Methode. Zur Auswahl standen mehrere verschiedene Konfigurationen. Diese wurden in Abschnitt 2.3.2 vorgestellt. An dieser Stelle werden die Konfigurationen nochmals aufgeführt:

- Der Controller des EduBots ist an die serielle Schnittstelle des Steuerrechners angeschlossen. Die Ansteuerung des EduBots basiert entweder auf dem Befehlsprotokoll des Controllers, oder auf der grafischen Anwendung der Firma Robotica Limited.
- Der Controller des EduBots ist an den mobilen Pioneer Roboter angeschlossen. Der Pioneer Roboter ist an die serielle Schnittstelle des Steu-

errechners angeschlossen. Für die Ansteuerung des EduBots seitens des Steuerrechners wird entweder auf die Schnittstelle des Betriebssystems des Pioneer Roboters, auf das Befehlsprotokoll des Controllers, oder auf die Schnittstelle der Software-Bibliothek ARIA zurückgegriffen.

Die Wahl einer Konfiguration muss die funktionalen Anforderungen an die Steuerungssoftware berücksichtigen, da die Fähigkeiten der Steuerungssoftware von der gewählten Konfiguration limitiert werden. Die funktionalen Anforderungen wurden aus dem Inhalt der Abschnitte 3.6 und 3.7 abgeleitet:

- An-/Ausschalten Der Betrieb des Roboters erfordert die Fähigkeit der Steuerungssoftware, den Roboter an- und auszuschalten.
- Ansteuerung der Servomotoren Die Steuerungssoftware muss in der Lage sein, einzelne Glieder des Manipulators allein, als auch mehrere Glieder synchron zu bewegen.
- Vorgabe der Geschwindigkeit Die Geschwindigkeit der Bewegungen des EduBots muss durch die Vorgabe von Geschwindigkeitswerten variierbar sein.
- Verfolgung linearer Bahnen Die Software muss zu Translationen des Greifer entlang linearen Bahnen im dreidimensionalen, kartesischen Raum \Re^3 befähigen. Zeitgleich muss die Orientierung der Backen konstant gehalten werden können.
- Automatisierung Die Steuerungssoftware muss einen vorgegebenen Aktionsplan automatisiert ausführen können.

Einfluss auf die getroffene Wahl der Konfiguration hatte neben den genannten funktionalen Anforderungen eine zusätzlich festgelegte Zielvorgabe: in Abschnitt 1.3 wurde erwähnt, dass für die Berechnung von Aktionsplänen auf eine bereits bestehende Anwendung zurückgegriffen werden kann. Diese Anwendung basiert auf der Programmiersprache Java und ermöglicht dem Benutzer neben der Spezifikation von polygonalen Werkstücken die Berechnung entsprechender Aktionspläne in einer grafischen Oberfläche. Als Zielvorgabe wurde festgelegt, die zu entwickelnde Steuerungssoftware in die Java-Anwendung zu integrieren. Motiviert wurde diese Entscheidung durch das angestrebte Szenario, indem der Benutzer Werkstücke spezifiziert, entsprechende Aktionspläne berechnet und diese anschließend mit dem EduBot ausführt, ohne die grafische Oberfläche dafür verlassen zu müssen.

Unter Berücksichtigung der Vorgaben wurden die folgenden Entscheidungen getroffen: der Servocontroller des EduBots wird mit einem seriellen Kabel direkt an die serielle Schnittstelle des Steuerrechners angeschlossen.

56

Der Controller wird wie üblich mit dem EduBot verbunden. Für die Implementierung der Steuerungssoftware wird die Programmiersprache Java gewählt, da diese Wahl die geplante Integration der Software in die Java-Anwendung zur Berechnung von Aktionsplänen auf natürliche Weise unterstützt. Für die Ansteuerung des Roboters seitens der Steuerungssoftware ist das Befehlsprotokoll des Controllers vorgesehen. Abbildung 4.1 stellt das Befehlsprotokoll des Controllers dar [20]. Die einfach gehaltene Funktionalität des Protokolls entspricht allerdings nicht den funktionalen Anforderungen an die Steuerungssoftware. Daher soll die fehlende Funktionalität aufbauend auf dem Befehlsprotokoll in der Steuerungssoftware implementiert werden. Die Programmiersprache Java bietet derzeit keine Mittel für die serielle Kommunikation an. Für die Kommunikation zwischen der Software und dem Controller des Roboters wird eine offizielle Erweiterung der Programmiersprache Java, das Java Communication API, vorgeschlagen. Implementierungen der Schnittstelle sind auf der Internetseite der Firma Sun Microsystems [23] für die Betriebssysteme Microsoft Windows und Sun Solaris erhältlich (Stand 31.05.2005). Abbildung 4.2 illustriert die Konfiguration.

Bytefolge	Beschreibung	Antwort
0xFF, 0x0C, 0x01	Servo Power ON	Keine
0xFF, 0x0C, 0x00	Servo Power OFF	Keine
0xFF, 0x80	Abfrage der Versions-	Versionsnummer, ge-
	nummer des Controllers	folgt von 0x0D, 0x0A
0xFF, 0x85	Lese Status der digita-	0-7 für Ports 0,1 und
	len Ports	2, gefolgt von 0x0D,
		0x0A
0xFF, 0x0s, p	Setze Servo $s=0-7$ auf	Keine
	Position $p=0x00-0xFF$	

Abbildung 4.1: Das Befehlsprotokoll des Controllers

Von einem Anschluß des EduBots an einen mobilen Pioneer Roboter wurde generell abgesehen, um den technischen Aufwand gering zu halten und die Orientierung der Werkstücke gemäß den Ideen des RISC-Ansatzes von Canny und Goldberg [7] mit einfacher Hardware zu erreichen. Dies spiegelt sich auch im Konzept der Umsetzung aus Kapitel 3 wieder, welches den stationären Betrieb des EduBots vorsieht.

Die grafische Anwendung der Firma Robotica Limited zur Ansteuerung des EduBots bietet keine offene Software-Schnittstelle an. Die umfassenden Fähigkeiten der Anwendung können nicht von einer externen Software aus nutzbar gemacht werden. Daher ermöglicht dieser Ansatz die Integration der zu entwickelnden Steuerungssoftware in die bestehende Java-Anwendung zur Berechnung von Aktionsplänen nicht und schied für eine Umsetzung aus.



Abbildung 4.2: Konfiguration zur Steuerung des EduBots.

4.2 Machbarkeitstest

Nach der Entscheidung für eine Konfiguration wurde ein Machbarkeitstest durchgeführt, um grundsätzliche Probleme des technischen Ansatzes frühzeitig zu erkennen. Das Ziel des Machbarkeitstests war die Implementierung eines Prototyps der Robotersteuerungssoftware in der Programmiersprache Java. Dieser Prototyp sollte die durch das Befehlsprotokoll des Controllers gegebene Grundfunktionalität für die Ansteuerung des Roboters bereitstellen.

Der Machbarkeitstest wurde auf einem Computer unter dem Betriebssystem Microsoft Windows XP durchgeführt. Für die Implementierung der Software wurden die aktuelle Java 2 Platform Standard Edition, Version 5.0 und das Java Communication API in der Version 2.0 benutzt. Es entstand ein Prototyp, welcher eine Schnittstelle für das An-/Ausschalten des EduBots sowie die Ansteuerung einzelner Servomotoren anbietet. Um die entstandene Software auf Fehlerfreiheit zu testen, wurde eine grafische Oberfläche in Java implementiert, welche dem Benutzer die Funktionalität der Schnittstelle über grafische Steuerungselemente direkt nutzbar macht. Die erfolgreiche Durchführung des Machbarkeitstests festigte die Entscheidung,

4.2. MACHBARKEITSTEST

das Konzept aus Abschnitt 3 mit dem gewählten technischen Ansatz zu realisieren. Im Folgenden werden einige Erkenntnisse beschrieben, welche während der Phase des Machbarkeitstests gewonnen wurden.

Die für diese Arbeit vorliegende Version des EduBots wird von der Firma ActivMedia Robotics als Werkzeugaufsatz zu einem mobilen Pioneer Roboter geliefert. Im Lieferumfang des EduBots befindet sich ein dreiadriges Datenkabel, welches üblicherweise den Controller des Roboters mit dem mobilen Pioneer Roboter verbindet. Das Kabel endet auf der Seite des Pioneer Roboters mit einem zehnpoligen Pfostenstecker. An diesem Ende des Kabels ist nun der Steuerrechner anzuschliessen. Für die serielle Kommunikation besitzen Computer in der Regel eine neunpolige, serielle Buchse nach dem RS232 Standard, welcher z. B. von Campbell [6] beschrieben wird. Für den Anschluss des EduBots an den Steuerrechner wurde ein Adapterkabel hergestellt, welches auf einer Seite mit einer zehnpoligen Pfostenbuchse und auf der anderen Seite mit einem neunpoligen seriellen Stecker gemäß dem RS232-Standard ausgestattet ist. Abbildung 4.3 zeigt das enstandene Kabel.



Abbildung 4.3: Ein Adapterkabel für den Anschluss des Controllers an einen Computer

In der vorliegenden Version ist die Stromversorgung des EduBots ebenfalls über einen mobilen Pioneer Roboter vorgesehen. Für den geplanten Aufbau musste eine alternative Stromquelle gefunden werden. Auf Anfrage teilte die Firma Robotica Limited mit, der Controller benötige von Haus aus eine Stromstärke von zehn Ampere und eine elektrische Spannung von fünf Volt. In der von der Firma ActivMedia vertriebenen Version des EduBots sei der Controller dagegen in weitere Hardware eingebettet. Daher würde eine Spannung von zwölf Volt benötigt.

Zur Realisierung dieser Anforderung wurde für den Betrieb des EduBots ein wiederaufladbarer Blei-Akkumulator mit einer Kapazität von sieben Amperestunden und einer Spannung von zwölf Volt verwendet.

4.3 Materialien

Der EduBot wurde mit drei Schrauben auf einem kleinen Beistelltisch eines schwedischen Möbelaustatters montiert. Der Greifer arbeitet auf einer tiefer liegenden Werkebene. Die Oberfläche der Werkebene ist relativ glatt und entspricht damit den Ansprüchen bezüglich der möglichst positionsunabhängigen Reibung. Abbildung 4.4 zeigt den EduBot auf dem Sockel.



Abbildung 4.4: Der EduBot wurde auf einen Sockel montiert

Weiterhin galt es an jedem Finger des Greifers eine lineare Backe zu befestigen. Um den Greifer durch die Modifikation nicht nachhaltig zu beschädigen, wurden beide Finger abgeschraubt und durch Holzelemente gleicher Form ersetzt (siehe Abbildung 4.5 links). Nun sollte je Holzelement eine Backe befestigt werden. Die Form der originalen Finger und die Position der Durchbohrungen für die fixierenden Schrauben konnten nicht präzise genug reproduziert werden. Die Holzfinger und die daran befestigten Backen stan-

60

4.4. ARCHITEKTUR DER SOFTWARE

den infolgedessen nicht parallel zueinander. Es wurde daher entschieden, die originalen Finger und eine unschädliche, aber möglichst stabile Art der Befestigung für die linearen Backen zu verwenden.

Letztendlich wurde handelsüblicher Holzleim benutzt, um zwei Holzbacken mit je einer sieben Millimeter breiten Aussparung an den Fingern des Greifers zu befestigen. An eine der Backen wurde eine Laufschiene aus Metall geschraubt. Auf dem Laufwagen der Schiene wurde Sandpapier angebracht, um eine hohe Reibung zwischen Werkstücken und dem Laufwagen zu erzielen. Die Laufschiene wurde versenkt montiert, um die maximale Distanz der beiden Backen zueinander im geöffneten Zustand des Greifers möglichst wenig zu vermindern. Abbildung 4.6 zeigt eine Laufschiene des verwendeten Typs, Abbildung 4.5 zeigt rechts den modifizierten Greifer, welcher während den Testreihen im Rahmen dieser Arbeit zum Einsatz kam.



Abbildung 4.5: Links: zwei Finger des Greifers wurden durch Holzelemente ersetzt. Rechts: der modifizierte Greifer



Abbildung 4.6: Eine Laufschiene für den Greifer

4.4 Architektur der Software

Aufbauend auf dem entwickelten Prototyp für die Steuerungssoftware wurde weitere Funktionalität zur Ansteuerung des EduBots implementiert, um die funktionalen Anforderungen zu erfüllen. Der folgende Abschnitt beschreibt die Architektur der entstandenen Software. Abbildung 4.7 gibt einen Überblick über die verschiedenen Pakete und ihre Abhängigkeiten. Das Paket jEduLib enthält die eigentliche, allgemein verwendbare Steuerungssoftware für den EduBot. Diese verwendet für die serielle Kommunikation das Paket javax.comm, welches die Implementierung des Java Communication API darstellt. Das Paket swinggui beinhaltet eine separate Java-Anwendung, welche die Funktionalität der Steuerungssoftware über eine grafische Oberfläche nutzbar macht. Das Paket appsSwingGui.JawGripper enthält die Anwendung für die Berechnung von Aktionsplänen und nutzt das Paket jEduLib, um den EduBot anzusteuern.



Abbildung 4.7: Die einzelnen Software-Pakete und ihre Abhängigkeiten

4.4.1 Das Paket jEduLib

Dieser Abschnitt skizziert den Aufbau des Pakets jEduLib. Die zentrale Programmklasse des Pakets ist die Klasse EduBot250. Diese implementiert die Schnittstelle EduBotInterface und stellt allgemeine Funktionalität zur Steuerung des Roboters bereit. Instanzen der Klasse EduBot250 besitzen jeweils eine Referenz auf eine Instanz der Klasse SerialPort. SerialPort kapselt die Funktionalität des Java Communication API und ermöglicht die Kommunikation über die serielle Schnittstelle. Bewegungsaufträge für den EduBot werden als Instanzen der Klasse MovementRequest dargestellt und an eine Warteschlange, den MovementScheduler, übergeben. Die Warteschlange sorgt für die sequentielle Abarbeitung eingehender Aufträge. Abbildung 4.8 verdeutlicht die Zusammenhänge durch ein Klassendiagramm in der Notation der Unified Modeling Language [13].

4.4.2 Das Paket appsSwingGui.JawGripper

Das Paket appsSwingGui.JawGripper wurde vom Verfasser dieser Diplomarbeit im Rahmen des Studiums an der Universität Bonn in Java entwickelt.

62



Abbildung 4.8: Klassendiagramm des Pakets jEduLib

Im Mittelpunkt des Pakets steht die Klasse MainFrame, welche eine grafische Oberfläche erzeugt. Die Oberfläche erlaubt dem Benutzer, polygonale Werkstücke in einem Editorfenster zu spezifizieren. Die Berechnung eines Aktionsplans übernimmt nach der Spezifizierung eines Werkstücks die Klasse JawGripperAlgorithm. Anschließend wird der Aktionsplan in der grafischen Oberfläche dargestellt. Die Ausführung des Plans durch einen Greifer kann mit Hilfe der Klasse AnimateFrame schrittweise grafisch visualisiert werden. Das Paket wurde im Rahmen der Diplomarbeit um die Klasse EduBot erweitert. Diese Klasse nutzt die Steuerungssoftware des Pakets jEduLib, um die verschiedenen Bewegungen für die Durchführung eines Aktionsplans mit dem EduBot zu erzielen. Die Klasse unterteilt den Bewegungsablauf des EduBots in mehrere Teilaufgaben. Für die Durchführung der Teilaufgaben werden verschiedene Methoden bereitgestellt. Der Aufruf der passenden Methoden wurde in die Klasse AnimateFrame integriert, sodass neben der Visualisierung eines Aktionsplans am Computer gleichzeitig die Ausführung des Plans mit einem EduBot möglich ist. Abbildung 4.9 skizziert die Zusammenhänge in einem Klassendiagramm.

