

Lehrstuhl für Montagesystemtechnik und Betriebswissenschaften
der Technischen Universität München

Aufgabenorientierte Offline-Programmierung von Industrierobotern

Doris Kugelman

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der
Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing. Dr. h. c. F. Pfeiffer

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. G. Reinhart

2. Hon.-Prof. Dr.-Ing. G. Hirzinger

Die Dissertation wurde am 19.01.1999 bei der Technischen Universität München
eingereicht und durch die Fakultät für Maschinenwesen am 12.06.1999 ange-
nommen.

Forschungsberichte

iwb

Band 127

Doris Kugelman

***Aufgabenorientierte
Offline-Programmierung
von Industrierobotern***

***herausgegeben von
Prof. Dr.-Ing. G. Reinhart***

Herbert Utz Verlag

UTZ

Forschungsberichte IWB

Berichte aus dem Institut für Werkzeugmaschinen
und Betriebswissenschaften
der Technischen Universität München

herausgegeben von

Univ.-Prof. Dr.-Ing. Gunther Reinhart
Technische Universität München
Institut für Werkzeugmaschinen und Betriebswissenschaften (iwb)

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Kugelman, Doris:

Aufgabenorientierte Offline-Programmierung von Industrierobotern /

Doris Kugelman. -

München : Utz, Wiss., 1999

(Forschungsberichte IWB ; 127)

Zugl.: München, Techn. Univ., Diss., 1999

ISBN 3-89675-615-X

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, der Entnahme von Abbildungen, der Wiedergabe auf photomechanischem oder ähnlichem Wege und der Speicherung in Datenverarbeitungsanlagen bleiben, auch bei nur auszugsweiser Verwendung, vorbehalten.

Copyright © Herbert Utz Verlag GmbH 1999

ISBN 3-89675-615-X

Printed in Germany

Herbert Utz Verlag GmbH, München

Tel.: 089/27791-00 · Fax: 089/27791-01

Geleitwort der Herausgeber

Die Produktionstechnik ist für die Weiterentwicklung unserer Industriegesellschaft von zentraler Bedeutung. Denn die Leistungsfähigkeit eines Industriebetriebes hängt entscheidend von den eingesetzten Produktionsmitteln, den angewandten Produktionsverfahren und der eingeführten Produktionsorganisation ab. Erst das optimale Zusammenspiel von Mensch, Organisation und Technik erlaubt es, alle Potentiale für den Unternehmenserfolg auszuschöpfen.

Um in dem Spannungsfeld Komplexität, Kosten, Zeit und Qualität bestehen zu können, müssen Produktionsstrukturen ständig neu überdacht und weiterentwickelt werden. Dabei ist es notwendig, die Komplexität von Produkten, Produktionsabläufen und -systemen einerseits zu verringern und andererseits besser zu beherrschen.

Ziel der Forschungsarbeiten des *iwb* ist die ständige Verbesserung von Produktentwicklungs- und Planungssystemen, von Herstellverfahren und Produktionsanlagen. Betriebsorganisation, Produktions- und Arbeitsstrukturen sowie Systeme zur Auftragsabwicklung werden unter besonderer Berücksichtigung mitarbeiterorientierter Anforderungen entwickelt. Die dabei notwendige Steigerung des Automatisierungsgrades darf jedoch nicht zu einer Verfestigung arbeitsteiliger Strukturen führen. Fragen der optimalen Einbindung des Menschen in den Produktentstehungsprozeß spielen deshalb eine sehr wichtige Rolle.

Die im Rahmen dieser Buchreihe erscheinenden Bände stammen thematisch aus den Forschungsbereichen des *iwb*. Diese reichen von der Produktentwicklung über die Planung von Produktionssystemen hin zu den Bereichen Fertigung und Montage. Steuerung und Betrieb von Produktionssystemen, Qualitätssicherung, Verfügbarkeit und Autonomie sind Querschnittsthemen hierfür. In den *iwb*-Forschungsberichten werden neue Ergebnisse und Erkenntnisse aus der praxisnahen Forschung des *iwb* veröffentlicht. Diese Buchreihe soll dazu beitragen, den Wissenstransfer zwischen dem Hochschulbereich und dem Anwender in der Praxis zu verbessern.

Gunther Reinhart

Vorwort

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftliche Mitarbeiterin am Institut für Werkzeugmaschinen und Betriebswissenschaften (*iwb*) der Technischen Universität München.

Herrn Prof. Dr.-Ing. Gunther Reinhart, dem Leiter dieses Instituts, gilt mein besonderer, herzlicher Dank für die wohlwollende Förderung und großzügige Unterstützung meiner Arbeit. Bei Herrn Prof. Dr.-Ing. Dr. h.c. Joachim Milberg, dem ehemaligen Institutsleiter, bedanke ich mich sehr herzlich für die gebotene Möglichkeit zur Promotion am *iwb* und die anfängliche Betreuung der Arbeit.

Bei Herrn Prof. Dr.-Ing. Gerd Hirzinger, dem Leiter des Instituts für Robotik und Systemdynamik des Deutschen Zentrums für Luft- und Raumfahrt (DLR) Oberpfaffenhofen, möchte ich mich für die Übernahme des Korreferates und die aufmerksame Durchsicht der Arbeit sehr herzlich bedanken.

Darüberhinaus bedanke ich mich bei allen Mitarbeiterinnen und Mitarbeitern des Instituts sowie allen Studenten, die mich bei der Erstellung meiner Arbeit unterstützt haben, recht herzlich.

Schließlich bedanke ich mich bei meinem Mann Werner für das entgegengebrachte Verständnis, die Unterstützung und die kritische Durchsicht dieser Arbeit und bei meinen Eltern, die mir die erforderliche Ausbildung ermöglichten.

München, im Juli 1999

Doris Kugelmann

Inhaltsverzeichnis

1 Einführung	1
1.1 Einleitung	1
1.2 Zielsetzung dieser Arbeit	3
1.3 Vorgehensweise	3
2 Stand der Technik	5
2.1 Systeme zur Offline-Programmierung von Industrierobotern	6
2.1.1 3D-Simulationssysteme	6
2.1.2 Systeme zur impliziten Programmierung	8
2.2 Abweichungen zwischen simulierter und real erforderlicher Roboterbewegung	17
2.2.1 Abweichungen innerhalb des realen Systems	18
2.2.2 Abweichungen zwischen realem System und Simulationsmodell	19
2.3 Auslegung, Berücksichtigung und Kompensation von Toleranzen und Ungenauigkeiten	26
2.3.1 Rechnergestützte Produkttolerierung und Fügeanalyse	26
2.3.2 Online-Kompensation von Ungenauigkeiten in der Roboterbewegung und im Aufbau der Montagezelle	29
2.3.3 Berücksichtigung von Ungenauigkeiten in Offline- Programmiersystemen	29
2.4 Zusammenfassung	31
3 Konzeption des Gesamtsystems zur aufgabenorientierten, modellgenauigkeitgerechten Offline-Programmierung	33
3.1 Zielsetzung	33
3.2 Systemaufbau	34
4 Modell der Roboterzelle	36
4.1 Simulationsmodell	36

4.1.1 Geometrische Beschreibung	37
4.1.2 Kinematikmodell	38
4.1.3 Steuerungsmodell	38
4.1.4 Kollisionsprüfung	40
4.2 Parametrisierbares Geometriemodell	41
4.2.1 Ausgangssituation	41
4.2.2 Zerlegung einer Geometrie	43
4.2.3 Abschätzung von Ungenauigkeiten	44
5 Handhabungsplanung	53
5.1 Greifplanung	56
5.1.1 Greifertypen	57
5.1.2 Bestimmung geeigneter Greifkonfigurationen	58
5.1.3 Roboterspezifische Bewertung der Greifkonfigurationen	63
5.2 Planung kollisionsfreier Feinbewegungen	65
5.2.1 Anpassung des Objektkoordinatensystems	68
5.2.2 Bestimmung der Objektbahn	69
5.2.3 Transformation in eine Roboterbewegung und Anpassung	80
5.3 Bahnplanung für weiträumige Transferbewegungen	82
5.3.1 Bahnsuche	82
5.3.2 Modell der PTP-Bewegung für die Bahnplanung und Kollisionskompensation	85
5.3.3 Kollisionsüberprüfung mit Hüllkörpern für den verwendeten Bahnsuchalgorithmus	94
5.4 Übergreifende Handhabungsplanung	102
5.5 Überprüfung und Umgestaltung vorhandener Roboterprogramme	108
6 Automatischer Layoutabgleich	109
6.1 Problemstellung beim Layoutabgleich	109
6.2 Konzeption eines Systems zum Layoutabgleich	113

6.2.1 Definition der Variationsaufgabe.....	119
6.2.2 Automatische Variation zur Ermittlung einer gültigen Layoutvariante für das eingefahrene Roboterprogramm	122
6.2.3 Konzeption des Gesamtsystems zum Layoutabgleich	124
7 Anwendungsbeispiele.....	127
7.1 Greifplanung mit tolerierten oder unterschiedlichen Greifobjekten.....	127
7.2 Planung von Feinbewegungen	129
7.3 Störungsbehebung durch einen autonomen, mobilen Roboter.....	130
7.4 Maschinenentladung.....	134
7.5 Kommissionierung einer Montagepalette.....	136
8 Zusammenfassung und Ausblick	139
Literaturverzeichnis	144

1 Einführung

1.1 Einleitung

Im Zeitalter des Internet mit seinen virtuellen Märkten vollzieht sich ein Wandel von lokalen Märkten, die von einzelnen Unternehmen bedient werden, zu einem globalen Markt, der von allen erfolgreichen Konkurrenten bedient wird. In diesem Wettbewerb bestehen nur Unternehmen, die sowohl den Vorteil eines Produktivitätsfortschrittes als auch den Vorteil innovativer, den Kundenwünschen gerechter Produkte gegenüber ihren Konkurrenten für sich verbuchen können. Innovationen stellen die Bodenschätze der deutschen Unternehmen für den globalen Markt dar (REINHART 1997).

Innovative Produkte können auf Dauer aber nur durch eine Verkürzung der Entwicklungszeiten und Produktzyklen zum richtigen Zeitpunkt, vor dem Konkurrenten angeboten werden. Denn Innovationswettbewerb ist auch Zeitwettbewerb (MILBERG 1997). Ein großer Schritt in diese Richtung erfolgte mit dem flächendeckenden Einsatz von CAD-Systemen. Die Hersteller erhalten die Produkte ihrer Zulieferer im Entwicklungsstadium in Form von CAD-Daten. Der Datenaustausch wird durch die Verwendung elektronischer Medien schneller, und es bietet sich v. a. die Möglichkeit zu Untersuchungen am CAD-Modell des Gesamtproduktes, die früher erst an einem physikalischen Prototypen durchgeführt werden konnten. Die Unternehmen haben es sich zum Ziel gesetzt, die physikalischen Prototypen oder Physical Mock-Ups weitestgehend durch Digital Mock-Ups in Form von Rechnermodellen des Produktes zu ersetzen, da Digital Mock-Ups wandlungsfähiger sind und die Detektion von Problemen am Digital Mock-Up zu einem früheren Zeitpunkt und damit kostengünstiger erfolgt (BARTHOLD 1998, REINHART & VON PRAUN 1998). Ein Digital Mock-Up umfaßt auch die Simulation verschiedener Prozesse wie z. B. das Produktverhalten unter äußeren Kräften mit Finite-Elemente-Simulation oder die Planung von Montagevorgängen mit 3D-Simulation, wofür 3D-CAD-Daten benötigt werden. Mit der Durchsetzung von 3D-CAD verbreitete sich auch die Anwendung der 3D-Simulation (REINHART & FELDMANN 1997). Die meisten CAD-Systeme bieten heutzutage die Möglichkeit

der Simulation von Programmen für Produktionsmaschinen mit numerischer Steuerung zur Herstellung des konstruierten Produktes. Bereits seit längerem gibt es einige eigenständige NC-Programmiersysteme, die ebenfalls 3D-CAD-Daten verwenden (WILLERT 1996). Beim Punktschweißen im Karosserierohbau werden die Programme offline im Simulationssystem erstellt und anschließend in der Zelle eingefahren. Das Potential der Verkürzung der Stillstandszeiten des Roboters durch die 3D-Simulation wird jedoch bisher bei weitem nicht ausgeschöpft. Ursachen dafür sind die mangelnde Akzeptanz der Simulationsergebnisse durch die Zellenbetreuer, der vergleichsweise hohe Aufwand zur Erstellung eines Modells der Zelle und die mangelnde Unterstützung des Planers bei der Offline-Programmierung. Simulationssysteme arbeiten meist unter der Annahme idealer Voraussetzungen (HEINEMEIER 1993, S. 11) und berücksichtigen keine Ungenauigkeiten oder Toleranzen. Dies führt im ungünstigsten Fall zur Unbrauchbarkeit der mit dem Simulationssystem offline erstellten Roboterprogramme und damit zu einer ablehnenden Haltung der Roboterbetreuer gegenüber der Offline-Programmierung. Zudem wird das an die Gegebenheiten der realen Zelle angepaßte Programm nur selten in die Planung bzw. das Simulationsmodell zurückgeführt und auch ein Abgleich des Modells mit dem realen Zellenaufbau erfolgt nur selten. Das Simulationsmodell wird somit nach der einmaligen Nutzung zur ersten Offline-Programmierung und zur Planung des Zellenaufbaus nie mehr genutzt. Das Aufwand-Nutzen-Verhältnis für die Simulation und Modellerstellung ist für diesen Fall sehr ungünstig. Außerdem findet eine Unterstützung des Planers z. B. durch Automatismen und Hilfsfunktionen in den meisten Systemen gar nicht oder nur unzureichend statt. Meist wird in der Simulation immer noch eine Teach-In-Programmierung wie am realen Roboter durchgeführt, so daß die Programmierung zwar die Stillstandszeiten des Roboters reduziert, da sie offline erfolgt, die Möglichkeiten zur Vereinfachung des Programmiervorgangs durch die Simulation aber nicht genutzt oder vom System nicht geboten werden.

1.2 Zielsetzung dieser Arbeit

Zielsetzung dieser Arbeit ist es aufzuzeigen, wie die Effektivität des Simulationseinsatzes bei der Offline-Programmierung gesteigert werden kann. Dazu wird ein Konzept zur benutzerfreundlichen Offline-Programmierung durch unterstützende Automatismen vorgestellt, das ein einfaches Einfahren der offline generierten Programme durch die Berücksichtigung von Ungenauigkeiten und Toleranzen in der Simulation gewährleistet. Für die Weiterverwendung der Simulationsmodelle für spätere Umlanungen oder Umstellungen in der Zelle wird zudem ein benutzerfreundliches System zur Anpassung des Simulationslayout an das Layout der realen Zelle mittels der eingefahrenen Programme konzipiert.

Betrachtet wird dabei insbesondere die Anwendung zur Offline-Programmierung von Industrierobotern in der Handhabung und Montage, dem größten Einsatzgebiet für Industrieroboter neben dem Schweißen (SCHWEIZER 1996).

1.3 Vorgehensweise

Zur Konzeption eines benutzerfreundlichen, effektiven Systems zur Offline-Programmierung wird zunächst ein Überblick über den Stand der Technik und Forschung bei der Roboter-Offline-Programmierung, die Abweichungen zwischen Modell und Realität sowie bestehende Ansätze zur Berücksichtigung von Abweichungen und Ungenauigkeiten in Kapitel 2 gegeben.

Kapitel 3 stellt ein Gesamtkonzept für ein System zur modellgenauigkeitsgerechten, benutzerfreundlichen Offline-Programmierung vor.

Das System baut auf einem Bewegungssimulationsmodell der Roboterzelle einschließlich eines parametrisierten Geometriemodells auf, das in Kapitel 4 betrachtet wird.

Kapitel 5 behandelt die automatische, benutzerfreundliche Offline-Generierung von Roboterprogrammen mit Hilfe von Modulen zur Greifplanung, Planung von Feinbewegungen und Planung weiträumiger Transferbewegungen.

Eine Vorgehensweise zum einfachen Abgleich des Simulationsmodells mit dem eingefahrenen Roboterprogramm wird in Kapitel 6 konzipiert.

Die Erprobung des Systemkonzepts mit einem Systemprototypen anhand einiger Anwendungsbeispiele beschreibt Kapitel 7.

In Kapitel 8 werden die Ergebnisse dieser Arbeit zusammengefaßt und ein Ausblick auf die mögliche Weiterentwicklung von Systemen zur Offline-Programmierung gegeben.

2 Stand der Technik

Bei der Programmierung von Industrierobotern wird zwischen dem Teach-In- oder Playback-Verfahren, der textuellen Programmierung und der aufgabenorientierten Programmierung unterschieden.

- Beim Teach-In wird der Roboter in die gewünschte Stellung bewegt und diese im Programm abgespeichert. Erfolgt die Programmierung am Roboter selbst, wird die Stellung im allgemeinen mit dem Programmierhandgerät angefahren. In einzelnen Fällen, insbesondere bei Robotern kleinerer Baugröße, können auch die Achsbremsen gelöst werden und die gewünschte Stellung wird durch Ziehen und Schieben an den Bestandteilen der Kinematik eingenommen. Für die Programmierung ohne Nutzung des realen Roboters, die sog. Offline-Programmierung, existieren computergestützte Systeme, die eine Programmierung nach dem Teach-In-Verfahren gestatten. Die Roboterstellung wird dabei mit Hilfe eines grafischen Modells am Bildschirm visualisiert.
- Bei der textuellen Programmierung wird der Programmcode durch Editieren erzeugt. Dies ist sowohl innerhalb der realen Anlage mit dem Editor der Robotersteuerung als auch offline mit einem beliebigen ASCII-Text-Editor möglich. Diese Art der Programmierung kann jedoch nur von erfahrenem Fachpersonal durchgeführt werden, da z. B. auch die Zahlenwerte für die einzelnen Roboterstellungen eingegeben werden müssen und so eine genaue Kenntnis der Roboterkinematik und der Beschreibung von Stellungen erforderlich ist.
- Die aufgabenorientierte Programmierung dagegen ermöglicht auch weniger erfahrenen Personen die Erzeugung von Roboterprogrammen. Sie erfolgt in der Regel offline in einem computergestützten System, dem die Aufgabe in einer Hochsprache vorgelegt wird. Soll der Roboter z. B. ein Teil an einen bestimmten Zielort bringen, so muß nur noch das Teil und der Zielort vorgegeben werden. Die Zerlegung der Aufgabe in Einzelanweisungen, wie Greifen des Teils, Bewegen des Teils zum Zielort, Absetzen des Teils und die weitere Detaillierung dieser Einzelanweisungen in Form von Basisbefehlen der Robotersteuerung erfolgt automatisch. Der Programmierer muß also nur die Aufgabe vorgeben und nicht explizit die einzelnen Schritte für den Roboter defi-

nieren. Da die Vorgabe der Aufgabe nicht explizit in Einzelanweisungen erfolgt, wird diese Art der Programmierung auch implizite Programmierung genannt.

Zur Offline-Programmierung werden heute i. a. computergestützte Systeme auf Basis eines sog. 3D-Bewegungssimulationssystems eingesetzt, die eine Programmierung mit dem Teach-In-Verfahren unterstützen und die Kontrolle editierter Programme gestatten. Funktionen zur aufgabenorientierten Programmierung sind in kommerziellen Systemen nur in sehr geringem Umfang enthalten, im Rahmen von Forschungsarbeiten wurden bereits einige aufgabenorientierte Programmiersysteme entwickelt. Programme, die mit den im folgenden vorgestellten Systemen zur Offline-Programmierung erzeugt wurden, müssen jedoch entweder in der realen Zelle angepaßt werden, oder das Modell der Produktionszelle muß vor der Programmierung mit der realen Zelle abgeglichen werden. Deshalb werden nach der Vorstellung der Programmiersysteme im folgenden Abschnitt die Abweichungen zwischen Modell und realer Zelle behandelt sowie der Stand der Technik und der Forschung bei der Kompensation dieser Abweichungen dargestellt.

2.1 Systeme zur Offline-Programmierung von Industrierobotern

2.1.1 3D-Simulationssysteme

In den 80er Jahren wurden eine Reihe von 3D-Simulationssystemen entwickelt, deren Haupteinsatzgebiet die Offline-Programmierung von Industrierobotern und der Entwurf automatisierter Zellen war (FOITZIK 1990, HUCK 1990, WLOKA 1991, WRBA 1990). Die 3D-Bewegungssimulation (kurz 3D-Simulation) basiert auf einem Modell von Roboter und Zelle. Die Grundbausteine dieses Modells sind ein Geometrie-, ein Kinematik- und ein Steuerungsmodell. Die ersten 3D-Bewegungssimulationssysteme waren deshalb zumeist in ein 3D-CAD-System

integriert. Spezielle Applikationen wie z. B. für das Laserschneiden werden auch heute z. T. noch direkt an ein CAD-System gekoppelt.

Während die Systeme anfangs nur auf leistungsfähigen Grafikworkstations eingesetzt wurden, entstanden in den letzten Jahren immer mehr PC-geeignete 3D-Simulationssysteme und auch seit längerem erhältliche Systeme bieten zumeist eine PC-Version an. Auf dem Markt erhältliche Systeme sind z. B. AnySIM (LINNEN 1997, LINNEN & WUNSCH 1998), Cosimir (FREUND & ROBMANN 1997, FREUND ET AL. 1993), GRASP (SORENTI 1996), IGRIP und Ultra von DENEBOBOTICS, INC. (1997), ROBCAD von TECNOMATIX TECHNOLOGIES LTD (1998), SILMA (JAMES & GUPTILL 1993) und Workspace (OWENS 1996). Für die Offline-Programmierung verfügen die Systeme meist über eine Nachbildung des Sprachumfangs bestimmter Robotersteuerungen, so daß ein Roboterprogramm in der Roboterzielsprache erzeugt werden kann. Ein Roboterprogramm in der Roboterzielsprache wird auch als RC-Programm bezeichnet. RC steht für Robot Control (Robotersteuerung). Zusätzlich oder alternativ bieten die meisten Systeme die Möglichkeit der Programmierung in einer systemeigenen Hochsprache mit einem entsprechenden Postprozessor zur Programmierung (DILLMANN 1998). Auch die genormte Robotersprache IRL (BORGOLTE 1993, HEB 1994) ist weit verbreitet (z. B. FREUND ET AL. 1993, FREUND ET AL. 1994).

Zur Simulation der Bewegungen eines Roboterprogramms wird entweder der Bewegungsinterpolator des Simulationsmodells verwendet oder die Software der Originalrobotersteuerung eingebunden. Die Software der Originalsteuerung wird z. B. bei roboterspezifischen Simulationssystemen wie KRSIM (FISCHER & JACOB 1998, KUKA ROBOTER GMBH 1997, PRASSLER ET AL. 1998, S. 198) und Rapid (PRASSLER ET AL. 1998, S. 178) oder der Verwendung von sog. RCS-Modulen (ANGERBAUER ET AL. 1996, BERNHARDT ET AL. 1994) eingesetzt. Bei der Programmierung selbst kann der Benutzer wie am realen Roboter durch Teach-In Programmpunkte erzeugen.

Inzwischen wird diese Art der Programmierung auch in immersiven Virtual-Reality-Umgebungen, bei denen der Benutzer in die virtuelle Welt eintauchen kann, angeboten (FREUND & ROBMANN 1997, DREWS & WEYRICH 1997, NEUGEBAUER ET

AL. 1994). Dies ist aber nur in Ausnahmefällen sinnvoller als der Einsatz der 3D-Simulation. Denn immersive Virtual Reality stellt heute oft noch zu hohe Anforderungen an die Hardware (LINDEMANN ET AL 1997) und der zusätzliche Nutzen durch immersive Virtual Reality bei der Roboter-Offline-Programmierung wurde bisher noch nicht geprüft (REINHART & ROBGODERER 1998).

2.1.2 Systeme zur impliziten Programmierung

Zur Vereinfachung der Offline-Programmierung wurden in den vergangenen Jahren einige Systeme zur impliziten Programmierung entwickelt, die zumeist in ein 3D-Simulationssystem integriert oder mit diesem verknüpft wurden. Neben Systemen zur Programmierung von Robotern in der Laserbearbeitung wie z. B. den Systemen von BAUER (1998), GEIGER ET AL. (1998), HOLLENBERG (1995) oder PEPPER (1998) wurden vor allem Systeme zur impliziten Programmierung von Montage- und Handhabungsvorgängen entwickelt (HECK 1995, HÖRMANN & WERLING 1989, LOZANO-PÉREZ ET AL. 1989, NNAJI 1993, REINISCH 1992, SATO ET AL. 1995, SCHELL 1997, STETTER 1993, SUN 1998, VORMANN 1995, WEEKS 1997). Dabei muß vom Benutzer nur das zu greifende Objekt und dessen Ziellage spezifiziert werden. Aus diesen Angaben wird dann automatisch ein kollisionsfreies Roboterprogramm erzeugt.

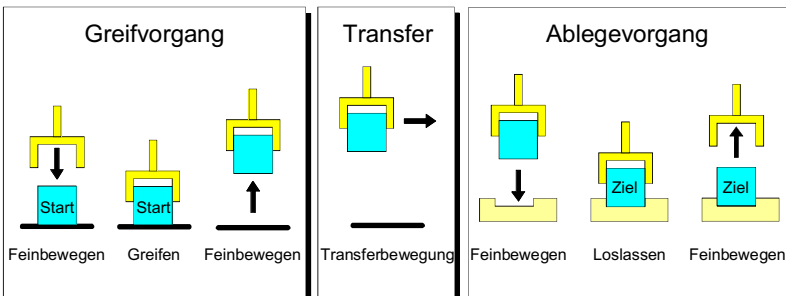


Bild 2-1: Aufbau eines Handhabungsvorgangs nach HÖRMANN & HÖRMANN (1990)

Die implizite Programmierung von Handhabungsvorgängen basiert auf der Zerlegung eines Handhabungsvorgangs in drei Teilvorgänge: den Greif-, den Transfer- und den Ablegevorgang (Bild 2-1). Der Transfervorgang wird mit einer weiträumigen Roboterbewegung durchgeführt. Zur Planung des Transfervorgangs kommen automatische Bahnplanungsverfahren zum Einsatz, die ausgehend von der Anfangs- und der Endposition des Transfervorgangs eine kollisionsfreie Roboterbewegung erzeugen. Für den Greifvorgang werden automatische Greifplanungsverfahren eingesetzt. Zu der ermittelten Greifkonfiguration, der Lage des Objekts relativ zum Greifer, wird dann eine kollisionsfreie Anrückbewegung und eine kollisionsfreie Feinbewegung zum Verlassen des Greifbereichs ermittelt. Mit Hilfe der Greifkonfiguration und der Ziellage des zu transportierenden Objekts wird anschließend analog der Ablegevorgang geplant, der im wesentlichen aus einer Feinbewegung in die Ablegeposition und einer weiteren Feinbewegung zum Verlassen der Ablegeposition besteht.

Bei einer Feinbewegung werden die Roboterhand, der Greifer und das zu transportierende Teil in der Regel nahe an anderen Objekten vorbei bewegt. Sie unterliegen somit im Vergleich zu den anderen Komponenten des Roboters einer erhöhten Kollisionsgefahr. Bei einer Transferbewegung können größere Abstände zu möglichen Hindernissen eingehalten werden als bei einer Feinbewegung, die z. B. einen Fügevorgang enthält. Bewegungsbahnen für Feinbewegungen werden deshalb meist mit anderen Strategien geplant als Bahnen für Transferbewegungen.

2.1.2.1 Greifplanungsverfahren

Mögliche Greifkonfigurationen müssen geometrische und physikalische Randbedingungen einhalten. Die meisten Verfahren zur Greifplanung suchen deshalb zunächst nach geometrischen Elementen an dem zu greifenden Objekt, die für das Greifprinzip des verwendeten Greifers geeignet sind. Mit einem Zweifingerparallelgreifer z. B. kann ein Objekt an zwei zueinander parallelen Flächen gegriffen werden. Auf der Bestimmung geeigneter Geometrieelemente basieren u. a. die Greifplanungsverfahren von HÖRMANN & HÖRMANN (1990), NNAJI

(1993), RAO & GOLDBERG (1993), REINISCH (1992), RÖHRDANZ (1998), SCHELL (1997), STETTER (1993) und WEEKS (1997). Die meisten der genannten Arbeiten setzen einen Zweifingerparallelgreifer voraus. Geeignete Geometrieelemente werden entweder mit einem eigenen Algorithmus automatisch bestimmt, oder es wird auf eine featurebasierte Geometriebeschreibung in einem CAD-System zurückgegriffen, bei der geeignete Geometrieelemente bereits als sog. Features definiert sind.

Für die ermittelten geeigneten Geometrieelemente werden dann bestimmte Griffe oder Greifkonfigurationen und deren sog. Greifsicherheit bestimmt. Die Greifsicherheit wird beeinflusst durch das Gewicht des zu greifenden Objekts, die maximal vom Greifer aufgebrauchte Greifkraft, die wirkenden Reibungskräfte, die Angriffspunkte des Greifers am Werkstück sowie die Erreichbarkeit der Position durch den Roboter (HÖRMANN & HÖRMANN 1990, RÖHRDANZ 1998, SCHELL 1997, STETTER 1993, VAN BRUGGEN ET AL. 1993, WEEKS 1997). SCHELL (1997) betrachtet zusätzlich die Beschleunigung der Handhabungsbewegung, RÖHRDANZ (1998) die Stabilität der Ablegeposition des Objekts bzw. der resultierenden Baugruppe. Die jeweilige Greifsicherheit der einzelnen Greifkonfigurationen dient als Bewertungskriterium zur Auswahl der besten Greifkonfiguration, die in der Regel automatisch erfolgt. Beim Einsatz der Greifplanung zur impliziten Programmierung von Handhabungsvorgängen wird im nächsten Schritt die sich aus der Greifkonfiguration und der Ziellage des Objekts ergebende Ablegekonfiguration betrachtet, die vor allem für den Roboter erreichbar sein muß. Kann die Ablegekonfiguration vom Roboter angefahren werden, so erfolgt die Planung der Feinbewegungen für das Greifen und Ablegen.

2.1.2.2 Verfahren zur Planung von Feinbewegungen

Für die Planung von Feinbewegungen, bei denen der Roboter sehr genau und nahe an Hindernissen vorbei bewegt werden muß, finden sich nur wenige Ansätze in der Literatur. Es wird entweder versucht, die Feinbewegung durch eine Linearbewegung (STETTER 1993) zu modellieren bzw. aus ähnlichen Basisbewegungen zusammensetzen. Oder es wird eine Bahnplanung für das abzusetzende

Objekt einschließlich Greifer durchgeführt und später in eine Roboterbewegung zurücktransformiert, die dann hinsichtlich Kollision überprüft wird wie z. B. bei HÖRMANN & WERLING (1989). Andere Ansätze wiederum befassen sich mit dem sog. Bolzen-Loch-Problem und lösen dieses durch die Ermittlung von Bewegungsfreiräumen (HUCK 1990) oder die Simulation eines Kraft-Momenten-Sensors (SUN 1998).

2.1.2.3 Bahnplanungsverfahren

Für die Bahnplanung weiträumiger Roboterbewegungen, bei denen meist ein Mindestabstand zu vorhandenen Hindernissen eingehalten werden kann, existieren zahlreiche Arbeiten (z. B. ADOLPHS & NAFZIGER 1990, BAGINSKI 1996, BARRAQUAND & LATOMBE 1991, BORGOLTE 1994, GLAVINA 1991, HEINE & SCHNARE 1991, HÖRMANN 1987, KUBOTA ET AL. 1996, LI 1989, LOZANO-PÉREZ 1987, LÜTH 1993, PEPER 1998, RALLI & HIRZINGER 1994, VAN DE LOGT ET AL. 1996). Dabei wird eine kollisionsfreie Roboterbewegung zwischen der vorgegebenen Startposition und der gewünschten Zielposition des Roboters mit einem Bahnplanungsverfahren ermittelt. Bei Bahnplanungsverfahren ist zu unterscheiden zwischen globalen und lokalen Verfahren.

Globale Verfahren berücksichtigen den gesamten Roboterarbeitsraum bei der Suche nach einer kollisionsfreien Bahn. Die Suche stützt sich auf einen sog. Hinderniskonfigurationsraum, der für jede Stellung des Roboters Informationen über etwaige Kollisionen enthält. Der Hinderniskonfigurationsraum muß vor der Bahnsuche berechnet werden. Lokale Verfahren dagegen betrachten nur den Bereich des Arbeitsraums, in dem sich der Roboterendeffektor momentan befindet. Sie verfügen über bestimmte Strategien zur Generierung von Roboterbahnen, die z. B. die Entfernung zur Zielstellung verringern. Die generierten Bahnen werden während der Planung einer Kollisionsprüfung unterzogen und je nach Ergebnis dieser Prüfung verworfen oder beibehalten. Bei lokalen Verfahren entfällt der Aufwand zur Berechnung des Hinderniskonfigurationsraums, sie benötigen jedoch bei schwierigen Problemstellungen oft sehr lange Zeit zur Bestimmung einer kollisionsfreien Bahn.

Zur Berechnung eines Hinderniskonfigurationsraums wird der gesamte Arbeitsraum des Roboters mit möglichst hoher Auflösung diskretisiert. Eine Roboterposition kann bei einem Sechsauslenkarmroboter entweder durch die Achswinkelwerte der einzelnen Gelenke oder die kartesischen Koordinaten des Endeffektors einschließlich der Raumwinkel, die kartesische Position des Werkzeugkoordinatensystems des Roboters, beschrieben werden. Letztere Art der Beschreibung erfordert u. U. jedoch zusätzlich die Angabe eines weiteren Parameters zur eindeutigen Festlegung der Stellung (s. auch Kap. 4). Der Hinderniskonfigurationsraum wird entsprechend der gewählten Darstellung als Gelenkwinkelhinderniskonfigurationsraum oder als kartesischer Hinderniskonfigurationsraum bezeichnet (LÜTH 1993).