4.5 Implementierung der Steuerungssoftware

Zu Beginn der Implementierungsphase existierte ein Prototyp der Steuerungssoftware, welcher aus einer einfachen Version der Klasse EduBot250 sowie der Klasse SerialPort bestand. Der Prototyp ermöglichte das Anund Ausschalten des Roboters und erlaubte die Ansteuerung einzelner Servomotoren durch die Vorgabe von Gelenkpositionswerten. Aufbauend auf dem



Abbildung 4.9: Klassendiagramm des Pakets appsSwingGui.JawGripper

bestehenden Ansatz wurde die Software um weitere Funktionalität erweitert und in dem Paket jEduLib zusammengefasst. Dieser Abschnitt beschreibt einige Aspekte der Weiterentwicklung des Prototyps zu einer allgemein verwendbaren Steuerungssoftware.

4.5.1 Eigenschaften der Software

Im Folgenden werden einige grundlegende Eigenschaften der Software dahingehend beschrieben, wie diese erzielt wurden.

Gelenkpositionsangaben Das Befehlsprotokoll des Controllers sieht keine Möglichkeit vor, die aktuellen Gelenkpositionswerte des EduBots zu erfragen. Die Steuerungssoftware erlaubt diese Abfrage dennoch: um einen Servomotor in Bewegung zu versetzen, ist die serielle Übermittlung einer speziellen Bytefolge an den Controller nötig. Diese Bytefolge gibt Aufschluss über die zu bewegende Achse sowie über den gewünschten Gelenkpositionswert. Die Steuerungssoftware speichert für jeden der sechs Servomotoren den letzten für den jeweiligen Servomotor übermittelten Gelenkpositionswert. Die sechs gespeicherten Werte können über eine Methode der Klasse EduBot250 jederzeit abgefragt werden. Diese Methode geht davon aus, dass die Servomotoren die gewünschte Position einnehmen können und dies innerhalb einer kurzen Zeitspanne tun. Mechanische Hindernisse können den Manipulator davon abhalten, die gewünschte Position einzunehmen. In dieser Situation ergeben sich Abweichungen zwischen der Soll- und der Ist-Position eines oder mehrerer Servomotoren. In allen anderen Fällen stimmt die von der Steuerungssoftware gelieferte Positionsangabe mit den tatsächlich von den Servomotoren eingenommenen Positionswerten überein.
- Salut des Arms Nachdem der EduBot über den entsprechenden Befehl des Protokolls angeschaltet wurde, begeben sich die Servomotoren als erstes automatisch mit der maximalen Geschwindigkeit in eine vom Controller vorgegebene Position. Dies ist im Normalfall die Position, welche der EduBot inne hatte, bevor er ausgeschaltet wurde. Eine Ausnahme bildet der Fall, dass die Stromzufuhr des Controllers seit dem letzten Ausschalten des EduBots unterbrochen wurde. In diesem Fall "vergisst" der Controller die gespeicherten Gelenkpositionswerte der Servomotoren und steuert den Manipulator nach dem Einschalten mit maximaler Geschwindigkeit in eine ausschweifende, sogenannte Salut-Position. Um unbeabsichtigte Kollisionen des Manipulators mit Objekten in seiner Umgebung zu vermeiden, wird dieses Verhalten von der Steuerungssoftware unterbunden. Der Ablauf der Einschaltprozedur sieht vor, im ausgeschalteten Zustand des EduBots einen Bewegungsbefehl abzusetzen. Der Controller speichert daraufhin die übergebenen Positionswerte, welche einer festen Homeposition entsprechen. Als nächstes wird der Roboter durch die Steuerungssoftware angeschaltet. Der Controller manövriert den Manipulator, soweit er sich nicht in dieser befindet, in die Homeposition. Wird der EduBot vor dem Anschalten in die Homeposition gebracht, kann er ohne Bedenken in Betrieb genommen werden.
- Geschwindigkeitsdrosselung Die Servomotoren des EduBots führen Bewegungen standardmäßig mit maximaler Geschwindigkeit aus. Um die funktionale Anforderung betreffend der Geschwindigkeitsregelung zu erfüllen (vgl. Abschnitt 4.1), wird die Bewegungsgeschwindigkeit der Servomotoren von der Steuerungssoftware gedrosselt (vgl. [20], Seite 14-15). Die Steuerungssoftware realisiert die Bewegung einer Achse nicht mittels einem einzelnen Bewegungsbefehl, sondern als Folge von mehreren, inkrementellen Befehlen an den Controller. Ein Servomotor wird z. B. anhand von 100 sukzessiv abgesetzten Bewegungsbefehlen von der Gelenkposition 100 in die Position 200 beordert. Zwischen der Absetzung zweier inkrementeller Bewegungsbefehle erfolgt jeweils eine Pause, deren Länge die Geschwindigkeit der Bewegung beeinflusst. Auf diese Weise ergibt sich eine kontinuierlich erscheinende Bewegung, deren Geschwindigkeit variiert werden kann.
- Synchrone Achsenbewegungen Die Steuerungssoftware verfügt über die Möglichkeit, mehrere Achsen gleichzeitig anzusteuern. Zu diesem Zweck werden die Bewegungsbefehle für mehrere Achsen zeitlich verzahnt abgesetzt. Nachdem für jede zu bewegende Achse ein Bewegungsbefehl übermittelt wurde, erfolgt die für die Geschwindigkeitsdrosselung notwendige Wartezeit. Im Anschluss werden die nächsten Bewegungsbefehle an den Controller gesendet, um danach erneut für

eine gewisse Dauer zu pausieren. Abbildung 4.10 bildet den zeitlichen Ablauf der Ansteuerung des Controllers ab.



Abbildung 4.10: Zeitlicher Ablauf der Ansteuerung

4.5.2 Vorwärtskinematik

Die Vorwärtskinematik des EduBots wurde entsprechend der Denavit-Hartenberg Konvention verwirklicht. Da dieses Thema bereits in Abschnitt 2.2.3 ausführlich abgehandelt wurde, werden die einzelnen Schritte der Durchführung im Folgenden nur kurz beschrieben.

- Zu Beginn wurde ein kartesisches Weltkoordinatensystem mit dem Ursprung im ersten Gelenk des Manipulators festgelegt. Den restlichen Gelenken und dem Greifer des Manipulators wurde übereinstimmend mit den Denavit-Hartenberg Konventionen jeweils ein lokales kartesisches Koordinatensystem zugeordnet. Abbildung 4.11 zeigt den Edu-Bot, das blaue Weltkoordinatensystem, das grüne Werkzeugkoordinatensystem und die roten Gelenkkoordinatensysteme.
- Die Denavit-Hartenberg Parameter des EduBots wurden gemäß dem Abschnitt 2.2.3 ermittelt. Hierbei wurden die genauen Abmessungen des Roboters aus der Spezifikation des Herstellers[22] verwendet. Die Tabelle 4.12 stellt die ermittelten Denavit-Hartenberg Parameter dar. Die Winkel θ_i entsprechen den variablen Gelenkwinkeln des Manipulators und sind daher nicht angegeben.
- Die Vorwärtskinematik benötigt als Ausgangspunkt der Berechnungen die Gelenkwinkel des Manipulators. Die Gelenkwinkel sind nicht bekannt, da die Servomotoren mit Gelenkpositionswerten angesteuert werden, welche nicht den Winkeln entsprechen. Für jede Achse wurde eine Ausgleichskonstante ermittelt. Diese Konstanten können für die Transformation von Gelenkpositionswerten des Manipulators in entsprechende Gelenkwinkel verwendet werden. Der Verwendung von Ausgleichskonstanten liegt die Annahme zugrunde, dass durch die

Anderung um genau einen Gelenkpositionswert einer Achse der entsprechende Achswinkel – unabhängig von der aktuellen Position des Manipulators und der Ladung des Greifers – um einen konstanten Betrag variiert wird.

Die Steuerungssoftware unterstützt die Angabe der Position des Greifers oder auch der Gelenke relativ zu dem Weltkoordinatensystem in kartesischen Koordinaten. Beispielhaft soll die Ermittlung der Weltkoordinaten des Greifers beschrieben werden: die Software berechnet aus den Gelenkpositionswerten des Manipulators mittels der Ausgleichskonstanten die aktuellen Gelenkwinkel. Die Gelenkwinkel ermöglichen die Transformationen der Darstellung von Vektoren zwischen benachbarten Frames als Matrizenmultiplikation. Die Transformationen benötigen die übrigen, fixen Denavit-Hartenberg Parameter als konstante Attribute. Die Position des Greifers <u>PSfrag replacements</u> <u>Psfrag replacements</u>. Vektor wird durch die Verkettung der einzelnen Denavit-

Hartenberg Transformationen sukzessiv von Frame zu Frame in eine Darstellung bezüglich des Weltkoordinatensystems transformiert.



Abbildung 4.11: Koordinatensysteme des EduBots

An dieser Stelle soll angemerkt werden, dass die Messung der Ausgleichskonstanten für die Umrechnung zwischen Gelenkwinkeln und Gelenkpositionswerten Probleme bereitete. Für das vierte Gelenk wurde beispielsweise beobachtet, dass eine Änderung der Gelenkpositionswerte von 238 auf 129 einer Rotation um 90 Grad entspricht. Gemäß der Annahme von konstanten Winkelinkrementen der Gelenke sollte eine Drehung des Gelenks von 129 auf den Positionswert 20 ebenfalls einer Rotation um 90 Grad entsprechen. Beobachtet wurde ein Rotationswinkel von etwa 90 Grad erst, als der Gelenkpositionswert 13 eingenommen wurde. Auch das Gelenk fünf wies derartige

i	θ_i	d_i	α_i	a_i
1	—	0 mm	90 Grad	$68,75 \mathrm{~mm}$
2	—	0 mm	0 Grad	$160 \mathrm{mm}$
3	—	0 mm	-90 Grad	0 mm
4	—	$137{,}75~\mathrm{mm}$	-90 Grad	0 mm
5	_	0 mm	-90 Grad	100 m

Abbildung 4.12: Ermittelte Denavit-Hartenberg Parameter des EduBots

Abweichungen auf, welche im Widerspruch zu der Annahme von konstanten Winkelinkrementen stehen. Weiterhin wurde die Wiederholungsgenauigkeit des Roboters von \pm 5mm deutlich: das mehrmalige Anfahren des gleichen Gelenkpositionswertes eines Gelenks resultierte teilweise in unverkennbar unterschiedlichen Gelenkwinkeln. Aufgrund der Wiederholungsungenauigkeit wurden die Konstanten mehrmals ermittelt und die Ergebnisse durch Mittelung zu einem Endwert zusammengefasst.

Neben der Problematik, dass die Ausgleichskonstanten möglicherweise nicht optimal gewählt wurden, verursachen die genannten Gründe für feste Ausgleichskonstanten in vielen Fällen Abweichungen zwischen den tatsächlichen Gelenkwinkeln und den berechneten Winkeln. Daher besteht bezüglich der von der Vorwärtskinematik berechneten Position und Orientierung des Greifers Unsicherheit. Dies ist bei der gegebenen Wiederholungsgenauigkeit des Manipulators nicht überraschend.

4.5.3 Rückwärtskinematik

Die Steuerungssoftware unterstützt die Angabe von Zielpositionen für den Greifer des EduBots in kartesischen Koordinaten. Aus den kartesischen Koordinaten werden Gelenkpositionswerte berechnet, welche den Greifer in die vorgegebene Zielposition bringen. Zu diesem Zweck wurde ein numerisches Verfahren [11] implementiert. Der verwendete, iterative Algorithmus ist in Abbildung 4.5.1 dargestellt.

Intuitiv wird auf den Endpunkt des Manipulators eine Kraft ausgeübt, welche in einer Bewegung des Endpunkts in Richtung der Zielposition resultiert. Diese Bewegung wird durch simultane Änderungen aller Gelenkpositionswerte erreicht. Der Algorithmus endet im Allgemeinen mit dem Erreichen der Zielposition.

Die Berechnung eines neuen Gelenkpositionswertes für das Gelenk i in Schritt 4 des Algorithmus wird im Folgenden beschrieben. Sei a ein Vektor entlang der Gelenkachse i, b sei ein Vektor entlang des Gliedes i, r sei ein zu a und b orthogonaler Vektor und f sei ein Vektor vom Endpunkt des Manipulators zu der Zielposition (siehe Abbildung 4.13(i)). Ist der Vektor fparallel zu a oder b, so kann das Gelenk i keinen Beitrag zu einer Annäherung

Algorithmus 4.5.1 Algorithmus für die inverse Kinematik
Eingabe:
Zielposition ω des Greifers,
Anzahl n der Gelenke des Manipulators.
1: while es gibt Fortschritt do
2: for $i = 1$ to n do
3: Berechne mit der Vorwärtskinematik die aktuellen Weltkoordinaten
des Gelenks i
4: Berechne für Gelenk i aus seinen Weltkoordinaten und der Zielpo-
sition ω einen neuen Gelenk positionswert, welcher den Greifer der
Zielposition näher bringt
5: Setze den Gelenkpositionswert von Gelenk i auf den neu berechne-
ten Wert
6: end for
7: end while
8: gebe die aktuellen Gelenkpositionswerte zurück

an die Zielposition leisten. Die zu berechnende relative Drehbewegung von Gelenk i wird daher als proportional zum Sinus des Winkels zwischen a und f sowie proportional zum Sinus des Winkels zwischen b und f angenommen. Weiterhin soll die Geschwindigkeit der Bewegung des Endpunkts langsamer werden, je kleiner die Distanz zwischen dem Endpunkt und der Zielposition wird. Die Geschwindigkeit der relativen Drehbewegung von Gelenk i kann also proportional zu |f| gesetzt werden. Die Richtung der relativen Bewegung von Gelenk i soll positiv sein, wenn das Vektorskalarprodukt der Vektoren r und f positiv ist, andernfalls ist die Richtung der relativen Bewegung negativ. Es ergeben sich die folgenden Formeln für die Berechnung des neuen Gelenkpositionswertes x_{neu} für Gelenk i:

$$x_{relativ} = |f| \sin(\langle a, f\rangle) \sin(\langle b, f\rangle) sign(\langle r, f \rangle)$$

$$(4.1)$$

$$x_{neu} = x_{aktuell} + x_{relativ} \tag{4.2}$$

Dabei ist $x_{aktuell}$ der aktuelle Gelenkpositionswert von Gelenk *i* und der Wert von $x_{relativ}$ entspricht der durchzuführenden, relativen Bewegung des Gelenks *i*.

Für den Fall, dass die Gelenkachse i parallel zu Glied i verläuft (vgl. Abbildung 4.13(ii)), ergibt sich die folgende Änderung für die Berechnung des neuen Gelenkpositionswertes: der Vektor b verlaufe nun nicht entlang Glied i, sondern reiche von Gelenk i zum Endpunkt des Manipulators. Die gegebenen Formeln sind mit dieser Änderung weiterhin gültig.

Das beschriebene Verfahren wurde um die Möglichkeit erweitert, gewünschte Zielpositionswerte für einzelne Gelenke als Randbedingung vorzugeben. Die betreffenden Gelenke nehmen in Algorithmus 4.5.1 nicht an



Abbildung 4.13: Berechnung neuer Gelenkpositionswerte

der Berechnung der Bewegung teil, sondern nehmen bereits zu Anfang des Algorithmus den vorgegebenen Positionswert ein.

Weiterhin wird auf die Beschränkung des Bewegungsspielraums der einzelnen Gelenke des EduBots wie folgt eingegangen: liegt ein neu berechneter Gelenkpositionswert x_{neu} außerhalb des zulässigen Bereichs, so wird der aktuelle Gelenkpositionswert des betreffenden Gelenks auf den zulässigen Grenzwert gesetzt.

In einigen Situationen stellt die gefundene Lösung nicht das Optimum dar. Dieser Fall liegt vor, wenn der Manipulator die Zielposition mit der gegebenen Lösung nicht erreicht, obwohl das Ziel mit geeigneten Gelenkpositionswerten erreichbar ist. Die Ursache liegt in solchen Fällen in der Beschränkung der Gelenkpositionswerte des EduBots durch min/max-Werte. Die gefundene Lösung zeichnet sich dadurch aus, dass die Bewegungsfreiheit von mindestens einem Gelenk des Manipulators während der iterativen Annäherung durch den jeweiligen min/max-Wert beschränkt wird. Die Bewegung des Gelenks kann während den folgenden Iterationen des Algorithmus nicht wie gewünscht fortgesetzt werden, wobei auch andere Gelenke nicht zu einer weiteren Annäherung an die Zielposition beitragen können. Der Algorithmus ist darauf ausgerichtet, den Greifer stetig an die Zielposition anzunähern. In der implementierten Version besteht keine Möglichkeit, durch eine Abkehr von diesem Prinzip aus dem lokalen Optimum zu entkommen. Diese Situation wird in Abbildung 4.14 anhand eines einfachen Beispiels dargestellt.

Streng genommen ist das Verfahren nicht der Disziplin der inversen Kinematik zuzuordnen, da es weder die Angabe einer gewünschten Orientierung des Greifers erlaubt, noch alle prinzipiell vorhandenen Lösungen der Aufgabenstellung berechnet. Welche Lösung für das Erreichen der Zielposition im Fall von Mehrdeutigkeiten gefunden wird, hängt von den initia-

70



Abbildung 4.14: Der Endpunkt des Manipulators kann sich dem Ziel aufgrund des rot dargestellten Bewegungsspielraums des Gelenks nicht weiter annähern. Die gefundene Lösung der inversen Kinematik stellt nur ein lokales Optimum dar.

len Gelenkwerten zu Anfang der Berechnungen ab. Für die Umsetzung von Goldbergs Methode ist das beschriebene Verfahren dennoch geeignet: der EduBot arbeitet mit dem Greifer ausschließlich im Bereich der Werkebene. In der eingenommenen Haltung besitzen die Gelenke einen großen Bewegungsspielraum. Beschränkungen einzelner Gelenke durch min/max-Werte können durch ausgleichende Bewegungen anderer Gelenke egalisiert werden. Im Kontext der praktischen Umsetzung werden daher global optimale Lösungen gefunden, sodass die Zielposition erreicht wird. Das Verfahren bildet die Grundlage für die Verfolgung linearer Bahnen im dreidimensionalen, kartesischen Raum. Dieses Thema wird im nächsten Abschnitt erläutert.

4.5.4 Verfolgung linearer Bahnen

Mit Hilfe der Rückwärtskinematik ergibt sich die Möglichkeit, mit dem Greifer des EduBots einer linearen Bahn im dreidimensionalen Raum \Re^3 zu folgen. Zu diesem Zweck werden die kartesischen Koordinaten des Start- und Zielpunkts schrittweise linear interpoliert. Nach der Interpolation werden die ermittelten kartesischen Koordinaten mittels der inversen Kinematik des Manipulators in adäquate Gelenkpositionswerte umgerechnet. Die so generierten Gelenkpositionswerte werden als Steuersignale an den EduBot gesendet (vgl. Algorithmus 4.5.2).

Die Verfolgung linearer, kartesischer Bahnen mit einem seriellen Gelenkarmroboter ist im Allgemeinen nicht immer möglich (Craig, [10]), auch wenn der Start- und Zielpunkt im Arbeitsraum des Roboters liegt. So ist es durchaus möglich, dass nicht alle Punkte auf der Geraden zwischen dem Start- und Zielpunkt im Arbeitsraum des Roboters liegen. Weiterhin kann an bestimmten Positionen, sogenannten *Singularitäten*, die für die weitere Verfolgung der Bahn notwendige Geschwindigkeit eines Gelenks unendlich

Algorithmus 4.5.2 Lineare Translation des Greifers

Eingabe: Start
position τ_{start} und Endposition τ_{end} der Translation in kartesischen Ko
ordinaten

- 1: $i \leftarrow 0$
- 2: while i < 1 do
- 3: Erhöhe i um ein kleines δ
- 4: Berechne mittels τ_{start} , τ_{end} und *i* den nächsten Interpolationsschritt
- 5: Transformiere die berechneten kartesischen Koordinaten in Gelenkpositionswerte
- 6: Bewege den Manipulator gemäß der berechneten Gelenkpositionswerte
- 7: end while

groß werden. Da dies mit einem realen Roboter nicht möglich ist, ergibt sich unweigerlich eine Abweichung von der korrekten Bahn. Aufgrund solcher inhärenter Probleme sollte die kartesische Interpolation mit entsprechender Vorsicht verwendet werden.

4.5.5 Ergebnis

Aufbauend auf dem Befehlsprotokoll des EduBots 250 wurde eine allgemein verwendbare Robotersteuerungssoftware in der Programmiersprache Java implementiert. Diese stellt Funktionalität zur Steuerung des Roboters als Schnittstelle zur Verfügung. Die Software ermöglicht das An- und Ausschalten des Roboters sowie die synchrone Ansteuerung mehrerer Servomotoren. Eine Bewegung des Manipulators lässt sich durch die Vorgabe von Gelenkpositionswerten oder durch die Angabe von kartesische Koordinaten der Zielposition für den Greifer erzielen. Weiterhin wurde eine Geschwindigkeitsdrosselung implementiert, welche die Geschwindigkeit des EduBots variabel gestaltet. Die aktuelle Position des Greifers lässt sich in Gelenk- und Weltkoordinaten erfragen. Über die inverse Kinematik besteht die Möglichkeit, linearen Bahnen im dreidimensionalen Raum \Re^3 zu folgen. Nebenbei entstand eine Java-Anwendung, welche die Funktionalität der Software über Bedienelemente einer grafischen Oberfläche nutzbar macht. Die Vorwärtskinematik liefert teilweise ungenaue Resultate, welche sich in der mangelnden Präzision der Rückwärtskinematik fortsetzen. (vgl. Abschnitt 4.5.2, Seite 67).

4.6 Implementierung der Steuerungskomponente

Die in der Programmiersprache Java implementierte, allgemeine Steuerungssoftware für den EduBot aus Abschnitt 4.5 erfüllt die funktionalen Anforderungen, welche in Kapitel 3 für eine Umsetzung von Goldbergs Methode genannt wurden. Im Rahmen dieser Arbeit wurde der Bewegungsablauf zur Orientierung von Werkstücken aus Kapitel 3 aufbauend auf der Steuerungssoftware mit dem EduBot 250M umgesetzt. Hierfür wurde u. a. ein Startund ein Endpunkt für die Translation, wie in Abschnitt 3.7 beschrieben, festgelegt. Die Länge der Translationsbahn des Greifers betrug demnach etwa 28 Zentimeter.

Im Laufe der Implementierung entstand die Java-Klasse EduBot, welche dem Paket zur Berechnung von Aktionsplänen, appsSwingGui.JawGripper, hinzugefügt wurde. Die Klasse unterteilt den Bewegungsablauf des EduBots in mehrere Teilaufgaben. Für die Durchführung der Teilaufgaben werden verschiedene Methoden bereitgestellt. Der Aufruf der passenden Methoden wurde in die Anwendung zur Berechnung von Aktionsplänen integriert, sodass neben der Visualisierung eines Aktionsplans am Computer gleichzeitig die Ausführung des Plans mit einem EduBot möglich ist.

Die Bezeichnung des seriellen Ports für die Kommunikation mit dem EduBot wird der Java-Anwendung beim Start als Parameter übergeben. Eine geeignete Translationsgeschwindigkeit des Greifers ist u. a. von dem Werkstück abhängig (vgl. Abschnitt 3.5). Daher ist beim Start der Anwendung neben der Bezeichnung des seriellen Ports die gewünschte Translationsgeschwindigkeit zu nennen. Werden keine Parameter angegeben, entfällt die Ansteuerung des Roboters während der Visualisierung eines Aktionsplans. Im Folgenden wird kurz auf nicht offensichtliche Details der Umsetzung eingegangen:

Translation entlang einer linearen, kartesischen Bahn Die Translation des Greifers wird mittels der Rückwärtskinematikfunktion der Steuerungssoftware und dem Algorithmus 4.5.2 (Seite 72) bewerkstelligt. Der Greifer wird in die Ausgangsposition manövriert. In dieser Haltung steht der Greifer wie gefordert lotrecht über der Werkebene. Der Gelenkpositionswert des fünften Gelenks wird während der Translation mittels der Angabe von Randbedingungen¹ konstant gehalten, um die lotrechte Orientierung des Greifers über der Werkebene aufrecht zu erhalten. Weiterhin muss die Orientierung der Backen während der Translation des Greifers gleichbleibend der Normalstellung entsprechen. Aus der Rotation um das erste Gelenk des Manipulators ergibt sich eine Abweichung der Orientierung der Backen. In jedem Interpolationsschritt des Algorithmus wird daher, abhängig von dem Positionswert des ersten Gelenks, ein Gelenkwert für das vierte Gelenk berechnet und als Randbedingung für das vierte Gelenk verwendet.

Bemerkt werden soll an dieser Stelle, dass während der linearen Translation Abweichungen des Greifers von der geplanten Bahn beobach-

¹vgl. Abschnitt 4.5.3

tet wurden. Dies wird zurückgeführt auf die bereits beschriebenen Ungenauigkeiten der Vorwärtskinematik des Manipulators (vgl. Abschnitt 4.5.2 auf Seite 67), welche sich in Ungenauigkeiten der implementierten Rückwärtskinematik fortsetzen. Weiterhin variiert die Orientierung der Backen des Greifers im Verlaufe der Translation leicht. Diese Beobachtung stärkt die Vermutung, dass die Rotationsgeschwindigkeit der Gelenke des EduBots – in diesem Fall der Gelenke eins und vier – aufgrund nicht konstanter Winkelinkremente je nach dem aktuell eingenommenen Gelenkpositionswert leicht variiert.

Positionierung des Werkstücks Das Drehspektrum von Gelenk vier ist durch die vom Hersteller angegebenen min/max-Werte auf etwa 170 Grad beschränkt. Dies sorgte insofern für eine Überraschung, da auf den Internetseiten des Herstellers für die Rotationsgelenke des Manipulators ein Wert von je 180 Grad angegeben wird. Für die Rotation des Werkstücks werden 180 Grad Bewegungsfreiheit benötigt. Eine diesbezügliche Anfrage an den Support der Firma ActivMedia Robotics ergab, dass die min/max-Werte in kleinen Schritten verändert werden könnten. Dabei solle beobachtet werden, ob die Bewegung des Servomotors zittrig würde. In diesem Fall müssten die Werte konservativer gewählt werden.

Die min/max-Werte von Gelenk vier wurden daraufhin gemäß den gegebenen Anweisungen revidiert, wodurch die Bewegungsfähigkeit des Gelenks auf einen Bereich von etwa 183 Grad verbessert wurde.

4.7 Tests

Der abschließende Schritt der Umsetzung von Goldbergs Methode sah das Testen der entstandenen Lösung vor. Anhand verschiedener Werkstücke wurden Aktionspläne mit der Java-Anwendung berechnet und mit dem Gelenkarmroboter EduBot 250M ausgeführt. Während der Ausführung eines Aktionsplans wurde die tatsächliche Orientierung des Werkstücks mit der angezeigten Orientierung in der Visualisierung am Steuerrechner verglichen, um die korrekte Ausführung des Aktionsplans zu überprüfen.

Während der Translation des Greifers begeben sich die Werkstücke in eine stabile Orientierung. Diese Orientierungen entsprechen den lokalen Minima der Radiusfunktion des Werkstücks (vgl. Kapitel 2.1). Ein Werkstück kann daher einige stabile Orientierungen aufweisen. Für jede stabile Orientierung des zu testenden Werkstücks wurde ein Testlauf ausgeführt, wobei das Werkstück so platziert wurde, dass es sich im Verlauf der ersten Translation des Greifers in die gerade betrachtete stabile Orientierung begab.

4.7. TESTS

4.7.1 Vorbereitungen

Aus Holz wurden zehn verschiedene Werkstücke gefertigt und deren Maße notiert. Die jeweilige Masse der Werkstücke wurde als gleichverteilt angenommen, sodass der Massenschwerpunkt der Werkstücke von der Java-Anwendung zur Berechnung von Aktionsplänen automatisch ermittelt werden konnte. Abbildung 4.15 zeigt die Grundfläche der hergestellten Werkstücke, Aufschluss über die geometrischen Daten der Werkstücke in Form von Punktkoordinaten gibt die Abbildung 4.16. Die dabei verwendete Ein-

 $\frac{\text{PSfrag replacenheits}}{\text{I.75 Zentimetern.}}$ werkstücke besaßen eine Höhe von 1.75 Zentimetern.



Abbildung 4.15: Die verwendeten Werkstücke

Werkstück	Geometriedaten
1	(0,29) $(0,20)$ $(20,20)$ $(20,0)$ $(35,0)$ $(35,29)$
2	(0,0) $(24,0)$ $(29,19)$ $(15,42)$ $(0,13)$
3	(0,0) (35,0) (35,18) (0,18)
4	(0,10) $(27,0)$ $(36,09)$ $(36,10)$
5	(0,0) $(0,15)$ $(37,16)$ $(37,10)$ $(30,10)$ $(30,0)$
6	(0,0) $(29,0)$ $(29,10)$ $(19,10)$ $(19,30)$ $(09,30)$
	(09,10) $(0,10)$
7	(0,0) $(31,0)$ $(18,19)$
8	(0,0) $(17,0)$ $(26,16)$ $(11,25)$ $(2,19)$
9	(1,0) $(34,0)$ $(33,5)$ $(8,4)$ $(7,11)$ $(0,11)$
10	(0,0) $(32,0)$ $(32,1)$ $(1,11)$

Abbildung 4.16: Geometriedaten der verwendeten Werkstücke

Im Zuge der Vorbereitungen wurde eine auf die Werkstücke abgestimmte Translationsgeschwindigkeit bestimmt (vgl. Abschnitt 3.5), welche den durch die quasi-statische Annahme von Goldbergs Methode auftretenden Fehler gering halten sollte. Dazu wurde der Reibungskoeffizient μ zwischen mehreren Werkstücken und der Werkebene experimentell ermittelt. Für die untersuchten Werkstücke ergaben sich Reibungskoeffizienten von 0.3513 bis 0.4545. Mit einem erlaubten translatorischen Fehler von 0.1 Zentimetern ergab sich aufgrund der Formel 3.6 als obere Schranke für die Geschwindigkeit des Greifers je nach Werkstück $\frac{8.3 cm}{s}$ bis $\frac{9.4 cm}{s}$. Für die Translation des Greifers wurde daraufhin eine Geschwindigkeit von $\frac{5.5 cm}{s}$ festgelegt. Im Rahmen der Tests wurde der Java-Anwendung ein Geschwindigkeitswert von 35 als Parameter übergeben. Mit diesem Wert entsprach die Geschwindigkeit des Greifers während der Translation etwa der Vorgabe von $\frac{5.5 cm}{s}$.