Für jeden Punkt des diskretisierten Arbeitsraums bzw. für jede entsprechende Stellung des Roboters wird ein Kollisionstest durchgeführt. Schließlich erhält man die Menge aller kollisionsfreien diskreten Roboterstellungen und die Menge aller kollisionsbehafteten diskreten Roboterstellungen. Die Menge der kollisionsbehafteten Roboterstellungen ist die Transformation der Hindernisse im Arbeitsraum des Roboters von einem Inertialkoordinatensystem in den gewählten Konfigurationsraum des Roboters. Die Hindernisse sind in Bild 2-2 in den diskretisierten Konfigurationsräumen dargestellt. Im kartesischen Hinderniskonfigurationsraum sind die Koordinaten des Tool-Center-Points (TCP) schwarz markiert, für die der Roboter mit dem Hindernis kollidiert. Das graue Hindernis wurde dieser Darstellung zur Verdeutlichung überlagert, auch der graue Bereich ist kollisionsbehaftet. Bei der Darstellung im Gelenkwinkelkonfigurationsraum sind alle kollisionsbehafteten Gelenkwinkelkonfigurationen schwarz markiert bis auf die beiden grauen Felder, die den zwei im Bild links oben dargestellten Roboterstellungen entsprechen. Die erste Achse hat dabei den Wert $q_1 \approx 50^\circ$, die zweite Achse die Werte $q_2 \approx 0^\circ$ und $q_2 \approx 70^\circ$.

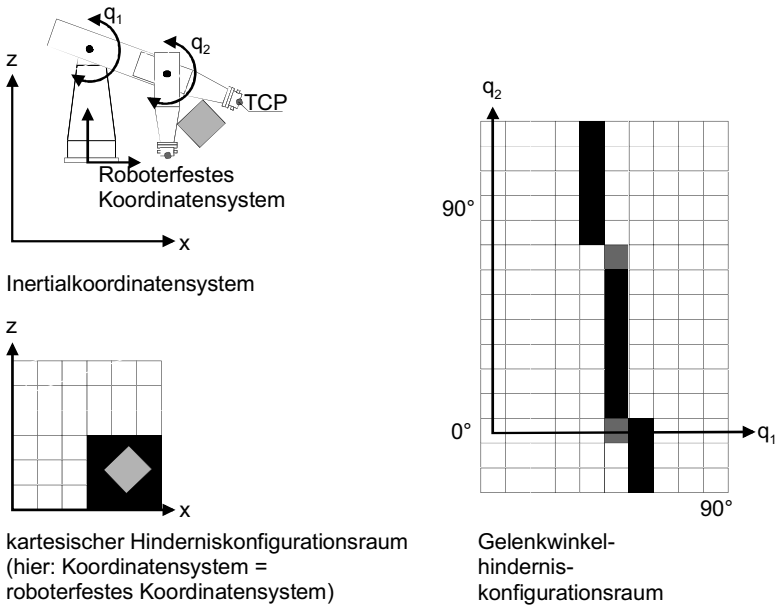


Bild 2-2: Hindernis im Inertialkoordinatensystem und beiden Konfigurationsräumen für das Beispiel eines Roboters mit zwei Drehachsen

Bei der Suche nach einer kollisionsfreien Bewegungsbahn für den Roboter werden benachbarte kollisionsfreie Punkte im Hinderniskonfigurationsraum zu einer Bahn verbunden. Die dazwischenliegenden Stellungen sind aufgrund der i. a. verwendeten, ausreichend feinen Diskretisierung ebenfalls kollisionsfrei. Die Berechnung des Hinderniskonfigurationsraums ist jedoch aufgrund der feinen Diskretisierung sehr zeitaufwendig, insbesondere wenn mehr als drei Roboterachsen betrachtet werden. Wird z. B. bei der Berechnung der Hindernisse in einem Gelenkwinkelkonfigurationsraum jede Achse durch n diskrete Stellungen dargestellt, so ist bei drei Achsen die Überprüfung von n^3 Roboterstellungen erforderlich, während bei sechs Achsen bereits n^6 Roboterstellungen hinsichtlich Kollision überprüft werden müssen. Die Anzahl der Kollisionstests wurde gegenüber dem Fall mit 3 Achsen quadriert.

Da jedoch die Betrachtung von mehr als drei Achsen bei vielen Anwendungen gefordert ist und der Hinderniskonfigurationsraum zudem bei jeder Änderung der Hinderniskonstellation angepaßt bzw. neu berechnet werden muß, wurden mehrere Bahnplanungsverfahren entwickelt, die umgekehrt vorgehen. D. h. es werden zuerst Bahnen vorgeschlagen und dann überprüft, ob diese Bahnen kollisionsfrei sind, bis eine kollisionsfreie Gesamtbahn gefunden wird. Diese Verfahren werden als lokale Verfahren bezeichnet, da sie zumeist eine lokale Strategie zur Hindernisumgehung verfolgen und nicht global im gesamten Arbeitsraum nach einer Bahn suchen. Einen Überblick über lokale Verfahren geben BARRAQUAND & LATOMBE (1991). Ein weiteres lokales Verfahren ist das sog. Gleiten von GLAVINA (1991). Lokale Verfahren benötigen zwar meist weniger Rechenzeit, bergen jedoch das Risiko einer "Sackgasse" in sich. Deshalb werden immer mehr gemischte Ansätze, die sich bei der Suche nicht auf die direkte lokale Umgebung beschränken, eingesetzt (z. B. GLAVINA 1991, KUBOTA ET AL. 1996, KUGELMANN 1994B, STETTER 1993). Die meisten dieser Verfahren basieren auf dem Ansatz von GLAVINA (1991), der zufällig sog. Zwischenziele im Arbeitsraum erzeugt und versucht, die Zwischenziele zu Bahnstücken zu verbinden, bis sich aus den Bahnstücken eine Gesamtlösung von der Start- zur Zielposition ergibt. Bei der Verbindung der Zwischenziele wird zunächst die Bahn hinsichtlich Kollision überprüft, die in dem von den einzelnen Achsen des Roboters aufgespannten Raum einer Gerade entspricht. Ist diese, im sog. Gelenkwinkelraum geradlinige Verbindung kollisionsbehaftet, so wird mit einem lokalen Verfahren (dem oben erwähnten Gleiten) nach einer kollisionsfreien Verbindung gesucht.

Die meisten Bahnplanungsverfahren wurden in ein 3D-Bewegungssimulationssystem integriert, da dort verschiedene Hinderniskonfigurationen und Bahnplanungsaufgaben einfach modelliert und mit der angebotenen Kollisionsrechnung getestet werden können. Deshalb bieten einige wenige käufliche Simulationssysteme auch ein Bahnplanungsverfahren im Systemumfang sowie eine Programmierschnittstelle zur Implementierung weiterer Verfahren an.

2.1.2.4 Überblick über realisierte Systeme zur impliziten Programmierung

Implizite Verfahren zur Programmierung von Handhabungsvorgängen werden von auf dem Markt erhältlichen 3D-Simulationssystemen fast gar nicht angeboten, obwohl einige Forschungsarbeiten zu diesem Thema existieren wie z. B. die im folgenden vorgestellte Auswahl.

LOZANO-PÉREZ ET AL. (1989) stellten ein System zur aufgabenorientierten Planung von Pick-and-Place-Vorgängen vor. Dabei wurde nicht zwischen Feinbewegung und weiträumiger Transferbewegung unterschieden und die Greifplanung war auf Zweifingerparallelgreifer und Objekte mit parallelen Greifflächen beschränkt. Jedoch enthielt dieses aufgabenorientierte System bereits die Strategie des Umgreifens. D. h. wenn keine gemeinsame Greifkonfiguration für Start- und Ziellage des Objekts gefunden wurde, wurde das Objekt an einer Zwischenposition abgesetzt und erneut gegriffen.

An der Universität Karlsruhe wurde ebenfalls ein System für Roboter mit Zweifingerparallelgreifern entwickelt (HÖRMANN 1987, HÖRMANN & HÖRMANN 1990, HÖRMANN & WERLING 1989, HÖRMANN & WERLING 1990). Bei diesem System wurden die Feinbewegungen separat in Abhängigkeit von der Geometrie des Greifers und des zu transportierenden Objektes geplant, ein evtl. erforderliches Umgreifen wurde jedoch nicht berücksichtigt.

REINISCH (1992) kombiniert ein Greifplanungsverfahren, das u. a. auf der Definition von sog. Greiffeatures beruht, mit einem Bahnplanungsverfahren, das an das Verfahren von ADOLPHS & NAFZIGER (1990) angelehnt ist. Zusätzlich zu einem System zur impliziten Programmierung benutzt diese Arbeit auch Sensorik und steuerungstechnische Möglichkeiten zur Einhaltung bzw. Anpassung der geplanten Bewegung.

STETTER (1993) stellt ein Verfahren zur impliziten Roboterprogrammierung vor, das für die Online-Programmerstellung für einen autonomen mobilen Roboter konzipiert ist. Die Greifplanung wurde für Zweifingerparallelgreifer implementiert. Feinbewegungen im Greif- und Ablegebereich werden als Linearbewegun-

gen modelliert. Für die Planung weiträumiger Roboterbewegungen kommt ein Verfahren, das auf den Arbeiten von GLAVINA (1991) basiert, zum Einsatz. Alle Algorithmen wurden in das am iw b entwickelte 3D-Bewegungssimulationssystem USIS (Universal Simulation System, WOENCKHAUS ET AL. 1994) integriert. Für den späteren Einsatz des Systems für den autonomen mobilen Roboter MOBROB des iw b wurden zusätzlich Sensormodelle in USIS implementiert, z. B. das Modell einer CCD-Kamera (STETTER 1993). Da sowohl USIS als auch AnySIM (Linner 1997) auf demselben Systemkern aufbauen, USIS jedoch zusätzliche prototypische Module enthält, wird im folgenden der Begriff AnySIM/USIS verwendet.

SCHELL (1997) stellt kein System zur impliziten Programmgenerierung im eigentlichen Sinne vor, nutzt jedoch die Methoden der impliziten Programmgenerierung zur Überprüfung der Greifbarkeit und Greifsicherheit bei Handhabungsvorgängen. Er betrachtet insbesondere die Werkstückhandhabung in Fertigungszellen und plant deshalb nicht kollisionsfreie Bewegungsbahnen für den verwendeten Roboter sondern überprüft lediglich einzelne Positionen z. B. am Ende einer Fügebewegung auf Kollision. Das System soll zur frühzeitigen Erkennung von Fehlern im Zellaufbau, wie z. B. ungeeigneten oder falsch konstruierten Betriebsmitteln, eingesetzt werden. Als Basis für die Applikation dient das 3D-Simulationssystem ROBCAD.

RÖHRDANZ (1998) bearbeitet die Aufgabenstellung der impliziten Programmierung nicht vollständig, da er die Generierung kollisionsfreier Bewegungen nicht behandelt. Sein System ist jedoch eines der wenigen, das die Stabilität des Objekts an der Ablegeposition berücksichtigt und über Strategien zum Umgreifen eines Objekts verfügt.

WEEKS (1997) stellt ein System zur impliziten Programmierung eines SCARA-Roboters vor mit einem Konfigurationsraumverfahren zur Bahnplanung. Er verwendet bei scheinbar unlösbaren Greifplanungsaufgaben auch andere, in der Zelle zur Verfügung stehende Zweifingergreifer.

2.2 Abweichungen zwischen simulierter und real erforderlicher Roboterbewegung

Zwei der Haupteinsatzgebiete der 3D-Simulation sind die Planung von Zellenlayouts und die Offline-Programmierung von Industrierobotern. 3D-Simulation wird zur Offline-Programmierung jedoch nicht flächendeckend eingesetzt, da die Zeit zum Einfahren der offline erstellten Programme zum Teil sehr hoch ist. Dies ist der Fall, wenn nicht nur geringfügige Positionskorrekturen für die Roboterbewegungen erforderlich sind, sondern ganze Programmteile umstrukturiert werden müssen, da erhebliche Abweichungen zwischen der simulierten Roboterbewegung und der in der realen Zelle zur Erfüllung der jeweiligen Aufgabe erforderlichen Roboterbewegung auftreten.

Abweichungsursachen werden im folgenden vorgestellt, wobei unterschieden wird zwischen Effekten, die bereits bei der Programmierung am realen System zu Abweichungen führen und Effekten, die bei der Offline-Programmierung noch hinzukommen. Abweichungen sind bei der Programmierung unvermeidlich.

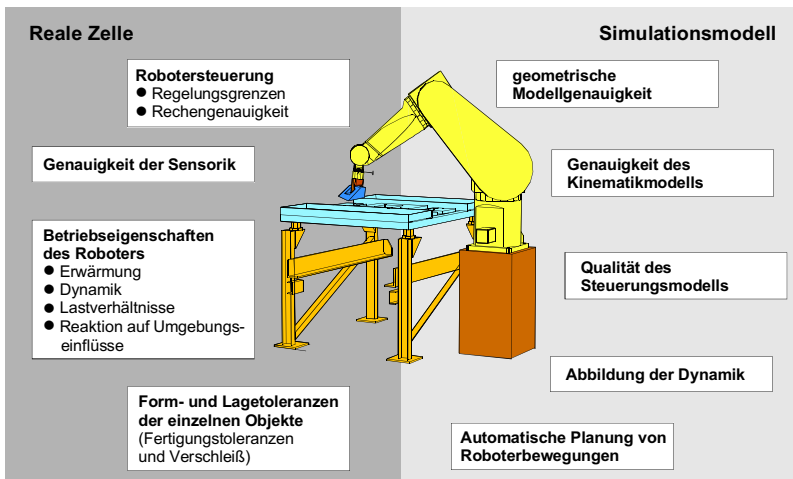


Bild 2-3: Faktoren, die zu Abweichungen im Prozeß führen

Es existieren aber einige Methoden und Verfahren, um Abweichungen aufzudecken und zu kompensieren, die in Kapitel 2.3.2 und Kapitel 2.3.3 behandelt werden. Ursachen für Abweichungen zwischen der ausgeführten, programmierten Roboterbewegung und der für den Prozeß erforderlichen Bewegung sind in Bild 2-3 dargestellt.

2.2.1 Abweichungen innerhalb des realen Systems

Auch wenn ein Roboterprogramm am realen Roboter erstellt wurde, kann es später zu Abweichungen zwischen der Soll- und der Ist-Bewegung des Roboters kommen. Eine Ursache dafür liegt im Aufbau der Zelle. Alle Komponenten unterliegen Fertigungstoleranzen und nutzen sich nach einer gewissen Zeit ab. Die Abnutzung oder der Austausch von Komponenten beeinflusst die absolute Lage des zu montierenden Produkts, die sich aus den Relativlagen seiner Trägerelemente zusammensetzt. Außerdem unterliegen die einzelnen Komponenten des Produkts selbst auch Fertigungstoleranzen. Ändert sich der Aufbau der Zelle wie beschrieben, so werden andere absolute Roboterarbeitspunkte zur Durchführung der Handhabungsaufgabe benötigt, und es muß nachgeteicht werden.

Andererseits kann der Roboter absolute Positionen in seinem Arbeitsraum nur mit einer bestimmten Genauigkeit anfahren. Diese Genauigkeit wird beeinflusst durch die Robotersteuerung, die u. a. Regelungsgrenzen und Rechenungenauigkeiten unterliegt, durch die Genauigkeit der Sensorik z. B. zur Einstellung und Messung der Achswinkelwerte und durch die Betriebseigenschaften des Roboters. Im Betrieb benötigt der Roboter eine gewisse Zeit zur Erwärmung, während dieser Zeit unterliegen die tatsächlich angefahrenen Positionen einer bestimmten Drift. Eine Drift kann aber auch durch Umgebungseinflüsse verursacht werden wie starke Temperaturschwankungen in der Umgebung. Außerdem hängt der Bahnverlauf einer Bewegung von dynamischen Einflußgrößen wie z. B. der zulässigen Maximalgeschwindigkeit ab. Mechanische Fehler wie Spiel, Hysterese und Reibung führen ebenfalls zu Abweichungen. Und auch die Lastverhältnisse können zu einer Verlagerung der angefahrenen Roboterpositionen führen wegen der statischen Nachgiebigkeit des Roboters.

2.2 Abweichungen zwischen simulierter und real erforderlicher Roboterbewegung

Kenngrößen für die Genauigkeit von Industrierobotern sind in der EN 29 283 (1993) enthalten, die die internationale Norm ISO 9283 in der deutschen Übersetzung einschließt.

Die beschränkte Bewegungsgenauigkeit des Roboters und die geometrische Veränderlichkeit der Zelle z. B. bei einer groben Tolerierung des zu montierenden Produkts führen zu Unsicherheiten bei der Programmierung am realen Roboter. Eine weitere Ursache für Bahnabweichungen besteht darin, daß die Programmierung mit einer gegenüber dem Normalbetrieb verringerten Geschwindigkeit durchgeführt wird. Auf Programmtests kann also weder bei Offline- noch bei Online-Programmierung verzichtet werden.

2.2.2 Abweichungen zwischen realem System und Simulationsmodell

Bei der Offline-Programmierung kommt jedoch hinzu, daß ein Modell der Roboterzelle verwendet wird. Die Qualität des Modells und der Planungsfunktionen entscheidet letztendlich über die Einsetzbarkeit der auf dem Modell basierenden Roboterprogramme.

Wie in Bild 2-3 dargestellt, läßt sich die Modellqualität aus der geometrischen Modellgenauigkeit, der Qualität des Kinematik- und des Steuerungsmodells sowie der Berücksichtigung dynamischer Aspekte im Modell ableiten.

2.2.2.1 Geometrisches Modell

Abweichungen des geometrischen Modells der Roboterzelle werden durch unterschiedliche Aufstellung der Objekte und unterschiedliche Objektgeometrien in der realen Zelle und im Simulationsmodell verursacht. Die in einer Roboterzelle vorhandenen Objekte unterliegen Fertigungstoleranzen. Die Toleranzinformation ist zwar im CAD-System vorhanden, wird aber im 3D-Bewegungssimulationssystem in der Regel vernachlässigt. Bei der Bewegungssimulation wird mit den Nennmaßen der Bauteile gearbeitet. Die Schwankungen in der Bauteilgröße werden so z. B. bei Kollisionsuntersuchungen nicht beachtet.

Die Summierung von Toleranzen in einer Baugruppe kann zudem zu Abweichungen in der Positionierung eines Objekts führen, auch wenn die Objektgeometrie selbst in Simulation und Realität genau übereinstimmt. Dies gilt ebenso für magazinierte, mit der Realität übereinstimmende Objekte, falls die Geometrie oder der Aufstellort des Magazins sich von den realen Gegebenheiten unterscheiden. Abweichungen in der Platzierung treten außerdem vor allem dann auf, wenn die Objektfixierung nachträglich angebracht wird, z. B. zwei Platten beim Aufbau der Roboterzelle miteinander verbohrt werden. Insbesondere das Verankern von Zellenkomponenten in der Aufstellfläche bringt enorme Abweichungen von der geplanten Aufstellposition mit sich. Je größer die Abweichungen der Objektgeometrien und der Aufstellpositionen der Objekte sind, desto größer sind auch die Abweichungen der realen Position des Roboterwerkzeugs von seiner realen Sollposition. Die Roboterbewegung muß also erheblich angepaßt werden. U. U. müssen dazu ganze Bewegungsabschnitte geändert werden, wenn z. B. der reale Arbeitspunkt nur durch eine Veränderung der Achskonfiguration des Roboters erreicht werden kann, weil sich eine oder mehrere Achsen an den Grenzen des Verfahrbereichs befinden.

2.2.2.2 Kinematikmodell und Dynamikabbildung

Die Genauigkeit der Abmessungen hat insbesondere auch beim kinematischen Modell des Roboters erheblichen Einfluß auf die Unterschiede zwischen realen und simulierten Roboterstellungen. Dynamische Einflüsse z. B. durch die vom Roboter bewegte Last oder das Überschwingen beim Anfahren einer Position werden bei üblichen Simulationsmodellen nicht berücksichtigt.

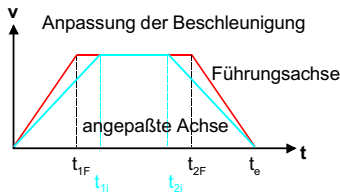
2.2.2.3 Steuerungsmodell

Die gängigen Grundbausteine der Bewegungssimulation sind ein geometrisches und ein kinematisches Modell des Roboters sowie ein Modell seiner Steuerung. Das verwendete Modell der Robotersteuerung ist deshalb ein weiterer wesentlicher Faktor für Bewegungsabweichungen.

2.2 Abweichungen zwischen simulierter und real erforderlicher Roboterbewegung

Die meisten Simulationssysteme verwenden als Steuerungsmodell eine selbst entworfene Nachbildung, die auf der Bahn basiert, die in der Robotersteuerung der Regelung vorgegeben wird. Bei der RCM3-Steuerung beispielsweise sind dies bei einer Point-To-Point-Bewegung (PTP-Bewegung), mit der der Roboter möglichst schnell sein Ziel erreichen soll, Rampenprofile für die Geschwindigkeiten der einzelnen Achsen. Dabei wird die Achse zuerst konstant beschleunigt bis auf eine vorgegebene Geschwindigkeit, verfährt dann mit konstanter Geschwindigkeit und wird schließlich abgebremst (Bild 2-4).

Die Achsen bewegen sich nach dem vorgegebenen Geschwindigkeitsprofil mit der eingestellten Maximalgeschwindigkeit und Maximalbeschleunigung vom Anfangs- zum Endwert. Beim Synchron-Betrieb werden Geschwindigkeit und / oder Beschleunigung zusätzlich so angepaßt, daß alle Achsen gleichzeitig mit der Bewegung beginnen und diese dann auch gleichzeitig wieder beenden. Die Geschwindigkeiten und / oder Beschleunigungen der einzelnen Achsen i werden dazu an die von der langsamsten Achse F benötigte Bewegungszeit angepaßt. Die langsamste Achse bestimmt also den Bewegungsverlauf und wird deshalb auch Führungssachse genannt. In Bild 2-4 sind drei Möglichkeiten zur Anpassung des Geschwindigkeitsverlauf einer Achse an die Vorgaben der Führungssachse dargestellt. Eine Herabsetzung der Maximalgeschwindigkeit wie im Bild links unten dargestellt wird z. B. bei der RCM3-Steuerung durchgeführt. Das dargestellte Geschwindigkeitsprofil ist jedoch nur ein von der Steuerung angestrebter Sollwert, da verschiedene Randbedingungen wie maximal verfügbare Motormomente oder dynamische Schwingungen die genaue Einhaltung des Geschwindigkeitsprofils nicht gestatten. Diese Ungenauigkeiten wirken sich ebenso wie die Genauigkeiten des Roboters nach EN 29 283 beim Einfahren von offline in einem Robotersimulationssystem erstellten Programmen aus.



Mögliche Geschwindigkeitsprofile für Robotersteuerungen (3 Beispiele für Rampenprofile, Synchron-PTP)

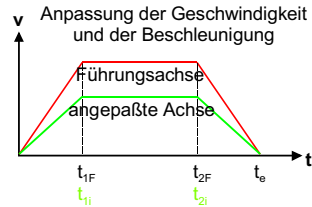
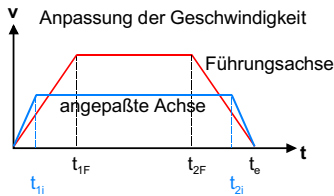


Bild 2-4: Beispiele für Rampenprofile der Geschwindigkeit beim Synchron-PTP-Betrieb

Zudem ist eine exakte Nachbildung der Sollrampenprofile oder anderer dynamisch günstigerer Sollbahnen, wie sie in neueren Steuerungen üblich sind, im Steuerungsmodell meist nicht möglich, da die genaue Funktionsweise der Steuerung nicht veröffentlicht wurde, z. B. bei der RCM3-Steuerung die für ein bestimmtes Rampenprofil wirklich verwendeten Beschleunigungen und Geschwindigkeiten nicht bekannt sind, oder Effekte wie z. B. der Einsatz mechanischer Bremsen zusätzlich zu den Motormomenten der einzelnen Achsen nicht berücksichtigt werden. Ein besseres Steuerungsmodell liefern sog. RCS-Module. RCS steht für Robot Control Simulation. In dem Projekt "RRS - Realistische Roboter-Simulation" wurde für einzelne Robotersteuerungen ein Steuerungsmodell in Form von Software als Aufsatz für auf dem Markt erhältliche 3D-Bewegungssimulationssysteme entwickelt (BERNHARDT ET AL. 1994). An diesem Projekt waren die Steuerungshersteller ABB, Comau, Fanuc, KUKA, Renault Automation, Siemens und VW, die 3D-Simulationssystemhersteller Dassault Systèmes, Deneb, Silma und Tecnomatix sowie als Initiatoren die Automobilhersteller Audi, BMW, Ford, Mercedes, OPEL, P.S.A. und Volvo beteiligt. Das Projektmanagement wurde vom Fraunhofer-Institut für Produktionsanlagen und Konstruk-

2.2 Abweichungen zwischen simulierter und real erforderlicher Roboterbewegung

tionstechnik (IPK-Berlin) übernommen. Mit den RCS-Modulen kann die kinematische Bahnplanung der Steuerung simuliert werden, da sie als Kernstück die Transformationen und die Bewegungsinterpolation der Original-Steuerungssoftware enthalten (BERNHARDT ET AL. 1994). Während hier sozusagen die kinematische Bahnplanung aus der Steuerung isoliert und als Softwaremodul zur Verfügung gestellt wurde, besteht ein anderer Lösungsansatz darin, die reale Robotersteuerung an das Simulationssystem zu koppeln (KUKA ROBOTER GMBH 1997, PRASSLER ET AL. 1998, S. 178). Diese Ansätze bilden zwar die vorgegebenen Sollbahnen sehr genau ab, durch die Regelung der Steuerung verursachte Bahnabweichungen sind jedoch nur schwer oder gar nicht zu berücksichtigen.

2.2.2.4 Automatische Planung von Roboterbewegungen

Bisher noch wenig verbreitet ist der industrielle Einsatz automatischer Funktionen zur Erzeugung kollisionsfreier Roboterbewegungen, sog. Bahnplanungsalgorithmen. Die von den Simulationssystemherstellern angebotenen Bahnplaner eignen sich oft nur für sehr einfache Probleme, bei denen die Bahnplanung gegenüber der manuellen Programmierung kaum einen Vorteil bietet. Die Möglichkeit zur Integration eigener Bahnplanungsmodule über die Programmierschnittstelle des Simulationssystem wird hauptsächlich in der Forschung genutzt. Außerdem fehlt hier oft die Umsetzung einer mathematisch idealen Bahn in eine mit realen Standardsteuerungen ausführbare Roboterbewegung.

Viele Bahnplanungsverfahren setzen die kollisionsfreien Roboterbahnen aus Teilbewegungen zusammen, die sich für eine mathematische Beschreibung und Lösung des Problems eignen. Dies hat jedoch zur Folge, daß die Robotersteuerung meist keinen Bewegungsmodus anbietet, mit dem die ermittelte kollisionsfreie Bahn abgefahren werden kann. Da bei industriellen Anwendungen der Einsatz eines speziellen Controllers mit dieser Fähigkeit in der Regel keinen Sinn ergibt, werden die ermittelten Bahnen durch verfügbare Bewegungsarten, z. B. PTP-Bewegungen angenähert. Dies führt jedoch u. U. zu erheblichen Abweichungen, so daß eine kollisionsfreie Bewegung nicht mehr gewährleistet werden kann. Die Abweichungen zwischen der simulierten Bewegung und der Bewegung

des realen Roboters zwischen explizit angegebenen Positionen werden hier bereits durch das eingesetzte Programmierverfahren verursacht. Es weichen nicht nur die Bewegungen voneinander ab, bereits die geforderten Sollbahnen, die Gerade im Gelenkwinkelraum in der Simulation und die PTP-Sollbahn für den realen Roboter, unterscheiden sich stark.

Für eine Steuerung mit rampenförmigem Geschwindigkeitsverlauf (Bild 2-4) und Anpassung der Geschwindigkeiten der Folgeachsen an die Bewegung der Führungsschse wie in (WANNER 1989) beschrieben wird diese Abweichung der Sollbahnen in Kapitel 5.3.1 näher untersucht.

Der Unterschied zwischen der Geraden im Gelenkwinkelraum und der von der Steuerung angestrebten Sollbahn kann nicht vernachlässigt werden. In vielen Fällen ist zwar die Kollisionsgefahr für den Roboterarm relativ gering, der Endeffektor ist aber sehr gefährdet. Betrachtet man als Rechenbeispiel die Sollbahn eines Roboters des Typs Manutec R15S1030 der Firma Siemens bei einer PTP-Bewegung von der in Grad angegebenen Achswinkelstellung $A = (0 \mid 20 \mid 20 \mid 0 \mid 0 \mid 0)$ in die Achswinkelstellung $B = (60 \mid 42,5 \mid 57 \mid 0 \mid 0 \mid 0)$, so ergeben sich für die Handwurzelpunktkoordinaten radiale und vertikale Abweichungen von bis zu 15 mm gegenüber der entsprechenden Geraden im Gelenkwinkelraum (s. Bild 2-5). Beide Abweichungen erreichen fast gleichzeitig ihren Maximalwert, so daß ein maximaler Gesamtversatz von bis zu 20 mm auftreten kann. Bei der Einplanung eines unzureichenden Abstandes können also auch schon bei relativ kleinen Verfahrenswegen erhebliche Abweichungen auftreten, die unter Umständen von der Bahnplanung nicht erkannte Kollisionen mit sich bringen.

Dieser Effekt kann sich noch vergrößern, wenn der Roboter im Überschleifmodus verfährt. Bei der automatischen Bahnplanung ist wie an diesem Beispiel aufgezeigt die Abweichung zwischen dem Bewegungsmodell der Bahnplanung und der von der Steuerung angestrebten Sollbahn ein erhebliches Problem. Diese Abweichungen müssen deshalb zur Gewährleistung einer kollisionsfreien Bewegung bei der Planung berücksichtigt werden.

2.2 Abweichungen zwischen simulierter und real erforderlicher Roboterbewegung

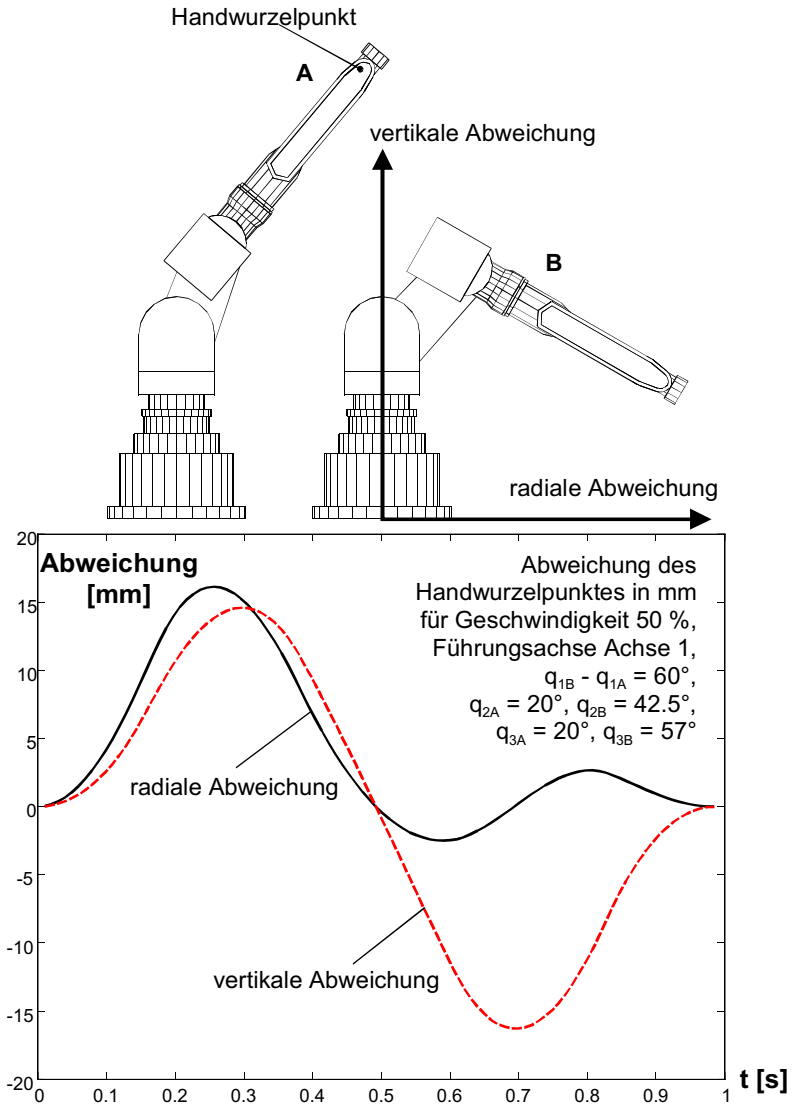


Bild 2-5: Beispiel für auftretende Abweichung der Handwurzelpunktkoordinaten

Aber auch die Ungenauigkeiten der Bewegung des realen Roboters, Ungenauigkeiten im simulierten Zellenlayout und im in der Simulation verwendeten Steuerungsmodell können wie in den vorigen Abschnitten erläutert die Weiterverwendung offline erzeugter Roboterprogramme erschweren, was u. U. sogar die erneute Programmierung online bedeuten kann.

2.3 Auslegung, Berücksichtigung und Kompensation von Toleranzen und Ungenauigkeiten

Es genügt nicht, die bei vielen Bahnplanungsalgorithmen verwendeten mathematisch idealisierten Bewegungsmodelle durch Sollbahnen zu ersetzen, die von der realen Robotersteuerung verfolgt werden können. Sondern es müssen auch Abweichungen und Unsicherheiten bei der geometrischen Modellierung von Produktbaugruppen und Roboterzellen sowie Bewegungsungenauigkeiten des Roboters bei der Programmierung berücksichtigt werden.

Mögliche Abweichungen im geometrischen Aufbau von Produktbaugruppen oder einer ganzen Roboterzelle können durch Toleranzen beschrieben werden. Deshalb wird im folgenden der Stand der Forschung bei der Tolerierung von Bauteilen in der Konstruktion und der Ermittlung der entsprechenden Auswirkungen auf Fertigung und Fügevorgänge dargelegt. Anschließend werden Möglichkeiten zur Kompensation von Bewegungsungenauigkeiten des Roboters, die unabhängig vom geometrischen Aufbau der Arbeitszelle sind, betrachtet. Dabei wird unterschieden zwischen Verfahren, die die Genauigkeit der Bewegung online während der Bewegung des realen Roboters erhöhen, und Verfahren, die die Offline-Programmierung unterstützen.

2.3.1 Rechnergestützte Produkttolerierung und Fügeanalyse

Der Rechnereinsatz in Konstruktion und Planung brachte eine Entlastung des Mitarbeiters von Routinetätigkeiten und wiederholter Neugenerierung gleicher Daten mit sich. Auf der anderen Seite entstand aber das Problem der rechnerge-

rechten Formulierung von Sachverhalten. Erfahrungswissen und Intuition sind auch heute noch Eigenschaften, die von einem Computerprogramm in absehbarer Zeit nicht in dem Maße erreicht werden können wie vom Menschen.

Erfahrung und Verständnis des Konstrukteurs bei der Auslegung von Bauteilen sind wesentliche Faktoren für eine erfolgreiche Konstruktion. Ein guter Konstrukteur hat z. B. ein Gefühl für die Tolerierung von Bauteilen entwickelt. Er kann intuitiv abschätzen, welche Toleranzen mit akzeptablen Kosten auf den zur Verfügung stehenden Maschinen erreicht werden und welche Toleranzen für eine montagegerechte Konstruktion und die Funktion des Produkts eingehalten werden müssen. Diese Erfahrung auch einem jungen Konstrukteur zur Verfügung zu stellen ist Gegenstand zahlreicher Forschungsarbeiten (z. B. LINDEMANN & IRLINGER 1996). Die meisten auf dem Markt vorhandenen CAD-Systeme verfügen inzwischen über eine parametrische und featureorientierte Bauteilmodellierung und Tolerierungsfunktionen für Bauteile. Ein Problem ist jedoch, daß die bisher in der Konstruktion übliche Tolerierung nach den entsprechenden DIN-Normen (BÖTTCHER 1986) oder ANSI Y14.5 für eine rechnergestützte Überprüfung und Optimierung nur bedingt geeignet ist (GUILFORD & TURNER 1993, JUSTER 1992, WEBER & STARK 1994). Mit dem STEP-Partialmodell 47 wurde basierend auf den bestehenden Normen eine Schnittstelle zum Austausch von Toleranzdaten definiert. Wegen der Nachteile der derzeitigen Normen bei der rechnergestützten Toleranzanalyse und -synthese wurden an mehreren Forschungsinstituten eigene Tolerierungsmodelle entwickelt (wie z. B. STEWART 1993 oder DESROCHERS & RIVIÈRE 1997), über die WEBER & STARK (1994) und JUSTER (1992) einen guten Überblick geben. Den Arbeiten zur Toleranzanalyse und -synthese ist gemein, daß sie entweder ein eigenes Modell benutzen oder sich auf Maßtoleranzen beschränken. Es existieren zahlreiche Arbeiten zur fertigungsgerechten Auslegung von Bauteiltoleranzen (z. B. KLEIN 1994, LEE & JOHNSON 1993, LEE & WEI 1998, PING ET AL. 1995, SKOWRONSKI & TURNER 1997, SRINIVASAN ET AL. 1996). Mehrere Arbeiten beschäftigen sich mit Systemen zur automatischen Tolerierung, die den Anforderungen der Montierbarkeit und Fügbarkeit von Baugruppen Rechnung tragen und die Auswirkungen von Einzeltoleranzen auf die gesamte Toleranzkette betrachten (GUPTA ET AL. 1995,

HSU ET AL. 1993, LATOMBE ET AL. 1997, LIU & HU 1995, LIU ET AL. 1996, VARGHESE ET AL. 1996).

Weiterer Forschungsbedarf besteht in der Integration der Einzelergebnisse zu einem Gesamtsystem. In diese Richtung zielt das teilautomatische System von REICHENEDER (1997) zur Tolerierung von Produkten, das die Randbedingungen der Fertigung, die Montierbarkeit und die Funktionsfähigkeit des Produktes gleichzeitig berücksichtigt.

Auf dem Gebiet der automatischen montagegerechten Tolerierung existieren auch schon erste Arbeiten zur automatischen Generierung von Montagefolgen für Baugruppen. Handelt es sich um eine manuelle Montage, so ist v. a. die Fügefähigkeit der Bauteile über den ganzen Toleranzbereich ausschlaggebend, die erforderliche Bewegung der Bauteile wird vom Menschen mit Hilfe seiner Sinne an die jeweiligen Istmaße angepaßt. Bei einer hybriden oder vollautomatisierten Montage oder Handhabung dagegen müssen der Zellenaufbau und die Bewegungsautomaten oder Roboter für die durch die Produkttoleranzen gegebenen Randbedingungen geeignet sein. Ziel ist es, mit einer Roboterbewegung den Fügevorgang für den ganzen Toleranzbereich der Baugruppe durchführen zu können. Die Toleranzgrenze für das Produkt, innerhalb der aufgrund der Nachgiebigkeit des Systems noch zuverlässig montiert oder gehandhabt werden kann, ist jedoch nur sehr schwer zu bestimmen.

Dagegen könnte ein Toleranzmodell einer Roboterarbeitszelle bei der Zellenplanung dazu genutzt werden, mögliche Lagen der zu handhabenden Bauteile relativ zum Roboter zu bestimmen und damit zu untersuchen, ob z. B. das Bauteil in allen Lagen für den Roboter erreichbar ist. Bei der Offline-Programmierung kann damit die sicherste Stellung des Roboters, die für alle Lagen geeignet ist, ermittelt werden.

2.3.2 Online-Kompensation von Ungenauigkeiten in der Roboterbewegung und im Aufbau der Montagezelle

Eine bestimmte Stellung kann vom Roboter allerdings auch nur begrenzt genau angefahren werden wie in Kapitel 2.2.1 beschrieben. Die Genauigkeitsgrenzwerte müssen für die Aufgabe des Roboters ausreichend sein, oder es müssen zusätzliche Mittel wie z. B. Sensorik, die die erforderliche Anpassung der Roboterbewegung zur Einhaltung einer bestimmten Bahn auch bei Temperaturschwankungen ermittelt, eingesetzt werden. Gerade die Temperaturdrift ist ein großes Problem. Die Roboterhersteller investieren deshalb verstärkt in die Erforschung von Möglichkeiten der Verringerung temperaturbedingter Positionsabweichungen von Industrierobotern (HEISEL ET AL. 1997, REINHART ET AL. 1998).

Größere Toleranzen bei Roboterbewegungen oder Bauteilen werden teilweise auch durch komplizierte Greifer und Werkzeuge (KUGELMANN 1993) oder spezielle Sensorik (MILBERG ET AL. 1997, ZELLER 1995) kompensiert. PFEIFFER (1996) kombiniert ein kompliziertes Greifsystem mit einem Kraftmomentensensor und einer besonderen Steuerung zum Fügen eines Bolzens in eine Bohrung. Eine zuvor mit einem dynamischen Robotermodell berechnete und bzgl. der auftretenden statischen und dynamischen Belastungen optimierte Roboterbewegung wird mit Hilfe der Sensordaten überwacht und bei Überschreitung bestimmter Toleranzwerte für die auftretenden Kräfte und Momente korrigiert.

Ungenauigkeiten in der Lage von Werkstücken werden mit geeigneter Sensorik, wie z. B. einer integrierten Bildverarbeitung (KROTH 1998, PAPANIKOLOPOULOS & SMITH 1998, PRASSLER ET AL. 1998, S. 178) erfaßt und die Roboterbewegung entsprechend angepaßt. Diese Ungenauigkeiten müssen für eine erfolgreiche Offline-Programmierung bereits im Programmiersystem berücksichtigt werden.

2.3.3 Berücksichtigung von Ungenauigkeiten in Offline-Programmiersystemen

Abweichungen der Objektgeometrien und der Aufstellpositionen der Objekte in der Roboterzelle werden mit einer sog. Zellenkalibrierung erfaßt. Im einfachsten

Fall werden dabei vom realen Roboter verschiedene markante Positionen in der Zelle angefahren und das dazu benutzte Roboterprogramm ins Simulationssystem übertragen. Dort erfolgt dann eine interaktive Korrektur der Objektpositionen (WRBA 1990, OSTERWINTER 1992). Zum Anfahren der Punkte in der realen Zelle werden entweder spezielle Meßspitzen oder Tastsensoren eingesetzt. Für einen automatischen Abgleich der simulierten Zelle mit dem realen Layout werden oft Bildverarbeitungssysteme eingesetzt wie z. B. bei SHIRINZADEH ET AL. (1994). Dabei muß zuerst die verwendete Kamera kalibriert werden. Anschließend werden einzelne Marken (MOCTEZUMA 1996) oder ganze Objekte (KUGELMANN 1994A) lokalisiert und das Simulationslayout durch eine entsprechende Objektplatzierung angepaßt.

Abweichungen des Roboters vom bei der Offline-Programmierung verwendeten Robotermodell werden mit einer sog. Roboterkalibrierung erfaßt. Dabei werden die spezifischen Parameter des Roboters bestimmt und im Modell entsprechend berücksichtigt, z. B. durch Umdefinieren des Basiskordinatensystem des Roboters im Simulationsmodell in bestimmten Bereichen des Arbeitsraums. Für die Roboterkalibrierung und die Fehlerkompensation wurden zahlreiche Verfahren entwickelt (BEHRENS & THATER 1992, DUELEN & SCHRÖER 1991, HIDALGO & BRUNN 1998, KÜHN & VISSER 1992, LIN & LU 1997, SCHRÖER 1993, SIEGLER 1988, THATER & BEHRENS 1993, VISSER ET AL. 1998). Dabei wird vor allem die Genauigkeit des Roboters bei Bewegungen, insbesondere auf definierten Bahnen erfaßt, die von der Regelung des Roboters, Spiel in den Getrieben, Steifigkeiten der einzelnen Achsen und weiteren Faktoren abhängt. Getriebeispiel und Steifigkeiten können nur statistisch erfaßt werden wie z. B. bei TÖNSHOFF ET AL. 1993. Die entsprechenden Meßwerte für Abweichungen werden dann bei der Offline-Programmierung berücksichtigt (z. B. WEISEL 1996). Dabei werden meist die Parameter eines Robotermodells zur Kalibrierung mit Hilfe der Meßwerte berechnet. SCHRÖER (1993) berücksichtigt z. B. auch Kalibrierungsparameter für das dynamische und elastische Verhalten des Roboters.

Für eine einfache Kalibrierung muß bei der Roboterprogrammierung ein möglichst realitätsgetreues Bewegungsmodell verwendet werden, damit bei der Kalibrierung lediglich Abweichungen gegenüber einem ausgezeichneten Bezugsro-

boter berücksichtigt werden müssen und nicht Unzulänglichkeiten des Roboter-simulationsmodells. XU & WURST (1992) z. B. vermessen und berücksichtigen einen von der Idealkinematik abweichenden Kinematikaufbau. CACCAVALE & CHIACCHIO (1994), FERRETTI ET AL. (1994) oder GAUTIER ET AL. (1994) bestimmen dynamische Parameter des Roboters. In einigen Robotermodellen zur Offline-Programmierung oder Entwicklung neuer Roboter wurde auch ein dynamisches Robotermodell implementiert (PRITSCHOW ET AL. 1993, ROOS ET AL. 1997, STETTER 1994, TAUBER 1990), in dem z. B. wie bei STETTER (1994) und TAUBER (1990) die Regelkreise der einzelnen Achsen mit Hilfe der Mehrkörpersimulation modelliert sind, und damit die Abweichungen der vom Regler bewirkten Positionierung von der Idealpositionierung zur Verfügung stehen. Derartige Ansätze sind sehr zeitaufwendig und die Parameterbestimmung für Roboterkinematik und -regelung muß nicht nur für jeden Robotertyp, sondern für jeden einzelnen Roboter durchgeführt werden. Die Bewegungen der durch analytische Methoden verfeinerten Modelle wiesen zudem immer noch starke Unterschiede zur realen Bewegung auf. Bessere Ergebnisse sind von der Verwendung eines originalsteuerungsgetreuen Regelungsmodells wie bei ROOS ET AL. (1997) zu erwarten.

2.4 Zusammenfassung

Für die Offline-Programmierung von Industrierobotern gibt es eine Reihe von Systemen. Insbesondere die 3D-Simulation ist weit verbreitet. Eine einfache Offline-Programmierung bietet die 3D-Simulation in Kombination mit impliziten Programmierverfahren. Allerdings besteht trotz Kalibrierungsverfahren und verfeinerten Robotermodellen weiterhin das Problem des Einfahrens der offline generierten Programme. Eine Kalibrierung ist oft zeitaufwendig und muß für jeden Roboter und jeden Zellenaufbau neu durchgeführt werden. Auch mit der Kalibrierung werden nicht alle Ungenauigkeiten erfaßt. Ein Nachteachen ist deshalb in den meisten Fällen unumgänglich. Bei großen Abweichungen muß u. U. sogar die Struktur des Programms geändert werden. Der Nutzen der Offline-Programmierung ist dann sehr gering. Wenig benutzerfreundlich ist zudem in den meisten

Fällen der Abgleich der simulierten mit der realen Roboterzelle. Wenn ein Programm einmal eingefahren wurde, wird deshalb meist das geänderte Programm nicht in das Simulationsmodell zurückgeführt und auch die bei der Programmkorrektur aufgedeckte erforderliche Anpassung des Simulationsmodells bleibt aus.

Die Folge davon ist, daß bei einer Umstellung einzelner Arbeitsvorgänge in einer Zelle eine Offline-Programmierung meist nicht rentabel ist, sondern gleich online geteacht werden muß. Die Möglichkeiten der Offline-Programmiersysteme, insbesondere die Unterstützung von Benutzern mit weniger Erfahrung oder die automatische Erstellung und Optimierung von Bewegungssequenzen, werden deshalb bisher nur unzureichend genutzt.

Zum gewinnbringenden Einsatz der Offline-Programmierung ist es daher erforderlich, die Programme bereits im Offline-Programmiersystem für Anpassungen auszulegen und dabei insbesondere Ungenauigkeiten zu berücksichtigen. Die vorgestellten Ansätze zur rechnergestützten Tolerierung und Fügeanalyse befassten sich zwar bereits mit Aufgabenstellungen wie z. B. der Montierbarkeitsanalyse. Bisher beschränken sich die meisten Arbeiten jedoch auf einzelne Teilaspekte, ein integriertes System für Toleranz- und Montierbarkeitsuntersuchungen steht nicht zur Verfügung. Deshalb können derartige Ansätze noch nicht für die Berücksichtigung der Ungenauigkeiten in einer Roboterzelle bei der Offline-Programmierung eingesetzt werden.

Auch die vorgestellten Ansätze zur Simulation von Bewegungsungenauigkeiten eines Roboters basieren entweder auf einem zu stark vereinfachten Modell oder sind zu aufwendig. Deshalb werden Ungenauigkeiten, die die grundsätzliche Eignung des Roboterprogramms beeinträchtigen, beim Programmmentwurf offline bisher nicht berücksichtigt. Dies ist jedoch für eine sinnvolle Vorbereitung der Programme erforderlich. Außerdem muß eine einfache Möglichkeit zur Rückführung der Änderungen beim Einfahren in das Simulationsmodell angeboten werden, damit das Modell wirklich gepflegt wird und so für die weitere Programmierung zur Verfügung steht.

3 Konzeption des Gesamtsystems zur aufgabenorientierten, modellgenauigkeitsgerechten Offline-Programmierung

3.1 Zielsetzung

Zielsetzung dieser Arbeit ist es aufzuzeigen, wie dem Benutzer ein integriertes System zur Offline-Programmierung zur Verfügung gestellt werden kann, das automatisch einfahrergerechte Programme generiert und den Benutzer bei der Rückführung eingefahrener Programme in das Modell der Zelle unterstützt.

Die automatische Programmgenerierung basiert auf den Methoden der impliziten Roboterprogrammierung, die den Benutzer von Routinetätigkeiten entlastet und zudem die Möglichkeit einer automatischen Optimierung von Programmbestandteilen bietet. Unsicherheiten wie Ungenauigkeiten bei der Roboterbewegung und im Zellenaufbau werden in dem vorgestellten System bereits bei der Planung berücksichtigt. So wird automatisch ein Roboterprogramm erstellt, das beim Einfahren einfach durch geringfügige Positionskorrekturen angepaßt werden kann ohne die Struktur des Programms verändern zu müssen.

Nach dem Einfahren des offline erstellten Roboterprogramms in der realen Zelle wird dieses zurück in die Simulation übertragen. Mit Hilfe eines Anpassungsmoduls wird das Simulationsmodell der Zelle über das eingefahrene Programm automatisch mit dem realen Zellenaufbau abgeglichen. Der Benutzer hat dabei die Möglichkeit über das Setzen bestimmter Parameter auf die Zellenumgestaltung Einfluß zu nehmen. Dies ist erforderlich, da es für den Abgleich keine eindeutige Lösung gibt.

3 Konzeption des Gesamtsystems zur aufgabenorientierten, modellgenauigkeitgerechten Offline-Programmierung

3.2 Systemaufbau

Die Basis des Systems zur Offline-Programmierung bildet ein 3D-Bewegungssimulationssystem mit einem Modell der Roboterzelle, das auf einer parametrisierbaren geometrischen Darstellung beruht, sowie Funktionen zur Bewegungssimulation und Kollisionsrechnung (Bild 3-1).

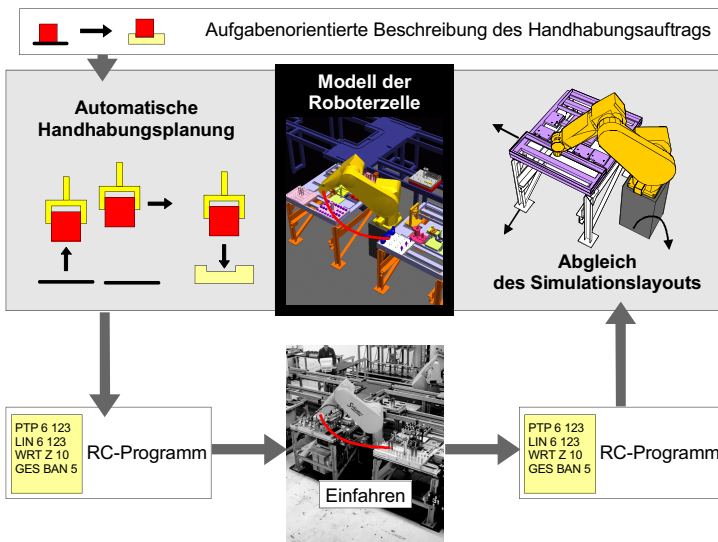


Bild 3-1: Gesamtsystem zur Offline-Programmierung

Auf das Modell der Roboterzelle einschließlich der Simulationsfunktionen greift ein Modul zur automatischen Handhabungsplanung zurück, das zu einer aufgabenorientierten Beschreibung des Handhabungsauftrags des Roboters automatisch ein kollisionsfreies Roboterprogramm erstellt.

Das RC-Programm wird an die reale Robotersteuerung übertragen, in der Zelle eingefahren und im Produktionsbetrieb verwendet. Um das Simulationsmodell der Zelle für die Planung und Vorbereitung weiterer Handhabungsaufgaben für

die reale Zelle sinnvoll einsetzen zu können und den Aufwand für das Einfahren bei der zukünftigen Erstellung von Programmen zu reduzieren, müssen die beim Einfahren des Programms erfolgten Korrekturen in das Simulationsmodell zurückgeführt werden. Dies ist in der Regel sehr aufwendig, da Positionsabweichungen zwischen dem in der Simulation erzeugten und dem eingefahrenen RC-Programm nicht eindeutig Positionsabweichungen und Toleranzen bestimmter Zellenkomponenten zugeordnet werden können. Es obliegt der Einschätzung des Planers, der den Modellabgleich durchführt, wie er diese Aufgabe löst. Der Planer erzeugt beim Abgleich meist einige Layoutvarianten, bis er eine für das eingefahrene Programm gültige Lösung ermittelt. Das integrierte System zur Offline-Programmierung soll deshalb das Simulationslayout automatisch an die beim Einfahren erfolgten Änderungen im RC-Programm anpassen.

Die automatische Anpassung des Layouts sowie die automatische Generierung leicht einzufahrender Roboterprogramme und die dazu erforderliche Integration eines parametrisierbaren Geometriemodells in das Simulationsmodell der Roboterzelle sind Schwerpunkt dieser Arbeit. Das Modell der Roboterzelle einschließlich einer Funktion zur Überprüfung der Einhaltung von Sicherheitsabständen mit vergrößerten Objektgeometrien wird im folgenden Kapitel behandelt. Die Module zur automatischen Handhabungsplanung werden in Kapitel 5 vorgestellt. Kapitel 6 beschreibt ein Konzept für den automatischen Layoutabgleich.

4 Modell der Roboterzelle

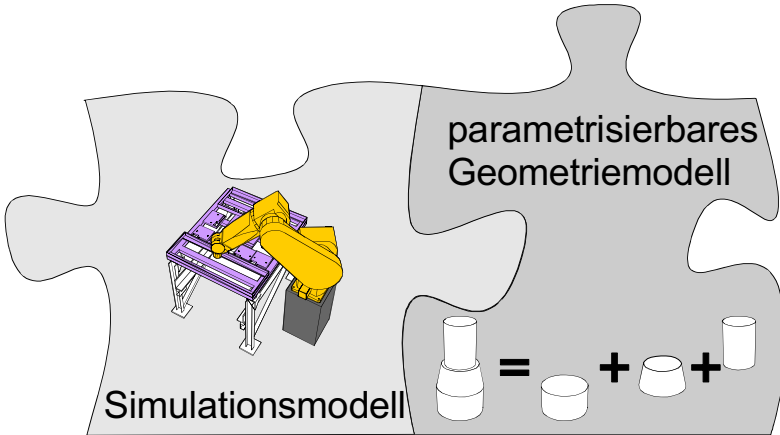


Bild 4-1: Basissystem

Basis des Systems zur Offline-Programmierung ist das 3D-Bewegungssimulationssystem AnySIM/USIS, in dem ein Simulationsmodell der Roboterzelle abgelegt ist, Funktionen zur Kollisionsprüfung zur Verfügung stehen und veränderliche Geometrien durch ein parametrisierbares Geometriemodell berücksichtigt werden.

4.1 Simulationsmodell

Das Simulationsmodell setzt sich aus den geometrischen Daten der Roboterzelle, einer kinematischen Beschreibung des Roboters und der anderen in der Zelle vorhandenen Betriebsmittel sowie je einem Steuerungsmodell zusammen.

4.1.1 Geometrische Beschreibung

Geometrische Daten des Simulationsmodells sind die CAD-Daten der einzelnen Komponenten der Roboterzelle und die Anordnung der Komponenten innerhalb der Zelle. Diese Anordnung wird als Simulationslayout bezeichnet. Die Geometriebeschreibung der Komponenten wird entweder über Schnittstellen aus einem CAD-System übernommen oder steht insbesondere bei Standardkomponenten in eigenen Simulationsbibliotheken zur Verfügung. Im Layout der Roboterzelle ist ein Inertialkoordinatensystem definiert, auf das sich die Positionen der einzelnen Objekte beziehen (Bild 4-2). Die Position eines Objekts ist durch die Koordinaten $(x \mid y \mid z \mid \alpha \mid \beta \mid \gamma)$ beschrieben. Diese beziehen sich auf die Lage des Koordinatensystems, in dem die Geometriebeschreibung des Objekts z. B. in einem CAD-System generiert wurde. Dieses Koordinatensystem wird im folgenden als Objektkoordinatensystem bezeichnet

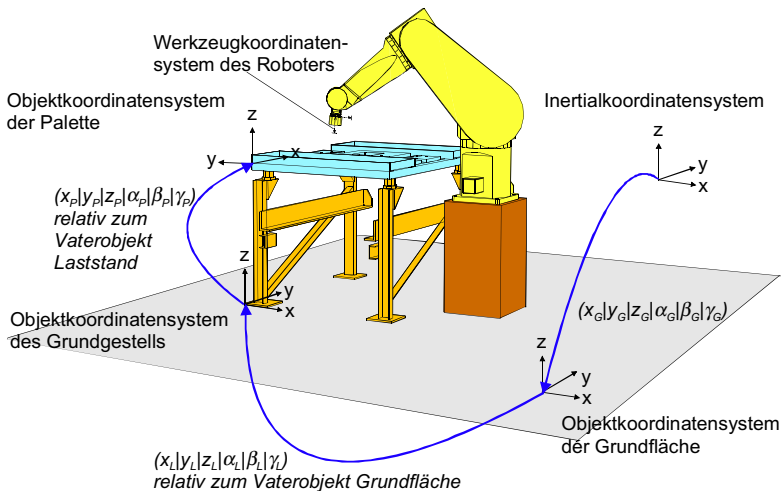


Bild 4-2: Definition der Objektposition im Simulationslayout

Damit sich bei der Platzierung von Objekten oder bei der Simulation von Bewegungen alle Objekte mitbewegen, die an einem Basisobjekt angebracht sind, können zudem Verbindungen definiert werden. Dabei wird das sog. "Sohnobjekt" mit dem "Vaterobjekt" verbunden (SCHRÜFER 1992). Die Koordinaten $(x | y | z | \alpha | \beta | \gamma)$ beziehen sich dann nicht mehr auf das Inertialsystem sondern auf das Objektkoordinatensystem des Vaterobjekts. Zur Platzierung eines Objekts wird die Objektposition als homogene Transformationsmatrix dargestellt, da bei dieser Darstellung die aktuelle Positionsmatrix nur mit der Platzierungsmatrix multipliziert werden muß, um die neue Positionsmatrix zu erhalten.

4.1.2 Kinematikmodell

Zur Beschreibung von Kinematiken für die 3D-Simulation werden Denavit-Hartenberg-Matrizen verwendet. Kinematikbeschreibungen, z. B. von Industrierobotern, sind wie Geometriedaten in Bibliotheken abgelegt. Bei der Modellierung einer neuen Kinematik können Kinematikmodellierer benutzt werden, in denen die einzelnen Komponenten einer Kinematik zueinander plaziert und über Gelenke verbunden werden. Für Kinematiken kann zusätzlich die Lage eines Werkzeugkoordinatensystems (Bild 4-2) angegeben werden. Mit Hilfe dieses Kinematikmodells können dann z. B. die Achsen eines Roboter per Hand einzeln bewegt werden. Die aktuelle Positionierung des Werkzeugkoordinatensystems kann durch Multiplikation der Denavit-Hartenberg-Matrizen der einzelnen Roboterachsen berechnet werden. Dabei handelt es sich um die sog. Vorwärtstransformation. Mit der Rückwärtstransformation dagegen werden aus der Position des Werkzeugkoordinatensystems, das auch Endeffektorkoordinatensystem genannt wird, die Winkelstellungen der einzelnen Achsen berechnet.

4.1.3 Steuerungsmodell

Die Rückwärtstransformation in der Simulation muß der der realen Robotersteuerung entsprechen. Dies betrifft auch zusätzliche Angaben, die u. U. zur Berechnung der wirklichen Achswinkel mit Hilfe der Rückwärtstransformation erforderlich sind. Bei einem Sechssachsenknickarmroboter z. B. ist zwar die Angabe

einer Roboterstellung durch die Achswinkel eindeutig, für eine eindeutige Bestimmung der Achsstellung aus der Lage des Werkzeugkoordinatensystems ist jedoch zusätzlich ein sog. Stellungsparameter erforderlich, da eine Endeffektorposition mit unterschiedlichen Achswinkelstellungen angefahren werden kann (Bild 4-3).

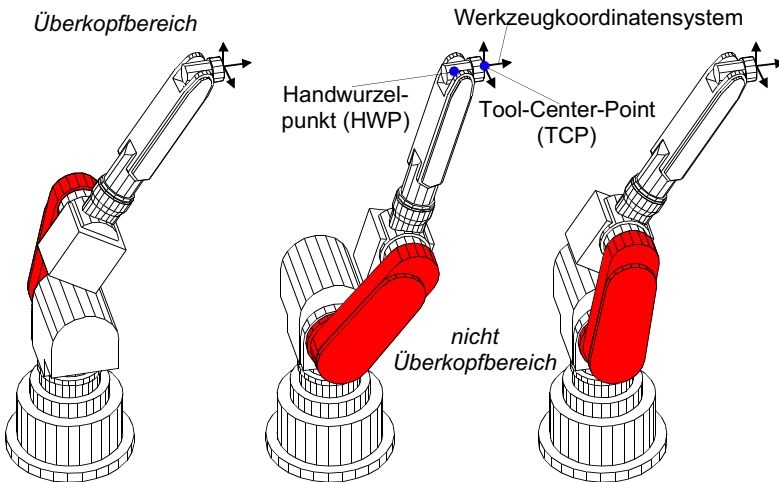


Bild 4-3: Unterschiedliche Winkelstellungen bei gleicher Endeffektorposition

Außerdem ist für manche Kinematiken die Rückwärtstransformation nur numerisch zu lösen. Für die Rückwärtstransformation von Sechsbachsenknickarmrobotern, bei denen sich die drei Handachsen in einem Punkt, dem sog. Handwurzelpunkt schneiden, existiert eine analytische Lösung. Sonst wird die Rückwärtstransformation numerisch gelöst.

Die Rückwärtstransformation ist Basis der Steuerungsnachbildung, die zudem Angaben über das Beschleunigungs- und Geschwindigkeitsverhalten der einzelnen Roboterachsen bei verschiedenen Bewegungsarten wie z. B. der Point-To-

Point-Bewegung (PTP-Bewegung) oder der Linearbewegung beinhaltet. Außerdem verfügt sie über einen Interpreter, der die einzelnen Befehle eines Roboterprogramms für die Simulation einer Bewegung übersetzt und einen Interpolator, der aus einer Anfangs- und einer Endstellung die Zwischenstellungen des Roboters während der Bewegung mit Hilfe der hinterlegten Geschwindigkeitsprofile berechnet.

Mit dem Steuerungsmodell können wie beim realen Roboter mit Teach-In-Programmierung, d. h. durch Bewegen des Roboters in eine Position und Abspeichern dieser Position im RC-Programm, RC-Programme erzeugt werden. Außerdem ist der simulierte Roboter dadurch in der Lage RC-Programme (Programme der Originalsteuerung) abzuarbeiten und die entsprechenden Bewegungen auszuführen. Mit der Simulation allein ist jedoch lediglich eine visuelle Überprüfung des RC-Programms und der korrespondierenden Roboterbewegungen hinsichtlich Kollision möglich.

4.1.4 Kollisionsprüfung

3D-Simulationssysteme verfügen deshalb über ein Kollisionsrechnungsmodul, mit dem für eine Momentaufnahme der Bewegung ein Kollisionscheck durchgeführt werden kann. In dem 3D-Simulationssystem AnySIM/USIS sind die Objektgeometrien wie auch in vielen anderen Simulations- oder Animationssystemen als Polygonflächenmodelle abgelegt. Die Kollisionsrechnung basiert auf einem Polygonflächenschnittverfahren. Sie ist hierarchisch aufgebaut, um die Anzahl der Schnittoperationen zu reduzieren und so mit wenig Rechenaufwand eine Kollisionsprüfung durchführen zu können (SCHRÜFER 1992).

4.2 Parametrisierbares Geometriemodell

4.2.1 Ausgangssituation

Die Betrachtung der Ausprägungen einer Objektgeometrie innerhalb des vorgegebenen Toleranzbereichs ist in AnySIM/USIS nicht vorgesehen, angeboten wird lediglich eine Funktion zum Kopieren einer Geometrie mit je einem bestimmten Streckungsfaktor für die drei Hauptrichtungen. Dies ist jedoch für die Betrachtung kritischer Toleranzbereiche bei der Montage nicht ausreichend, da z. B. bei der Vergrößerung eines Bauteiles auch seine Bohrungen mit vergrößert werden. So kann man dann zwar die Auswirkungen des vergrößerten Grundkörpers auf Kollisionen bei der Annäherung anderer Objekte überprüfen, das Fügen eines Bolzens in eine Bohrung ist durch die vergrößerte Bohrung jedoch wesentlich unkritischer als bei der Originalgeometrie.

Außerdem werden CAD-Daten bei der Übernahme in ein Simulationssystem meist zu facettierten Flächen konvertiert und verlieren so auch eventuell im CAD-System vorhandene Parameter- und Toleranzinformationen. Zur Berücksichtigung von Ungenauigkeiten bei der Bewegungssimulation sind Parameter- und Toleranzinformationen jedoch erforderlich. Mit Durchsetzung der STEP-Normen (insbesondere STEP 47) ist zu erwarten, daß tolerierte Bauteile direkt in einer parametrisierten Beschreibung aus dem CAD-Modell übernommen werden können, wobei allerdings noch das prinzipielle Problem der Beschreibung von Form- und Lagetoleranzen besteht (s. auch WEBER & STARK 1994). Da in den nächsten Jahren ein großer Fortschritt sowohl bei den angebotenen Systemen als auch den Datenschnittstellen zu erwarten ist, wird im Rahmen dieser Arbeit ein vorläufiger, stark vereinfachter Ansatz für ein parametrisiertes Geometriemodell verfolgt. Anhand dieses Geometriemodells wird der Nutzen eines mit Parameter- und Toleranzinformationen versehenen Geometriemodells für die 3D-Simulation aufgezeigt. Es soll jedoch nicht mit den oben genannten Forschungsansätzen konkurrieren, sondern später durch einen umfassenderen Ansatz ersetzt werden.

Relativ einfach aufgebaute Geometrielemente lassen sich meist in geometrische Primitive bzw. Formelemente zerlegen (QAMHIYAH ET AL 1996, SAKURAI &

DAVE 1996, ZHANG ET AL. 1997), wie z. B. Quader, Zylinder, Spate, gerade Prismen, Kugeln, Kegel und Kegelstümpfe, die zu einem Grundprimitiv addiert oder von diesem subtrahiert werden. Ist die Zerlegung eines Objekts in derartige Primitive bekannt, so können die für eine Geometrieänderung über den Toleranzbereich relevanten Maße bestimmt werden. Für einen geraden Kegel sind dies z. B. der Radius der Grundfläche und die Höhe. Wählt man den Bereich, in dem diese Maße variieren, verhältnismäßig groß, so sind in dem Schwankungsbereich, der durch das kleinstmögliche und das größtmögliche Primitiv bestimmt wird, auch Form- und Lagetoleranzen berücksichtigt. Bei einem Kegel wäre dies z. B. eine Abweichung des Höhenvektors von der Normale der Grundfläche oder eine Abweichung des Aufpunktes des Höhenvektors. Geometrische Schwankungen komplizierter Bauteile, die z. B. Freiformflächen enthalten, sind auf diese Art und Weise nicht ohne weiteres zu erfassen. Vereinfachend wird hier deshalb die Gesamtgeometrie so weit möglich in Primitive zerlegt, und die verbleibende Geometrie wird mit der bereits oben erwähnten Funktion symmetrisch zum geometrischen Schwerpunkt geschrumpft und vergrößert. Der Nachteil, daß Negativelemente wie z. B. Aussparungen mit Freiformflächen bei einer Geometrievergrößerung mitvergrößert bzw. bei einer Gesamtverkleinerung entsprechend verkleinert werden, ist bei der Gesamtgenauigkeit des Systems zu vernachlässigen, da in den meisten Fällen derartige komplexe Objektgeometrien nur an der Baugruppe selbst oder an wenigen produktspezifischen Betriebsmitteln auftreten, während die meisten Objekte in einer Roboterzelle, insbesondere auch die einzelnen geometrischen Bestandteile des Roboters, auf Primitive zurückgeführt werden können.