4.7.2 Ergebnisse

14 von 32 Testläufen wurden fehlerfrei abgeschlossen. In den übrigen Fällen kam es zu Abweichungen der Bewegungen des Werkstücks von dem beabsichtigten Verhalten. Eine aufgetretene Unregelmäßigkeit bedeutete nicht generell den Fehlschlag des Verfahrens. So endeten einige Werkstücke nach fehlerhaften Orientierungsvorgängen durchaus in der gewünschten Endorientierung.

Die Tabelle 4.17 gibt Aufschluss über die erzielten Ergebnisse. Die Spalte *Testläufe* enthält die Anzahl der durchgeführten Testläufe je Werkstück. Die Spalte *Fehlerfrei* zählt Testläufe, welche ohne jegliche Abweichungen vom beabsichtigten Verhalten des Werkstücks beendet wurden. In der Spalte *Fehler* werden diejenigen Testläufe aufgeführt, welche Abweichungen vom geplanten Verhalten des Werkstücks aufwiesen, die aber dennoch erfolgreich in der beabsichtigten Orientierung des Werkstücks endeten. Unter *Fehlschlag* werden Testläufe gezählt, welche aufgrund von einem aufgetretenem Fehlverhalten mit einer falschen Orientierung des Werkstücks abgeschlossen wurden. Die Testläufe von Werkstück zwei werden zwar aufgeführt, aber nicht gewertet, da das Werkstück mit 4.2 Zentimetern einen zu großen Durchmesser aufwies (siehe auch Abschnitt 4.7.3).

Die folgenden Abweichungen vom geplanten Verhalten des Werkstücks wurden beobachtet:

- Obwohl der eingesetzte Greifer gemäß den Ideen von Goldberg wenig Reibung aufwies (vgl. Abschnitt 3.3), trat während eines Testlaufs von Werkstück neun eine Verklemmung des Werkstücks zwischen den Backen des Greifers auf. Die Verklemmung führte im weiteren Verlauf zu einem Fehlschlag des Testlaufs.
- 2. Das Werkstück zwei, welches mit 4.2 Zentimetern den größten Durchmesser unter allen verwendeten Werkstücken aufwies, kollidierte in mehreren Situationen mit dem Greifer. In diesen Fällen traf eine Backe des geöffneten Greifers während der Abwärtsbewegung unbeabsichtigt auf das Werkstück. Infolge der Kollisionen kam das Werkstück

4.7. TESTS

Werkstück	Testläufe	Fehlerfrei	Fehler	Fehlschlag
1	4	1	2	1
2	4	2	0	2
3	4	4	0	0
4	3	0	2	1
5	3	3	0	0
6	4	0	1	3
7	3	1	0	2
8	5	3	0	2
9	3	0	0	3
10	3	2	0	1
Gesamt*	32	14	5	13

Abbildung 4.17: Testresultate (*Testläufe von Werkstück zwei wurden nicht gewertet)

regelmäßig mit einer Seitenfläche auf der Werkebene zu liegen. Dieser Umstand führte zu einem Fehlschlag des Orientierungsvorgangs.

Derartige Kollisionen wurden beobachtet, wenn sich der geöffnete Greifer nach der Translation von oben der Werkebene näherte, um das Werkstück für den anschließenden Greifvorgang zwischen den Backen zu positionieren. Weiterhin kam es zu diesen Zwischenfällen, wenn das Werkstück auf den nächsten Drehwinkel des Aktionsplans gedreht werden sollte. Auch hier nähert sich der geöffnete Greifer der Werkebene im Zuge des Umgreifens von oben (siehe Algorithmus 3.7.1 auf Seite 50).

- 3. Einige theoretisch stabile Orientierungen erwiesen sich in der Praxis als nicht stabil. Werkstücke gelangten in diesem Fall während der Translation des Greifers in die vermeintlich stabile Orientierung, um die Rotationsbewegung anschließend bis zum Erreichen einer benachbarten stabilen Orientierung fortzusetzen (siehe Abbildung 4.18).
- 4. Nachdem sie während der Translation des Greifers einen Kontakt mit der Backe hergestellt hatten, rotierten manche Werkstücke entgegengesetzt der geplanten Richtung und endeten daher in einer nicht beabsichtigten Orientierung (siehe Abbildung 4.19).

Ein weiterer, potentieller Problemfall des Verfahrens trat während der Tests nicht auf: es besteht ein gewisses Risiko, dass das Werkstück während einer Translation des Greifers von der linearen Backe abrutscht und auf der Werkebene liegen bleibt. Die Position der Werkstücke war in diesem Sinne zu keinem Zeitpunkt während der Tests als kritisch einzustufen. Weiterhin ergaben die Tests keinen Einlass zu der Annahme, die Translationsdistanz des Greifers wäre zu gering gewählt worden. In allen beobachteten Situationen wurde das Werkstück während der Translation des Greifers vor dem Erreichen des Endpunktes der Translation in eine stabile Orientierung überführt.



Abbildung 4.18: (i) Ein Kontakt zwischen dem Werkstück und dem Greifer entsteht, (ii) das Werkstück befindet sich in der geplanten Orientierung, (iii) die Rotation des Werkstücks setzt sich unplanmäßig fort (iv) und endet in einer benachbarten stabilen Orientierung.



Abbildung 4.19: (i) Ein Kontakt zwischen dem Werkstück und dem Greifer entsteht, (ii) das Werkstück rotiert wie geplant gegen den Uhrzeigersinn, (iii) das Werkstück rotiert unbeabsichtigt im Uhrzeigersinn.

4.7.3 Bewertung

Die Verklemmung des Werkstücks neun zwischen den Backen des Greifers ist nach Ansicht des Autors auf die Reibung zwischen der Laufschiene und ihrem Gleiter zurückzuführen. Die geringe Kraft, welche der Greifer des EduBots auf das Werkstück und den Gleiter der Laufschiene ausübt, reicht in ungünstigen Situationen nicht aus, um den Gleiter in Bewegung zu versetzen. Abhilfe kann der Einsatz von reibungsärmeren Laufschienen schaffen. Zu bedenken ist, dass mit solchen Laufschienen der ideale, reibungsfreihe Greifer nur angenähert werden kann. Es ist davon auszugehen, dass ein Greifer, welcher während des Greifvorgangs eine stärkere Kraft aufbringen kann, das Eintreten von Verklemmungen weiter reduzieren kann. In diesem Sinne ist der EduBot 250M, welcher über einen eher schwachen Antrieb verfügt, für die Durchführung des Verfahrens nicht gut geeignet.

Die Bewegungen des geöffneten Greifers, der sich der Werkebene lotrecht von oben nähert, wurden auf eine Weise realisiert, welche eine möglichst zentrierte Position des Werkstücks zwischen den Backen gewährleisten sollte. Aufgrund der Wiederholungsgenauigkeit des EduBots von \pm 5mm kann es allerdings zu Abweichungen der Position des Greifers kommen. Der Problemfall zwei lässt erkennen, dass für Werkstücke mit einem Durchmesser von 4.2 Zentimetern eine Kollision mit dem Greifer unter Nutzung des implementierten Bewegungsablaufs nicht ausgeschlossen werden kann. Die maximale Größe von zu orientierenden Werkstücken wurde aufgrund dieser Erkenntnisse auf vier Zentimeter festgelegt. Die Resultate der Testläufe von Werkstück zwei wurden in der Gesamtwertung nicht berücksichtigt.

Nach jeder Translation des Greifers sollte das Werkstück planmäßig in einer der stabilen Orientierungen enden. Mit der Anzahl der Winkel des Aktionsplans steigt die Anzahl der durchzuführenden Translationen und somit steigt für eine fest gewählte, stabile Orientierung des Werkstücks tendenziell die Wahrscheinlichkeit, dass diese bestimmte Orientierung während der Ausführung des Aktionsplans realisiert wird. Wurde eine in der Praxis nicht stabile Orientierung während der Berechnung des Aktionsplans als stabil eingestuft, so wird diese mit steigender Anzahl der Winkel des Aktionsplans tendenziell irgendwann während der Durchführung des Plans erzielt und verursacht aufgrund ihrer Instabilität ein Fehlverhalten. Dass viele der verwendeten Werkstücke eine derartige, falsch eingestufte Orientierung aufwiesen, erklärt eine Vielzahl der in der Tabelle 4.17 dokumentierten Fehler und Fehlschläge.

Neben dem Auftreten des beschriebenen Problemfalls drei hatte der Problemfall vier einen großen Anteil an den beobachteten Unstimmigkeiten. Ziel weiterer Bemühungen war es, das Auftreten der Problemfälle drei und vier zu vermeiden, da hier viel Potential zum Senken der Fehlerquote gesehen wurde. Einer genaueren Betrachtung der Ursachen der Probleme drei und vier, Vorschlägen zu deren Lösung sowie der Beschreibung von erneuten Testläufen ist mit Kapitel 5 ein eigenes Kapitel gewidmet.

4.8 Zusammenfassung

In diesem Kapitel wurde die Umsetzung des Konzepts zur Orientierung von polygonalen Werkstücken aus Kapitel 3 mit einem EduBot 250M beschrieben. Der verwendete Roboter wurde von der Universität Bonn zur Verfügung gestellt und dient in der genutzten Version in erster Linie als Zubehör für einen mobilen autonomen Roboter aus der Pioneer-Familie der Firma ActivMedia Robotics. Für die Umsetzung im Rahmen dieser Arbeit wurde der EduBot ohne einen Pioneer-Roboter betrieben, um den Aufwand an benötigter Hardware gering zu halten. Der EduBot wurde daher direkt von einem Steuerrechner über die serielle Schnittstelle auf Basis des Befehlsprotokolls des Controllers des EduBots angesteuert. Für die Verbindung zwischen dem Robotercontroller und dem Steuerrechner wurde ein serielles Adapterkabel hergestellt, die Stromversorgung des EduBots wurde über einen Akkumulator gewährleistet. Der Roboter wurde fest auf einen Sockel montiert und mit einem reibungsarmen Greifer gemäß Abschnitt 3.3 ausgestattet.

Für die Ansteuerung des Roboters wurde eine allgemein verwendbare Steuerungssoftware in der Programmiersprache Java implementiert. Für die serielle Kommunikation verwendet die Software eine Standard-Erweiterung der Sprache Java, das Java Communication API 2.0 in einer Implementierung für Microsoft Windows. Die entstandene Steuerungssoftware erweitert die Möglichkeiten des simplen Befehlsprotokolls des EduBots u. a. um eine Geschwindigkeitsregelung, um die simultane Ansteuerung mehrerer Achsen und um die Implementierung der Vorwärts- und Rückwärtskinematik des Manipulators. Aufbauend auf der allgemeinen Steuersoftware wurden die Bewegungen zur Orientierung von polygonalen Werkstücken in einer Java-Klasse implementiert. Die Klasse wurde in eine bereits bestehende Java-Anwendung zur Berechnung von Aktionsplänen integriert. Der Benutzer hat in dieser Anwendung die Möglichkeit Polygone zu spezifizieren, entsprechende Aktionspläne berechnen und diese anschließend mit dem EduBot ausführen.

Als Nebenprodukt der Implementierungsphase entstand eine grafische Oberfläche, welche die allgemeine Steuerungssoftware des EduBots nutzt und dem Benutzer die direkte Ansteuerung des Roboters über grafische Bedienelemente ermöglicht.

In durchgeführten Tests mit zehn Werkstücken wurden einige Schwächen der entstandenen Umsetzung sichtbar. Ein Werkstück erwies sich mit einem Durchmesser von 4.2 Zentimetern als zu groß, sodass die Testläufe dieses Werkstücks als nicht relevant für eine Bewertung erachtet wurden. Ein weiteres Werkstück neigte zu Verklemmungen zwischen den Backen des Greifers. Manche Werkstücke rotierten während der Translation des Greifers entgegengesetzt der beabsichtigten Richtung oder wiesen instabile Orientierungen auf, welche während der Berechnung des Aktionsplans als stabil klassifiziert wurden.

Die Tests führten andererseits auch zu positiven Erkenntnissen: eine Translationsdistanz von 28 Zentimetern reichte in allen Testläufen aus, um das Werkstück während der Translation des Greifers in eine stabile Orientierung zu bringen. Ein seitliches Wegrutschen eines Werkstücks von der Backe während der Translation wurde in keinem Testlauf beobachtet.

4.8. ZUSAMMENFASSUNG

Von den 32 gewerteten Testläufen wurden 14 Läufe fehlerfrei beendet, 13 Testläufe schlugen fehl, die restlichen 5 Testläufe endeten trotz beobachteter Fehlverhalten in der korrekten Orientierung des Werkstücks. Insgesamt sieben von den neun gewerteten Werkstücken wiesen während der Testläufe Fehler auf.

Zusammenfassend kann gesagt werden, dass sich Goldbergs Verfahren zur Orientierung polygonaler Werkstücke im Falle eines unpräzisen Roboters wie dem EduBot 250M, oder im Falle von unpräzisen Geometriedaten der Werkstücke als wenig robust erweist. Weiterhin können Verklemmungen des Werkstücks aufgrund der geringen Kraft des Greifers nicht ganz ausgeschlossen werden. Die Umsetzung des Verfahrens erscheint in dieser Form als wenig befriedigend. Im nächsten Kapitel wird genauer auf einige der beobachteten Probleme eingegangen, weiterhin werden entsprechende Lösungswege entwickelt und umgesetzt.



Abbildung 4.20: Links oben: der EduBot schiebt ein Werkstück mit dem Greifer. Rechts oben: die Translation ist beendet, das Werkstück wird gegriffen. Links unten: der Greifer setzt das Werkstück für die nächste Translation ab. Rechts unten: der Greifer nähert sich dem Werkstück von oben, um es zwischen den Backen zu positionieren

Kapitel 5

Robustheit von Aktionsplänen

Der erste Teil dieses Kapitels erläutert einige negative Auswirkungen, welche sich aus der Existenz von Unsicherheit für Goldbergs Verfahren zur Orientierung polygonaler Werkstücke ergeben können. Im Rahmen dieser Arbeit wurden Ideen entwickelt und umgesetzt mit dem Ziel, die Orientierung polygonaler Werkstücke im Vorhandensein von Unsicherheit zu ermöglichen. Der Grundgedanke besteht in der Berechnung von robusten Aktionsplänen. Während der Berechnung wird den negativen Auswirkungen durch geeignete Techniken vorgebeugt. Dieses Kapitel widmet sich dem erarbeiteten Ansatz und beschreibt die erzielten Resultate. Sämtliche Betrachtungen setzen zwingend die in Abschnitt 3.1 gemachten Annahmen voraus, insbesondere also die Annahme eines quasi-statischen Bewegungsmodells und die Annahme von geschwindigkeits- sowie positionsunabhängiger Reibung zwischen dem Werkstück und der Werkebene.