Da Toleranz- und Parameterinformationen bisher kaum in 3D-Bewegungssimulationssystemen verfügbar sind und nicht von den Original-CAD-Daten übernommen werden können, muß die Generierung derartiger Informationen im Simulationssystem ermöglicht werden. Ein Teilschritt bei der Informationsgenerierung ist die Zerlegung einer Geometrie in Primitive. Zur Beschränkung des Aufwands zur Informationsgenerierung ist deshalb eine Funktion zur automatischen Bestimmung der Zerlegung einer Geometrie in Primitive erforderlich.

4.2.2 Zerlegung einer Geometrie

Die Geometriedaten liegen als Polygonflächenmodelle vor. Kreise z. B. sind als reguläre Vielecke dargestellt. Ein Kegel wird von einem als Vieleck dargestellten Grundkreis und mehreren Dreiecken, die seinen Mantel bilden, begrenzt (Bild 4-4 links).

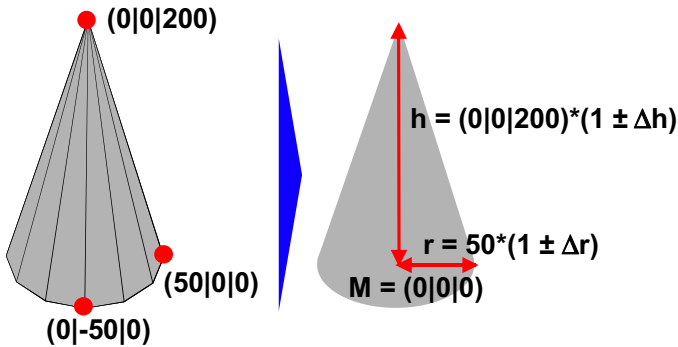


Bild 4-4: Darstellung eines Kegels als Polygonflächenmodell und parametrisiert

Bei der automatischen Geometriezerlegung einfach zu erkennende Körper sind

- Zylinder,
- Zylinderspate,
- gerade Prismen und Prismenspate mit einem Parallelogramm, einem Dreieck oder einer Ellipse als Grund- und Deckfläche,
- Kegel und Kegelstümpfe.

Das in Bild 4-5 links dargestellte Bauteil z. B. kann aus zwei Zylindern und einem Kegelstumpf zusammengesetzt werden.

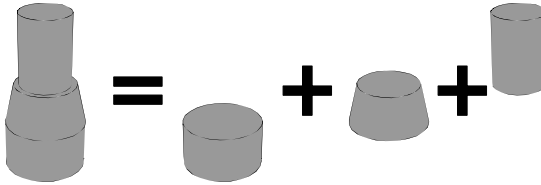


Bild 4-5: Zerlegung einer Geometrie in Elementarkörper

Für eine parametrisierte Darstellung mit Elementargeometrien werden zunächst die Polygonflächen eines Objekts automatisch als Kreise, Ellipsen, Parallelogramme und Dreiecke klassifiziert. Parallele Kreisflächen werden dann dahingehend untersucht, ob sie mit ihren Nachbarflächen einen Zylinder, einen Zylinderspat oder einen Kegelstumpf bilden. Bei Kreisflächen wird zudem untersucht, ob sie die Grundfläche eines Kegels sind. Bei zwei Parallelogrammen, Ellipsen oder Dreiecken in zueinander parallelen Ebenen wird geprüft, ob sie zusammen mit angrenzenden Flächen ein Prisma bilden. Dabei werden die charakteristischen Größen für die jeweilige Elementargeometrie in einer Datenstruktur abgelegt. Beim Kegel sind dies der Höhenvektor und sein Aufpunkt sowie der Radius des Grundkreises. Zudem wird bei der Geometrieanalyse festgestellt, ob es sich um ein positives oder ein negatives Geometrieelement handelt. Ein negatives Geometrieelement ist z. B. ein Zylinder, der eine Bohrung repräsentiert.

Mit Hilfe der Geometriezerlegung können dann Objekte in ihre Elementarkörper zerlegt werden, die charakteristischen Maße der Elementarkörper geändert werden, die Elementarkörper im Gesamtobjekt neu plaziert werden und schließlich wieder zum Gesamtobjekt zusammengesetzt werden.

4.2.3 Abschätzung von Ungenauigkeiten

Zur Abschätzung der Auswirkung von Ungenauigkeiten auf das Kollisionsverhalten muß aus den parametrisierten Geometriedaten die größte sich ergebende Geometrie berechnet werden. Negative Elementargeometrien werden dabei mit

den kleinstmöglichen Abmaßen, positive Elementargeometrien mit den größtmöglichen Abmaßen dargestellt (Bild 4-6). Restgeometrien, die mit dem einfachen Verfahren nicht weiter in Elementargeometrien zerlegt werden konnten, werden wie positive Elementargeometrien behandelt. Formtoleranzen oder Objektverdrehungen werden durch einen Aufschlag bei der Vergrößerung berücksichtigt.

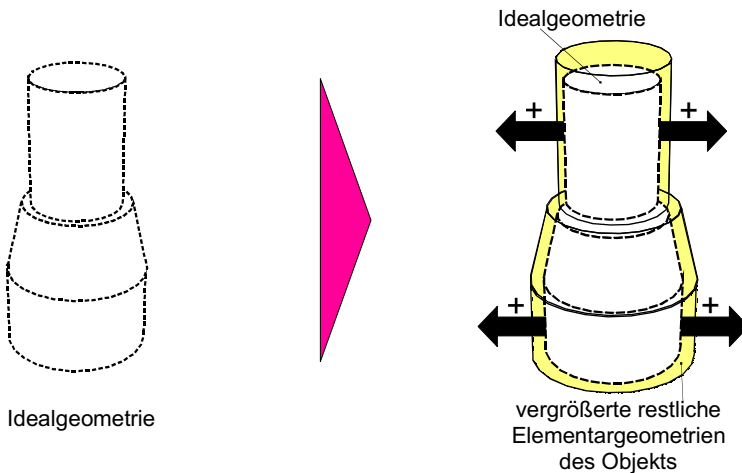


Bild 4-6: Vergrößerung der Elementargeometrien

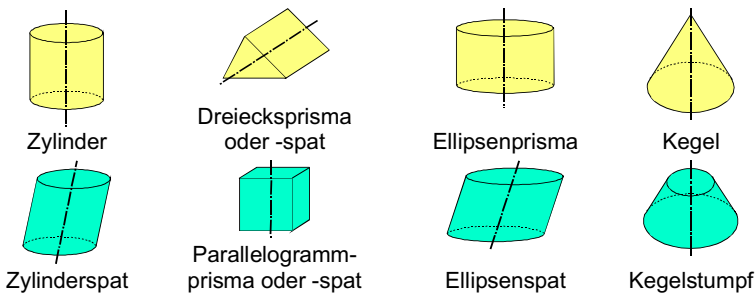


Bild 4-7: Betrachtete Elementargeometrien und ihre Symmetrieachsen

Die betrachteten Elementargeometrien besitzen mindestens eine Symmetrieachse, die entsprechend toleriert ist. Neben der Symmetrieachse bestimmt mindestens eine weitere ebene Fläche, die durch zwei Hauptachsen mit charakteristischen Abmessungen gekennzeichnet ist, die Gesamtgeometrie (Bild 4-7).

Aus den angegebenen Toleranzmaßen oder den entsprechenden Allgemeintoleranzen nach DIN 7168 (1970) kann dann die größte bzw. die kleinste Elementargeometrie berechnet werden. Die Planläufigkeit der ebenen Fläche wird durch Addition der entsprechenden Differenz zur neuen Höhe des Objekts berücksichtigt. Eine Schiefstellung der Symmetrieachse wird durch die Erhöhung der Abmaße der Hauptachsen kompensiert usw. Ein Zylinder mit den Maßen $h = 100 \text{ mm}$ und $r = 50 \text{ mm}$ ohne Toleranzangabe wird für die erweiterte Kollisionsrechnung mit Berücksichtigung von Ungenauigkeiten dann folgendermaßen dargestellt. Bei grober Toleranzklasse nach DIN 7168 gilt $h = 100 \pm 0,8 \text{ mm}$, $r = 50 \pm 0,8 \text{ mm}$. Der rechte Winkel der Symmetrieachse zu den Flächen darf um $\pm 25^\circ$ abweichen. Nach Toleranzklasse T ergibt sich eine Rundlauf- und Planlauf-toleranz von $0,5 \text{ mm}$ sowie eine Symmetrietoleranz von $1,0 \text{ mm}$.

Bei der Winkelabweichung verschiebt sich der Aufpunkt bzw. die Spitze um maximal $h \cdot \tan(25^\circ) = 0,7 \text{ mm}$. Dies liegt also innerhalb der Tolerierung des Radius von $\pm 0,8 \text{ mm}$ und innerhalb der Symmetrietoleranz von $1,0 \text{ mm}$. Die Effekte der Rundlauf- und der Planlauf-toleranz liegen weit unterhalb der zulässigen linearen Abweichungen. D. h., es genügt für die vergrößerte Geometrie für die Kollisionsrechnung einen Zylinder mit Radius $50,8 \text{ mm}$ und Höhe $100,8 \text{ mm}$ zu verwenden, für die Minimalgeometrie einen Zylinder mit Radius $49,2 \text{ mm}$ und Höhe $99,2 \text{ mm}$. Dies erklärt sich auch durch die nationale Gültigkeit des Hüllprinzips nach DIN 7167, nach dem die Maßtoleranz das Maximum-Material-Maß für das jeweils idealisierte Formelement bestimmt, das das wirkliche Formelement umhüllt und innerhalb dessen die wahren Konturen liegen müssen (PAHL & FLEMMING 1997, S. 32). Allerdings haben die Form- und Lagetoleranzen sehr wohl eine Auswirkung auf Elemente, die auf diese Elementargeometrie aufgesetzt sind. Schließt sich z. B. bei einer Exzenterwelle ein zweiter Zylinder an den so tolerierten an, so kann die Planlauf-toleranz einen Einfluß auf die Achstolerierung des folgenden Zylinders haben, der bei

ungünstiger Verteilung schon eine Grundschieflage erhält, die in erster Linie von der Art der Bemaßung abhängt. Diese Fehler, die durch eine unsachgemäße Bemaßung entstehen, sollen hier allerdings nicht überprüft werden.

Wegen der Polygonflächendarstellung in AnySIM/USIS müssen bei der Vergrößerung für die Kollisionsrechnung bei Rotationskörpern zusätzlich die durch die Polygonisierung entstandenen Sekantenfehler berücksichtigt werden. Der Radius bzw. die Hauptachsenabmessungen einer Ellipse müssen zusätzlich um den Sekantenfehler erhöht werden.

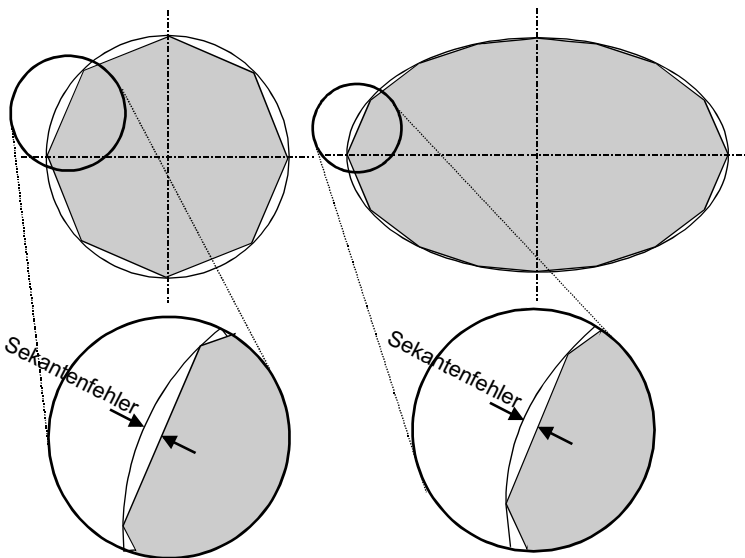


Bild 4-8: Berücksichtigung des Sekantenfehlers

Die Berechnung der vergrößerten (oder analog der verkleinerten) Elementargeometrien ist aufgrund der parametrisierten Darstellung und der gültigen Allgometoleranzen definiert. Schwieriger zu erfassen sind die Auswirkungen der Tolerierung von allgemeinen Elementen mit Freiformflächen. Für die Kollisionsrechnung für Objekte mit Freiformflächen soll deshalb ein Ansatz gewählt wer-

den, der eine umschreibende Geometrie ermittelt, über die die enthaltenen Geometrieelemente auf keinen Fall hinausragen, auch wenn damit gleichzeitig in Kauf genommen werden muß, daß u. U. zu große Geometrien verwendet werden.

Dazu wird zunächst die größtmögliche, bei einer Verdrehung auftretende Verschiebung eines Objektpunktes berechnet und aus der zusammen mit der größtmöglichen linearen Verschiebung entstehenden Gesamtverschiebung die Erweiterung des Objekts in den drei Hauptrichtungen ermittelt. Im Gegensatz zur Geometrievergrößerung von Elementargeometrien können hier nicht die Verschiebungsbeträge der linearen Verschiebung und der aus einer Verdrehung resultierenden Verschiebung verglichen werden und der größere der beiden Werte gewählt werden. Freiformflächen haben in der Regel eine sehr komplizierte Geometriebeschreibung, die nur sehr schwer nach DIN 7168 zu tolerieren ist. Eine evtl. im CAD-System definierte Tolerierung wird nicht ins Simulationssystem übertragen. Durch die Addition der beiden Verschiebungsanteile wird eine sichere Hülle ermittelt, während bei der Wahl des größeren Anteils z. B. wegen ungünstiger Positionierung einer Toleranz auf der Objektgeometrie u. U. keine wahre Hülle entsteht. D. h. es wird eine Hülle berechnet, die zwar viele Fälle einhüllt, nicht aber alle im Toleranzbereich zulässigen Geometrien.

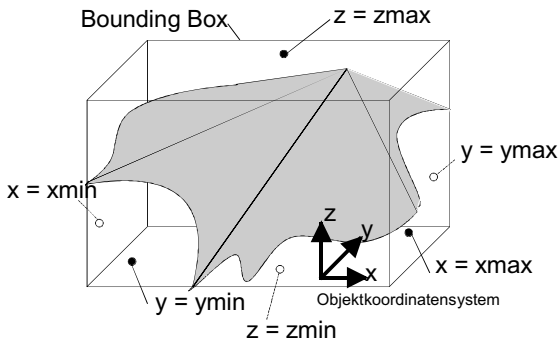


Bild 4-9: Bounding Box eines Objekts

Zur Vereinfachung der Berechnung der Verschiebungen wird ein Hüllquader des Objekts, die sog. Bounding Box (SCHRÜFER 1992, S. 101) verwendet (Bild 4-9). Für die Berücksichtigung von Winkeltoleranzen wird die Bounding Box des Objekts um alle Achsen des Objektkoordinatensystems gedreht. Der Drehpunkt ist der geometrische Schwerpunkt der Bounding Box. Hatte die Bounding Box ursprünglich die Abmaße $x_0 = x_{\max} - x_{\min}$, $y_0 = y_{\max} - y_{\min}$ und $z_0 = z_{\max} - z_{\min}$, so liegt eine um die Winkel α , β und γ gedrehte Bounding Box innerhalb einer Bounding Box mit den Abmaßen $x' = x_0 + \Delta x$, $y' = y_0 + \Delta y$ und $z' = z_0 + \Delta z$ (Bild 4-10), wobei sich x , y und z wie folgt berechnen:

- $x' = (x_0 \cos \beta + (y_0 \sin \alpha + z_0 \cos \alpha) \sin \beta) \cos \gamma + (y_0 \cos \alpha + z_0 \sin \alpha) \sin \gamma$
- $y' = (x_0 \cos \beta + (y_0 \sin \alpha + z_0 \cos \alpha) \sin \beta) \sin \gamma + (y_0 \cos \alpha + z_0 \sin \alpha) \cos \gamma$
- $z' = (y_0 \sin \alpha + z_0 \cos \alpha) \cos \beta + x_0 \sin \beta$

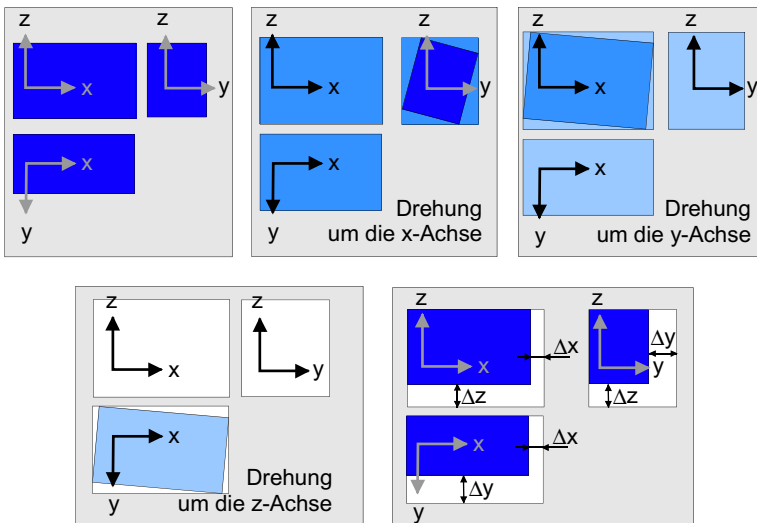


Bild 4-10: Vergrößerung der Bounding Box bei Drehung

Für ein Beispiel mit $x = 100$ mm, $y = 10$ mm, $z = 30$ mm und den Winkeln $\alpha = 1^\circ$, $\beta = 0,5^\circ$, $\gamma = 1^\circ$ (entsprechend den mittleren Allgometoleranzen nach DIN 7168) ergibt sich $x' = 100,4$ mm, $y' = 12,3$ mm und $z' = 31,0$ mm. Es gilt also $\Delta x_{\text{rot}} = 0,4$ mm, $\Delta y_{\text{rot}} = 2,3$ mm und $\Delta z_{\text{rot}} = 1,0$ mm.

Für die Gesamtabweichung müssen zudem noch die Abweichungen der linearen Abmaße nach DIN 7168 berücksichtigt werden. Dabei ergibt sich für das Beispiel

- $\Delta x_{\text{ges}} = \Delta x_{\text{lin}} + \Delta x_{\text{rot}} = 0,3$ mm + $0,4$ mm = $0,7$ mm.
- $\Delta y_{\text{ges}} = \Delta y_{\text{lin}} + \Delta y_{\text{rot}} = 0,2$ mm + $2,3$ mm = $2,5$ mm.
- $\Delta z_{\text{ges}} = \Delta z_{\text{lin}} + \Delta z_{\text{rot}} = 0,3$ + $1,0$ mm = $1,3$ mm.

Zur Berücksichtigung der Verschiebung im Polygonflächenmodell werden alle Polygonpunkte für die Berechnung einer vergrößerten Geometrie um $0,5 \Delta x_{\text{ges}}$ in x-Richtung, um $0,5 \Delta y_{\text{ges}}$ in y-Richtung und um $0,5 \Delta z_{\text{ges}}$ in z-Richtung vom Schwerpunkt weg verschoben. Bei sehr großen Flächen wird die Fläche zunächst in mehrere Teilflächen unterteilt und anschließend das beschriebene Verfahren zur Ermittlung der Hüllgeometrie angewendet. So wird das Problem der Verwendung zur großer Hüllkörper und damit der Schwierigkeit der Bewertung irrelevanter Kollisionen verringert.

Die Betrachtung von Toleranzketten ist nicht trivial und mit einer nachträglich parametrisierten Geometriedarstellung nur schwer zu verwirklichen. Zur Abschätzung für die Kollisionsrechnung werden die doppelten Toleranzen verwendet, da bei einer sachgemäßen Tolerierung so eine entsprechende Toleranzhülle erzeugt wird, die für eine überschlagsmäßige Betrachtung für die Kollisionsrechnung ausreichend ist.

Damit ist es möglich, bei der Planung eines Handhabungsvorgangs festzustellen, ob eine Kollision sicher ausgeschlossen werden kann. Dies ist der Fall, wenn für die so vergrößerten Objektgeometrien keine Kollision auftritt. Tritt schon bei den kleinstmöglichen Objektgeometrien Kollision auf, so kann die Handhabung nicht auf die überprüfte Art und Weise erfolgen. In allen anderen Fällen kann keine sichere Aussage getroffen werden. Die Verwendung verkleinerter und vergröß-

Größere Objektgeometrien bieten gegenüber Abstandsuntersuchungen einen Zeitvorteil, da nur einmal die veränderte Geometrie berechnet werden muß und dann die Durchführung von Kollisionstests genügt. Bei der Berechnung von Abständen dagegen müssen jedes Mal erneut für die Flächen der beiden untersuchten Objekte die einzelnen Abstände zueinander berechnet werden. Eine Abstandsrechnung ist außerdem aufwendiger als ein Kollisionstest.

Die Abschätzung von Ungenauigkeiten über eine Zerlegung von Elementargeometrien eignet sich auch zur Untersuchung von Greifvorgängen, bei denen einerseits ein Kontakt zwischen Objekt und Greifer hergestellt werden muß, andererseits, die Greiferfinger kollisionsfrei an die Greifposition bewegt werden müssen. Mit Hilfe des hier vorgestellten Verfahrens kann die Absicherung einer Greifposition mit zwei Kollisionsuntersuchungen erfolgen.

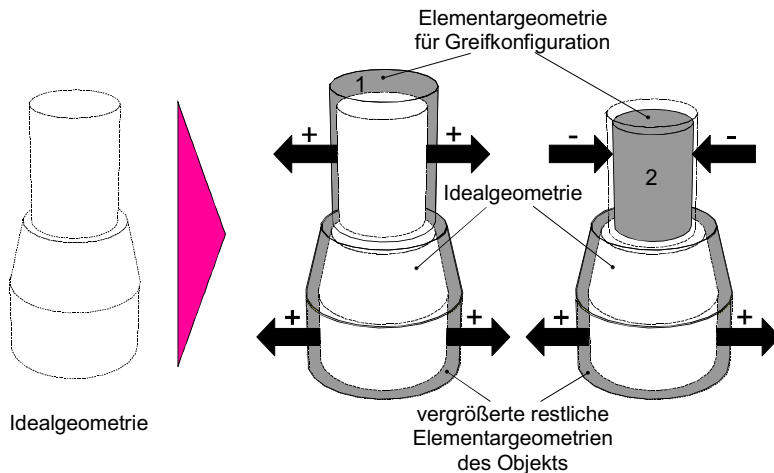


Bild 4-11: Berücksichtigung von Toleranzen beim Greifen

Dazu wird zunächst die größtmögliche Objektgeometrie in Kombination mit der größtmöglichen Geometrie des geöffneten Greifers betrachtet. Tritt hier keine

Kollision auf, so kann der Greifer kollisionsfrei vom Roboter in Greifposition plaziert werden. Für einen sicheren Griff müssen die innerhalb des Toleranzbereichs kleinsten Greiferfinger auch die entsprechende kleinste äußere Objektabmessung sicher umschließen können bzw. sich sicher in die größte Bohrung einspreizen können. Dazu wird die Objektgeometrie mit Ausnahme des Elements, an dem der Greifer angreift, vergrößert und die Elementargeometrie des Angriffselements verkleinert (Möglichkeit 2 in Bild 4-11). Für diese Geometrieausprägung muß ein Kontakt innerhalb des Greifbereichs hergestellt werden können.

Diese Bewertung von Greifpositionen ermöglicht eine sichere Greifplanung, die im Rahmen der automatischen Planung von Handhabungsvorgängen im folgenden Kapitel beschrieben wird.

5 Handhabungsplanung

Die automatische Handhabungsplanung erstellt aus einer aufgabenorientierten Beschreibung eines Handhabungsauftrags automatisch ein kollisionsfreies Roboterprogramm. Die erforderlichen Angaben bei der aufgabenorientierten Beschreibung sind das zu handhabende Objekt, seine Start- und seine Zielposition. Stehen in der Zelle mehrere Roboter zur Verfügung, so muß zusätzlich der einzusetzende Roboter definiert werden, bei mehreren verfügbaren Greifern ist diese Angabe nicht zwingend erforderlich, da bei der Erzeugung des Roboterprogramms auch mehrere Greifer bzgl. ihrer Eignung für die Handhabungsaufgabe getestet werden können.

Die Angaben können entweder direkt im Simulationssystem interaktiv eingegeben werden oder es kann eine textuelle Aufgabenbeschreibung verwendet werden. Für das Simulationssystem AnySIM/USIS wurde eine derartige Beschreibung für eine Schnittstelle zu einem der Simulation übergeordneten Montageplanungssystem von CUIPER & ROßGODERER (1997) entwickelt, die Module des im folgenden vorgestellten Systems zur Handhabungsplanung in einem System zur automatischen Generierung eines optimalen Zellenlayouts mit RC-Programm auf Basis eines Montageablaufplans nutzen.

Die automatische Handhabungsplanung besteht aus einem Modul zur übergeordneten Handhabungsplanung, das seinerseits auf Module zur Greifplanung, Feinbewegungsplanung und Bahnplanung zurückgreift (Bild 5-1). Das übergeordnete Modul zur Handhabungsplanung legt die Reihenfolge der Planung der einzelnen Schritte fest, die nicht der Reihenfolge des Handhabungsablaufs entspricht.

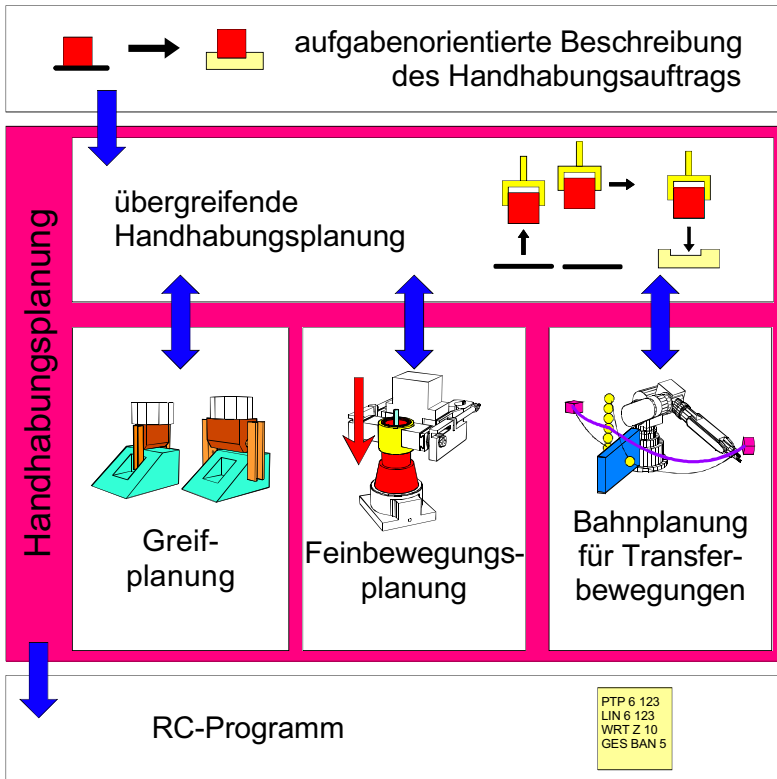


Bild 5-1: System zur automatischen Handhabungsplanung

Bei einem typischen Handhabungsvorgang (Bild 5-2) nähert sich der Roboter zunächst mit einer weiträumigen Transferbewegung der Greifposition, fährt diese mit einer Feinbewegung exakt an, greift das Objekt und verläßt mit einer weiteren Feinbewegung den Greifbereich. Mit einer weiteren Transferbewegung und einer sich daran anschließenden Feinbewegung nimmt er die Ablegeposition ein, setzt das Objekt ab und verläßt mit einer Feinbewegung den Ablegebereich.

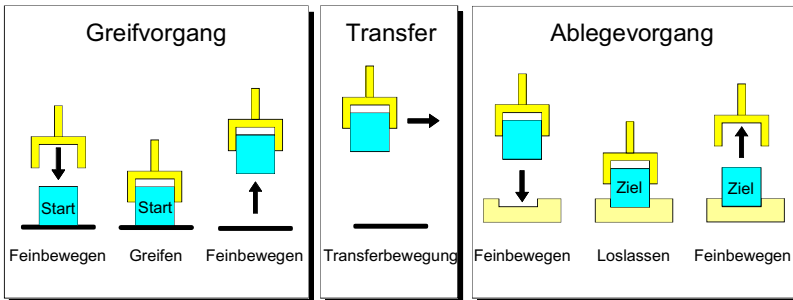


Bild 5-2: Standardstruktur eines Handhabungsvorgangs

Bei der Planung dieses Handhabungsvorgangs müssen zuerst der Greifvorgang und der Ablegevorgang bestimmt werden. Die Feinbewegungen werden dabei in Abhängigkeit von der Greif- und der Ablegeposition geplant. Nach Festlegung des Greif- und des Ablegevorgangs wird das Programm schließlich mit der Planung der Transferbewegungen vervollständigt. Außerdem müssen manchmal Planungsschritte wiederholt werden, da die Lösungsvariante des vorhergehenden Planungsschrittes (z. B. der Greifplanung, die die Greifposition bestimmt) die Lösung eines nachfolgenden Planungsschrittes (z. B. der Planung einer gültigen Feinbewegung) ausschließt. Diese Steuerung des Planungsablaufs und die damit verbundene Strukturierung des Handhabungsvorgangs wird von der übergreifenden Handhabungsplanung übernommen. Sie hat zudem die Aufgabe, Eingaben vom Benutzer anzufordern, falls die Planung nicht automatisch erfolgen kann oder Einschränkungen für die Sicherheit der Lösung hingenommen werden müssen, beispielsweise eine Position an den Grenzen des Roboterarbeitsraums durchfahren wird.

Im folgenden werden zunächst die untergeordneten Planungsmodule und abschließend die übergreifende Handhabungsplanung vorgestellt.

5.1 Greifplanung

Die Greifplanung bestimmt zuerst sog. Greifkonfigurationen zum Greifen des Objekts. Eine Greifkonfiguration beschreibt die relative Lage des Greifers zum Werkstück, wenn dieses vom Greifer gegriffen ist. Je nach Greifertyp existieren in Abhängigkeit von Greifer- und Objektgeometrie mehrere geometrisch mögliche Greifkonfigurationen (Bild 5-3). Sie müssen jedoch auch die Kriterien der Griffstabilität und der Kollisionsfreiheit erfüllen, um für die Greifaufgabe angewandt werden zu können. Sind diese Kriterien erfüllt, so ist zusätzlich zu überprüfen, ob die zum Herstellen der jeweiligen Greifkonfiguration erforderliche Roboterposition erreichbar ist und ob das Objekt mit dieser Greifkonfiguration vom Roboter in die Ablegeposition gebracht werden kann. Erst dann ist die Greifplanung abgeschlossen.

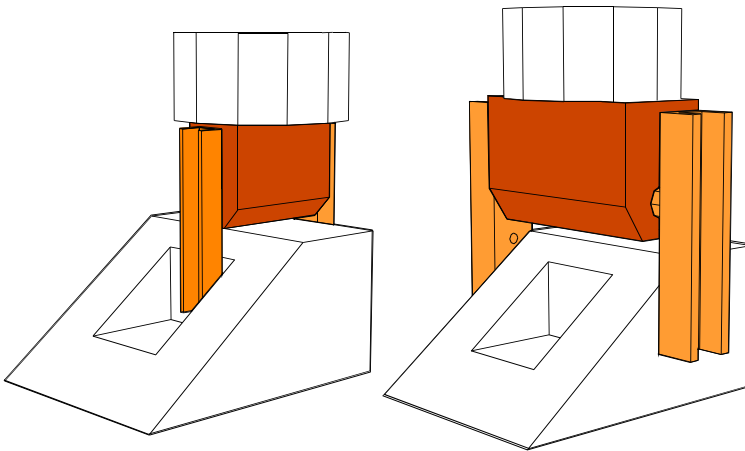


Bild 5-3: mögliche Greifkonfigurationen zum Greifen des abgebildeten Objekts

5.1.1 Greifertypen

Nach STETTER (1993) und KUGELMANN (1991) lassen sich Greifer in Standardgreifer und Spezialgreifer unterteilen. Zu den Standardgreifern zählen Parallelgreifer, Dreipunktgreifer (z. B. Drei-Finger-Außengreifer) und Vakuumgreifer. Die meisten Standardgreifer ausgenommen der Vakuumgreifer bestehen aus einem Joch und beweglich darin gelagerten Fingern (Bild 5-4). Spezialgreifer sind z. B. Röhrengreifer oder die der menschlichen Hand nachempfundenen Multifingergreifer. Weiterhin ist zu unterscheiden, nach welchem Prinzip der Griff entsteht, d.h. durch Formschluß oder durch Kraftschluß. Beim Formschluß berühren sich Greifer und Objekt meist großflächig, was dadurch erreicht wird, daß z. B. die Greiferfinger die Form der negativen Geometrie des zu greifenden Objekts besitzen. Beim Kraftschluß muß der gesamten Gewichtskraft durch die Greifkraft entgegengewirkt werden (STETTER 1993, S. 45). Dies erfolgt durch Haftreibung zwischen Fingern und Objekt. Beim Formschluß dagegen können durch entsprechende Gestaltung der Greiferfinger auch Normalkräfte zum Halten des Objekts genutzt werden.

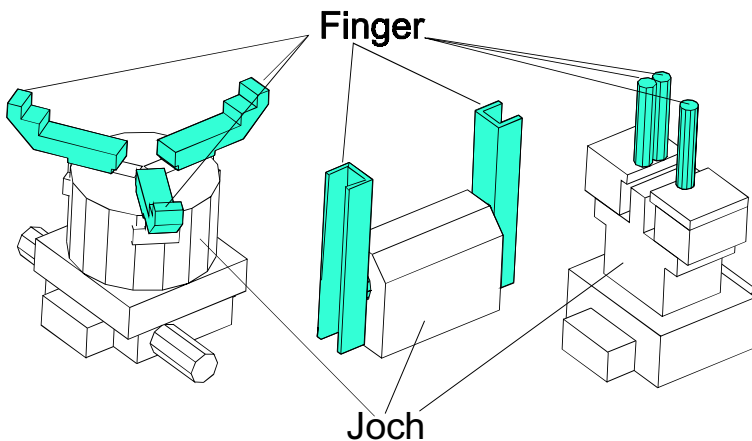


Bild 5-4: Greifer mit Joch und beweglichen Fingern

5.1.2 Bestimmung geeigneter Greifkonfigurationen

Insbesondere beim Greifen durch Formschluß, aber auch bei kraftschlüssigen Verbindungen werden die Greifkonfigurationen oft bereits bei der Konstruktion des Objekts und des Greifers beschränkt, z. B. durch die Definition zugelassener Greifflächen. Eine Schnittstelle zur Übernahme dieser Greifflächen aus einem CAD-System wurde von ROßGODERER ET AL. (1995) entwickelt. Eine bestimmte gewünschte Greifkonfiguration für eine Paarung von zu greifendem Teil und Greifer kann auch durch Festlegung einer Relativkonfiguration dieser beiden Objekte im Simulationslayout definiert werden und steht dann für weitere Planungsschritte zur Verfügung (ROßGODERER ET AL. 1995). Derartige Relativkonfigurationen dienen grundsätzlich zur Speicherung von Relativplatzierungen zwischen zwei Objekten.

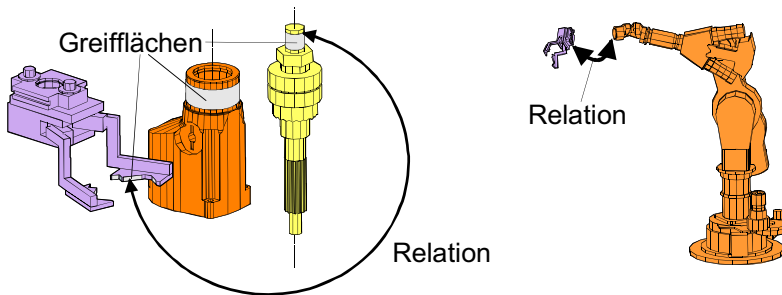


Bild 5-5: Verwendung von Funktionselementen (z. B. Greifflächen) und Relationen in der 3D-Simulation (aus ROßGODERER ET AL. 1995)

Besteht jedoch kein Vorwissen in Form von Relativkonfigurationen über mögliche Greifkonfigurationen, so muß eine Greifplanung durchgeführt werden. Bei Verwendung eines Zweifingerparallelgreifers mit Greifbacken, deren Greifflächen eben und senkrecht zur Richtung der aufgebrachtten Greifkraft sind, existieren in der Regel viele unterschiedliche Greifkonfigurationen. Für diesen Greifertyp hat STETTER (1993) ein Verfahren zur Greifplanung entwickelt, das zunächst

die zum Greifen geeigneten geometrischen Elemente des Objekts bestimmt (Bild 5-6 oben).

Dies sind parallele Flächen (Bild 5-6 oben links), Flächen und dazu parallele Linien (Bild 5-6 oben in der Mitte), Flächen und ein Punkt (z. B. Spitze und Grundfläche eines Kegels oder einer Pyramide wie in Bild 5-6 oben rechts), zueinander parallele Linien usw. Mit diesem Verfahren können Greifkonfigurationen für Objekte mit parallelen Innen- und Außengeometrien bestimmt werden. Greifkonfigurationen, die aufgrund des Öffnungsbereichs des Greifers oder der maximal möglichen Greifkraft nicht geeignet sind, werden sofort wieder verworfen.

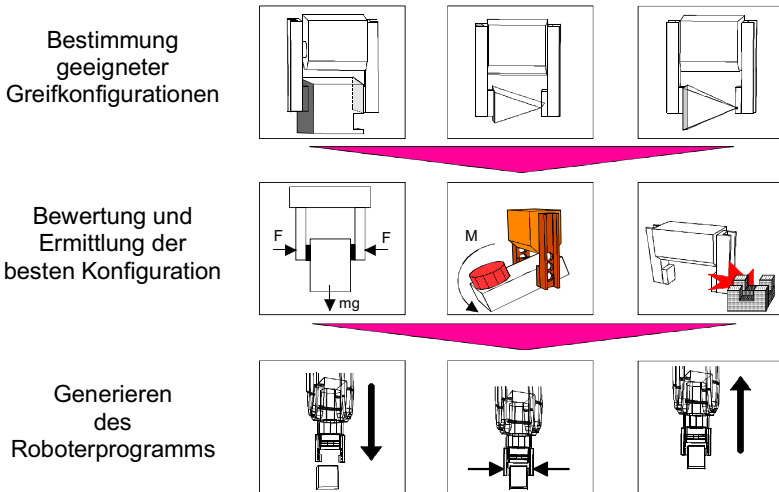


Bild 5-6: Greifplanungsverfahren von STETTER (1993)

Die verbleibenden, möglichen Konfigurationen werden entsprechend der gebotenen Stabilität in eine Liste sortiert, deren erstes Element die stabilste bestimmte Greifkonfiguration ist. Bei zwei parallelen Flächen werden dabei im Überlappungsbereich mehrere Angriffspunkte für den Greifer als mögliche Greifkonfigurationen betrachtet. Greifkonfigurationen mit einem Kraftangriffspunkt nahe dem

Objektschwerpunkt sind zwar stabiler. Damit die Aufgabe aber auch gelöst werden kann, wenn die schwerpunktnahen Konfigurationen aufgrund einer Kollision des Greifers mit Objekten in der Umgebung nicht geeignet sind, werden auch andere Konfigurationen betrachtet.

Zur Bewertung der Stabilität berechnet STETTER (1993) einen Beiwert, der die Wahrscheinlichkeit des Entgleitens des Objekts aus den Greiferfingern oder des Abkippens des Objekts in den Fingern charakterisiert. Dabei werden zunächst Griffe, die zwei parallele Flächen nutzen, stabiler eingestuft als Griffe, die eine Fläche und eine Kante nutzen, usw. Außerdem wird unterschieden, ob der Abstand der beiden parallelen Elemente nahe an einer der Grenzen des Greiferöffnungsbereichs liegt oder einem günstigeren Wert in der Mitte des Öffnungsbereichs entspricht. Mit einem Zweifingeraußengreifer kann an zwei parallelen Objektflächen, deren Abstand der geringsten Öffnungsweite entspricht, nicht die maximale Anpreßkraft aufgebracht werden. Bei einem Flächenabstand, der sich kaum von der größten Öffnungsweite unterscheidet, besteht die Schwierigkeit, den Greifer am Objekt zu positionieren ohne mit dem Objekt zu kollidieren. Der dritte Einflußfaktor für den Stabilitätswert ist der Abstand des Objektschwerpunkts zum Greifkraftangriffspunkt. Je weiter entfernt vom Schwerpunkt die Finger greifen, desto wahrscheinlicher ist ein Abkippen des Objekts in den Fingern.

Nach der Bewertung der Stabilität der kollisionsfreien Greifkonfigurationen legt STETTER (1993) schließlich die stabilste Greifkonfiguration zusammen mit einer Anfahr- bzw. Abrückposition für den Roboter senkrecht über der Greifposition in einem Roboterprogramm ab.

Viele Objekte werden jedoch auch an Bohrungen oder an äußeren zylindrischen Elementen gegriffen, wofür ein Zweifingerparallelgreifer mit ebenen Backen nur unzureichende Griffsicherheit bietet. So entspricht z. B. bei Innengreifern, auch wenn sie als Zweifingerparallelgreifer ausgeführt sind, die Geometrie meist nicht den bei STETTER (1993) getroffenen Voraussetzungen. Bei Drehteilen bzw. zylindrischen Formelementen werden in der Regel Dreipunktaußengreifer verwendet. Für das hier vorgestellte System zur integrierten Programmgenerierung werden deshalb für die Greifplanung die beiden Module von STETTER (1993) zur

Suche nach parallelen Geometrieelementen und zur Bewertung der Greifkonfigurationen mit einer Vielzahl weiterer Module in einem umfassenderen Greifplanungssystem kombiniert und an die Anforderungen der übergreifenden Handhabungsplanung angepaßt. Zusätzlich enthält dieses System beispielsweise Algorithmen zur Bestimmung von Greifkonfigurationen für Innengreifer an Bohrungen und für Drei-Finger-Außengreifer (Bild 5-7).

Beide Algorithmen suchen zunächst nach zylindrischen Geometrieelementen, vergleichen den Zylinderdurchmesser mit dem Öffnungsbereich des Greifers und platzieren anschließend den Greifer an dem zylindrischen Geometrieelement. Kollidiert hierbei der Greifer mit der Umgebung, so kann meist durch ein Drehen des Greifers um die Symmetrieachse des zylindrischen Geometrieelements eine kollisionsfreie Greifkonfiguration bestimmt werden. Die Greifplanung ermittelt also im ersten Schritt die zu einem bestimmten Greifertyp passenden Elementargeometrien des zu greifenden Objekts und wählt im nächsten Schritt, bei dem der wirklich verwendete Greifer dieses Typs dann feststeht, diejenigen aus, die mit ihren Abmessungen innerhalb der Öffnungsweite des verwendeten Greifers liegen. Bei zylindrischen Geometrieelementen werden bei dieser Auswahl gleich die kleinstmöglichen und die größtmöglichen Abmessungen (also der Variationsbereich) berücksichtigt. Bei einer Bohrung z. B. muß sowohl der minimale als auch der maximale Bohrungsdurchmesser, der sich durch die Tolerierung ergeben darf, im Öffnungsbereich des Greifers liegen. Dann werden für ein zylindrisches Element mehrere Greiferangriffsmöglichkeiten überprüft, die sich durch ein Drehen des Greifers um die Zylinderachse um diskrete Winkelschritte und ein Verschieben des Greifers entlang der Zylinderachse um diskrete Längenschritte ergeben. Die dabei ermittelten kollisionsfreien Greiferstellungen werden dann wie beim Verfahren von STETTER (1993) bzgl. der Griffstabilität bewertet und sortiert.

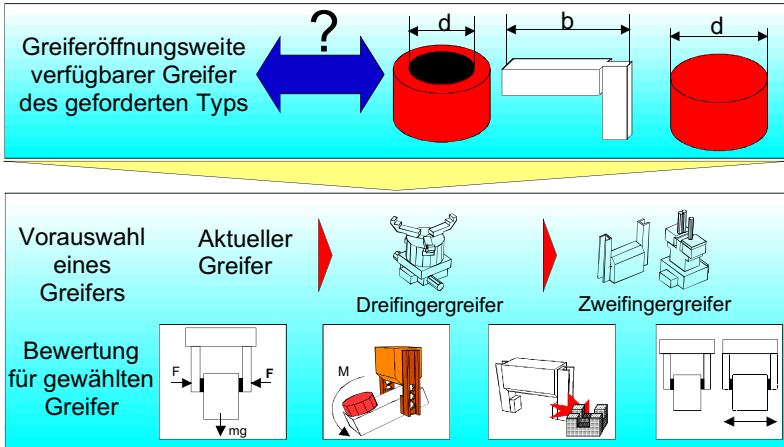


Bild 5-7: Ablauf der erweiterten Greifplanung

Bei zueinander parallelen Geometrieelementen wird deren maximaler und minimaler Abstand noch nicht unmittelbar nach der Bestimmung paralleler Geometrieelemente berücksichtigt, da die Geometrieelemente Teil unterschiedlicher Elementargeometrien sein können und eine Berechnung des minimalen und maximalen Abstands deshalb aufwendiger ist als bei zylindrischen Elementargeometrien.

Handelt es sich bei den Elementargeometrien um parallele Flächen, so müssen diese nach der Bewertung der Greifkonfiguration bzgl. der Griffstabilität dahingehend überprüft werden, ob sie auch bei den durch die Tolerierung möglichen Abstandsvariationen für eine Greifkonfiguration geeignet sind. Dies erfolgt im Rahmen der Überprüfung der Auswirkungen der Bauteiltoleranzen auf etwaige Kollisionen beim Greifen des Objekts wie in Kap. 4 beschrieben. Die Stabilitäts-güte des Griffs der Idealgeometrie wird vereinfachend auf den gesamten Toleranzbereich übertragen.

Ist die Tolerierung so ungünstig, daß sie kein sicheres Greifen des Objekts mit dem betrachteten Greifer erlaubt, wird dies über die übergreifende Handhabungs-

planung dem Benutzer mitgeteilt, der dann die Möglichkeit hat, einen Test mit engerer Tolerierung durchzuführen und u. U. konstruktive Maßnahmen zu ergreifen.

Um bei einer späteren Greifplanung für die gleiche Objektgeometrie die erneute Suche nach geeigneten Geometrieelementen, d. h. parallelen Geometrieelementen, zylindrischen Elementen und Bohrungen, zu verkürzen, ist die Möglichkeit gegeben, die für einen bestimmten Greifer extrahierten Elemente persistent abzulegen. Gespeichert werden dabei z. B. nach einer Greifplanung für einen Zweifingergreifer alle parallelen Geometrieelemente und die zugehörige Öffnungsweite des Greifers. Bei einer erneuten Greifplanung mit einem Zweifingergreifer müssen dann nur die seinem Öffnungsbereich entsprechenden Geometriepaare ausgewählt werden, die Geometrieanalyse entfällt.

Außerdem kann nach einer Greifplanung für einen Greifer die beste Greifkonfiguration als Relativkonfiguration (wie in Bild 5-5 beschrieben) gespeichert werden und steht so für weitere Planungsschritte zur Verfügung. Eine Mehrfachplanung für die gleiche Aufgabe wird somit vermieden.

5.1.3 Roboterspezifische Bewertung der Greifkonfigurationen

Die ermittelten möglichen Greifkonfigurationen berücksichtigen die durch die Objektgeometrie und den Greifer gegebenen Randbedingungen. Die zum Greifen des Objekts an der Startlage benötigte Greiferstellung muß aber auch durch entsprechende Positionierung des Roboters erreicht werden können. Deshalb wird im nächsten Schritt für alle Greifkonfigurationen eine zugehörige Roboterposition bestimmt. Greifkonfigurationen, deren Position nicht vom Roboter angefahren werden kann, werden verworfen. Durch die Relativposition des Greifers zum Objekt und die Relativposition des Greifers zur Roboterhand ist die Lage des Werkzeugkoordinatensystems des Roboters, also die Position seines Tool Center Points (TCP) bestimmt. Wie bereits in Kapitel 4 erwähnt, kann jedoch eine TCP-Position von den meisten Robotertypen mit unterschiedlichen Achswinkelstellungen des Roboters erreicht werden (Bild 5-8).

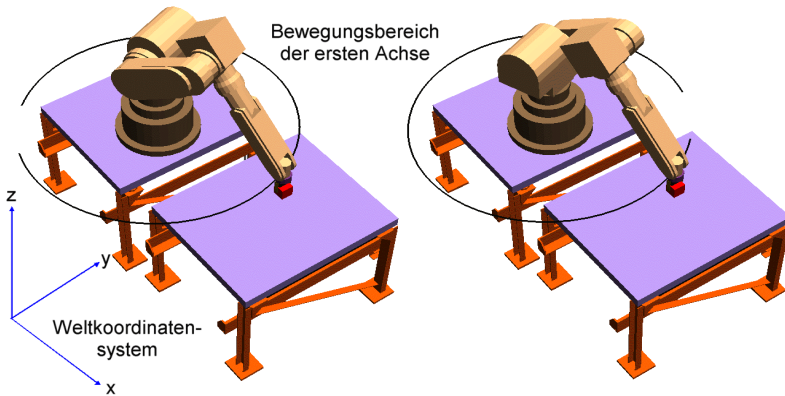


Bild 5-8: Greifkonfiguration und mögliche Achswinkelstellungen für die entsprechende TCP-Position

Bei der Darstellung einer Roboterstellung als TCP-Position muß deshalb zusätzlich ein sog. Stellungsparameter angegeben werden, der eine eindeutige Zuordnung der entsprechenden Achswinkel bei der Rücktransformation der TCP-Position ermöglicht. In ungünstigen Fällen kann die Wahl des Stellungsparameters die Bewegungsmöglichkeiten des Roboters erheblich einschränken. Befindet sich wie bei der in Bild 5-8 rechts dargestellten Roboterstellung z. B. die erste Achse nahe ihrem Anschlag in positive Drehrichtung, so kann sich der Roboter nur noch wenig in die y-Richtung des Weltkoordinatensystems bewegen. In der links dargestellten Stellung dagegen befindet er sich im Überkopfbereich und kann noch sehr weit in positiver y-Richtung des Weltkoordinatensystems verfahren. Es muß deshalb überprüft werden, welche Achswinkelstellungen des Roboters zu der zu einer Greifkonfiguration gehörigen TCP-Position des Roboters ausreichende Bewegungsfreiheit im Bereich um diese Position bieten. Dazu muß vor allem der Abstand der einzelnen Gelenkwinkelstellungen zu den Verfahrgrenzen überprüft werden und schließlich die Stellung mit der größten Bewegungsfreiheit gewählt werden. Der Abstand des TCP selbst zu den Grenzen des Roboterarbeitsraums kann als zusätzliches Kriterium für die Auswahl einer geeigneten Greifkonfiguration herangezogen werden, da bei einer TCP-Position nahe der Grenzen die

Möglichkeiten der Feinbewegungen zum Erreichen und Verlassen der Greifstellung des Roboters stark eingeschränkt sind. Deshalb bestimmt letztendlich der verbleibende Verfahrbereich der Roboterachsen von der Greifposition aus zusammen mit dem Abstand der Roboterposition von der Arbeitsraumbegrenzung und der bereits erfolgten Stabilitätsbewertung die Einordnung der Greifkonfiguration. Die Greifkonfiguration die bei allen drei Kriterien am besten abschneidet, wird zunächst bei den anschließenden Planungsschritten weiterverfolgt. Die anderen Konfigurationen werden jedoch für den Fall beibehalten, daß mit der betrachteten Greifkonfiguration keine erfolgreiche Bewegungsplanung durchgeführt werden kann. Zuerst werden dabei die Feinbewegungen geplant.

5.2 Planung kollisionsfreier Feinbewegungen

Für die Planung dieser Feinbewegungen ist ein geeignetes Planungsmodul erforderlich. In der Umgebung der Greifposition sind in der Regel einige weitere Zellenkomponenten wie z. B. Magazine oder Montagebetriebsmittel angeordnet. D. h. die Dichte der Hindernisse, mit denen der Greifer oder das zu greifende Objekt bei einer Bewegung kollidieren können, ist sehr hoch. Sie müssen deshalb sehr genau bewegt werden, was mit entsprechenden Feinbewegungen erfolgt. Neben den Restriktionen, die durch den Arbeitsraum des Roboters und seine Bewegungsmöglichkeiten gegeben sind, sind dabei vor allem die Geometrie des Objekts und des Greifers sowie der Hindernisse in der unmittelbaren Umgebung ausschlaggebend. Ein sperriges Objekt ist wesentlich schwerer kollisionsfrei durch eine dichte Ansammlung von Hindernissen zu bewegen als ein kompaktes Objekt mit sehr kleinen Abmessungen. Bei derartigen Feinbewegungen ist deshalb zunächst eine Objektbahn zu bestimmen und anschließend aus dieser eine Roboterbahn abzuleiten.

Bei einem Großteil gängiger Pick-And-Place-Bewegungen erfolgt die Feinbewegung als Linearbewegung in vertikaler Richtung, vor allem bei der Verwendung von SCARA-Robotern. In der Montage weit verbreitet sind zudem lineare Fügebewegungen. Fügebewegungen werden bereits bei der Konstruktion indirekt vom Konstrukteur festgelegt. Durch eine explizite Angabe der Fügerichtungen

durch einen Vektor stehen sie für die Gestaltung der Feinbewegung zur Verfügung. Bei der Feinbewegungsplanung wird deshalb zunächst untersucht, ob eine FÜgerichtung explizit angegeben ist, sonst wird die vertikale Richtung als FÜgerichtung angenommen. Dann wird untersucht, ob der Roboter mit Greifer und gegriffenem Objekt so weit in negativer FÜgerichtung von der Greifposition weg bewegt werden kann, daß der Abstand von den Hindernissen in der unmittelbaren Umgebung der Greifposition einen vorgegebenen Mindestwert erreicht. Gleichzeitig ist sicherzustellen, daß bei der Linearbewegung die einzelnen Achsen nicht zu nahe an ihre Grenzen gefahren werden, da es sonst zu Problemen beim späteren Einfahren des Programms auf der realen Anlage kommen kann.

Ist diese einfache Lösung der Gestaltung der Feinbewegung als eine Linearbewegung nicht möglich, so muß eine komplexere Objektbahn mit Hilfe eines automatischen Bahnplanungsverfahrens gesucht und in eine entsprechende Roboterbewegung transformiert werden. Eine hohe Bahngenauigkeit, wie sie bei dieser Aufgabenstellung erforderlich ist, erreicht ein Roboter in der Regel bei Linear- oder Zirkularbewegungen. Bei einer Linearbewegung bewegt sich der TCP des Roboters auf einer Gerade im Weltkoordinatensystem und das Werkzeugkoordinatensystem des Roboters sowie die am Roboter befindlichen Werkzeuge und Werkstücke werden bei einer Änderung der Raumwinkel stetig entlang dieser Gerade verdreht. Zwischen der Ausgangsposition $(x_A^{\text{Roboter}} | y_A^{\text{Roboter}} | z_A^{\text{Roboter}} | \alpha_A^{\text{Roboter}} | \beta_A^{\text{Roboter}} | \gamma_A^{\text{Roboter}})$ und der Zielposition $(x_Z^{\text{Roboter}} | y_Z^{\text{Roboter}} | z_Z^{\text{Roboter}} | \alpha_Z^{\text{Roboter}} | \beta_Z^{\text{Roboter}} | \gamma_Z^{\text{Roboter}})$ wird die Bewegung also linear interpoliert. Es empfiehlt sich deshalb, die Feinbewegung aus mehreren Linearbewegungen des Roboters zusammensetzen. D. h. für die zunächst zu planende Objektbewegung, daß sie aus Teilbewegungen zusammengesetzt sein muß, die durch eine Anfangsposition $(x_A^{\text{Objekt}} | y_A^{\text{Objekt}} | z_A^{\text{Objekt}} | \alpha_A^{\text{Objekt}} | \beta_A^{\text{Objekt}} | \gamma_A^{\text{Objekt}})$ und eine Zielposition $(x_Z^{\text{Objekt}} | y_Z^{\text{Objekt}} | z_Z^{\text{Objekt}} | \alpha_Z^{\text{Objekt}} | \beta_Z^{\text{Objekt}} | \gamma_Z^{\text{Objekt}})$ bestimmt werden, zwischen denen linear interpoliert wird. Ein Beispiel für eine derartige Bewegung ist in Bild 5-9 gegeben, bei dem zwischen den Positionen P_1 und P_2 sowie den Positionen P_2 und P_3 interpoliert wird. Der Ursprung des Objektkoordinatensystems bewegt sich dabei auf der gestrichelt eingezeichneten Bahn.

Die Wahl des Objektkoordinatensystems ist entscheidend für den Verlauf der Bewegung. Enthält die Bewegung einen rotatorischen Anteil, so bewegt sich der Ursprung des Objektkoordinatensystems auf einer Geraden, während alle anderen Punkte jeweils eine nicht-lineare Bahn im Weltkoordinatensystem beschreiben. Im Beispiel läßt sich dies anhand der schwächer eingezeichneten falschen Objektbahn nachvollziehen. Diese geradlinige Bahn für den Objektpunkt, der am weitesten vom Ursprung des gewählten Objektkoordinatensystems entfernt liegt, wird nicht eingehalten. Für die beiden Zwischenpositionen liegt er nicht auf der geradlinigen Verbindung von Anfangs- und Endposition. Die geradlinige Bahn wird nur eingehalten, wenn dieser Objektpunkt als Ursprung des Objektkoordinatensystems definiert wird.

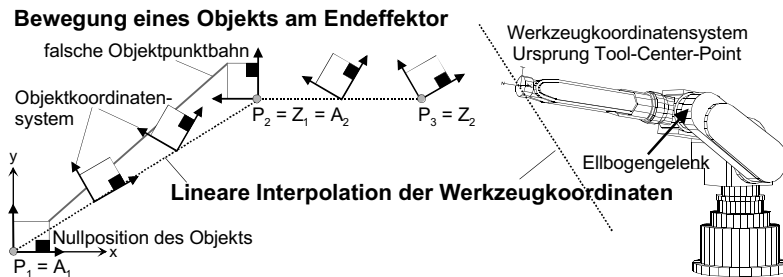


Bild 5-9: Bahnbeschreibung durch lineare Interpolation von Objektpositionen

Um die Objektbewegung in Linearbewegungen des Roboters transformieren zu können, muß das Objekt deshalb zunächst so verändert werden, daß der Ursprung des Objektkoordinatensystems mit dem Ursprung des Werkzeugkoordinatensystems übereinstimmt, wenn das Objekt vom Greifer gehalten wird. Statt das Objektkoordinatensystem anzupassen, könnte man auch mit einer Werkzeugkorrektur das Werkzeugkoordinatensystem des Roboters anpassen. Werkzeugkorrekturen sind jedoch roboterspezifisch, weshalb hier die Anpassung des Objektkoordinatensystems bevorzugt wird.

5.2.1 Anpassung des Objektkoordinatensystems

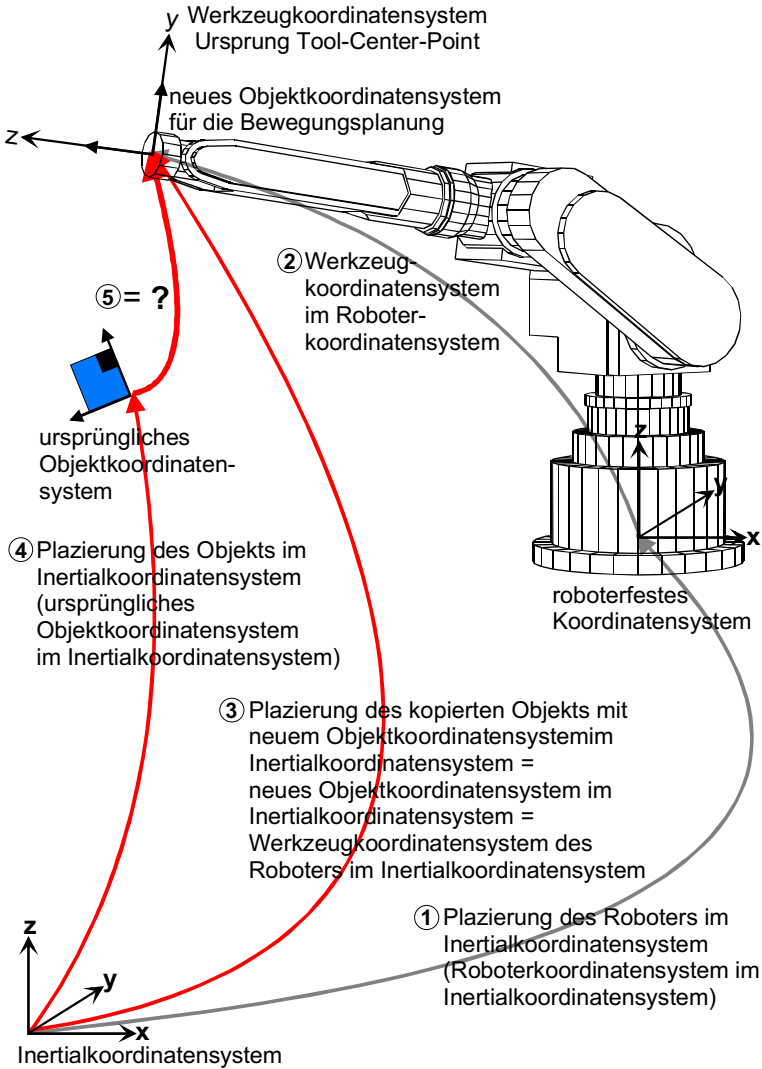


Bild 5-10: Anpassung des Objektkoordinatensystems

Die geometrische Beschreibung von Objekten im Simulationssystem AnySIM/USIS bezieht sich auf ein Objektkoordinatensystem (Kap. 4.1.1). Alle Punkte, aus denen sich die facettierten Objektaußenflächen zusammensetzen, sind in diesem Koordinatensystem angegeben. Zur Platzierung des Objekts im Simulationslayout wird die Lage des Objektkoordinatensystems im Simulationslayout angegeben. Für die Planung der Feinbewegung muß das Objektkoordinatensystem identisch mit dem Werkzeugkoordinatensystem sein. Deshalb wird ein neues Objekt mit dem Werkzeugkoordinatensystem als Objektkoordinatensystem erzeugt, das sich mit dem ursprünglichen Objekt überdeckt. Dazu muß die geometrische Beschreibung des ursprünglichen Objekts im ursprünglichen Koordinatensystem in das neue Objektkoordinatensystem transformiert werden.

Die Bestimmung der Transformation, die mit einem Fragezeichen in Bild 5-10 dargestellt ist, erfolgt über die im Simulationsmodell verfügbaren Transformationsmatrizen. Das neue Objektkoordinatensystem stimmt mit dem Werkzeugkoordinatensystem überein. Die Lage des Werkzeugkoordinatensystems in Inertialkoordinaten ③ läßt sich aus der Roboterstellung, die die Lage des Werkzeugkoordinatensystems in Roboterkoordinaten ② enthält, und der Lage des Roboterkoordinatensystems ① bestimmen. Die Lage des Roboterkoordinatensystems ist durch die Beschreibung der Platzierung des Roboters im Simulationslayout bekannt. Durch die Beschreibung der Platzierung des ursprünglichen Objekts im Simulationslayout ist das ursprüngliche Objektkoordinatensystem in Inertialkoordinaten ④ definiert. Sowohl das neue als auch das ursprüngliche Objektkoordinatensystem können damit als Transformationsmatrix in Inertialkoordinaten dargestellt werden und somit zur Berechnung der Transformation der Geometrie vom ursprünglichen ins neue Koordinatensystem ⑤ genutzt werden. Für den weiteren Verlauf der Bahnplanung wird die neu erstellte Objektkopie verwendet und anschließend wieder gelöscht.

5.2.2 Bestimmung der Objektbahn

Nach der Anpassung des Objektkoordinatensystems erfolgt die Bestimmung der Objektbahn. Um auch Kollisionen des Greifers mit der Umgebung auszuschließen

ßen, wird dieser bereits bei der Bahnsuche an das bewegte Objekt gehängt und so bei der Kollisionsprüfung mitberücksichtigt. Als Zielposition wird die Objektposition gewählt, die sich ergibt, wenn der Tool-Center-Point der Greifposition in das Ellbogengelenk des Roboters (Bild 5-9) verschoben wird. Dieser automatische Vorschlag ist sinnvoll, da beim Entfernen eines Objekts aus der Greifposition die Bewegung Richtung Ellbogengelenk des Roboters in vielen Fällen einer Bewegung weg von den Hindernissen in der Umgebung der Greifposition entspricht. Alternativ kann auch vom Planer eine Zielposition vorgegeben werden.

Für die Planung von Objektbahnen werden häufig Potentialfeldverfahren eingesetzt, die aber oft nur für Objekte mit kompakter Geometrie gute Ergebnisse liefern, während für Objekte mit aufgesetzten Stiften, Stutzen oder ähnlichen Geometriekomponenten keine Bahn mehr gefunden wird. Dies liegt einerseits daran, daß die Objekte meist als punktförmige Geometrie betrachtet werden und die Hindernisse um die Abmaße des Objekts vergrößert werden, was in ungünstigen Fällen zu einer unverhältnismäßig starken Verengung von Durchgängen führt (Bild 5-11).

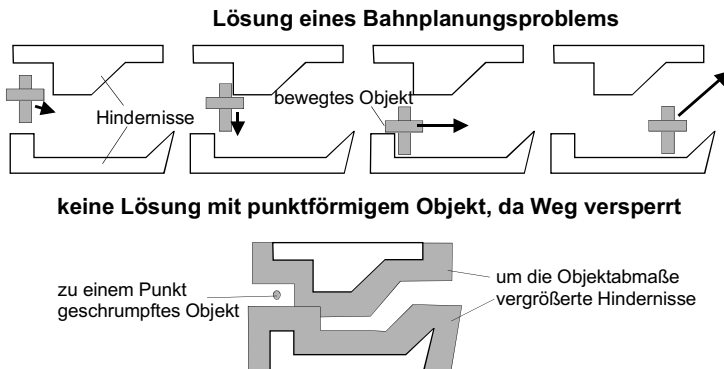
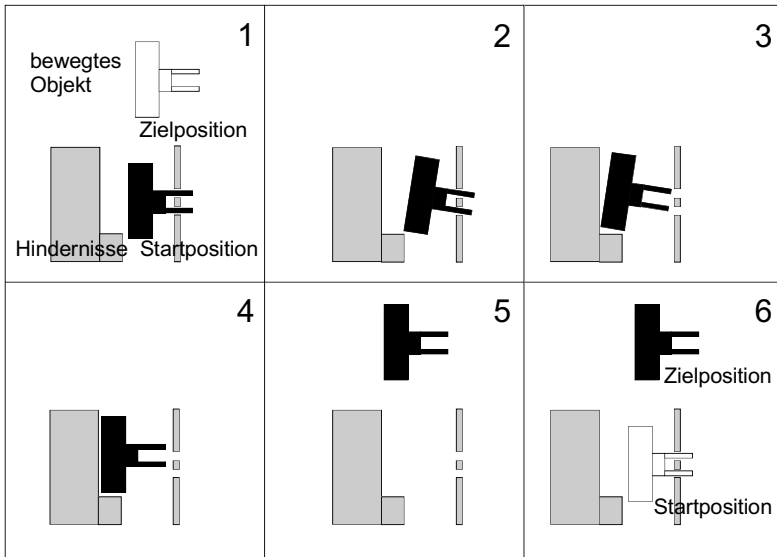


Bild 5-11: Beispiel, das mit punktförmigem Objekt und größeren Ersatzgeometrien für die Hindernisse nicht gelöst werden kann

Lösung



keine Lösung mit Potential wegen fehlender Drehbewegung

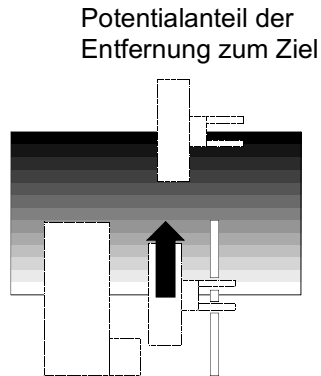
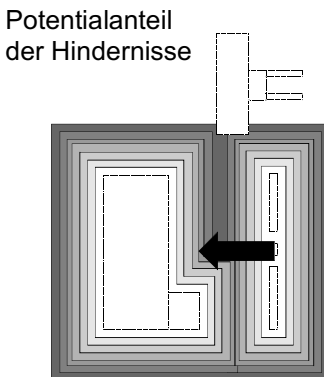


Bild 5-12: Beispiel für eine mit Potentialfeldmethoden nur schwer zu lösende Aufgabe

Wird andererseits auf die Verwendung eines punktförmigen Objektes verzichtet, so liefert die Berechnung des Potentialfeldes oft Ergebnisse, die automatisch nur schwer richtig interpretiert werden können, wie in dem in Bild 5-12 dargestellten Fall.