5.1 Die Bewegung eines geschobenen Werkstücks

In Abschnitt 3.4 wurde bereits erwähnt, dass die Bewegung eines polygonalen Werkstücks, welches von einer linearen Backe geschoben wird, unter gewissen Annahmen einer Rotation entspricht. Weiterhin wurden Angaben zu der Geschwindigkeit der Rotation gemacht. Dieser Abschnitt beschäftigt sich mit weiteren Details der Bewegung des Werkstücks.

Um die Rotationsrichtung des Werkstücks zu bestimmen, verwendet Brost [5] sogenannte *Reibungskegel (friction cones*). Ein Reibungskegel existiert an jedem Punkt-Flächen-Kontakt und wird berandet durch zwei Strahlen R_l und R_r , welche mit der Normale der am Kontakt beteiligten Fläche einen Winkel von $\alpha = \tan^{-1}(\mu)$ bilden (siehe Abbildung 5.1). Der Wert μ stellt den Reibungskoeffizienten zwischen der am Kontakt beteiligten Fläche und dem Punkt dar.



Abbildung 5.1: Ein Reibungskegel, definiert durch die Strahlen R_l und R_r mit Winkel α zu der gestrichtelten Flächennormale

Brost betrachtet weiterhin den Strahl R_p , welcher die Translationsrichtung der linearen Backe angibt. Für die Rotationsrichtung des Werkstücks während der Translation der Backe lässt sich die folgende, anschauliche Regel angeben: verlaufen mindestens zwei der drei Strahlen links des Massenschwerpunkts des Werkstücks, so rotiert das Werkstück im Uhrzeigersinn. Liegen dagegen mindestens zwei der drei Strahlen rechts des Massenschwerpunkts, so rotiert das Werkstück entgegen dem Uhrzeigersinn. Einen Sonderfall stellt die Situation dar, in der einer der Strahlen genau auf den Massenschwerpunkt zeigt und die beiden anderen Strahlen auf unterschiedlichen Seiten des Massenschwerpunkts liegen. In diesem Fall entspricht die Bewegung des Werkstücks einer Translation. Das Werkstück in der Abbildung 5.1 rotiert also im Uhrzeigersinn, da die Strahlen R_l und R_p links des Massenschwerpunkts verlaufen.

Neben der Ermittlung der Rotationsrichtung eines Werkstücks können mit Hilfe dieser Methode die Kanten des Werkstücks als stabil oder instabil klassifiziert werden. Hierfür sind an den beiden Endpunkten der betrachteten, an der linearen Backe ausgerichteten Kante des Werkstücks die Reibungskegel wie beschrieben auszuwerten. Votieren beide Kegel für die gleiche Rotationsrichtung des Werkstücks, so ist die Kante als instabil einzustufen. Votieren die beiden Reibungskegel dagegen für unterschiedliche Rotationsrichtungen, so ist die betrachtete Kante stabil (siehe Abbildung 5.2).

5.2 Implikationen von Unsicherheit für Goldbergs Verfahren

Für die Ermittlung und Angabe von technischen Leistungsdaten industrieller Roboter existieren einige Kriterien und genormte Ermittlungsverfahren [16]. Der Begriff der Wiederholungsgenauigkeit eines Manipulators bei-



Abbildung 5.2: Zwei Reibungskegel klassifizieren die kontaktierende Kante als stabil: der linke Reibungskegel votiert für eine Rotation im Uhrzeigersinn, der rechte Kegel votiert für eine Rotation gegen den Uhrzeigersinn.

spielsweise beschreibt die Genauigkeit, mit welcher ein Roboter eine feste Position wiederholt anfahren kann. Da die Steuerung eines Gelenkarmroboters in diesem Sinne fehlerbehaftet ist, entsteht Unsicherheit über die tatsächliche Position und Orientierung des Greifers. Eine weitere Form von Unsicherheit im Kontext der Manipulation von Werkstücken ergibt sich aus der Tatsache, dass die Maße und die Massenverteilung der Werkstücke (und damit die Position des Massenschwerpunkts) produktionsbedingt leichten Schwankungen unterliegen. Im Folgenden sollen mit Hilfe des vorhergehenden Abschnitts die Konsequenzen derartiger Unsicherheit für das Verfahren von Goldberg erläutert werden.

Die Rotationsrichtung eines Werkstücks und die Stabilität von Werkstückkanten lässt sich mittels der Betrachtung der Position des Massenschwerpunkts und des Verlaufs der drei Strahlen R_p , R_l und R_r bestimmen. Im Umkehrschluss können Umstände, welche eine Richtungsänderung der Strahlen bewirken, die Bewegung des Werkstücks qualitativ beeinflussen. Variationen der Translationsrichtung der linearen Backe z. B. resultieren per Definition in einer Änderung der Richtung des Strahls R_p . Weiterhin geht mit einer Variation der Orientierung der linearen Backe offensichtlich eine Änderung der Richtung des Strahlen R_l und R_r einher. Variationen der Geometrie und der Massenverteilung des Werkstücks, welche die Position des Massenschwerpunkts verändern, können offensichtlich ebenfalls die Art der Bewegung des Werkstücks beeinflussen.

Goldbergs Verfahren geht während der Planung von Aktionsplänen von bekannten Geometriedaten der Werkstücke und von einer präzisen Robotersteuerung aus. Ideale Bedingungen sind in der Praxis nicht zu erwarten. Unsicherheit über die Geometrie und Massenverteilung der Werkstücke, ebenso wie Unsicherheit über die Translationsrichtung und die Orientierung des Greifers kann dazu führen, dass die Rotationsrichtung eines geschobenen Werkstücks sowie die Stabilität von Werkstückkanten von den während der Berechnung des Aktionsplans angenommenen Sachverhalten abweicht. In der Folge entstehen im Verlauf des Orientierungsvorgangs ungeplante Orientierungen des Werkstücks, ein Fehlschlag des Verfahrens kann daher nicht ausgeschlossen werden.

Durch eine Erhöhung der Präzision der Robotersteuerung sowie durch eine Bestimmung der tatsächlichen Werkstücksmaße lässt sich die Unsicherheit reduzieren. Akella und Mason [1] begegnen der herrschenden Unsicherheit bezüglich der Werkstückmaße auf eine andere Weise: sie berücksichtigen explizit Toleranzen für die Position des Massenschwerpunkts und für die Position der Werkstückecken. Die gegebene Geometrie eines Werkstücks und die entsprechenden Toleranzen definieren eine Menge von unendlich vielen Werkstücken. Das Ziel der Arbeit von Akella und Mason ist die Berechnung eines Aktionsplans, welcher alle Instanzen einer solchen Menge orientiert. Sie beschreiben eine Klasse von Werkstücken, welche mit ihrem Ansatz unter Unsicherheit betreffend der Werkstückmaße – mit oder ohne die Verwendung von Sensorik – erfolgreich orientiert werden kann.

5.3 Anpassung von Goldbergs Ansatz

Im Rahmen dieser Arbeit wurde versucht, die Robustheit von Goldbergs Algorithmus zur Berechnung von Aktionsplänen durch gezielte Modifikationen zu verbessern. Zu diesem Zweck wurden eigene Ideen entwickelt und in der Umsetzung von Goldbergs Verfahren verwirklicht. Im Folgenden wird der Ansatz genauer beschrieben.

5.3.1 Rotationsrichtung des Werkstücks

Das erste Ziel war die Vermeidung von Situationen, in denen das Werkstück nach der Entstehung eines Kontaktes mit der linearen Backe des Greifers entgegengesetzt der beabsichtigten Richtung rotiert. Im Kontext von Goldbergs Verfahren ist die Translationsrichtung des Greifers orthogonal zu der Orientierung der Backe. Sei g die Gerade, welche durch den Massenschwerpunkt des Werkstücks und durch den Kontaktpunkt zwischen dem Werkstück und der Backe verläuft (siehe Abbildung 5.3). Kritische Orientierungen des Werkstücks zeichnen sich durch einen kleinen Winkel zwischen dem Strahl R_p und der Geraden g aus. Für kleine Winkel kann es aufgrund von geringen Abweichungen der tatsächlichen Position des Massenschwerpunkts oder der Translationsrichtung des Greifers zu einer qualitativen Veränderung der relativen Lagen des Strahls R_p und des Massenschwerpunkts kommen. Dies zieht eine Anderung der Rotationsrichtung des Werkstücks nach sich, und die berechnete Schiebefunktion des Werkstücks liefert falsche Resultate. Derartige, kritische Situationen sind daher möglichst zu vermeiden.

Der ursprüngliche Algorithmus von Goldberg wurde in Abschnitt 2.1 vorgestellt. Die Idee des Verfahrens ist, eine Folge von t-Intervallen der Transferfunktion t zu berechnen, mit deren Hilfe die Orientierung des Werkstücks während der Ausführung des Aktionsplans trichterartig eingeschränkt wird,

5.3. ANPASSUNG VON GOLDBERGS ANSATZ

bis eine Endorientierung erreicht wird. An dieser Stelle soll zur Erinnerung nochmals der Algorithmus für die Berechnung der Folge von t-Intervallen aufgeführt werden (siehe Algorithmus 5.3.1). In jeder Iteration der while-Schleife wird ein Element der Folge von größer werdenden t-Intervallen berechnet. Es ist zu beachten, dass in diesem Algorithmus die Greiffunktion sdurch die Transferfunktion t ersetzt wurde (vgl. Abschnitt 2.1.6). Dies führt zu einer Änderung der Bezeichnungen s-Intervall und s-Image zu t-Intervall und t-Image.



Abbildung 5.3: Eine kritische Orientierung des Werkstücks

Algorithmus 5.3.1 Berechnung einer Folge von t-Intervallen

Eingabe: planares, konvexes Polygon P

- 1: Berechne die Transferfunktion t von P
- 2: Finde das größte t-Intervall Θ_1 , auf dem t konstant ist
- 3: $i \leftarrow 1$
- 4: while es existiert ein t-Intervall Θ mit $| t(\Theta) | < | \Theta_i |$ do
- 5: Θ_{i+1} sei das größte dieser Intervalle
- $6: \quad i \leftarrow i+1$
- 7: end while
- 8: Gebe die Liste $(\Theta_1, \Theta_2, \ldots, \Theta_i)$ zurück

Die berechneten t-Intervalle werden während der Ausführung eines Aktionsplans in umgekehrter Reihenfolge verwendet. Die Abbildung 5.4 verdeutlicht die Situation während der Ausführung des Aktionsplans: die Orientierung des Werkstücks liegt in dem Intervall Θ_{i+1} und wird durch eine Translation des Greifers und einen Greifvorgang in das t-Image $t(\Theta_{i+1})$ überführt. Der nächste Schritt soll das Intervall Θ_i in das t-Image $t(\Theta_i)$ überführen.

Wie lassen sich die kritischen Orientierungen des Werkstücks ermitteln, um diese zu vermeiden? Der Übergang von einer Rotationsrichtung des Werkstücks zu der entgegengesetzten Rotationsrichtung in Abbildung 5.3 erfolgt, wenn der Massenschwerpunkt des Werkstücks den Strahl R_p überschreitet. Diese Situation auf die Schiebefunktion p des Werkstücks übertragen entspricht der Überschreitung einer Intervallgrenze der Schiebefunktion. Es gilt $t := s \circ p$. Man macht sich leicht klar, dass die Schiebefunktion p die Intervallgrenzen von t vorgibt, während die beiden Funktionen p und s die Werte von t auf diesen Intervallen bestimmen. Eine kritische Orientierung des Werkstücks ist also dann gegeben, wenn die vor der Translation des Greifers realisierte Orientierung des Werkstücks nahe des linken oder des rechten Randes des t-Intervalls Θ_i liegt. In diesem Fall liegt die realisierte Orientierung ebenso nahe an einer Intervallgrenze der Schiebefunktion p.



Abbildung 5.4: Der Orientierungsvorgang im Detail

Das Auftreten von kritischen Orientierungen des Werkstücks kann durch einen Sicherheitsabstand zwischen den Rändern von $t(\Theta_{i+1})$ und Θ_i reduziert werden. Ein großer Sicherheitsabstand garantiert einen großen Winkel zwischen dem Strahl R_p und der Geraden g in der Abbildung 5.3, bzw. einen großen Abstand der Orientierung des Werkstücks von einer Orientierung, welche einer Intervallgrenze der Schiebefunktion p entspricht.

Sei $\epsilon := |\Theta_i| - |t(\Theta_{i+1})|$. Durch die Zentrierung des Intervalls $t(\Theta_{i+1})$ im Intervall Θ_i wird eine Fehlertoleranz von $\frac{\epsilon}{2}$ ermöglicht. Goldberg verwendet den Wert $\frac{\epsilon}{2}$ lediglich für die Zentrierung von Intervallen. In jeder Iteration der while-Schleife des Basisalgorithmus werden alle t-Intervalle Θ mit $|t(\Theta)| < |\Theta_i|$ gesucht, anschließend wird das größte der gefundenen Intervalle als Θ_{i+1} gewählt. Da die Wahl unabhängig von ϵ geschieht, kann ϵ für das gewählte Intervall sehr klein sein, was für die Durchführung des Verfahrens in der Praxis ein Risiko darstellt. Beträgt ϵ beispielsweise 0.01, so entspricht die Fehlertoleranz $\frac{\epsilon}{2}$ einem Winkel von etwa 0.28 Grad. Das bedeutet, das eine Abweichung der Orientierung des Werkstücks vor der entsprechenden Translation des Greifers um 0.3 Grad bereits zu einer falschen Rotationsrichtung des Werkstücks führen kann.

Dieser Umstand lässt sich ändern: der Basisalgorithmus ist an der entsprechenden Stelle zu modifizieren, um in jeder Iteration der while-Schleife alle Intervalle Θ mit $|t(\Theta)| + \chi < |\Theta_i|$ zu finden, wobei $\frac{\chi}{2}$ die gewünschte Fehlertoleranz darstellt. Dies schließt im Vergleich zu der ursprünglichen Vorgehensweise eventuell einige Intervalle aus der Lösungsmenge aus. Aus den gefundenen Intervallen wird anschließend nach den ursprünglichen Kriterien das Intervall Θ_{i+1} ausgewählt. Unabhängig von der Wahl des Intervalls gilt für die Fehlertoleranz $\frac{\epsilon}{2}$:

$$\frac{\epsilon}{2} = \frac{|\Theta_i| - |t(\Theta_{i+1})|}{2} > \frac{\chi}{2} \tag{5.1}$$

Bemerkung Die in diesem und dem folgenden Abschnitt besprochenen Änderungen haben einige weitergehende Konsequenzen, welche im Anschluss erläutert werden.

5.3.2 Stabilität von Werkstückkanten

Das zweite Ziel bestand in der Vermeidung von negativen Auswirkungen in Fällen, in denen sich eine als stabil eingestufte Werkstückkante während der Translation des Greifers als instabil herausstellt. Die entsprechende Situation ist nochmals in Abbildung 5.5 dargestellt.