Dabei muß ein Objekt aus Bohrungen herausgezogen werden und dabei gleichzeitig gekippt werden, um kollisionsfrei in die Zielposition zu gelangen wie oben im Bild dargestellt. Als Potential bietet sich hier die Überlagerung der beiden unten dargestellten Potentialfunktionen an. Das Objekt wird von den dunklen Stellen des Potentials angezogen, von den hellen Stellen abgestoßen. Links ist der Abstand zu den Hindernissen als Potential dargestellt, rechts die Entfernung zum Zielpunkt. Das Objekt soll sich von den Hindernissen wegbewegen und von der Zielposition angezogen werden. Nimmt man den günstigeren Fall an, daß beim Hindernispotential eine Bewegung nach links ermittelt wird, so muß dieser Bewegung die Bewegung nach oben des Potentials überlagert werden, das sich aus der Entfernung zur Zielposition bestimmt. Damit ergibt sich eine Bewegung nach links oben, die aber nicht kollisionsfrei durchgeführt werden kann. Eine Lösung kann nur gefunden werden, falls Objektdrehungen berücksichtigt werden. Dazu müßte aber erst ein Hinderniskonfigurationsraum innerhalb des Bewegungsraums berechnet werden, der neben den drei translatorischen Koordinaten auch alle rotatorischen Stellungen berücksichtigt und für jede Objektstellung angibt, ob diese kollisionsfrei ist oder nicht. Dies ist, wie bereits in Kap. 2 im Zusammenhang mit Bahnplanungsverfahren für Roboter erwähnt, sehr aufwendig.

Deshalb wurde für die Planung der Objektbahn der Bahnplanungsalgorithmus von GLAVINA (1991), der für die Suche einer kollisionsfreien weiträumigen Roboterbewegung entwickelt wurde, auf das Problem der kollisionsfreien Bewegung eines Objektes mit 6 Freiheitsgraden aus einer Hinderniszone hinaus übertragen. GLAVINA & HEIB (1993) beschreiben die Möglichkeit, den für weiträumige Roboterbewegungen entwickelten Algorithmus auf das Problem der Einbausimulation zu übertragen, indem der Roboter selbst bei der Kollisionsbetrachtung nicht berücksichtigt wird und nur als Bewegungssteuerung für das ein- oder auszubauende Objekt dient. Die Bahnsuche kann bei diesem Verfahren jedoch im Bereich von Stunden oder Tagen liegen. Ein Roboter hat im Gegensatz

zu einem einzelnen, frei beweglichen Objekt einen eingeschränkten Bewegungsraum. Deshalb wird im hier vorliegenden Ansatz zuerst eine Objektbahn bestimmt und anschließend in eine Roboterbewegung umgesetzt. Der Systembenutzer kann so erkennen, ob der Ein-/Ausbau selbst ein schwer zu lösendes Problem ist oder ob die Lösung durch den gewählten Griff oder die Kinematik des Roboters erschwert bzw. eingeschränkt wird. Außerdem ist die bei GLAVINA (1991), GLAVINA & HEIß (1993) ermittelte Roboterbewegung mathematisch idealisiert und deshalb mit den Standardbewegungsmodi eines Roboters nicht auszuführen (vgl. Kap. 2, S. 23 und 24, Kap. 5.3).

Bei GLAVINA (1991) werden einzelne Roboterstellungen durch die Gelenkwinkel des Roboters dargestellt. Einzelne Objektpositionen werden im hier gewählten Ansatz durch drei translatorische Koordinaten x , y , z und drei Raumwinkel α , β , γ beschrieben. Eine mit dem Verfahren ermittelte Bahn setzt sich analog zu dem für Roboter konzipierten Verfahren aus Teilbahnen zusammen, die mit einer bestimmten Objektposition beginnen und in einer bestimmten Objektposition enden. Die eigentliche Teilbahn besteht aus einer linearen Interpolation von der Anfangs- zur Endposition. Bei der Bahnsuche wird dabei zunächst die lineare Verbindung der Start- und der Zielposition des bewegten Objekts überprüft.

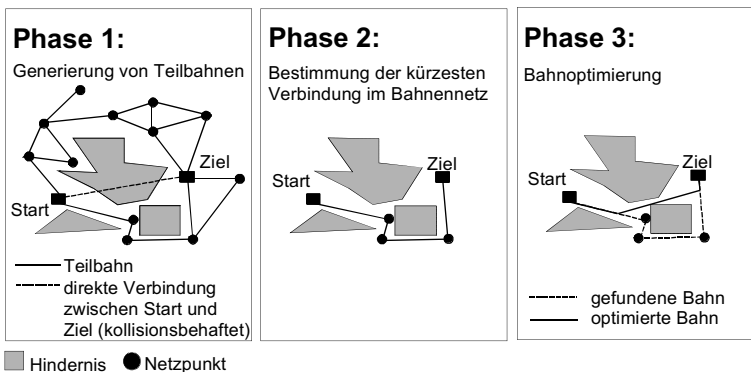


Bild 5-13: Bahnsuche

Ist diese kollisionsbehaftet, so werden weitere Objektpositionen und die mit anderen Objektpositionen begrenzten linearen Teilbahnen bzgl. Kollision überprüft. Schließlich erhält man ein zusammenhängendes Netz, das sowohl die Start- als auch die Zielposition des bewegten Objekts enthält (Bild 5-13). In diesem Netz wird dann die kürzeste Objektbahn extrahiert. Das Netz ist ein Graph, dessen Knotenpunkte den Graphknoten entsprechen und dessen Teilbahnen die Kanten eines Graphen repräsentieren (STETTER 1993). Die Länge der Teilbahn ergibt sich aus dem Betrag des die Teilbahn darstellenden Vektors

$$\vec{x}_{\text{Teilbahn}} = \vec{x}_{\text{Endposition}} - \vec{x}_{\text{Anfangsposition}} \quad \text{mit } \vec{x}_i = (x \mid y \mid z \mid \alpha \mid \beta \mid \gamma).$$

Mit der Länge der Teilbahn erhält der Graph eine Gewichtung. So kann die kürzeste Bahn schließlich mit Methoden der Graphentheorie (DOMSCHKE 1972, SEDGEWICK 1992) zur Suche des kürzesten Weges in einem gewichteten Graphen ermittelt werden. Die ermittelte kürzeste Bahn kann anschließend noch geglättet bzw. optimiert werden. Dazu werden die Teilbahnen, aus denen sich die Lösung zusammensetzt, durch Zwischenpositionen diskretisiert. Durch die Untersuchung möglicher Verbindungen zwischen Zwischenpositionen unterschiedlicher Teilbahnen wird versucht, die Ergebnisbahn weiter zu verkürzen.

Alle Netzknotenpunkte bzw. Objektpositionen der endgültigen Bahn müssen sich jedoch innerhalb des Arbeitsraums des Roboters befinden. Das Einhalten dieser Bedingung auf der gesamten Bahn muß zum Schluß der Planung noch einmal verifiziert werden und u. U. muß eine Bahnanpassung durchgeführt werden.

Zur Beginn der Bahnsuche stehen nur die Start- und die Zielposition des Objekts als Netzknotenpunkte zur Verfügung. Zur Erzeugung neuer Netzknotenpunkte und neuer von Netzknotenpunkten begrenzter Teilbahnen werden verschiedene Methoden angewandt. Bei der Überprüfung der Verbindungen zweier Netzknotenpunkte bzgl. Kollision werden gleichzeitig neue Positionen als Punkte in diesem Bahnnetz erzeugt. Ist z. B. die geradlinige Verbindung zweier Punkte kollisionsbehaftet, ein Geradenstück bei einem der beiden Verbindungsendpunkte aber kollisionsfrei, so wird dieses Geradenstück mit seinem zweiten Endpunkt als neuem Netzknotenpunkt in das Bahnnetz integriert. So lange die beiden Teilnetze mit der Start- und mit der Zielposition bei der Bahnsuche noch nicht zu-

sammenhängen, werden bei weiterem Bedarf an Positionen auch zufällig neue Netzknotenpunkte oder Zwischenpositionen erzeugt.

Ebenso generieren das im folgenden beschriebene Gleitverfahren von GLAVINA (1991) und weitere Algorithmen zur geometrischen Analyse der Einbausituation geeignete Zwischenpositionen. Das Gleitverfahren von GLAVINA (1991) wird zur lokalen Umgehung von Hindernissen eingesetzt .

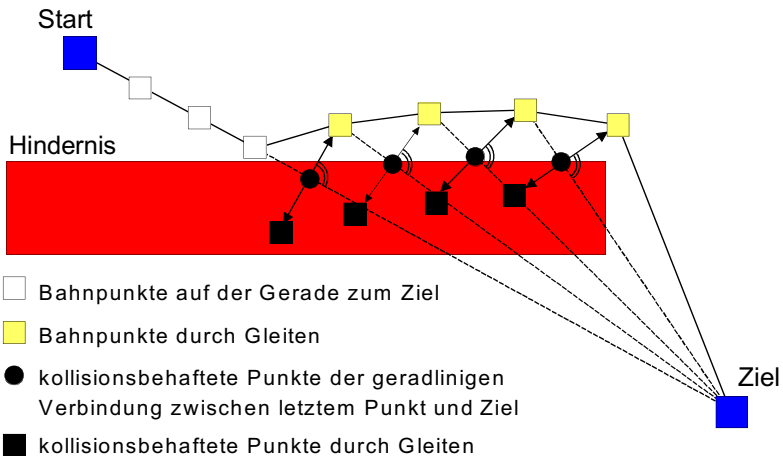


Bild 5-14: Das Gleitverfahren von GLAVINA (1991)

Bei diesem in Bild 5-14 dargestellten Gleitverfahren wird so lange auf der Gerade zum Ziel verfahren, bis eine Kollision auftritt. Zur Kollisionsprüfung wird die Gerade diskretisiert. Durch einen Seitenschritt in zu einer zu der Geraden orthogonalen Richtung wird ausgehend von dem kollisionsbehafteten Punkt eine neue Position erzeugt. Ist sowohl diese Position als auch ihre Verbindung zum letzten kollisionsfreien Punkt der bisherigen Bahn kollisionsfrei, so wird diese Position als neuer Bahnpunkt verwendet. Von diesem aus wird wieder geradlinig aufs Ziel zugegangen und evtl. ein weiterer Gleitpunkt erzeugt.

Der Lösungsraum wird bei diesem auf dem Verfahren von GLAVINA (1991) basierenden Verfahren im Gegensatz zur Hindernisvergrößerung beim Potentialfeldverfahren nicht eingeschränkt, jedoch weist die reine Übertragung des Bahnplanungsverfahrens von GLAVINA (1991) auf den Fall der Bewegung eines Starrkörpers gegenüber Potentialfeldverfahren den Nachteil auf, daß u. U. viele Geradenstücke getestet werden, die für eine Bahn nicht in Frage kommen, da die Bewegung nicht durch ein Potential zum Ziel hin gelenkt wird.

Das übertragene Verfahren wird deshalb durch spezielle Analysealgorithmen zur Beschleunigung der Bahnfindung ergänzt. Dabei werden Verhaltensmuster des Planers beim manuellen Bewegen des Objekts aus seiner Startlage in Automatismen übertragen. Löst ein Werker ein Objekt aus einer Baugruppe, so wird er zunächst versuchen, die Verbindungsstellen zu lokalisieren, z. B. Bolzen am zu lösenden Objekt und zugehörige Bohrungen an anderen Objekten der Baugruppe. Bis diese Verbindungselemente weit genug von der Baugruppe entfernt sind, sind die Bewegungsmöglichkeiten für das zu lösende Objekt meist sehr eingeschränkt. Bei der automatischen Bahnplanung werden deshalb zunächst die Geometrien der Hindernisse in unmittelbarer Nähe des zu bewegendes Objekts untersucht.

Besitzt ein Objekt eine Aussparung wie z. B. eine Bohrung, so ist die Oberfläche für die Aussparung durchbrochen. Bei einer Bohrung enthält die Oberfläche ein kreisförmiges Loch. Die aktuelle Oberfläche kann also durch Subtraktion einer Kreisfläche von der ursprünglichen Fläche ohne Bohrung erzeugt werden. Derartige Subtraktionsflächen werden im folgenden als Lochflächen bezeichnet. Zur Untersuchung der Geometrie der Hindernisse in unmittelbarer Nähe des zu bewegendes Objekts werden im ersten Schritt Lochflächen in den Hindernisoberflächen bestimmt und der Abstand des zu bewegendes Objekts zum Rand der Lochfläche berechnet. Ist dieser Abstand wesentlich kleiner als die Abmaße der Lochfläche, so muß untersucht werden, ob ein Teil des Objekts in dem Loch steckt, was einfach durch einen Schnitt der Lochfläche mit der Geometrie des zu bewegendes Objekts ermittelt werden kann.

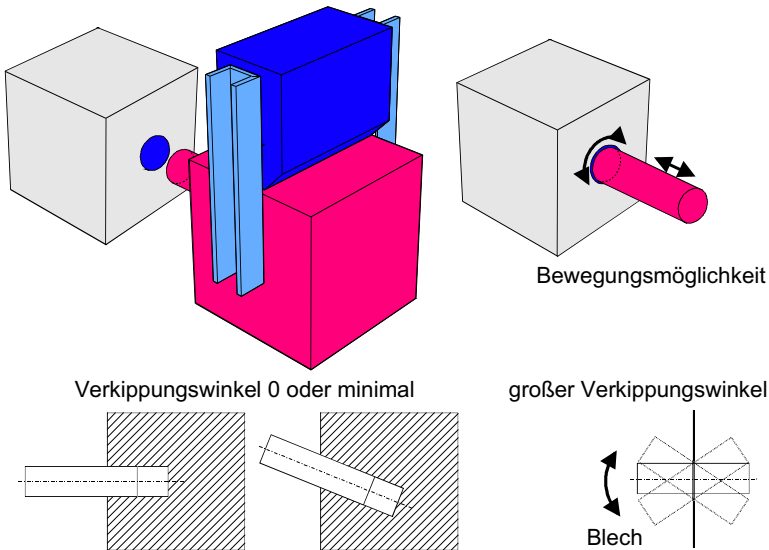


Bild 5-15: Bestimmung der Bewegungsfreiheit bei Geometrieelementen, die in Hindernissen stecken

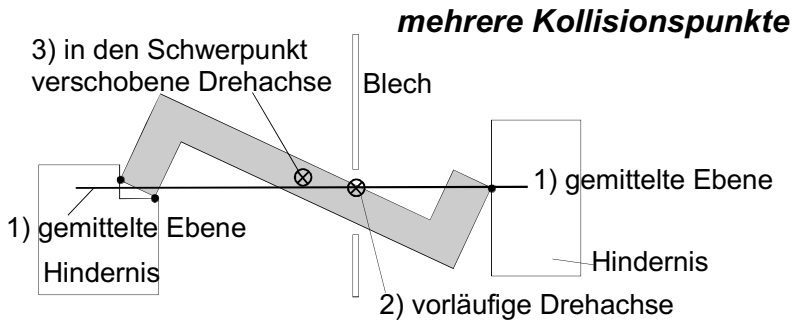
Im nächsten Schritt muß bestimmt werden, auf welcher Seite der Lochfläche das Objekt wegbewegt werden kann. Da das Objekt später vom Roboter und seinem Greifer wegbewegt wird, ist dies die Seite der Lochfläche, an der sich die Angriffspunkte des Greifers am Objekt befinden (Bild 5-15). Damit ist die Bewegungsmöglichkeit bereits eingeschränkt auf den Normalenvektor der Lochfläche in Richtung der zuvor bestimmten Objektseite. Dieser Vektor greift im Schwerpunkt der Lochfläche an und kann noch um diesen Schwerpunkt gekippt werden. Zur Bestimmung der zulässigen Verkippung wird im nächsten Schritt untersucht, ob es sich bei der Lochfläche um eine Aussparung in einem Blech handelt oder um den Beginn einer dreidimensionalen Aussparung, wie z. B. einer Bohrung oder einer Nut. Eine derartige dreidimensionale Aussparung ist in der Regel ein Prisma mit der Lochfläche als Grundfläche. Der mittlere Winkel, den die Kanten des Prismas im Material, die an der Lochfläche beginnen, mit der Lochfläche einschließen, bestimmt die Verkippung des zuvor bestimmten Richtungsvektors.

Neben der translatorischen Bewegung in Richtung des Richtungsvektors ist nun auch eine Verkipfung im Bereich des Verkipfungswinkels möglich. Handelt es sich bei dem Hindernis um ein Blech, so wird der zulässige Verkipfungswinkel auf $\pm 40^\circ$ festgelegt, da sich das Objekt bei günstiger Gestaltung um fast 45° gegenüber der Ebene der Lochfläche verkippen läßt.

Nach der Bestimmung der Bewegungsrichtung und des zulässigen Verkipfungswinkels wird das Objekt zunächst in diese Richtung bewegt, bis eine Kollision auftritt. Ist der zulässige Verkipfungswinkel ungleich Null, so wird getestet, ob durch eine translatorische Bewegung in die zuvor bestimmte Bewegungsrichtung und eine Drehung im zulässigen Verkipfungsbereich das Objekt wegbewegt werden kann.

Die Drehachse bestimmt sich dabei durch eine erweiterte Analyse des letzten Kollisionsfalls. Bei mehreren Kollisionspunkten werden diese durch eine Ebene gemittelt und die vorläufige Drehachse durch einen Schnitt dieser Ebene mit der Ebene der Lochfläche ermittelt (Bild 5-16 oben). Die Drehachse wird anschließend in den Schwerpunkt des Objekts parallel verschoben. Im Falle eines einzelnen Kollisionspunktes wird dieser mit dem Schwerpunkt der Lochfläche verbunden, das Lot auf die Ebene der Lochfläche gefällt und als Richtung die Normalenrichtung zu diesem Lot gewählt (Bild 5-16 unten), die in der Ebene der Lochfläche liegt. Diese Achse wird dann noch in den Objektschwerpunkt verschoben. Die Drehrichtung bestimmt sich entweder durch die Lage der Kollisionspunkte oder sie muß bei einer ungünstigen Punktverteilung durch Austesten ermittelt werden.

Läßt sich nicht wie bei einem Bolzen, der in einer Bohrung steckt, der Engpaß in der Ausgangs- bzw. der Endlage des Objekts bestimmen, so werden sog. Nahpositionen erzeugt, die durch translatorische Bewegung des Objekts entlang einer der Achsen des Werkzeugkoordinatensystems des Roboters für diese Lage oder durch eine Rotation um eine Achse dieses Koordinatensystems erzeugt werden. Dadurch wird das Objekt von seiner Ausgangslage wegbewegt und zumindest für eine der neuen Positionen mit großer Wahrscheinlichkeit auch von den Hindernissen, mit denen es sich verklemmen könnte, wegbewegt.



- Kollisionspunkte

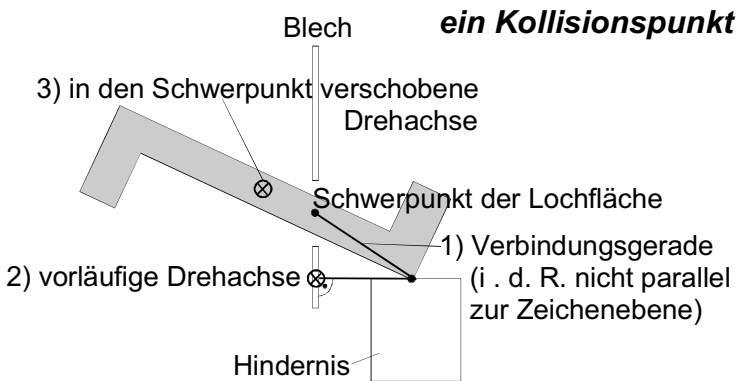


Bild 5-16: Bestimmung der Drehachse durch Kollisionspunkte

Die Analysestrategien, das Gleiten und die zufällige Erzeugung von Objektpositionen führen schließlich zu einer Start-Ziel-Verbindung im Teilbahnnetz. Diese Verbindung wird optimiert, bzgl. Übereinstimmung mit dem Roboterarbeitsraum überprüft und gegebenenfalls noch einmal angepaßt.

5.2.3 Transformation in eine Roboterbewegung und Anpassung

Die erzeugte Objektbahn muß schließlich in eine Roboterbewegung umgesetzt werden. Die Objektbahn besteht aus n Teilbahnen, die von je zwei Objektpositionen begrenzt werden, zwischen denen linear interpoliert wird. Die zugehörige Roboterbewegung setzt sich deshalb aus n Linearbewegungen zusammen, die von entsprechenden Roboterpositionen begrenzt sind. Für die Anfangsposition der Feinbewegung wurde die zugehörige Roboterstellung bereits bei der Greifplanung bestimmt. In dem in Bild 5-17 gezeigten Beispiel besteht die Objektbahn aus zwei Teilbahnen. Position ① ist bereits aus der Greifplanung bekannt. Für das Anfahren der Positionen ② und ③ durch den Roboter mit einer Linearbewegung müssen die zugehörigen Roboterpositionen bestimmt werden.

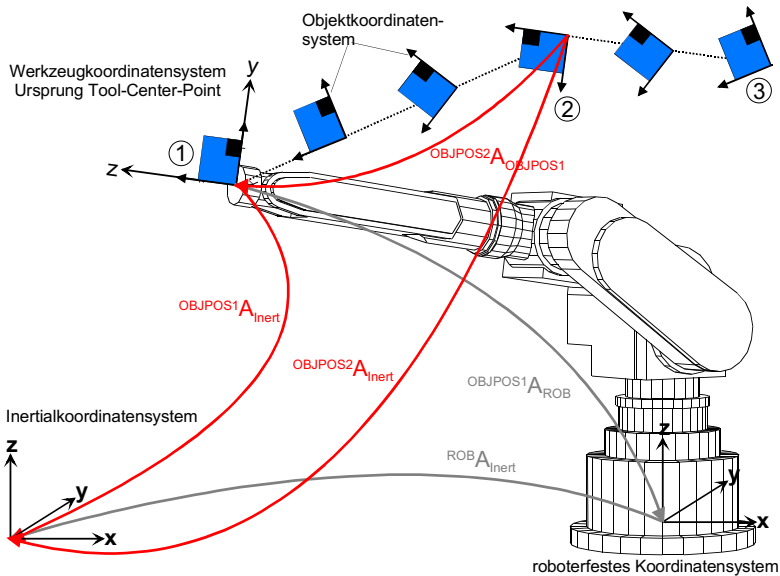


Bild 5-17: Transformation der Objektbewegung in eine lineare Roboterbewegung

Es gilt ${}^{OBJPOS2}A_{Inert} = {}^{OBJPOS2}A_{OBJPOS1} {}^{OBJPOS1}A_{Inert}$ bzw. für die Objektposition in Inertialkoordinaten $\vec{x}_{OBJPOS2} = {}^{OBJPOS1}A_{OBJPOS2} \cdot \vec{x}_{OBJPOS1}$. Die Matrix ${}^{OBJPOS2}A_{OBJPOS1}$ läßt sich deshalb aus den Positionen ① und ② des Objekts bestimmen. Sie beschreibt ebenfalls den Zusammenhang zwischen den Roboterpositionen ① und ② für ihre Darstellung in Inertialkoordinaten ($\vec{x}_{ROBPOS2} = {}^{ROBPOS1}A_{ROBPOS2} \cdot \vec{x}_{ROBPOS1}$) und wird deshalb zur Berechnung der Roboterposition ② aus der Roboterposition ① verwendet. Nach der Transformation der gesamten Roboterbahn muß die Roboterbewegung noch einmal ausgeführt werden, um einerseits Kollisionen des Roboters und andererseits unerlaubte Stellungswechsel aufzudecken. Außerdem wird dabei mit einer vergrößerten Objektgeometrie für Greifer und Objekt (wie bei der Greifplanung jedoch mit einem zusätzlichen geringen Aufschlag für die Bahnengenauigkeit des Roboters) gearbeitet, um Engpässe zu detektieren. Tritt ein unerlaubter Stellungswechsel oder eine Kollision des Roboters auf, so muß überprüft werden, ob die Bewegung des Objekts mit einem anderen Stellungsparameter ausgeführt werden kann. Bei einer Kollision, an der der Greifer oder das bewegte Objekt beteiligt sind, muß der Planer auf diesen Engpaß durch die übergeordnete Handhabungsplanung hingewiesen werden. Eine Umplanung beim Auftreten einer derartigen Kollision ist nicht sinnvoll, da gerade bei Fügebewegungen oft nur sehr enge Bewegungsräume zur Verfügung stehen, so daß schließlich die reale Zelle sehr genau eingerichtet werden muß und vielleicht sogar eine entsprechende Fügeensorik eingesetzt werden muß. Zum Teil können derartige Fügevorgänge mit vermeintlichen starken Kollisionen aber auch mit komplizierten Greifwerkzeugen sicher durchgeführt werden.

Nach der Planung der Feinbewegungen muß noch sichergestellt werden, daß auch die weiträumigen Roboterbewegungen zum Transfer zwischen den Feinbewegungen kollisionsfrei ausgeführt werden können. Dies wird mit einer automatischen Bahnplanung für den Roboter gewährleistet.

5.3 Bahnplanung für weiträumige Transferbewegungen

Die meisten in der Literatur angegebenen Bahnplanungsverfahren berechnen eine kollisionsfreie mathematische Bahn für den Roboter ohne zu berücksichtigen, ob diese Bahn wirklich in der angegebenen Form abgefahren werden kann. Die Regelung der einzelnen Achsen des Roboters kann dies nicht für jeden Fall gewährleisten wie in Kapitel 2 beschrieben.

Bei der Roboterprogrammierung werden für weiträumige Bewegungen vorzugsweise PTP-Bewegungen verwendet, da sie im Vergleich zu bahngesteuerten Bewegungen schneller vom Roboter ausgeführt werden können. Bei der Bahnplanung sollte deshalb im Idealfall eine Bahn bestimmt werden, die sich aus PTP-Bewegungen des Roboters zusammensetzt.

Die PTP-Bewegung ähnelt der Geraden im Gelenkwinkelraum, beinhaltet jedoch je nach Konstellation der von den einzelnen Achsen zurückzulegenden Wege erhebliche Abweichungen von der der Geraden entsprechenden Bahn, wie bereits in Kapitel 2 (Stand der Technik) erwähnt. Andererseits ist die Gerade im Gelenkwinkelraum eine Bewegungsbahn, die sich sehr gut zur Bahnsuche eignet und deshalb auch dem hier eingesetzten Bahnsuchalgorithmus zugrunde liegt. Nach der Vorstellung der für die Offline-Programmierung weiträumiger Roboterbewegungen eingesetzten Bahnplanung wird deshalb der Unterschied zwischen einer Bewegung auf der Gerade im Gelenkwinkelraum und einer PTP-Bewegung untersucht. Die Analyse der Bahnabweichung in Abhängigkeit des Fahrweges und der eingestellten Geschwindigkeitsparameter erfolgt in Kap. 5.3.2. Die maximalen Winkelabweichungen werden in Kap. 5.3.3 zur Auslegung von Hüllkörpern genutzt, mit denen überprüft werden kann, ob eine Bewegung trotz Abweichungen kollisionsfrei ist.

5.3.1 Bahnsuche

Für die Bahnplanung in veränderlichen Umgebungen eignet sich ein Planungsverfahren, das auf dem Verfahren von GLAVINA (1991) (s. auch Kap. 2.1.2.3) aufbaut. Die kürzeste Bewegung eines Roboters zwischen zwei Positionen ent-

spricht ihrer geradlinigen Verbindung in dem durch seine Achswinkel aufgespannten Gelenkwinkelraum \mathfrak{R}^n . Bei diesem Verfahren wird zunächst die direkte, geradlinige Verbindung zwischen der vorgegebenen Start- und Zielstellung des Roboters hinsichtlich Kollision getestet. Dazu wird diese Strecke im \mathfrak{R}^n durch Zwischenpositionen diskretisiert und an jeder Zwischenposition ein Kollisionstest durchgeführt. Tritt auf der gesamten Strecke keine Kollision auf, so ist die Bahnplanungsaufgabe bereits gelöst. Im Falle einer Kollision wird an der vorletzten kollisionsfreien Zwischenposition auf der Strecke ein Stützpunkt definiert und das Geradenstück zwischen der Startposition und dem Stützpunkt (Bild 5-18) für die weitere Bahnsuche gespeichert.

Zur Erzeugung eines weiteren Stützpunktes auf diese Weise wird zusätzlich von der Zielposition aus zur Startposition verfahren, bis Kollision auftritt. Durch die zufällige Erzeugung weiterer Stützpunkte entsteht so schließlich ein Bahnnetz, (wie bei der Planung der Feinbewegung in Kapitel 5.2.1) das zum Schluß eine Start-Zielverbindung enthält, die aus den Teilbahnen kombiniert werden kann.

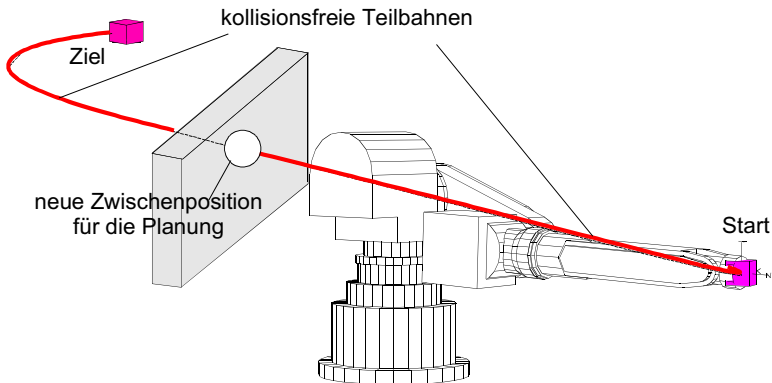


Bild 5-18: kollisionsfreie Geradenstücke zum Aufbau eines Netzes zur Bahnsuche

Neben dem Test geradliniger Verbindungen wird als weiteres Verfahren das Gleiten von GLAVINA (1991) eingesetzt, mit dem eine Bewegung an der Oberfläche eines Hindernisses entlang erzeugt wird. Dazu wird ein Seitschritt orthogonal zur bisherigen Bewegungsrichtung im \mathfrak{R}^n ausgeführt und erneut auf den gewünschten Endpunkt zugegangen. Tritt wieder eine Kollision auf, wird ein weiterer Gleitpunkt durch einen weiteren Seitschritt erzeugt (s. Kap. 5.2.1, Bild 5-14)

Beim Gleiten handelt es sich um ein lokales Suchverfahren, das nur bei konvexen Hindernissen funktioniert. Es muß deshalb nach einer bestimmten Anzahl von Versuchen abgebrochen werden, um nicht eine Endlosbewegung hervorzurufen. Für Hindernisse mit größerer Ausdehnung wurde das Gleiten deshalb mit einem sog. Ausweichen kombiniert (STETTER 1993), bei dem bei Abbruch des Gleitens einen Schritt in Bewegungsrichtung zurückgegangen wird und dann so lange orthogonal zu dieser Bewegungsrichtung verfahren wird, bis eine geradlinige, kollisionsfreie Verbindung zum Endpunkt möglich ist (Bild 5-19).

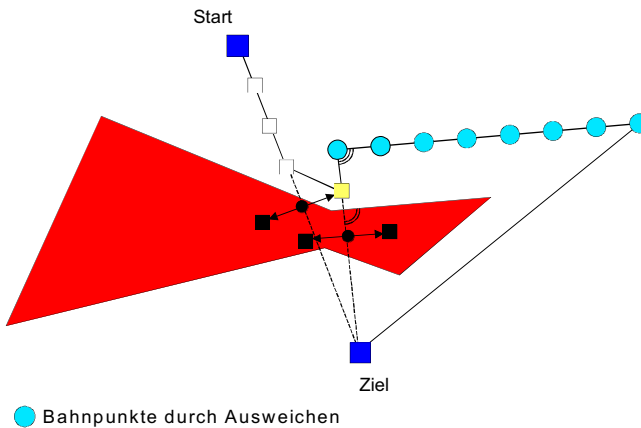


Bild 5-19: Ausweichen

Außerdem wurden wie bei STETTER (1993) beschrieben noch weitere Strategien zur Erzeugung von Netzstützpunkten entwickelt, da die rein zufällige Erzeugung von Stützpunkten oft nur nach sehr langer Zeit eine Start-Ziel-Bahn findet.

Für kleine Winkelwege und niedrige Geschwindigkeiten können die Geradenstücke zur Bewegungsausführung durch den realen Roboter einfach in PTP-Bewegungen des Roboters zwischen Anfangs- und Endposition umgesetzt werden, da dann die Abweichungen zwischen der Gerade im Gelenkwinkelraum und der tatsächlichen PTP-Bewegung relativ gering ist. In der Regel ist jedoch die PTP-Bewegung nicht zwingend kollisionsfrei, wenn dies für die Bewegung auf der Gerade im \mathfrak{R}^n zutrifft. Die maximalen Unterschiede zwischen beiden Bewegungsbahnen werden im folgenden untersucht. Dazu werden zunächst die Eigenschaften von PTP-Bewegungen betrachtet.

5.3.2 Modell der PTP-Bewegung für die Bahnplanung und Kollisionskompensation

Die Geschwindigkeitsprofile für PTP-Bewegungen vieler gängiger Roboter können durch Rampenprofile angenähert werden. Die Synchronisation der einzelnen Achsen erfolgt meist nach dem bei (WANNER 1989) als „ZEITGRENZEN 1“ bezeichnetem Modell, bei dem die Geschwindigkeiten aller Achsen an die der langsamsten Achse angepaßt werden. Die langsamste Achse wird deshalb auch als Führungssachse bezeichnet.