Abbildung 5.5: (i) Ein Kontakt zwischen dem Werkstück und dem Greifer entsteht, (ii) das Werkstück befindet sich in der geplanten Orientierung, (iii) die Rotation des Werkstücks setzt sich unplanmäßig fort (iv) und endet in einer benachbarten stabilen Orientierung.

Die Orientierung des Werkstücks wird nach der Translation des Greifers und nach dem anschließenden Greifvorgang planmäßig innerhalb eines bestimmten t-Image erwartet. Tritt während der Translation des Greifers ungeplant eine instabile Werkstückkante auf, so endet der anschließende Greifvorgang abhängig von der Situation eventuell mit einer unbeabsichtigten Orientierung des Werkstücks. Befindet sich die unbeabsichtigt realisierte Orientierung ebenfalls in dem erwarteten t-Image, so stellt die abweichende Orientierung des Werkstücks im weiteren Verlauf des Algorithmus kein Problem dar. In diesem Fall wird *spätestens* nach der vollständigen Abarbeitung des Aktionsplans wieder die planmäßige Orientierung des Werkstücks hergestellt. Vermieden werden müssen Situationen, in denen das Werkstück nach der Translation des Greifers und dem anschließenden Greifvorgang in einer Orientierung endet, welche außerhalb des erwarteten t-Image liegt. Eine Möglichkeit besteht in der Betrachtung der Stabilität von Orientierungen des Werkstücks während der Translation des Greifers. Ist die im planmäßig erwarteten t-Image am weitesten links liegende, als stabil eingestufte Orientierung, sowie die im t-Image am weitesten rechts liegende, als stabil eingestufte Orientierung auch unter Unsicherheit stabil, so kann das Werkstück während der Translation des Greifers nicht in eine Orientierung außerhalb dieses t-Image gelangen. Laut Rao und Goldberg minimiert ein Schiebevorgang den Radius des Werkstücks [31]. Stabile Orientierungen des Werkstücks entsprechen daher den lokalen Minima in der Radiusfunktion des Werkstücks. Gesucht ist ein Kriterium für die Bewertung der Stabilität von lokalen Minima in der Radiusfunktion unter Unsicherheit.

Es existiert ein einfaches Kriterium: die beiden Reibungskegel an den Endpunkten einer stabilen Werkstückkante votieren für entgegegengesetzte Rotationsrichtungen des Werkstücks (vgl. Abschnitt 5.1). Die stabile Werkstückkante wird instabil, wenn einer der beiden Reibungskegel sein Votum ändert. Der Übergang von einem Votum zu dem anderen geschieht für einen Reibungskegel bei einem Winkel zwischen dem Strahl R_p und der Geraden g von 0 Grad (siehe Abbildung 5.3). Wie im letzten Abschnitt bereits beschrieben wurde, trägt ein großer Winkel zwischen dem Strahl R_p und der Geraden g zu einer Reduzierung dieser Gefahr bei. Anders ausgedrückt: je größer der Abstand zwischen einem lokalen Minimum und einem benachbarten lokalen Maximum in der Radiusfunktion ist, desto größer müssen Abweichungen werden, damit die entsprechende Werkstückkante instabil wird. Der Abstand der beiden Extremstellen kann als Bewertungskriterium für die Stabilität der Werkstückkante verwendet werden.

Für alle t-Intervalle sind die zwei lokalen Minima der Radiusfunktion zu betrachten, welche in dem t-Intervall links und rechts am weitesten außen liegen (siehe Abbildung 5.6). Das rechts liegende Minimum wird anhand des Abstands zu dem lokalen Maximum auf der rechten Seite bewertet. Das links liegende Minimum wird mittels seines Abstands zu dem lokalen Maximum auf der linken Seite bewertet. Hat eines der beiden Minima nicht den geforderten Sicherheitsabstand δ zu dem entsprechenden Maximum, so wird das betrachtete t-Intervall im Algorithmus zur Berechnung der Folge von t-Intervallen nicht berücksichtigt. Enthält ein t-Intervall nur ein lokales Minimum der Radiusfunktion, so ist dieses Intervall während der Berechnung zu berücksichtigen, falls das Minimum zu beiden benachbarten Maxima jeweils einen Mindestabstand von δ aufweist.



Abbildung 5.6: Die Stabilität zweier lokaler Minima in einem t-Intervall wird anhand von Abstandskriterien bewertet.

5.3.3 Implementierung eines modifizierten Algorithmus

Werden die besprochenen Änderungen einfach in den Basisalgorithmus zur Berechnung der Folge von t-Intervallen integriert, so ergeben sich Probleme: durch die gegebenen Toleranzwerte $\frac{\chi}{2}$ und δ und aufgrund der beschriebenen Modifikationen fallen während der Berechnung der Folge von t-Intervallen eventuell Kandidaten weg, welche im ursprünglichen Algorithmus zu der Lösungsmenge gehört hätten. Es kann nicht garantiert werden, dass eine Lösung in Form einer Folge von t-Intervallen überhaupt existiert.

Auch die Entscheidung, ein bestimmtes t-Intervall Θ_i in die Lösungsmenge aufzunehmen, kann im weiteren Verlauf der Berechnung in einer Sackgasse enden: existiert in einer Iteration der while-Schleife kein Intervall Θ mit $|t(\Theta)| + \chi < |\Theta_i|$, so kann der Lösungsmenge in dieser Iteration kein weiteres t-Intervall hinzugefügt werden. Eventuell entspricht die Größe des der Lösungsmenge in der vorigen Iteration hinzugefügten t-Intervalls Θ_i nicht der Periode T der Transferfunktion. Für die Beendigung des Algorithmus ist diese Bedingung aber notwendig (vgl. Abschnitt 2.1.5). In diesem Fall muss eine getroffene Entscheidung rückgängig gemacht werden. Das t-Intervall Θ_i wird wieder aus der Lösungsmenge entfernt und der nächste Kandidat aus der vorigen Iteration wird als Θ_i verwendet. Dies geschieht so lange, bis in der while-Schleife der nächsten Iteration wieder mindestens ein Kandidat ermittelt werden kann, sodass die Lösungsmenge um Θ_{i+1} bereichert wird. Ist dies für kein Θ_i möglich, so muss noch ein weiterer Schritt rückgängig gemacht und das Intervall Θ_{i-1} aus der Lösungsmenge entfernt werden. Gelangt der Algorithmus auf diese Weise zurück bis zu der Entscheidung über Θ_1 , und wurden bereits alle Kandidaten für Θ_1 vergeblich ausprobiert, so existiert keine Lösung. Das beschriebene Schema entspricht einem Backtracking-Algorithmus. Die Abbildung 5.7 skizziert das Vorgehen. Allgemein wird versucht, aufgrund der bereits getroffenen Entscheidungen eine Lösung aufzubauen. Bereits geprüfte, nicht zum Ziel führende Wege werden grau dargestellt.

Auf der Basis der bisher beschriebenen Ideen wurde ein Backtracking-Algorithmus implementiert und in die Java-Anwendung für die Berechnung



Abbildung 5.7: Eine Lösung wird mittels Backtracking ermittelt.

von Aktionsplänen integriert. In der Anwendung können die beiden Fehlertoleranzen $\frac{\chi}{2}$ und δ beliebig gewählt werden, bevor ein Aktionsplän berechnet wird. Für einen berechneten Plan kann die tatsächlich erzielte Fehlertoleranz $\frac{\chi^*}{2}$ angezeigt werden. Diese entspricht dem niedrigsten Sicherheitsabstand zwischen einem t-Image $t(\Theta_{i+1})$ und einem t-Intervall Θ_i in dem berechneten Aktionsplan. Weiterhin lässt sich die Anzahl der t-Intervalle anzeigen, welche aufgrund der verwendeten Fehltertoleranz δ nicht in den Berechnungen des Aktionsplans berücksichtigt wurden.

Der implementierte Algorithmus gliedert sich im Wesentlichen in zwei Methoden. Die Hauptmethode (Algorithmus 5.3.2) berechnet die Transferfunktion t, deren Periode, sowie die Menge Ω derjenigen t-Intervalle, deren äußere lokale Minima auch unter der Fehlertoleranz δ stabil sind. In weiteren Berechnungen wird die Menge Ω anstatt der Menge aller t-Intervalle verwendet. Die while-Schleife der Hauptmethode (Zeilen 5 bis 11) entspricht der Entscheidung, welcher Kandidat als Intervall Θ_1 in die Lösungsmenge aufgenommen wird.

Die eigentliche Rekursion findet in der Methode SEARCHINTERVAL (Algorithmus 5.3.3) statt. Die Zeilen 1 bis 7 erstellen eine Kandidatenmenge für die jeweilige Rekursionstiefe, welche alle potentiellen Kandidaten für das Intervall Θ_i der Lösungsmenge enthält. Die Zeilen 11 bis 14 führen die Rekursion fort. Die Zeilen 15 bis 19 revidieren – falls nötig – die bereits getroffene Entscheidung über Θ_i zugunsten eines anderen Intervalls aus der Kandidatenmenge. Abschließend gibt die Methode SEARCHINTERVAL die Lösungsmenge an den Aufrufer zurück.

Im Folgenden soll ein Korrektheitsbeweis für den modifizierten Algorithmus skizziert werden. Durch die beschriebenen Modifikationen werden die Kandidatenmengen für die Intervalle Θ_i der Lösungsmenge im Vergleich zu Goldbergs Algorithmus höchstens verkleinert. Daher kann nicht garantiert werden, dass eine Lösung existiert. Die Berechnung bricht ab, wenn die Größe des zuletzt der Lösungsmenge hinzugefügten Intervalls Θ_i der Peri-

Algorithmus	5.3.2 Die	Hauptmethe	ode des	modifizierten	Algorithmus
-------------	------------------	------------	---------	---------------	-------------

Eingabe:

planares, konvexes Polygon P,

Fehlertoleranzen $\frac{\chi}{2}$ und δ .

- 1: Berechne die Transferfunktion t von P
- 2: Berechne die Periode T der Transferfunktion
- 3: Ermittle die Menge der kritischen Maxima, deren Abstand zu einem benachbarten Minimum kleiner ist als δ
- 4: Ermittle die Menge Ω der t-Intervalle, welche nicht an einem kritischen Maximum starten oder enden
- 5: while |solution.lastElement()| < T und Ω enthält ein konstantes t-Intervall do
- 6: **if** solution ist nicht leer **then**
- 7: Lösche das letzte Element aus *solution*
- 8: end if
- 9: Füge ein konstantes t-Intervall aus Ω zu solution hinzu und lösche dieses aus Ω
- 10: solution = SEARCHINTERVAL(Ω , solution, $\frac{\chi}{2}$)
- 11: end while
- 12: Gebe die Lösungsmenge solution zurück

ode T der Transferfunktion entspricht, oder wenn alle Möglichkeiten erfolglos ausprobiert wurden. Existiert mindestens eine Lösung für die gegebenen Fehlertoleranzwerte $\frac{\chi}{2}$ und δ , so wird diese bzw. eine der Lösungen aufgrund des eingesetzten Backtracking-Ansatzes gefunden und zurückgegeben. Anschließend wird aus der Folge von t-Intervallen wie üblich der Aktionsplan berechnet. Weiterhin führt die gefundene Lösung zu einem Aktionsplan, welcher die Orientierung des Werkstücks während der Ausführung trichterartig einschränkt. Die grundsätzlichen Ideen von Goldbers Algorithmus bleiben also unberührt, die berechneten Aktionspläne sind daher korrekt.

Für ein Polygon mit n Ecken berechnet Goldbergs ursprüngliches Verfahren in Zeit $O(n^2 \log n)$ einen Plan der Länge $O(n^2)$ (vgl. Abschnitt 2.1). Längere Pläne können auch mit dem modifizierten Verfahren nicht berechnet werden, da es $O(n^2)$ viele t-Intervalle gibt (Goldberg, [15]). Für die Werte $\frac{\chi}{2} = 0$ und $\delta = 0$ stimmt die Lösung des modifizierten Algorithmus mit Goldbergs Algorithmus überein. Daher sind die von dem modifizierten Algorithmus berechneten Pläne im Worst-Case gleichlang.

Die Worst-Case Laufzeit wird durch die Rekursion dominiert. Die Kandidatenmenge in SEARCHINTERVAL kann $O(n^2)$ viele t-Intervalle enthalten. Dies entspricht dem maximalen Verzweigungsgrad der Rekursion. Die Rekursionstiefe entspricht im Worst-Case der Planlänge und somit ebenfalls $O(n^2)$. Dies ergibt eine Worst-Case Laufzeit von

Algorithmus 5.3.3 Die rekursive Methode SEARCHINTERVAL
Eingabe:
Menge Ω mit t-Intervallen,
Lösungsmenge solution,
Fehlertoleranz $\frac{\chi}{2}$.
1: Erstelle eine leere Kandidatenmenge
2: Für alle t-Intervalle ω_i aus Ω do
3: Sei $image_i$ das entsprechende t-image zu dem t-intervall ω_i
4: if $image_i + \chi < solution.lastElement() $ then
5: Füge ω_i der Kandidatenmenge hinzu
6: end if
7: End
8: if die Kandidatenmenge ist leer then
9: Return solution
10: end if
11: Füge der Lösungsmenge <i>solution</i> ein Element der Kandidatenmenge hin-
zu und lösche es aus der Kandidatenmenge
12: if $ solution.lastElement() < T$ then
13: SEARCHINTERVAL(Ω , solution, $\frac{\chi}{2}$)
14: end if
15: while $ solution.lastElement() < T$ und die Kandidatenmenge ist nicht
leer \mathbf{do}
16: Ersetze das letzte Element in <i>solution</i> durch ein Element aus der
Kandidatenmenge
17: Lösche das zu <i>solution</i> hinzugefügte Element aus der Kandidaten-
menge

- 18: SEARCHINTERVAL(Ω , solution, $\frac{\chi}{2}$)
- 19: end while
- 20: Gebe die Lösungsmengesolutionzurück

$$O\left(n^{2^{n^2}}\right) = O\left(n^{2^{n^2}}\right). \tag{5.2}$$

In der Praxis ist der Verzweigungsgrad wesentlich geringer, da die Auswahlkriterien für die Kandidatenmengen der einzelnen Rekursionstiefen die meisten der $O(n^2)$ t-Intervalle verwerfen. Für die Berechnung von Aktionsplänen auf einem Pentium 4 war keine spürbare Wartezeit nötig.

5.4 Tests

In erneuten Testreihen mit den im Abschnitt 4.7 verwendeten Werkstücken sollten Erkenntnisse über die Wirksamkeit des modifizierten Verfahrens gewonnen werden. Das Auftreten von ungeplanten, instabilen Orientierungen wurde im Rahmen der erneuten Testreihen im Gegensatz zu den vorherigen Testläufen nicht als negativ gewertet, da diese instabilen Orientierungen zu einer Endorientierung des Werkstücks führten, welche aufgrund der Vorkehrungen für den Erfolg des Orientierungsvorgangs ungefährlich war. Eine Rotation des Werkstücks entgegengesetzt der geplanten Richtung wäre hingegen einer zu gering gewählten Fehlertoleranz $\frac{\chi}{2}$ anzulasten und negativ zu werten. Solche Situationen wurden neben den korrekten und den fehlgeschlagenen Testläufen in einer dritten Kategorie separat gezählt.