Zur Bestimmung der Führungssachse muß zunächst für jede Achse die mindestens für die Bewegung benötigte Zeit ermittelt werden, die von der Beschleunigung und der maximal zulässigen Geschwindigkeit abhängt. Die maximal zulässige Geschwindigkeit ist der vom Roboterprogrammierer bzw. -bediener eingestellte Prozentsatz p der maximalen Nenngeschwindigkeit v_i für alle Achsen $i = 1, 2, \dots, n$ bei der PTP-Bewegung. Die maximal zulässige Geschwindigkeit für Achse i beträgt deshalb $p \cdot v_i$. Diese Geschwindigkeit kann bei der Bewegung nur erreicht werden, wenn der zurückzulegende Winkelweg q_i größer als $(p^2 v_i^2)/A_i$ ist. A_i ist der Betrag der positiven und negativen Beschleunigung der Achse, die hier als

5 Handhabungsplanung

gleich vorausgesetzt wird. Ist der Weg kleiner, so berechnet sich die benötigte Zeit t_{ei} zu

$$t_{ei} = 2\sqrt{\frac{q_{ie}}{A_i}} \text{ für } q_i < \frac{p^2 v_i^2}{A_i}, \text{ sonst zu } t_{ei} = \frac{q_{ie}}{pv_i} + \frac{pv_i}{A_i}.$$

Die Achse mit der größten Zeit $t_{ei} = t_e$ ist die Führungsachse. Bei allen anderen Achsen berechnet sich dann die maximale Geschwindigkeit für die der Führungsachse folgende Bewegung zu

$$v_i' = \frac{A_i}{2} (t_e - \sqrt{t_e^2 - \frac{4q_{ie}}{A_i}}).$$

Die Achsen beschleunigen bis $t_{i1} = v_i' / A_i$, verfahren dann mit konstanter Geschwindigkeit und bremsen ab $t = t_e - t_{i1}$ wieder ab (Bild 5-20).

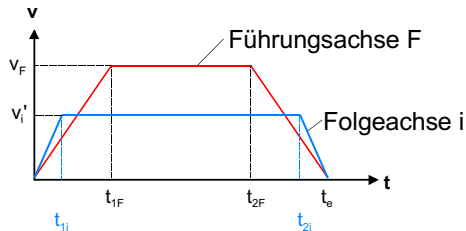


Bild 5-20: Geschwindigkeitsprofil bei der PTP-Bewegung im Fall ZEITGRENZEN I nach WANNER (1989)

Nur für den Fall, daß alle t_{i1} identisch wären, würde die Bewegung wirklich der Geraden im Gelenkwinkelraum entsprechen. Dann hätte jede Achse zu jeder Zeit den gleichen Anteil ihres Gesamtweges zurückgelegt. Für diesen Fall würde gelten

$$q_i = \frac{q_{ie}}{q_{je}} q_j = kq_j.$$

5.3 Bahnplanung für weiträumige Transferbewegungen

Bei den hier betrachteten Bewegungen, bei denen nur die Geschwindigkeit, nicht aber die Beschleunigung an die Bewegung der Führungsachse angepasst wird, sind die Zeitpunkte t_{1i} in der Regel jedoch für alle Achsen unterschiedlich und es kommt zu einer Abweichung

$$\Delta q_i = q_i - kq_j.$$

Betrachtet man zwei aufeinanderfolgende Achsen j und i mit $i = j+1$, so sind für die Abweichungen Δq_i zunächst die zwei Fälle $k < A_i / A_j$ und $k > A_i / A_j$ zu unterscheiden (Bild 5-21).

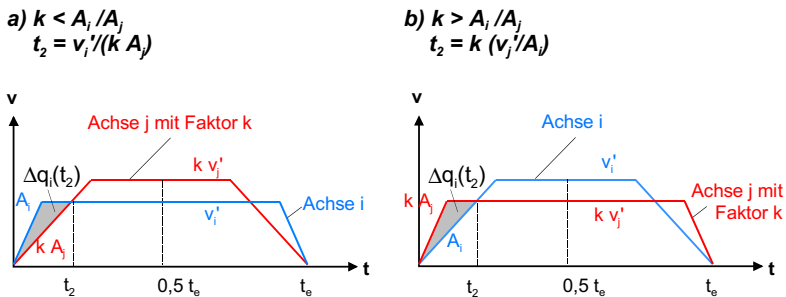


Bild 5-21: Geschwindigkeitsprofile der Achse i für das Rampenprofil (Achse i) und die Bewegung auf der Gerade im Gelenkwinkelraum (Achse j mit Faktor k) für die beiden Fälle a) $k < A_i / A_j$ und b) $k > A_i / A_j$

Für $k > A_i / A_j$ ist $\Delta q_i = q_i - kq_j$ zunächst negativ und erreicht sein Minimum bei $t_2 = k v_j' / A_i$. Der Betrag des Wertes von Δq_i an der Stelle t_2 entspricht der grauen Fläche im Bild. Δq_i steigt nach t_2 wieder an mit einem Nulldurchgang bei $t_e/2$, bis es bei $t_e - t_2$ den Wert $-\Delta q_i(t_2)$, seinen Maximalwert, erreicht. Danach fällt es wieder bis auf den Wert 0 an der Stelle t_e . Da sich durch die vereinfachende Annahme, die Abbremsbeschleunigung sei $-A_i$, ein zu $t_e/2$ symmetrischer Verlauf des Geschwindigkeitsprofils ergibt, ist für weitere Berechnungen die Betrachtung des Verlaufs bis $t_e/2$ ausreichend. Gleiches gilt für den Fall $k < A_i / A_j$, bei dem Δq_i jedoch zunächst positives Vorzeichen hat und bis auf sein Maximum bei $t_2 = v_i' / (k A_j)$ steigt, dann bis zu seinem Minimum bei $t_e - t_2$ sinkt und schließlich wie-

der bis auf den Wert 0 ansteigt. Für die beiden Fälle ergeben sich damit folgende Randbedingungen und Berechnungsformeln für den Betrag des Extremwerts von Δq_i :

a) $k < \frac{A_i}{A_j}$; $t_2 = \frac{kv_j'}{A_i}$;

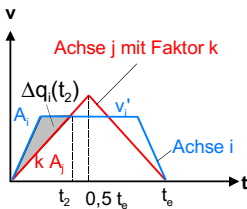
$$|\Delta q_i|_{\max} = q_i - kq_j = \frac{A_i}{2} \left(\frac{v_i'}{A_i} \right)^2 + v_i' \left(t_2 - \frac{v_i'}{A_i} \right) - \frac{kA_j}{2} t_2^2 = \frac{v_i'^2}{2} \left(\frac{1}{kA_j} - \frac{1}{A_i} \right)$$

b) $k > \frac{A_i}{A_j}$; $t_2 = \frac{v_i'}{kA_j}$;

$$|\Delta q_i|_{\max} = kq_j - q_i = \frac{kA_j}{2} \left(\frac{v_j'}{A_j} \right)^2 + kv_j' \left(t_2 - \frac{v_j'}{A_j} \right) - \frac{A_i}{2} t_2^2 = \frac{kv_j'^2}{2} \left(\frac{k}{A_i} - \frac{1}{A_j} \right)$$

mit $k = \frac{q_{ie}}{q_{je}}$, $v_{i/j}' = \frac{A_{i/j}}{2} \left(t_e - \sqrt{t_e^2 - \frac{4q_{ie/j}}{A_{i/j}}} \right)$ und $t_e = \frac{q_{Fe}}{pv_{Fe}} + \frac{pv_F}{A_F}$ bzw. $t_{ei} = 2\sqrt{\frac{q_{Fe}}{A_F}}$ bei unvollständiger Bewegung. (F ist der Index der Führungsachse.)

a) $k < A_i/A_j$
 $t_2 = v_i'/(kA_j)$



b) $k > A_i/A_j$
 $t_2 = k(v_j'/A_j)$

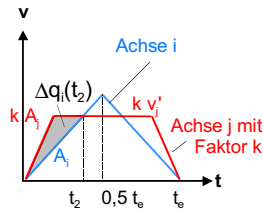


Bild 5-22: Fälle mit unvollständigen Rampenprofilen, in denen die Berechnung von $|\Delta q_i|_{\max}$ analog erfolgt

Diese Werte für $|\Delta q_i|_{\max}$ gelten auch für die in Bild 5-22 gezeigten Fälle a) und b), bei denen jeweils eine der beiden Achsen mit ihrer Geschwindigkeit keine vollständige Rampe beschreibt, da sie keine Bewegung mit konstanter Geschwindigkeit ausführt. Die Berechnung von $|\Delta q_i|_{\max}$ erfolgt wie nach den oben angegebene-

nen Formeln, wenn im Fall a) die Achse i keine Bewegung mit konstanter Geschwindigkeit ausführt bzw. im Fall b) die Achse j keine Bewegung mit konstanter Geschwindigkeit ausführt.

$|\Delta q_i|_{\max}$ hängt also sowohl von den Parametern der beiden aufeinanderfolgenden Achsen j und i als auch von den Parametern der Führungssachse F und dem maximal zulässigen Prozentsatz p der Nenngeschwindigkeit aller Achsen ab.

Es läßt sich jedoch nachweisen, daß einer der beiden Werte $|\Delta q_i|_{\max}(F = i)$ oder $|\Delta q_i|_{\max}(F = j)$ stets größer ist als die anderen Werte $|\Delta q_i|_{\max}(F \neq i, j)$ oder im Grenzfall den gleichen Wert annimmt:

Betrachtet man z. B. den Fall a) mit $k < A_i / A_j$, so ergeben sich für die drei Fälle folgende Formeln für $|\Delta q_i|_{\max}$:

$$1) \quad |\Delta q_i|_{\max}(F \neq i, j) = \frac{v_i'^2}{2} \left(\frac{1}{kA_j} - \frac{1}{A_i} \right) \text{ mit } v_i' \text{ wie oben}$$

$$2) \quad |\Delta q_i|_{\max}(F = i) = \frac{p^2 v_i^2}{2} \left(\frac{1}{kA_j} - \frac{1}{A_i} \right)$$

$$3) \quad |\Delta q_i|_{\max}(F = j) = \frac{v_i'^2}{2} \left(\frac{1}{kA_j} - \frac{1}{A_i} \right) \text{ mit } v_i' \text{ wie oben, aber } t_e \text{ abhängig von Achse } j$$

v_i' ist wegen der Reduzierung der Geschwindigkeit zur Synchronisation mit der Führungssachse immer kleiner als v_i . Für Paare $(q_{ie} | q_{je})$, bei denen i die Führungssachse gegenüber j ist, gibt es deshalb keine Führungssachse F gegenüber der Achse i, für die $|\Delta q_i|_{\max}$ von Fall 1) größer als $|\Delta q_i|_{\max}$ von Fall 2) ist. D. h. es müssen nun noch die Paare $(q_{ie} | q_{je})$ untersucht werden, bei denen j die Führungssachse gegenüber i ist und nachgewiesen werden, daß für diesen Fall stets $v_i'(F = j) \geq v_i'(F \neq j)$ ist.

$$v_i' = \frac{A_i}{2} \left(t_e - \sqrt{t_e^2 - \frac{4q_{je}}{A_i}} \right)$$

hängt über t_e von der jeweiligen Führungssachse $F \neq j$ oder $F = j$ ab. Die erste Ableitung von v_i' nach t_e errechnet sich zu

$$\frac{\partial v_i'}{\partial t_e} = \frac{A_i}{2} \left(1 - \frac{t_e}{\sqrt{t_e^2 - \frac{4q_{ie}}{A_i}}} \right)$$

und ist wegen des negativen Vorzeichen des zweiten Terms in der Klammer immer kleiner als 0, da dessen Betrag größer als 1 ist. D. h. v_i' sinkt mit wachsendem t_e . Damit $v_i'(F \neq j)$ größer als $v_i'(F = j)$ wird, müßte also $t_e(F \neq j) < t_e(F = j)$ gelten. Dies ist aber nicht möglich, da F für diesen Fall dann nicht mehr Führungsachse ist. Es genügt deshalb zur Bestimmung der $|\Delta q_i|_{\max}$ die Achsen i und j (oder entsprechend die Fälle 1 und 3) zu betrachten, da die Berücksichtigung der Fälle, in der andere Achsen Führungsachse sind, keine höheren Werte für $|\Delta q_i|_{\max}$ liefert.

Dieser Nachweis kann analog für den Fall b) $k < A_i / A_j$ mit

- 1) $|\Delta q_i|_{\max} (F \neq i, j) = \frac{k v_j'^2}{2} \left(\frac{k}{A_i} - \frac{1}{A_j} \right)$ mit v_j' wie oben
- 2) $|\Delta q_i|_{\max} (F = i) = \frac{k v_i'^2}{2} \left(\frac{k}{A_i} - \frac{1}{A_j} \right)$ mit v_j' wie oben, aber t_e abhängig von Achse i
- 3) $|\Delta q_i|_{\max} (F = j) = \frac{k p^2 v_j'^2}{2} \left(\frac{k}{A_i} - \frac{1}{A_j} \right)$

erbracht werden.

Zur Berechnung des größten auftretenden Unterschieds $|\Delta q_i|_{\max}$ muß dieser also in Abhängigkeit von den Achsen i, j und dem Geschwindigkeitsprozentsatz p bestimmt werden. Dies ist analytisch jedoch nicht möglich, da bei der Verwendung von Ableitungen zur Extremabestimmung die Nullstellen von Polynomen bis zu 5. Grades für den allgemeinen Fall ermittelt werden müßten.

Deshalb wird die Bestimmung von $|\Delta q_i|_{\max}$ im folgenden numerisch am Beispiel des Roboters Manutec R15S1030 der Firma Siemens durchgeführt. Dieser Roboter besitzt folgende charakteristische Werte:

5.3 Bahnplanung für weiträumige Transferbewegungen

Achse Nr.	1	2	3	4	5	6
Verfahrereich [°]	±165	±110	±135	±190	±120	±265
maximale Achsgeschwindigkeit [°/s]	200	100	200	356	400	280
Achsbeschleunigung [°/s ²]	260	156	443	1738	2539	1844

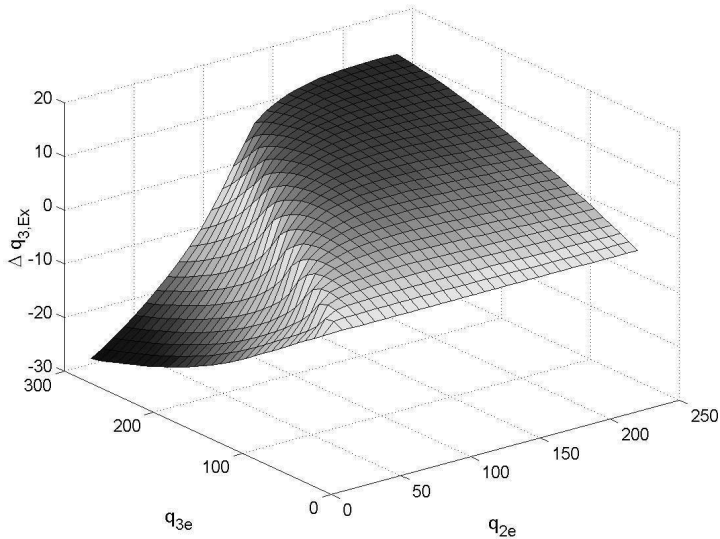


Bild 5-23: $\Delta q_{3,Ex}$ über q_{2e} und q_{3e} für den Roboter Manutec R15S1030

Die Fläche in Bild 5-23 entspricht dem Wert der maximalen bzw. minimalen Abweichung $\Delta q_{3,Ex}$ aufgetragen über die insgesamt zurückgelegten Winkelwege q_{2e} und q_{3e} für die Achsen $i = 3$, $j = 2$ und die Geschwindigkeiten und Beschleuni-

gungen des Roboters Manutec R15S1030. Für große q_3 und kleine q_2 gilt der Fall

b) mit $k = \frac{q_{ie}}{q_{je}} > \frac{A_i}{A_j}$.

In Bild 5-24 ist der Betrag $|\Delta q_3|_{\max}$ von $\Delta q_{3, Ex}$ aufgetragen. Die maximale Abweichung $|\Delta q_3|_{\max}$ beträgt $25,352^\circ$ bei $q_{2e} = 0,01^\circ$ und $q_{3e} = 270^\circ$. Die mittlere Abweichung beträgt $8,5^\circ$ und wird für $q_{3e} < 67,5^\circ$ und $q_{2e} < 55,0^\circ$ nicht überschritten.

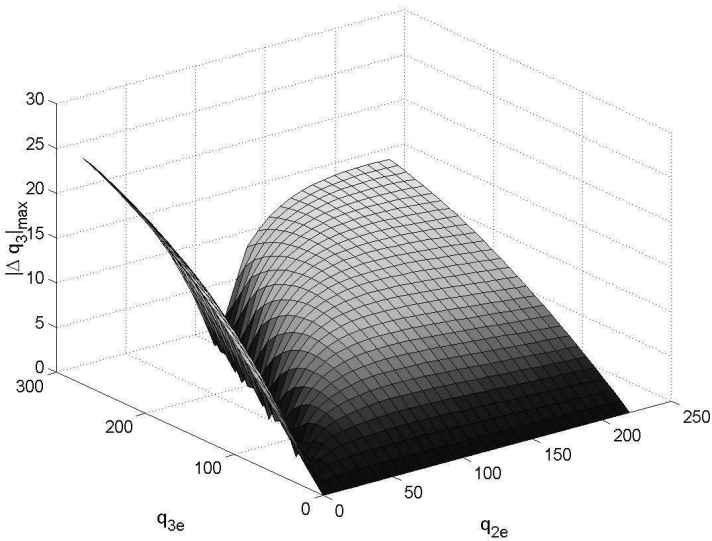


Bild 5-24: $|\Delta q_3|_{\max}$ über q_{2e} und q_{3e} für den Roboter Manutec R15S1030

Der Einfluß der Geschwindigkeit auf die Abweichung ist aus Bild 5-25 ersichtlich. Die bei einer Bewegung zulässige (und im Rahmen der Randbedingungen ausgeschöpfte) Maximalgeschwindigkeit ist als ein prozentualer Anteil p der nominalen Maximalgeschwindigkeit einer Achse definiert.

In Bild 5-25 sind diejenigen Werte von p über q_{2e} und q_{3e} aufgetragen, für die der größte Wert von $|\Delta q_3|_{\max}$ erzielt wird. Für kleine Werte von q_{2e} und q_{3e} ergibt sich

für $p < 100\%$ die größte Abweichung, sonst für $p = 100\%$. Hiermit wird noch einmal am Beispiel deutlich, daß neben den Verfahrenswegen und den maximalen Geschwindigkeiten und Beschleunigungen auf jeden Fall der vom Roboterbediener wählbare Prozentsatz p der Geschwindigkeitsparameter in die Abschätzung der ungünstigsten Abweichung eingehen muß. Die Abschätzung der maximalen Abweichung erfolgt numerisch durch einen Vergleich der Abweichungen über den Parameterdefinitionsbereich, da eine analytische Lösung nicht möglich ist (s. S. 90). Wegen des nicht unerheblichen Betrages der Abweichung $|\Delta q_3|_{\max}$ von bis zu 25° für die betrachteten Achsen muß die Abweichung auf jeden Fall bei der Kollisionskontrolle während der Bahnplanung berücksichtigt werden.

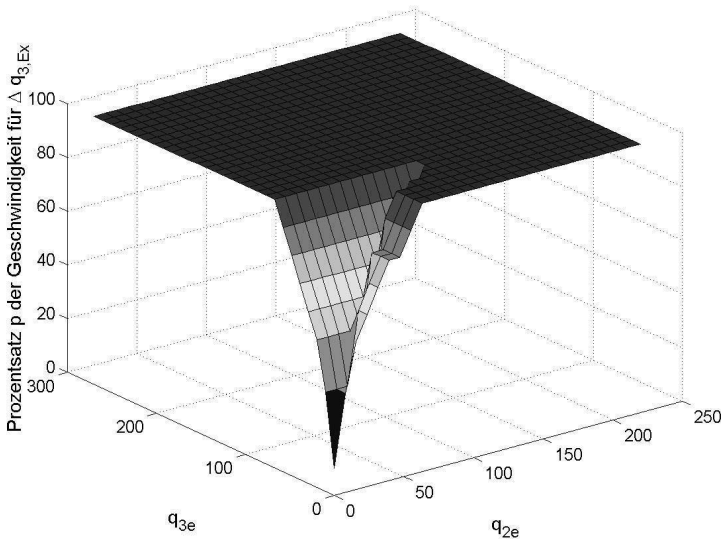


Bild 5-25: Prozentsatz p für maximale Abweichung

5.3.3 Kollisionsüberprüfung mit Hüllkörpern für den verwendeten Bahnsuchalgorithmus

Im ersten Schritt sollen dazu die einzelnen Komponenten des bewegten Roboters mit Kollisionshüllen umgeben werden, so daß zwar die Gerade im Gelenkwinkelraum bzgl. Kollision geprüft wird, durch die vergrößerte Geometrie aber auch die entsprechende PTP-Bewegung kollisionsfrei ist. Die Festlegung von Kollisionshüllen muß in Abhängigkeit des Kinematiktyps des Roboters erfolgen. Im folgenden soll die Festlegung von Kollisionshüllen deshalb für das Beispiel eines Sechssachsenknickarmroboters betrachtet werden. Für andere Bauformen muß das Vorgehen entsprechend übertragen werden.

Die ersten drei Achsen eines Sechssachsenknickarmroboters sind seine Hauptachsen, ihre Bewegung bewirkt die größten Positionsänderungen. Mit der vierten bis sechsten Achse, den sog. Neben- oder Handachsen erfolgt die Feineinstellung der Endeffektorposition. Für die Ermittlung von Kollisionshüllen für eine weiträumigen PTP-Bewegung eines Sechssachsenknickarmroboters wird deshalb zuerst die Auswirkung der ersten drei Achsen berücksichtigt. Die Berechnung des Unterschieds $|\Delta q_i|_{\max}$ für $i = j-1$ erfolgt mit den gleichen Formeln wie die oben hergeleitete Berechnung von $|\Delta q_i|_{\max}$ für $i = j+1$. Die Abweichung von Achse 1 gegenüber der von Achse 2 definierten Gerade im Gelenkwinkelraum ist in Bild 5-26 für den Roboter Manutec R15S1030 dargestellt.

Die Festlegung von Hüllgeometrien für die ersten drei Achsen gestaltet sich relativ einfach, wenn die Bewegung der zweiten Achse als mit der Bewegung des Roboters auf der Geraden im Gelenkwinkelraum identisch definiert wird. Als maximal zulässige Abweichung wird zunächst die jeweils mittlere Abweichung der Achswinkel der ersten und dritten Achse gewählt, die sich bei Betrachtung des gesamten Bewegungsbereichs ergibt. Dafür müssen die mittleren Abweichungen $|\Delta q_1|_{\max}$ und $|\Delta q_3|_{\max}$ sowie die zur Verhinderung der Überschreitung erforderlichen Wertegrenzen für die q_{ie} ($i = 1, 2, 3$) ermittelt werden.

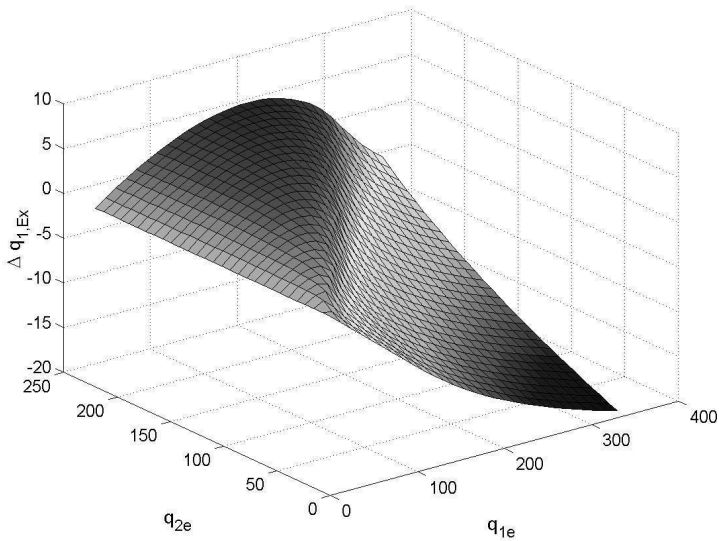


Bild 5-26: $\Delta q_{1,Ex}$ über q_{2e} und q_{1e} für den Roboter Manutec R15S1030

Betrachtet man für das Beispiels des Roboters Manutec R15S1030 die mittleren Abweichungen und die zulässigen Grenzwerte, so ergeben sich folgende Zahlenwerte:

Achsnr. i	$ \Delta q_i _{\max}$	$\overline{ \Delta q_i }$	Δq_i Grenz	Δq_2 Grenz
1	35,8°	9,6°	70,7°	47,1°
3	25,4°	8,5°	67,5°	55,0°

Während die maximale absolute Abweichung von der durch die zweite Achse bestimmten Geraden im Gelenkwinkelraum für die erste Achse 35,8° beträgt, beläuft sich die mittlere absolute Abweichung auf 9,6°. Die maximal möglichen Bewegungen sind durch die Winkelgrenzen der einzelnen Achsen festgelegt und betragen für die erste Achse 330° (±165°), für die zweite Achse 220° (±110°) und für die dritte Achse 270° (±135°). Beschränkt man diese Grenzen für die

erste Achse auf $70,7^\circ$ und für die zweite Achse auf $47,1^\circ$, so wird eine Abweichung von $9,6^\circ$ von der Geraden im Gelenkwinkelraum in der ersten Achse nicht überschritten. Dies gilt entsprechend für die Einhaltung einer maximalen Abweichung von $8,5^\circ$ in der dritten Achse (die der mittleren Abweichung über den gesamten Bereich entspricht) bei einer Begrenzung der Bewegung der dritten Achse auf $67,5^\circ$ und der ersten Achse auf $55,0^\circ$. Bei einer PTP-Bewegung wird im allgemeinen nur selten über den gesamten für den Roboter zugelassenen Winkelbereich einer Achse verfahren. Durch die Beschränkung des zulässigen Verfahrbereichs einer Achse für eine PTP-Anweisung wird deshalb einerseits die Bewegungsfreiheit des Roboters nicht wesentlich eingeschränkt und andererseits die Abweichung von der Gerade im Gelenkwinkelraum verringert. Die Abweichungen sind ausschlaggebend für die erforderliche Vergrößerung der Robotergeometrien, die dazu dient, Kollisionen aufgrund dieser Abweichung bei einer PTP-Bewegung zu vermeiden.

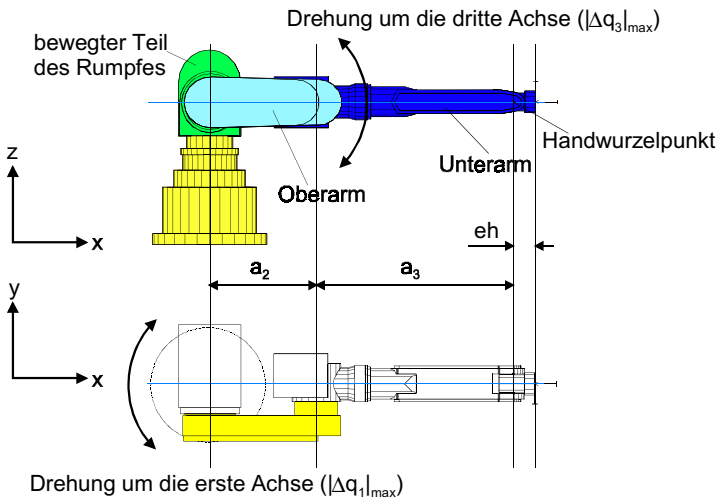


Bild 5-27: Bewegung der ersten und dritten Achse

Bei einer maximalen Abweichung von $|\Delta q_1|_{\max}$ der ersten Achse von der Geraden im Gelenkwinkelraum werden der bewegliche Teil des Roboterrumpfes, der Oberarm und der Unterarm um bis zu $\pm|\Delta q_1|_{\max}$ aus ihrer Sollposition herausgedreht (auch Bild 5-27). Von einer Positionsabweichung der dritten Achse ist lediglich der Unterarm betroffen.

Das vergrößerte Volumen für die Kollisionsrechnung muß demnach für den beweglichen Teil des Rumpfes und den Oberarm die Abweichung $|\Delta q_1|_{\max}$ der ersten Achse, für den Unterarm die Abweichung $|\Delta q_1|_{\max}$ der ersten Achse und die Abweichung $|\Delta q_3|_{\max}$ der dritten Achse kompensieren. Zur Berechnung des vergrößerten Volumens eignet sich für den Einfluß der Abweichungen der ersten Achse die dargestellte Roboterstellung mit horizontal ausgestrecktem Arm, bei der sich bis auf die 2. Achse mit $q_2 = 90^\circ$ alle Achsen in Nullstellung befinden, da die Vergrößerung dann nur in der horizontalen Ebene erfolgt. Die für die Abweichung der dritten Achse erforderliche Vergrößerung läßt sich in der Ebene der zugehörigen Seitenansicht berechnen, wobei die Wahl von q_2 frei ist.

Auf die Vergrößerung des beweglichen Teils des Roboterrumpfes kann verzichtet werden, falls nicht der Arbeitsraum des Roboters durch entsprechende Parameter eingegrenzt ist. Denn in der Regel, muß dieser Teil in seinem gesamten Drehbereich kollisionsfrei zu verdrehen sein. Der Unterarm erfährt in positiver oder negativer y-Richtung einen maximalen Versatz $\Delta y = (a_2 + a_3 + eh) \tan(|\Delta q_1|_{\max})$ mit a_2 , a_3 und eh wie in Bild 5-27 dargestellt. Außerdem kann er geringfügig in neg. x-Richtung verschoben werden, so daß der Ellbogen etwas nach hinten verlängert werden muß (Bild 5-28). Hier genügt eine Verlängerung von $\Delta l_x \leq a_2 \sin(|\Delta q_1|_{\max})$. Schwieriger gestaltet sich die Bestimmung der erforderlichen Vergrößerung für den Oberarm, da dieser nicht symmetrisch zur ersten Achse liegt. Eine vereinfachte Lösung bietet die Verwendung der Bounding Box und die Berechnung der vergrößerten Bounding Box für die Verdrehung wie in Kap. 4.

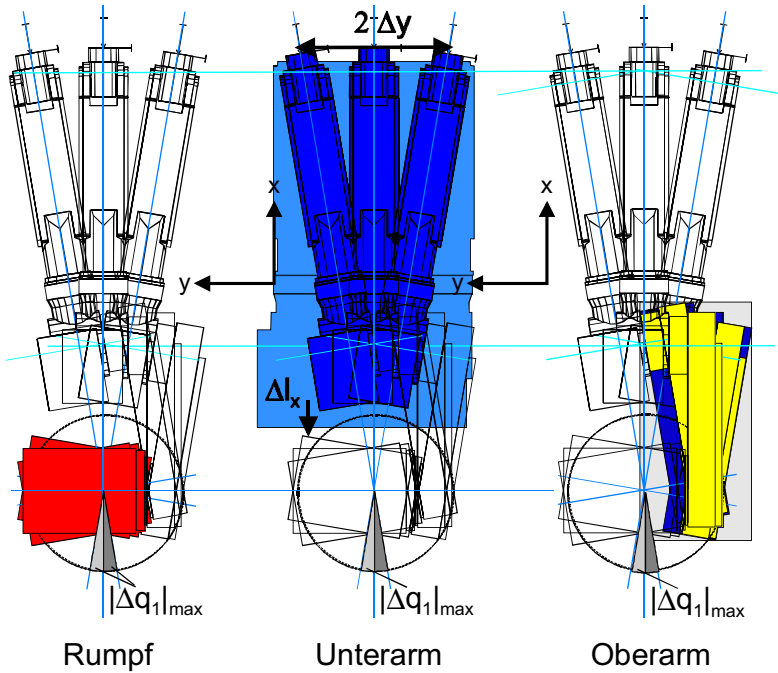


Bild 5-28: notwendige Vergrößerungen für die erste Achse

Aufgrund der Winkelabweichung $\Delta q_{3\max}$ in der dritten Achse erfährt der Unterarm in positiver oder negativer z -Richtung einen maximalen Versatz $\Delta z = (a_3 + e_h) \tan(|\Delta q_{3\max}|)$, der wie der Versatz von $|\Delta q_1|_{\max}$ in y -Richtung behandelt wird. Die Verschiebung in neg. x -Richtung wird bereits mit der großzügigen Verlängerung um Δl_x für die Abweichung in der ersten Achse kompensiert. Um auch bei Bewegungen in der vierten Achse Kollisionen des Unterarms zu vermeiden, wird schließlich das bei einer Drehung des Unterarms um 360° um die vierte Achse überstrichene Volumen als Hüllkörper des Unterarms verwendet.

Im nächsten Schritt müssen geeignete Hüllvolumina für die Mittelhand, den Werkzeugflansch sowie das Werkzeug bzw. den Greifer und das gegriffene Objekt bestimmt werden (Bild 5-29).

$$\begin{pmatrix} \Delta r \\ \Delta \varphi \\ \Delta z \end{pmatrix} = \begin{pmatrix} a_3 \sin(q_2 + q_3 + \Delta q_3) - a_3 \sin(q_2 + q_3) \\ \Delta q_1 \\ a_3 \cos(q_2 + q_3 + \Delta q_3) - a_3 \cos(q_2 + q_3) \end{pmatrix}.$$

Außerdem gilt für $|\Delta r|$ und $|\Delta z|$

$$|\Delta r| \leq 2a_3 \sin \frac{|\Delta q_3|_{\max}}{2} \text{ und } |\Delta z| \leq 2a_3 \sin \frac{|\Delta q_3|_{\max}}{2}.$$

Geeignete Hüllvolumina für die Handachsen, den Greifer und das Werkstück hängen stark von der aktuellen Winkelstellung der Handachsen ab. Als Hüllvolumen, das für alle Stellungen gilt, eignet sich deshalb eine Kugel. Der Kugelradius berechnet sich als Summe aus dem größtmöglichen Abstand eines Punktes der zu umhüllenden Geometrien vom Handwurzelpunkt und dem maximalen Versatz des Handwurzelpunktes $|v_{HWP}|_{\max}$. Dieser entspricht entweder den Grenzen für Δr und Δz wie oben berechnet oder er wird durch den tangentialen Versatz, der sich aus $\Delta \varphi = \Delta q_1$ ergibt, bestimmt zu

$$|v_{HWP}|_{\max} = \text{Max} \left(2a_3 \tan \frac{|\Delta q_3|_{\max}}{2}; 2(a_2 + a_3) \tan \frac{|\Delta q_1|_{\max}}{2} \right)$$

Mit Hilfe der beschriebenen Hüllvolumina kann nun für eine Bewegung auf einer Geraden im Gelenkwinkelraum mit q_i innerhalb der Wertegrenzen für $\Delta q_i < \Delta q_{i\max}$ die Kollision für entsprechende PTP-Bewegungen berücksichtigt werden. Setzt man die Wertegrenzen für die zulässigen q_i herab, so verringern sich die $\Delta q_{i\max}$ und auch die Hüllvolumina können verkleinert werden.