5.4.1 Ergebnisse

Das erste Werkstück wies in den ersten Testreihen eine instabile Orientierung auf, welche nach den Berechnungen hätte stabil sein müssen. Daher wurde die Fehlertoleranz δ in den erneuten Testreihen auf 0.2 gesetzt. Mit einer Fehlertoleranz $\frac{\chi}{2} = 0$ ergab sich ein Plan mit fünf Winkeln und einer erzielten Fehlertoleranz $\frac{\chi^2}{2}$ von etwa 0.175. Mit einer Fehlertoleranz $\frac{\chi}{2}$ von 0.18 wurde ein Plan mit sechs Winkeln berechnet, die erzielte Fehlertoleranz $\frac{\chi^2}{2}$ des Plans betrug etwa 0.23. Während der Ausführung des Plans mit sechs Winkeln trat kein Fehlverhalten auf. Dies stellt eine Verbesserung im Vergleich zu den ersten Testreihen dar. Durch eine weitere Erhöhung der Fehlertoleranz $\frac{\chi}{2}$ auf 0.24 konnte ein robusterer Plan mit sieben Winkeln und einer erzielten Fehlertoleranz $\frac{\chi^2}{2}^*$ von etwa 0.33 berechnet werden.

Das Werkstück zwei wurde nicht getestet, da es mit einem Durchmesser von 4.2 Zentimetern zu groß war.

Das dritte Werkstück wies in den ersten Testreihen keine instabilen Orientierungen auf, der Parameter δ wurde daher in den erneuten Testreihen auf 0 belassen. Mit $\frac{\chi}{2} = 0$ ergab sich der ursprüngliche Aktionsplan mit zwei Winkeln, welcher eine Fehlertoleranz $\frac{\chi^*}{2}$ von etwa 0.31 aufwies. Dieser relativ hohe Wert konnte erklären, warum für dieses Werkstück in den ersten Testreihen kein Fehlverhalten festgestellt wurde. Es konnte kein robusterer Plan berechnet werden. Für das dritte Werkstück wurden daher die Testresultate der ersten Testreihen übernommen.

Das Werkstück vier besaß in den ersten Testläufen eine fälschlicherweise als stabil eingestufte Orientierung. Der berechnete Plan bestand bei Werten von δ zwischen 0 und 0.7 immer aus den gleichen Winkeln, obwohl die Menge Ω bei Aktionsplänen mit $\delta > 0.1$ einige der kritischen t-Intervalle nicht enthielt. Daraus konnte geschlossen werden, dass bereits der Aktionsplan aus den ersten Testreihen als robust in Bezug auf die instabile Orientierung des vierten Werkstücks einzustufen war. Die erzielte Fehlertoleranz $\frac{\chi^*}{2}$ der mit $\frac{\chi}{2} = 0$ berechneten Aktionspläne betrug unabhängig von δ 0.189. Mit $\frac{\chi}{2} = 0.19$ ließ sich unabhängig von δ kein robusterer Aktionsplan berechnen. Die Durchführung der Testläufe mit dem ursprünglichen Aktionsplan wurde wiederholt. Die in der ersten Testreihe beobachteten Abweichungen beruhten auf der instabilen Orientierung des Werkstücks und wurden wieder beobachtet. Dieses Fehlverhalten wurde durch den modifizierten Algorithmus aber entschärft und daher in den erneuten Testreihen als fehlerfrei bewertet.

Das Werkstück fünf besaß ähnlich wie das Werkstück drei keine falsch eingestuften Orientierungen und gleichzeitig eine hohe erzielte Fehlertoleranz $\frac{\chi^*}{2}$ von etwa 0.38 für den dreiwinkligen Plan, welcher mit $\frac{\chi}{2} = 0$ berechnet wurde. Die festgestellte Robustheit des Plans deckte sich mit den Ergebnissen der ersten Testreihen, welche in drei Testläufen keinen Fehler ergaben. Die Resultate aus den ersten Testreihen wurden daher übernommen.

Ein fehlerhafter Testlauf und drei korrekte Läufe des sechsten Werkstücks stellten eine Verbesserung im Vergleich zu der ersten Testserie dar. Die erzielte Fehlertoleranz $\frac{\chi^*}{2}$ konnte von etwa 0.10 in den ersten Testläufen auf etwa 0.21 erhöht werden. Dieser Wert reichte augenscheinlich nicht aus, um die auftretenden Abweichungen der Werkstücksmaße und der Lage des Greifers gänzlich zu kompensieren.

Das Ergebnis des siebten Werkstücks enthielt einen fehlgeschlagenen Testlauf und zwei korrekte Läufe, was im Vergleich zu den ersten Testreihen eine Verbesserung darstellt. Die erzielte Fehlertoleranz $\frac{\chi^*}{2}$ konnte von etwa 0.01744 in den ersten Testläufen auf etwa 0.1616 erhöht werden.

Für das achte Werkstück wurde ein Plan mit den Werten $\delta = 0.4$ und $\frac{\chi}{2} = 0.14$ berechnet. Dies ergab eine erzielte Fehlertoleranz $\frac{\chi}{2}^*$ von etwa 0.149. Für das achte Werkstück ergaben die erneuten Testreihen trotz der Verwendung des robusteren Plans keine Veränderung der Resultate. Die maximal zu erzielende Fehlertoleranz von etwa 0.149 konnte offensichtlich keine korrekte Orientierung des Werkstücks gewährleisten.

Für das neunte Werkstück konnte im Vergleich zu der ersten Testserie kein robusterer Plan berechnet werden. Für dieses Werkstück wurde daher das Resultat von drei fehlgeschlagenen Testläufen aus den ersten Testreihen übernommen. Dieses schlechte Ergebnis lässt sich mit einer großen Abweichung von der planmäßigen Orientierung des Werkstücks während der Durchführung erklären. Diese kam u.a. durch Verklemmungen des Werkstücks im Greifer zustande.

Das zehnte Werkstück besaß in der ersten Testreihe eine fälschlicherweise als stabil eingestufte Orientierung. Der ursprüngliche Plan bestand aus drei Winkeln und erlaubte eine Fehlertoleranz $\frac{\chi^*}{2}$ von etwa 0.0856. Ein robuster Plan mit vier Winkeln erlaubte eine Toleranz $\frac{\chi^*}{2}$ von etwa 0.3976. Gemäß den Erwartungen waren alle drei neu durchgeführten Testläufe mit dem Werkstück zehn erfolgreich.

Die Abbildung 5.8 fasst die erzielten Resultate zusammen. In der Spalte *Winkel* wird die Anzahl der Winkel des Aktionsplans angegeben, in Klammern dahinter steht die Anzahl der Winkel des Plans aus der ersten Testserie. Von den 32 Testläufen waren 24 fehlerfrei, 7 schlugen fehl und ein Testlauf wies ein Fehlverhalten auf, endete aber in der korrekten Orientierung. Die Anzahl der Testläufe mit Fehlverhalten wurde im Vergleich zu der ersten Testserie von 5 auf einen gesenkt, die Anzahl der Fehlschläge wurde von 13 auf 7 gesenkt, andererseits erhöhte sich die Anzahl der korrekten Testläufe von 14 auf 24.

Werkstück	Winkel	Testläufe	Fehlerfrei	Fehler	Fehlschlag
1	6(5)	4	4	0	0
2	4 (4)	-	-	-	-
3	2(2)	4	4	0	0
4	3(3)	3	2	0	1
5	3(3)	3	3	0	0
6	6 (4)	4	3	1	0
7	4 (3)	3	2	0	1
8	6(5)	5	3	0	2
9	3(3)	3	0	0	3
10	4 (3)	3	3	0	0
Gesamt	-	32	24	1	7

Abbildung 5.8: Testresultate der zweiten Testserie

5.5 Zusammenfassung

In diesem Kapitel wurden beruhend auf den Arbeiten von Brost [5] einige Details betreffend der Bewegung eines polygonalen Werkstücks erläutert, welches von einer linearen Backe geschoben wird. Diese Erkenntnisse wurden für eine Analyse der gleichen Situation unter Unsicherheit genutzt. Die Analyse erbrachte einige Ursachen, welche die ungeplante Instabilität von Werkstückkanten sowie qualitative Änderungen der Rotationsrichtung des Werkstücks unter Unsicherheit erklären können. Aufbauend auf diesen Grundlagen wurden Änderungsvorschläge für Goldbergs Algorithmus zur Berechnung einer Folge von größer werdenden t-Intervallen erarbeitet, um eine höhere Robustheit von Aktionsplänen unter Unsicherheit zu erzielen.

Die Anderungsvorschläge wurden in die entstandene Umsetzung von Goldbergs Verfahren integriert. Der Benutzer hat die Möglichkeit, mittels zweier Fehlertoleranzen auf die Berechnung der Aktionspläne Einfluss zu nehmen. Der implementierte Algorithmus wurde in Abschnitt 5.3.3 vorgestellt.

Anschließend wurden mit den während der ersten Testserie verwendeten Werkstücken erneute Testläufe durchgeführt. Die verwendeten Aktionspläne besaßen im Vergleich zu der ersten Serie in einem Fall zwei zusätzliche Winkel, für die anderen Werkstücke ergaben sich Aktionspläne gleicher Länge oder mit einem zusätzlichen Winkel. Die Ergebnisse der insgesamt 32 Testläufe fielen im Vergleich zu der ersten Testserie deutlich besser aus: die Anzahl der Testläufe mit Fehlverhalten wurde im Vergleich zu der ersten Testserie von 5 auf einen gesenkt, die Anzahl der Fehlschläge wurde von 13 auf 7 gesenkt, andererseits erhöhte sich die Anzahl der korrekten Testläufe von 14 auf 24. Für einige Werkstücke konnte mit dem implementierten, modifizierten Algorithmus allerdings keine höhere Robustheit des Aktionsplans erzielt werden. Für diese Werkstücke ergaben sich nach wie vor Fehlschläge in der Durchführung.

Durch die Verwendung des modifizierten Algorithmus kann die Fehlerquote des Verfahrens unter Unsicherheit für bestimmte Werkstücke verbessert werden.

Kapitel 6

Fazit und Ausblick

100
Anhang A

Robotersteuerungssoftware jEduLib

Das Java-Paket jEduLib beinhaltet eine Robotersteuerungssoftware für den Gelenkarmroboter *EduBot 250M* der Firma *Robotica Limited*. Die Software kann ebenfalls für die Ansteuerung des baugleichen *Pioneer Arm* verwendet werden, welche die Firma *ActivMedia Robotics* als Zubehör zu den mobilen Robotern der Pioneer-Familie ausliefert. Die Software basiert auf der Programmiersprache Java und entstand durch Torsten Baumgartner im Rahmen der Diplomarbeit "Sensorfreie Orientierung von polygonalen Werkstücken mit dem EduBot 250".

A.1 Funktionen

Die folgende Liste gibt einen Überblick über die Funktionalität, welche die Steuerungssoftware anbietet:

- Eine Geschwindigkeitsregelung für die Anpassung der Achsgeschwindigkeit in 100 verschiedenen Stufen,
- die synchrone Ansteuerung einer beliebigen Untermenge der Achsen des Manipulators,
- die Abfrage der Roboterposition als Gelenkpositionwerte, Gelenkwinkel oder in kartesischen Koordinaten,
- die Ansteuerung von Zielpositionen, welche durch Gelenkpositionswerte oder durch kartesische Koordinaten gegeben sind,
- eine Warteschlange für die Entgegennahme und serielle Abarbeitung von Bewegungsanforderungen.

A.2 Installation

Das Paket jEduLib ist über den CVS-Server des Geometrie-Labors der Informatik Abteilung I, Universität Bonn erhältlich. Für eine Installation können die auf dem Server befindlichen Quelltexte kompiliert werden, alternativ existiert ein Jar-Archiv mit bereits kompilierten Java-Klassen. Der Pfad zu den kompilierten Java-Klassen bzw. zu dem Jar-Achiv ist in den Java-Klassenpfad aufzunehmen.

Das Paket jEduLib benötigt eine funktionsfähige Installation des Java Runtime Environments (JRE). Weiterhin wird für die serielle Kommunikation eine Standarderweiterung der Programmiersprache Java, das Java Communication API 2.0, benötigt. Das Java Runtime Environment ist auf den Internetseiten der Firma Sun Microsystems (http://java.sun.com/) erhältlich. Implementierungen des Java Communication API mit den entsprechenden Installationshinweisen stehen ebenfalls auf den Internetseiten der Firma Sun Microsystems (http://java.sun.com/) zum Download bereit. Alternativ kann die Implementierung für Microsoft Windows aus dem Verzeichnis der Steuerungssoftware auf dem CVS-Server des Geometrie-Labors verwendet werden.

Die Steuerungssoftware wurde ausschließlich unter Microsoft Windows entwickelt und getestet. Für die fehlerfreie Funktion unter anderen Betriebssystemen kann keine Garantie übernommen werden. Während der Entwicklung wurde die Java 2, Standard Edition, Version 5.0 unter Windows XP eingesetzt.

A.3 Schnittstelle zu der Robotersteuerung

Der EduBot wird über eine Instanz der Klasse jEduLib.EduBot250 angesteuert. Die Klasse bietet die folgenden, öffentlichen Methoden als Schnittstelle an:

• public EduBot250()

Standard-Konstruktor. Für die Verbindung zum Roboter wird der serielle Port "COM1" verwendet.

• public EduBot250(String portID)

Konstruktor. Für die Verbindung zum Roboter wird der serielle Port *portID* verwendet.

• public void powerOn()

Schaltet die Servomotoren des EduBots an.

Der Manipulator fährt automatisch und schnellstmöglich in die Home-Position. Daher sollte sich der EduBot vor dem Anschalten in einer Position nahe der Home-Position befinden.

A.3. SCHNITTSTELLE ZU DER ROBOTERSTEUERUNG

- public void powerOff() Schaltet die Servomotoren des Roboters aus.
- public boolean robotHasPower() Liefert *true*, falls der EduBot angeschaltet ist, ansonsten *false*.
- public void setRobotSpeed(int newSpeed) Setzt die Geschwindigkeit des Roboters auf den Wert *newSpeed*. Für *newSpeed* sind Werte von 1 bis 100 möglich.
- public int getRobotSpeed() Liefert die aktuelle Geschwindigkeit des Manipulators.
- public double[] getGripperPosition() Liefert ein Array mit der aktuellen Position des Greifers in kartesischen Koordinaten.
- public int[] getAxisValues() Liefert ein Array mit den sechs aktuellen Gelenkpositionswerten des Manipulators.
- public int[] getAxisAngles() Liefert ein Array mit den sechs aktuellen Gelenkwinkeln des Manipulators.
- public void moveHome() Fährt den EduBot in die Home-Position.
- public boolean moveAxis(int[] axis, int[] newValues, boolean moveLinear)

Bewegt eine bis maximal sechs Achsen synchron in eine Zielposition, welche durch Gelenkpositionswerte gegeben ist.

Das Array *axis* gibt an, welche Achsen zu bewegen sind. Im Array *newValues* stehen die gewünschten Endwerte für die genannten Achsen. Der boolsche Wert *moveLinear* gibt an, ob die Gelenkpositionswerte während der Bewegung im Raum der Gelenkpositionswerte linear oder nicht linear zu interpolieren sind.

• public void movePTP(double[] newPosition, boolean move-Linear)

Bewegt den Greifer in eine Zielposition, welche durch kartesische Koordinaten gegeben ist.

Das Array *newPosition* enthält die kartesischen Koordinaten der Zielposition. Der boolsche Wert *moveLinear* gibt an, ob die Gelenkpositionswerte während der Bewegung im Raum der Gelenkpositionswerte linear oder nicht linear zu interpolieren sind.

• public void movePTP(double[] newPosition, int[] constraints, boolean moveLinear)

Bewegt den Greifer in eine Zielposition, welche durch kartesische Koordinaten gegeben ist.

Das Array *newPosition* enthält die kartesischen Koordinaten der Zielposition. Der boolsche Wert *moveLinear* gibt an, ob die Gelenkpositionswerte während der Bewegung im Raum der Gelenkpositionswerte linear oder nicht linear zu interpolieren sind. Über das Array *contraints* können zusätzlich Randbedingungen angegeben werden. Die im Array gespeicherten Positionswerte werden von den entsprechenden Achsen in jedem Fall eingenommen, während die restlichen Achsen zu einer Annäherung an die Zielposition beitragen. Ein Arrayeintrag von -1 bedeutet, dass die jeweilige Achse unbeschränkt ist. Andere Arrayinhalte werden als Randbedingung interpretiert.

- public void openGripper(int speed) Öffnet den Greifer mit der Geschwindigkeit *speed*.
- public void closeGripper(int speed) Schließt den Greifer mit der Geschwindigkeit *speed*.

• public boolean robotIsMoving()

Liefert *true*, falls der EduBot zum Zeitpunkt des Methodenaufrufs eine Bewegungsanforderung ausführt, ansonsten *false*.

• public void haltRobot()

Unterbricht die Abarbeitung von Bewegungsanforderungen und löscht alle noch abzuarbeitenden Bewegungsanforderungen.

• public void waitForRobot()

Diese Methode gibt die Programmausführung an den Aufrufer zurück, sobald alle anstehenden Bewegungsanforderungen vollständig abgearbeitet wurden.

A.4 Grafische Bedienoberfläche

Die Funktionalität der Steuerungssoftware jEduLib kann über eine grafische Bedienoberfläche genutzt werden. Diese ist in dem Paket swinggui enthalten, welches ebenfalls auf dem CVS-Server des Geometrie-Labors erhältlich ist. Die grafische Oberfläche benötigt das Paket jEduLib (siehe Abschnitt A.2). Der folgende Aufruf über die Kommandozeile startet die Anwendung:

java swinggui.MainFrame <portID>

Der Parameter *portID* gibt den Namen des für die serielle Kommunikation zu verwendenden Ports an. Nach dem Start erscheint das Hauptfenster (siehe Abbildung A.1). Dieses gliedert sich in drei Teile:

- **Buttonleiste** Im oberen Teil des Fensters sind drei Buttons zu sehen. Mit diesen kann der Roboter an- und ausgeschaltet werden, in die Home-Position gefahren werden, außerdem kann ein Dialog für die Änderung der aktuellen Geschwindigkeit geöffnet werden.
- **Positionsanzeige** Unter der Buttonleiste ist die Positionsanzeige des Manipulators. Es existieren drei Reiter für die Darstellung der aktuellen Position des Manipulators als Gelenkwerte, als Gelenkwinkel, oder als kartesische Koordinaten des Greifers.
- Bedienelemente für Bewegungen Im unteren Teil des Hauptfensters sind die Bedienelemente für die Bewegungssteuerung zu sehen. Hier existiert ein Reiter für die direkte Ansteuerung einzelner Achsen des Manipulators. Weiterhin kann ein Dialogfenster für die Eingabe von gewünschten Gelenkpositionswerten geöffnet werden (siehe auch Abbildung A.2). Der zweite Reiter ermöglicht die Ansteuerung des Edu-Bots über kartesische Zielkoordinaten für den Greifer.

ver On/Off	Home	Speed		
Robot Positio	n			
Joint value	Cart	esian	Joint angle	
43.234 cm	2.1	290 cm] 16.004 cm	
43.234 cm	2.1	290 cm] [16.004 cm	
43.234 cm Move Robot Joint value	Cart	290 cm] [16.004 cm	
43.234 cm Move Robot Joint value	2.1	290 cm esian] [16.004 cm	
43.234 cm Move Robot Joint value 0 Move	Cart	290 cm esian 0] [16.004 cm	

Abbildung A.1: Das Hauptfenster der grafischen Oberfläche



Abbildung A.2: Dialog für die Eingabe von Gelenkpositionswerten

Anhang B

Java-Anwendung zur Berechnung von Aktionsplänen

Diese grafische Anwendung veranschaulicht einen Algorithmus zur Orientierung polygonaler Werkstücke ohne Sensoren von Kenneth Y. Goldberg. Die Anwendung entstand durch Torsten Baumgartner im Rahmen eines Programmierpraktikums an der Universtität Bonn. Während der Diplomarbeit "Sensorfreie Orientierung von polygonalen Werkstücken mit dem EduBot 250" wurde die Anwendung um eine Anbindung an den Gelenkarmroboter EduBot 250M und um die Möglichkeit der Berechnung von robusteren Aktionsplänen erweitert.

Die Anwendung ermöglicht die Spezifizierung von Polygonen durch den Benutzer. Anschließend kann für das vorgegebene Polygon ein universeller Aktionsplan berechnet werden, welcher dieses Polygon unabhängig von seiner initialen Orientierung in eine bestimmte Endorientierung überführt.

Die Berechnung des Aktionsplans beruht auf mehreren Funktionen. Deren Funktionsgraphen können in einem separaten Fenster angezeigt werden. Die Anwendung bietet die Möglichkeit der grafischen Simulation der Ausführung eines Aktionsplans. Zusätzlich kann simultan ein an den Rechner angesteuerter EduBot für die Ausführung des Aktionsplans angesteuert werden.

B.1 Installation

Die Klassen der Anwendung sind in dem Paket appsSwingGui.JawGripper zusammengefasst. Das Paket ist über den CVS-Server des Geometrie-Labors der Informatik Abteilung I, Universität Bonn erhältlich. Für eine Installation müssen die auf dem Server befindlichen Quelltexte kompiliert werden. Das Paket appsSwingGui.JawGripper benötigt eine Installation des Java Runtime Environments (JRE). Weiterhin wird eine Installation der Robotersteuerungssoftware jEduLib benötigt. Hinweise zur Installation der Robotersteuerungssoftware sind in dem Abschnitt A.2 gegeben. Das Java Runtime Environment ist auf den Internetseiten der Firma Sun Microsystems (http://java.sun.com/) erhältlich.

B.2 Bedienung

Der Aufruf der Java-Anwendung über die Kommandozeile geschieht folgendermaßen:

```
java appsSwingGui.JawGripper.MainFrame [<Port> <Speed>]
```

Die beiden Parameter *Port* und *Speed* sind optional. Werden keine Parameter angegeben, so entfällt die Ansteuerung des EduBots durch die Anwendung. Sind zwei Parameter angegeben, so werden diese als der zu verwendende serielle Kommunikationsport und die gewünschte Translationsgeschwindigkeit für den EduBot interpretiert.

Die Anwendung startet mit dem Hauptfenster (siehe Abbildung B.1 links). Dieses enthält in der linken Hälfte einen Polygoneditor. Der Editor erlaubt dem Benutzer die Spezifizierung von Polygonen mit der Maus. Über Kontextmenüs sind zudem weitergehende Funktionen erreichbar. Als nützlich erweist sich beispielsweise die Angabe von Punktkoordinaten in dem Eingabedialog, welcher in Abbildung B.2 dargestellt ist.



Abbildung B.1: Das Hauptfenster der Anwendung

In der oberen Hälfte enthält das Hauptfenster eine Buttonleiste, welche Zoomfunktionen zur Verfügung stellt. Textfelder im rechten Teil des Hauptfenster dienen der Angabe von Parametern des verwendeten Algorithmus zur

e.g. 1.5/2 0.1,-1.2 -4;6.3 :
3
polygon

Abbildung B.2: Dialog für die Eingabe von Punktkoordinaten

Berechnung von Aktionsplänen. Dies umfasst die Spezifizierung eines beliebigen, aber zwingend im spezifizierten Polygon liegenden Massenschwerpunkts ebenso wie zwei Parameter, welche Fehlertoleranzen darstellen. Mittels dieser Fehlertoleranzen können für einige Werkstücke Aktionspläne berechnet werden, welche eine hohe Robustheit in der Durchführung unter Unsicherheit aufweisen. Durch die Angabe einer geeigneten Rotationstoleranz können unter Umständen Situationen verhindert werden, in welchen das Werkstück während der Translation des Greifers entgegengesetzt der beabsichtigten Richtung rotiert. Eine geeignete Kantenstabilitäts-Toleranz verhindert negative Auswirkungen auf den Erfolg des Orientierungsvorgangs in Situationen, in denen eine Kante des Werkstücks während der Translation des Greifers ungeplant instabil wird.

Berechnete Pläne werden im rechten Teil des Hauptfensters angezeigt. Wird der Mauszeiger über dem Aktionsplan platziert und für einige Zeit nicht bewegt, so werden weitere Informationen betreffend der Robustheit des Aktionsplans angezeigt. Die Berechnung von Aktionsplänen beruht auf einigen Funktionen, welche für das spezifizierte Werkstück berechnet werden. Die Graphen der jeweiligen Durchmesser-, Greif-, Radius-, Schiebe- und Transferfunktion können nach der Berechnung eines Aktionsplans in einem separaten Fenster betrachtet werden (siehe Abbildung B.3).

Die Anwendung bietet dem Benutzer die Möglichkeit, die Ausführung von berechneten Bewegungsplänen in einer grafischen Darstellung am Rechner zu verfolgen. Hierfür wird in einem separaten Fenster das Werkstück sowie zwei schwarze Balken für den Greifer dargestellt (siehe Abbildung B.4). Die initiale Orientierung des Werkstücks kann vom Benutzer durch die Eingabe in ein Textfeld bestimmt werden. Über den Button "Start" bzw. "Next step" kann der Aktionsplan schrittweise ausgeführt werden, der Button "Reset" stellt den Startzustand wieder her. Ist ein EduBot 250 an den Rechner korrekt angeschlossen, ist die Robotersteuerungssoftware jEduLib korrekt installiert, und wurden der Anwendung zudem die entsprechenden Parameter übergeben, so wird während der visuellen Darstellung gleichzeitig der Gelenkarmroboter für die Ausführung des Aktionsplans angesteuert.



Abbildung B.3: Visualisierung von Funktionensgraphen



Abbildung B.4: Ausführung von berechneten Aktionsplänen

Literaturverzeichnis

- A. Akella and M. T. Mason. Parts orienting with shape uncertainty. pages 565–572, 1997.
- [2] D. Berkowitz and J. Canny. Designing parts feeders using dynamic simulation, 1996.
- [3] R. Berretty, K. Goldberg, L. Cheung, M. Overmars, G. Smith, and A. van der Stappen. Trap design for vibratory bowl feeders, 1999.
- [4] K. Böhringer, V. Bhatt, B. R. Donald, and K. Y. Goldberg. Algorithms for sensorless manipulation using a vibrating surface. *Algorithmica*, 26:389–429, 2000.
- [5] R. C. Brost. Planning robot grasping motions in the presence of uncertainty. Technical Report CMU-RI-TR-85-12, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, July 1985.
- [6] J. Campbell. V24/RS-232 Kommunikation. SYBEX-Verlag GmbH, Düsseldorf, 1984.
- [7] J. Canny and K. Goldberg. A risc approach to sensing and manipulation, 1993.
- [8] J. F. Canny and K. Y. Goldberg. Risc for industrial robotics: Recent results and open problems. In Proc. IEEE Internat. Conf. Robot. Autom., May 1994.
- [9] A. Castellet. Solving inverse kinematics problems using an interval method, 1998.
- [10] J. J. Craig. Introduction to Robotics: Mechanics and Control. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
- [11] H. Elias. Inverse kinematics. http://freespace.virgin.net/hugo.elias/ models/m_ik.htm, 2005.
- [12] M. Erdmann and M. T. Mason. An exploration of sensorless manipulation. In Proc. 3rd IEEE Internat. Conf. Robot. Autom., Apr. 1986.

- [13] M. Fowler. UML konzentriert. Addison-Wesley, 3. aufl. edition, 2004.
- [14] K. Goldberg and M. Furst. Low friction gripper. Patent, Mar. 1992.
- [15] K. Y. Goldberg. Orienting polygonal parts without sensors. Algorithmica, 10:201–225, 1993.
- [16] ISO. Iso 9283:1998. Manipulating industrial robots Performance criteria and related test methods.
- [17] T. Kamphans and R. Klein. Bewegungsplanung f
 ür Roboter. Vorlesungsskript, Universität Bonn, Institut f
 ür Informatik, 2001.
- [18] K. Lynch. Nonprehensile robotic manipulation: Controllability and planning, 1996.
- [19] D. Manocha and J. Canny. Efficient inverse kinematics for general 6r manipulators, 1994.
- [20] NN. Pioneer Arm. ActivMedia Robotics, 19 Columbia Drive, Amhert, NH 03031, 4 edition, September 2003.
- [21] NN. Aria robotic sensing and control libraries. http://www.activrobots.com/SOFTWARE/aria.html, 2005.
- [22] NN. Edubot 250m specification. http://www.robotica.co.uk/robotica/ images/pdfs/edubot250_spec_mobile.pdf, 2005.
- [23] NN. Java communications api. http://java.sun.com/products/javacomm/, 2005.
- [24] NN. Robotica ltd welcome to the world of robotics. http://www.robotica.co.uk/robotica/, 2005.
- [25] NN. Robots for researchers and developers. http://www.activrobots.com/, 2005.
- [26] M. A. Peshkin. Robotic Manipulation Strategies. Englewood Cliffs, NJ: Prentice Hall, 1990.
- [27] M. A. Peshkin and A. C. Sanderson. The motion of a pushed, sliding object, part 2: Contact friction. Technical Report CMU-RI-TR-86-07, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, April 1986.
- [28] M. A. Peshkin and A. C. Sanderson. The motion of a pushed sliding workpiece. *Internat. J. Robot. Res.*, 4(6):569–598, 1988.
- [29] J. Pfalzgraf. On geometric and topological reasoning in robotics. Annals of Mathematics and Artificial Intelligence, 19(3-4):279–318, 1997.

- [30] A. Rao and K. Goldberg. Shape from diameter: Negative results, 1993.
- [31] A. S. Rao and K. Y. Goldberg. Manipulating algebraic parts in the plane. *IEEE Trans. Robot. Autom.*, 11(4):598–602, Aug. 1995.
- [32] N. C. Singer. Utilizing dynamic and static stability to orient parts, 1985.
- [33] P. Wolfsteiner and F. Pfeiffer. Dynamics of a vibratory feeder, 1997.
- [34] K. E. Zanganeh. The contour method an interactive approach to the inverse kinematics of serial manipulators, 1997.

Erklärung

Hiermit erkläre ich, dass ich diese Arbeit selbständig durchgeführt habe, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Torsten Baumgartner Bonn, den 31. Januar 2006