Bei der Berechnung der mittleren Abweichungen für den Roboter Manutec R15S1030 ergeben sich Werte von bis zu 10° . Die für diese Abweichungen ermittelten Hüllkörper sind bereits so groß, daß in engen Bewegungsräumen für manche Fälle keine Bestimmung einer kollisionsfreien Bahn bei alleiniger Verwendung dieser Geometriemodelle möglich ist. Diese Geometriemodelle werden deshalb bei der Bahnplanung als maximale Hüllkörper verwendet, die damit verbundenen Wertebereiche für die zurückgelegten Winkelwege q_i als maximale Wertebereiche. Von diesen Wertebereichen wird wiederum die mittlere Abweichung gebildet und eine kleinere Größenstufe von Hüllkörpern erzeugt. Vor der

Überprüfung eines Geradenstückes im \mathfrak{R}^n wird dann die zu den Winkelwegen passende kleinste Hüllkörperstufe ausgewählt und für die Kollisionsrechnung verwendet. Tritt eine Kollision auf, so wird untersucht, wie weit mit den kleineren Stufen kollisionsfrei verfahren werden kann und gegebenenfalls der Winkelweg geteilt, damit die kleinere Stufe zulässig ist. Wird unter Verwendung der Hüllkörper keine kollisionsfreie Bahn gefunden, so wird das Netz um Wege ergänzt, die nur für die Nominalgeometrie des Roboters kollisionsfrei sind. Kann damit ein Weg gefunden werden, so werden im Roboterprogramm nur die mit den Hüllkörpern erzeugten Wege als PTP-Bewegungen abgespeichert, die anderen Wege werden in einem weiteren Planungsschritt durch bahngesteuerte Bewegungen angenähert, die bzgl. Kollision überprüft werden und ins Programm eingetragen werden.

Die Diskretisierung der Geradenstücke im \mathfrak{R}^n erfolgt wie bei STETTER (1993) entsprechend fein. Sie darf maximal den für die Hüllkörper gültigen Wert $|\Delta q_i|_{\max}$ annehmen, um eine kollisionsfreie Bahn zu gewährleisten. In der Literatur finden sich zwar einige Beispiele dafür, daß die Diskretisierungsschrittweite bei entsprechender Vergrößerung der Robotergeometrie heraufgesetzt werden kann (wie z. B. bei BAGINSKI 1997). Diese Beispiele versuchen jedoch lediglich die Kollisionsrechnung bei der Bahnplanung zu beschleunigen und behandeln nicht das Problem von Abweichungen.

Im Gegensatz zu den meisten in der Literatur beschriebenen Verfahren kann mit den Hüllkörpern für die Ausführung des Roboters ein kollisionsfreies Bewegungsprogramm für die Steuerung erzeugt werden. Andere Ansätze verwenden zur Ausführung von derart generierten Programmen oft eigene RC-Controller, die die Einhaltung des mathematisch vorgegebenen Weges gewährleisten oder sie gestatten u. U. sogar nur Bewegungen in einer Achse, um das Problem der Abweichung von der Geraden im Gelenkwinkelraum zu vermeiden. Derartige Ansätze sind im Gegensatz zum hier vorgestellten Ansatz für den Einsatz der Bahnplanung zur Offline-Programmierung einer Industrieanlage nicht geeignet. Eine andere Möglichkeit, kollisionsfreie PTP-Bewegungen zu erzeugen, ist die Verwendung eines realitätsgetreuen Steuerungsmodells (z. B. durch RCS-Module). Kollidiert der Roboter jedoch während einer PTP-Bewegung zwischen zwei Po-

sitionen, kann der Weg bis zur Kollisionsstelle nicht als kollisionsfrei vorausgesetzt werden, da sich die ursprünglich untersuchte längere Bahn von der kürzeren PTP-Bewegung bis zur Kollisionsstelle unterscheidet. Außerdem hängt die von der Roboterhand beschriebene Bahn auch von den für die Achsen eingestellten Parameter für Geschwindigkeit und Beschleunigung ab. Deshalb muß für diesen Fall sichergestellt sein, daß diese Parameter im realen Programm beibehalten werden oder es müssen unzählige Parameterkombinationen getestet werden. Die Verwendung von Hüllkörpern bei der Bahnsuche ist deshalb eine sichere Lösung mit vertretbarem Aufwand.

5.4 Übergreifende Handhabungsplanung

Für die automatische Durchführung der gesamten Planung eines Handhabungsvorgangs müssen die in den vorhergehenden Kapiteln vorgestellten Module zur Greifplanung, Planung von Feinbewegungen und Bahnplanung für weiträumige Transferbewegungen kombiniert werden. Falls unter den gegebenen Randbedingungen die Handhabung nicht möglich ist, müssen Alternativen untersucht werden. Es ist deshalb zusätzlich zu den speziellen Planungsmodulen eine übergreifende Handhabungsplanung erforderlich. Wie bereits in Kapitel 2 erwähnt, ist die für die Handhabung verwendete Greifkonfiguration eine entscheidende Randbedingung für die Planung des Ablegevorgangs sowie der Fein- und Transferbewegungen. Die Greifplanung ist deshalb der erste Vorgang bei der Handhabungsplanung.

Da in vielen Roboterzellen mehrere Greifer zur Verfügung stehen, muß dabei auch die Auswahl des geeigneten Greifers berücksichtigt werden. Die übergreifende Handhabungsplanung muß also zunächst den Greifertyp festlegen und einen in der Zelle verfügbaren Greifer dieses Typs auswählen (Bild 5-30). Mit diesen Vorgaben wird anschließend die Greifplanung durchgeführt. Ergebnis der Greifplanung sind eine oder mehrere Greifkonfigurationen, geordnet nach ihrer Güte, die durch die in Kap. 5.1 beschriebene Bewertung ermittelt wird.

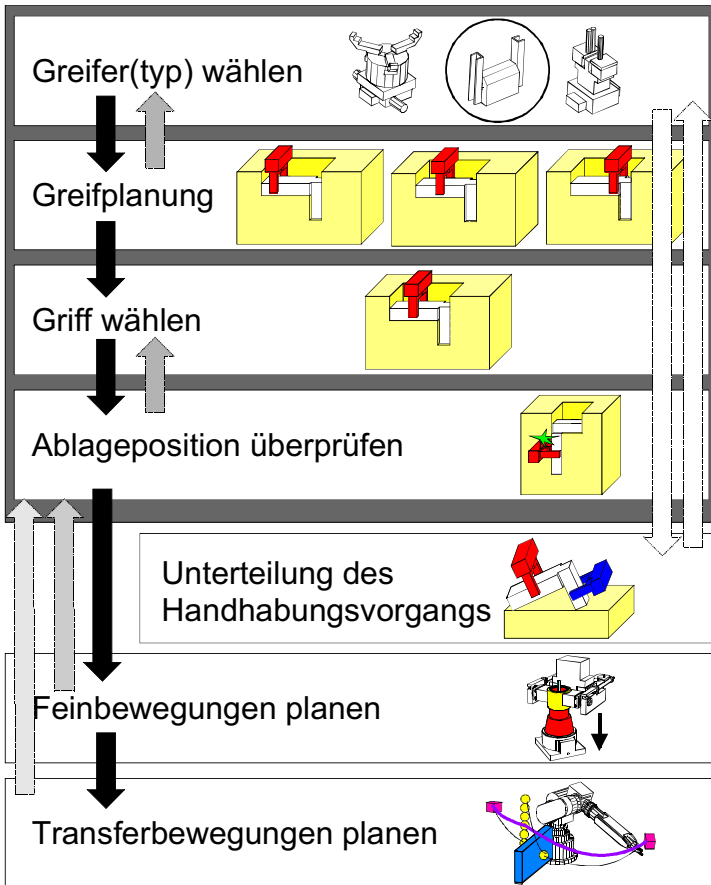


Bild 5-30: Übergreifende Handhabungsplanung

Im nächsten Schritt wird die Eignung der besten Greifkonfiguration für das Ablegen des Objekts an der gewünschten Position untersucht. Ist die Greifkonfiguration auch zum Ablegen geeignet, so werden die Feinbewegungen geplant und schließlich die weiträumigen Transferbewegungen zwischen den Feinbewegungen. Nach der vollständigen Planung wird das Ergebnis in einem Roboterpro-

gramm festgehalten, das dann sowohl in der Simulation als auch in der realen Zelle zur Ausführung des Transportvorgangs verwendet wird.

Der bisher beschriebene Ablauf entspricht dem idealen Ablauf einer Handhabungsplanung. Er ist in Bild 5-30 durch die schwarzen, senkrecht nach unten zeigenden Pfeile dargestellt. Oft ist jedoch nach der Bewertung des Planungsergebnisses nach dem jeweiligen Schritt eine Umplanung erforderlich. Umplanungen werden von der übergeordnete Handhabungsplanung eingeleitet, oberstes Ziel ist es, derartige Umplanungen von Beginn an zu vermeiden.

Bereits beim ersten Schritt, der Wahl des Greifertyps, können Umplanungen vermieden werden, indem ein Greifertyp gewählt wird, der für die Geometrie des zu greifenden Werkstücks geeignet ist. Dieser Zusammenhang ist jedoch vor der Untersuchung der Geometrie nicht zu erkennen. Die Auswahl des Greifertyps erfolgt deshalb nach folgenden Kriterien:

- Befindet sich bereits ein Greifer am Roboter, so wird die Geometrie zunächst für diesen Greifertyp untersucht. Dies ist sinnvoll, da man in vielen Fällen voraussetzen kann, daß der Planer bereits einen Greifer für die nächste Handhabungsaufgabe ausgewählt hat.

Wurde vom Planer kein Greifer vorbestimmt, so ist die Auswahl des richtigen Greifertyps wesentlich schwieriger und es kann nicht sichergestellt werden, daß wirklich die richtige Auswahl getroffen wurde.

- Spezialgreifer bieten meist eine höhere Griffstabilität als flexible Greifer und sind deshalb bevorzugt anzuwenden.

Aber auch bei Standardgreifern läßt sich eine Auswahlreihenfolge festlegen. Betrachtet man z. B. die drei Greifertypen eines Drei-Finger-Außengreifers, ein Zwei-Finger-Parallelaußengreifers und eines Innengreifers, so ist der flexibelste Greifer der Zwei-Finger-Parallelaußengreifer gefolgt vom Innengreifer.

- Bei der Greifplanung wird aber zuerst die Eignung des Drei-Finger-Außengreifers untersucht, um eine Greifkonfiguration mit möglichst hoher Griffsta-

bilität zu finden. Voraussetzung dafür ist, daß ein derartiger Greifer in der Zelle zur Verfügung steht.

- Enthält das zu greifende Objekt keine zylindrischen Elemente, so wird die Suche mit dem Innengreifer fortgesetzt.
- Nur wenn das Objekt auch mit dem Innengreifer nicht gegriffen werden kann, wird eine Greifplanung mit dem Zweifinger-Finger-Parallelaußengreifer fortgesetzt.

Wurden passende Greifelemente gefunden, so werden diese nun mit der Öffnungsweite eines vorhandenen Greifers dieses Typs verglichen und die sich daraus ergebenden möglichen Greifkonfigurationen nach der Griffstabilität sortiert in einer Liste zur Verfügung gestellt. Aus dieser Liste muß im nächsten Schritt eine Greifkonfiguration gewählt werden, mit der die folgenden Untersuchungen durchgeführt werden. Es wird immer die beste, noch nicht untersuchte Konfiguration gewählt. Sollte keine Konfiguration mehr zur Verfügung stehen, so muß entweder ein anderer verfügbarer Greifer desselben Typs betrachtet werden oder ein neuer Greifertyp gewählt werden, was durch den Pfeil von der Greifplanung nach oben im Bild dargestellt ist.

Für die gewählte Greifkonfiguration wird nun die Ablegekonfiguration, die sich aus Greifkonfiguration und Objektendposition berechnen läßt, hinsichtlich Kollision überprüft. Kann die Ablegeposition vom Roboter erreicht werden und weist sie keine Kollisionen auf, so wird mit der Planung der Feinbewegungen fortgefahren. Bei erfolgreicher Planung der Feinbewegungen erfolgt die Planung der Transferbewegungen. Wird bei der Bewegungsplanung keine kollisionsfreie Lösung gefunden, so wird die Handhabungsplanung mit der Auswahl einer neuen Greifkonfiguration fortgesetzt. Dies ist durch die beiden Pfeile von der Feinbewegungsplanung bzw. der Planung der weiträumigen Transferbewegungen dargestellt, die zurück in den Block der Schritte zur Festlegung von Greif- und Ablegeposition zeigen.

Existiert für alle zur Verfügung stehenden Greifer keine Greifkonfiguration, die sowohl für das Greifen als auch für das Ablegen verwendet werden kann, so muß

der Ablauf der Handhabung geändert werden. Hierbei bietet sich eine Unterteilung des Handhabungsvorgangs in zwei Handhabungsvorgänge mit Hilfe einer Zwischenposition an, an der das Objekt abgelegt wird (Bild 5-31, weiße Pfeile in Bild 5-30).

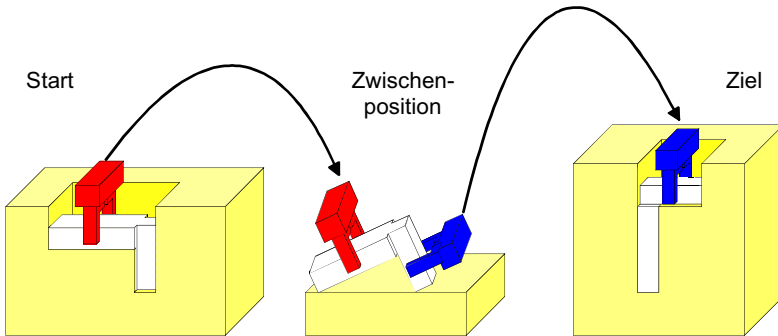


Bild 5-31: Lösung der Handhabungsaufgabe mit Hilfe einer Zwischenposition

Die Zwischenposition kann am besten vom Planer selbst festgelegt werden, der die Restriktionen für die Platzierung des Objekts an einer Zwischenposition kennt. Läuft die Planung allerdings im Hintergrund oder außerhalb der Arbeitszeit, sollte versucht werden, automatisch ein mögliches Ergebnis zu finden. Die Handhabungsplanung muß deshalb über ein einfaches Modul zur Suche nach Zwischenpositionen verfügen, das z. B. der Einschränkung unterliegen darf, daß das Objekt an der Zwischenposition auf einer ebenen, horizontalen Fläche aufliegt. Die Voraussetzung einer ebenen, horizontalen Fläche vereinfacht die Suche nach einer geeigneten Position, da nicht bestimmte Stellungen in einem Magazin o. ä. bekannt sein müssen. Zudem sind in fast allen Arbeitszellen ebene, horizontale Flächen vorhanden, an denen ein Objekt temporär abgelegt werden kann. Mögliche Orientierungen des Objekts auf der Auflagefläche sind dadurch gekennzeichnet, daß bestimmte Geometrielemente des Objekts in einer horizontalen Ebene liegen. Dies sind, wie in Bild 5-32 dargestellt, mindestens eine Fläche, zwei Kanten, eine Kante und ein Punkt oder drei Punkte.

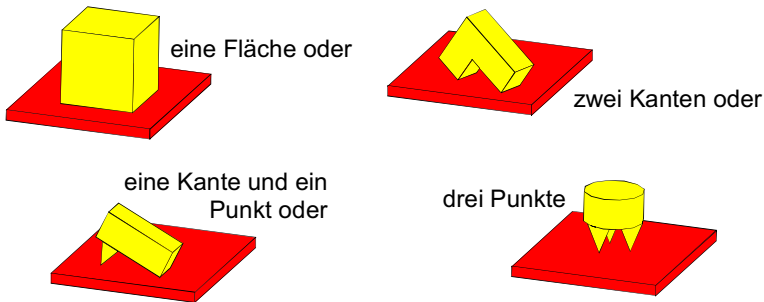


Bild 5-32: stabile Lagen

Ein Objekt kann allerdings auch zwei oder mehr Flächen in der horizontalen Auflageebene besitzen. Deshalb sagt ein einfacher Vergleich des Massenschwerpunkts mit dem geometrischen Schwerpunkt der mindestens vorhandenen Auflageelemente nichts über die Gesamtstabilität aus. Die Gesamtstabilität kann z. B. mit der Simulation physikalischer Effekte (STETTER 1993) überprüft werden. Dabei wird die Auswirkung der Gravitationskraft auf das Objekt in Ruhelage geprüft. Bewegt sich das Objekt, ist die Orientierung instabil, sonst stabil. Zur Bestimmung der genauen Position wird schließlich eine Orientierung ausgewählt und das Objekt im Überlappungsbereich von Roboterarbeitsraum und Auflage platziert. Die Zwischenposition muß mit einer möglichst guten Greifkonfiguration des Objekts erreicht werden können. Für den jeweiligen Greifer muß dann zusätzlich eine möglichst gute Greifkonfiguration ausgewählt werden, die sowohl an der Zwischenposition als auch an der Zielposition des Objekts verwendet werden kann. Können für einen Greifer nicht zwei derartige Konfigurationen ermittelt werden, so kann das Handhabungsproblem mit einem Greiferwechsel an der Zwischenposition gelöst werden. Für beide Lösungsansätze (mit und ohne Greiferwechsel an der Zwischenposition) müssen jedoch letztendlich die Fein- und Transferbewegungen für zwei Handhabungsvorgänge geplant werden.

Das vorgestellte, modular aufgebaute Verfahren ermöglicht eine sichere Neuplanung von Handhabungsvorgängen. Durch den modularen Aufbau kann es auch

zur Umplanung vorhandener Roboterprogramme eingesetzt werden, z. B. um eine Feinbewegung neu zu planen, da sich aufgrund einer Umgestaltung des Zellenlayouts nun mit dem alten RC-Programm Kollisionen ergeben.

5.5 Überprüfung und Umgestaltung vorhandener Roboterprogramme

Für einen sinnvollen Einsatz der 3D-Simulation zur Offline-Programmierung müssen nicht nur Programme von Grund auf neu erstellt werden, sondern auch vorhandene Programme hinsichtlich ihrer kollisionsfreien Funktionsfähigkeit überprüft werden. Dazu kann zunächst die Standardkollisionsüberprüfung des Simulationssystems eingesetzt werden. Sobald jedoch PTP-Bewegungen simuliert werden, sollte wie bei der Bahnplanung für Transferbewegungen ein erweitertes Geometriemodell verwendet werden, um auch Kollisionen zu erfassen, die sich z. B. beim Verstellen des Prozentsatzes der Maximalgeschwindigkeit, der als Geschwindigkeitsgrenze vorgegeben wird, ergeben. Treten erst mit dem erweiterten Geometriemodell Kollisionen auf, so muß der Planer entsprechend darauf hingewiesen werden und selbst entscheiden, ob das Bewegungsprogramm so belassen bleibt oder umgeplant werden soll. Treten beim Greifen, Ablegen, bei den Feinbewegungen oder bei den Transferbewegungen mit dem unvergrößerten Modell Kollisionen auf, so muß der Vorgang umgeplant werden. Das System muß dabei in der Lage sein, selbst aus dem bisher vorhandenen Programm die Aufgabenstellung für die Umplanung zu extrahieren, was aufgrund des einheitlichen Programmaufbaus relativ leicht zu realisieren ist. Bahngesteuerte Bewegungen nach oder vor einem Greif- oder Ablegevorgang müssen dabei als Feinbewegungen erkannt werden, andere Bewegungen als Transferbewegungen eingestuft werden. Auch wenn die Umplanung automatisch erfolgen kann, müssen die im aktuellen Programm aufgetretenen Fehler dem Benutzer dargestellt werden, um so z. B. zu vermeiden, daß das Programm nur nicht funktioniert, weil inzwischen das reale Werkzeug ausgetauscht wurde. Das Problem der Rückführung von RC-Programmen mit einem gleichzeitigen Abgleich von Simulation und Realität wird in Kapitel 6 näher behandelt.

6 Automatischer Layoutabgleich

6.1 Problemstellung beim Layoutabgleich

Der Aufwand für die Erstellung eines 3D-Simulationsmodells einer Roboterzelle ist trotz der Nutzung vorhandener CAD-Daten und Kinematikmodelle nicht zu vernachlässigen. Damit sich die Modellerstellung rechnet, sollte das Modell fortlaufend an den Zustand in der realen Zelle angepaßt werden. So kann es auch als Planungsbasis für die Erstellung weiterer RC-Programme eingesetzt werden. Abweichungen zwischen realer Zelle und Simulationslayout erfordern immer wieder Detailanpassungen der Roboterprogramme beim Einfahren. Die Gegenüberstellung des ursprünglichen und des eingefahrenen Programms gibt Aufschluß über diese Abweichungen.

Leider wird aber eine Anpassung des Simulationsmodells oft nicht durchgeführt. Schon bei Inbetriebnahme der Zelle werden lediglich die offline erstellten Programme als Basis verwendet, die angepaßten, eingefahrenen Programme und der reale Zellenaufbau aber nicht mehr ins Modell zurückgeführt. Dies liegt u. a. daran, daß zwar die Abweichungen in Form von unterschiedlichen Roboterstellungen beim Vergleich des ursprünglichen und des eingefahrenen Programms lokalisiert werden können, die Ermittlung der genauen Ursachen jedoch sehr kompliziert und aufwendig ist. Oft überlagern sich unterschiedliche Ursachen wie z. B. eine Positionsabweichungen von Objekten mit Bewegungstoleranzen des Roboters. Eine wesentliche Erleichterung des Modellabgleichs für den Planer bietet deshalb eine automatische oder alternativ eine manuelle, aber vom System unterstützte Anpassung des Planungsmodells an die realen Gegebenheiten.

Unterschiede zwischen realer und modellierter Zelle sind implizit im eingefahrenen RC-Programm des realen Roboters und im ursprünglich in der Simulation offline erstellten RC-Programm niedergelegt. Die Struktur eines offline erstellten RC-Programms muß beim Einfahren aufgrund von Abweichungen zwischen Modell und realer Zelle kaum geändert werden. Strukturelle Änderungen sind meist dann erforderlich, wenn die Struktur anfangs nicht mit der entsprechenden Sorg-

falt festgelegt wurde. Dies gilt sowohl für die Offline-Programmierung als auch für die Programmierung am realen Roboter.

Nach dem Einfahren eines offline erstellten Programms finden sich jedoch wichtige Unterschiede bei den Greif- und Ablegepositionen des Roboters sowie in der jeweils vor und nach diesen Positionen durchgeführten Feinbewegung, die durch die Greif- bzw. Ablegeposition und eine weitere Position bestimmt ist. Für einen Rückschluß von diesen Positionsabweichungen auf die vorzunehmenden Korrekturen im Modell werden zunächst die einzelnen Ursachen für Positionsunterschiede (Bild 6-1) und deren Auswirkungen betrachtet.

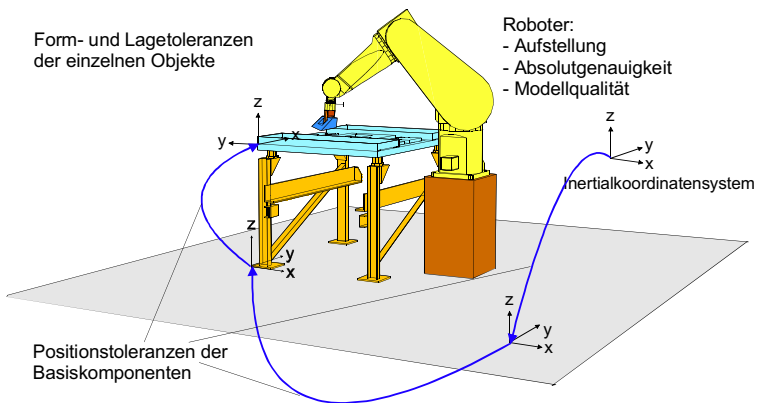


Bild 6-1: Ursachen für die Abweichungen zwischen realer und modellierter Zelle

Die Positionsunterschiede ergeben sich durch

- Differenzen zwischen der realen Roboterkinematik und dem Simulationsmodell,
- die Absolutgenauigkeit des realen Roboters,
- geometrische Toleranzen des Greifers,

- geometrische Toleranzen des bewegten Objekts,
- geometrische Toleranzen der Palettier- und Magaziniervorrichtungen sowie Positioniergenauigkeiten der Basiskomponenten der Zelle,
- eine unterschiedliche Aufstellung des Roboters in realer und modellierter Zelle und
- Toleranzen der einzelnen Bauteile und Kinematiken durch Erwärmung.

Betrachtet man eine einzelne Pick-and-Place-Operation, so treten mehrere Abweichungen zwischen der realen Zelle und ihrem Simulationsmodell auf. Die Position des zu greifenden Objekts in seiner Startlage weist eine Toleranz auf, die sich aus einer Toleranzkette ergibt, die von denjenigen Bauteilen beeinflusst wird, die direkt oder indirekt sog. Trägerobjekte des zu greifenden Objekts sind. Derartige Trägerobjekte sind z. B. der Hallenboden, ein darin verschraubtes Grundgestell mit Grundpalette und mehreren Werkstückträgern, von denen einer das zu greifende Objekt bereitstellt. Sind die Toleranzen der einzelnen Objekte in der Bauteilkette sowie die Fügeelemente bekannt, so können mögliche Werte für die Position des zu greifenden Objekts berechnet werden, die Berechnung der Gesamttoleranz für die Objektposition ist jedoch sehr schwer.

Häufig sind zwei Elemente miteinander verborgt und verschraubt, durch Bolzen verbunden, durch eine Führung verbunden oder in ein Formelement eingelegt. Betrachtet man das Beispiel in Bild 6-2, so ergeben sich bereits relativ komplizierte Gleichungen für die maximale Positionsabweichung des Objekts rechts im Bild, dessen Bohrungen über die Bolzen der Aufnahme links gesteckt werden.

Nimmt man z. B. an, daß sich die Achse der Bolzen bzw. Bohrungen um die x - und die y -Achse um $\Delta\alpha_x$ bzw. $\Delta\alpha_y$ aus dem Lot bewegen kann, der Aufpunkt der Achse Toleranzen Δx , Δy und Δz aufweist und die beiden Durchmesser um $\pm\Delta d$ vom Sollwert abweichen können, so kann es je nach Wahl der Bohrungs- und Bolzendurchmesser entweder zu einer sehr instabilen Fixierung kommen oder zu einer Verklammerung des oberen Bauteils.

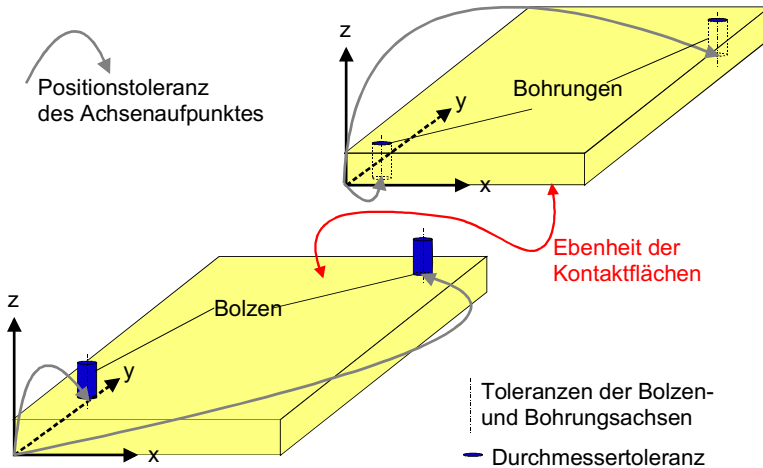


Bild 6-2: Einfaches Fügebeispiel mit schwieriger Berechnung der Auswirkung der Toleranzen auf die Position des oberen Bauteils

Betrachtet man z. B. den Fall, daß alle Maße und Flächenebenheiten mit Ausnahme der Winkel α_x der Bolzenachsen genau den Idealmaßen entsprechen und die beiden $\Delta\alpha_x$ für die Bolzenachsen einen identischen Wert annehmen, so ergeben sich abhängig von den Bolzen- und Bohrungsdurchmessern zwei mögliche Fixierungsfälle. Falls die Bolzendurchmesser klein genug und die Bohrungsdurchmesser groß genug sind, läßt sich das obere Objekt auch nach dem Fügen noch in der Ebene der Grundfläche verschieben. Dies gilt für

$$d_{Bohrung} > h_{Bolzen} \sin \Delta\alpha_x + \frac{d_{Bolzen}}{\cos \Delta\alpha_x} \quad (\text{Bild 6-3 links}).$$

Ist der Bohrungsdurchmesser kleiner, so ergibt sich ein Spalt zwischen den beiden Bauteilen, der maximal $l \sin \Delta\alpha_x$ groß wird (Bild 6-3 rechts). Bereits bei diesem vereinfachten Sonderfall sind also sehr viele unterschiedliche Bauteilpositionen denkbar. Diese beeinflussen z. B. ein Roboterprogramm, mit dem ein

weiteres Bauteil auf dem oberen Teil mit den Bohrungen abgelegt werden soll. Das Programm muß dann natürlich an die realen Abmaße angepaßt werden.

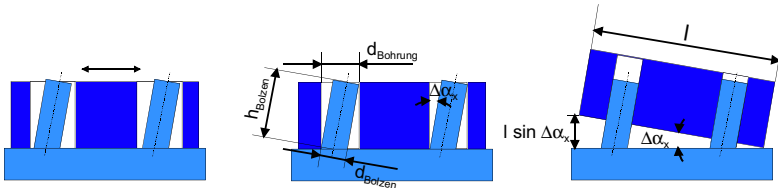


Bild 6-3: Unterschiedliche Fügepositionen abhängig von den aktuellen Maßwerten.

Die computersystemgerechte Tolerierung, bei der aus Einzeltoleranzen automatisch die Toleranz für die gesamte Baugruppe errechnet werden kann, ist zur Zeit noch Gegenstand der Forschung. Es existieren erste Systeme, die aus herkömmlichen Toleranzangaben die Toleranzketten bei der Montage von Baugruppen berechnen, diese berücksichtigen jedoch physikalische Randbedingungen wie z. B. das Verklemmen von Baugruppen noch nicht ausreichend oder überprüfen zwar die Zulässigkeit der angegebenen Toleranzen z. B. bzgl. der Freigängigkeit innerhalb einer Baugruppe ohne jedoch die Auswirkung auf die Bauteilplatzierung in einer Produktionszelle zu ermitteln. Die automatische Toleranzanalyse kann deshalb in ihrem derzeitigen Entwicklungsstand leider noch nicht als Basis für den automatischen Layoutabgleich genutzt werden.

6.2 Konzeption eines Systems zum Layoutabgleich

Bei einer realisierten Roboterzelle ist zu erwarten, daß verklebte Elemente weitgehend ausgeschlossen werden können, die Toleranzen bei Führungen eher gering sind, dafür aber bei weniger genau befestigten oder sogar frei aufliegenden Bauteilen sehr hohe Toleranzen berücksichtigt werden müssen. Würde man wirklich den maximalen Toleranzbereich einer Baugruppe berechnen wollen, so müßte man alle Kombinationen zulässiger Werte berücksichtigen, um zu einem

umfassenden Ergebnis für mögliche Positionierungen einzelner Elemente in der Roboterzelle zu kommen. Für die Anwendung zur Anpassung des Simulationsmodells der Roboterzelle ist deshalb eine vereinfachte, pauschale Betrachtung besser geeignet, bei der den einzelnen Objekten eine verhältnismäßig große Beweglichkeit zugewiesen wird, um die Idealgeometrie zu verwenden und den Einfluß von Toleranzketten vereinfacht als Linearsumme der Positionsabweichungen der einzelnen Bauteile darzustellen. Für die Istmaße der Bauteile bei der Fertigung wird in der Regel eine Normalverteilung zugrunde gelegt (KLEIN 1994), bei der die Wahrscheinlichkeit für die Idealmaße am höchsten ist. Die Darstellung der Toleranzkette als Linearsumme schränkt deshalb die Anwendung zur abschätzenden Ermittlung eines möglichen Reallayouts nicht ein.

Bei einer Roboterzelle mittlerer Größe kann für die freie Aufstellung eines Grundgestells oder Roboters (bzw. seines Sockels) auf dem Hallenboden oder bei seiner Verankerung in der Wand eine Toleranz von $\Delta x = \pm 3\text{mm}$, $\Delta y = \pm 3\text{mm}$, $\Delta z = \pm 3\text{mm}$, $\Delta \alpha = \pm 2^\circ$, $\Delta \beta = \pm 2^\circ$, $\Delta \gamma = \pm 2^\circ$ angenommen werden. Diese Toleranz wird ebenfalls bei temporären Verankerungen von großen Objekten die häufig durch den Materialfluß ausgetauscht werden, wie z. B. Paletten, die auf Grundgestellen aufliegen, vorausgesetzt. Bei allen anderen Betriebsmitteln wird eine Toleranz von $\pm 1\text{ mm}$ bzw. $\pm 0,5^\circ$ angenommen. Diese relativ großen Werte sind u. a. erforderlich, weil das Robotermodell selbst nicht verändert wird und so die meisten Toleranzen der Kinematik im Zellenmodell kompensiert werden müssen. Die Verlagerung der Kompensation in die Bauteilanordnung in der Zelle ist auch bei Kalibrationsverfahren zur Roboterprogrammierung mit 3D-Simulationssystemen üblich, da andere vergleichsweise einfache Lösungen nicht existieren.

Diese Vereinfachung des Abgleichsproblems führt nicht zur Unlösbarkeit, da das zur Aufgabe gehörige Gleichungssystem unterbestimmt ist. Es stehen lediglich die realen Roboterpositionen für die Aufstellung eines Gleichungssystems zur Verfügung. Variable sind jedoch

- die Positionen der Objekte im Layout, die direkt oder indirekt als Trägerelemente für zu handhabende Objekte und den Roboter eingesetzt werden,

- die Position des Greifers am Roboter,
- die Aufstellposition des Roboters,
- und die Greif- bzw. Ablegepositionen der einzelnen zu handhabenden Objekte.

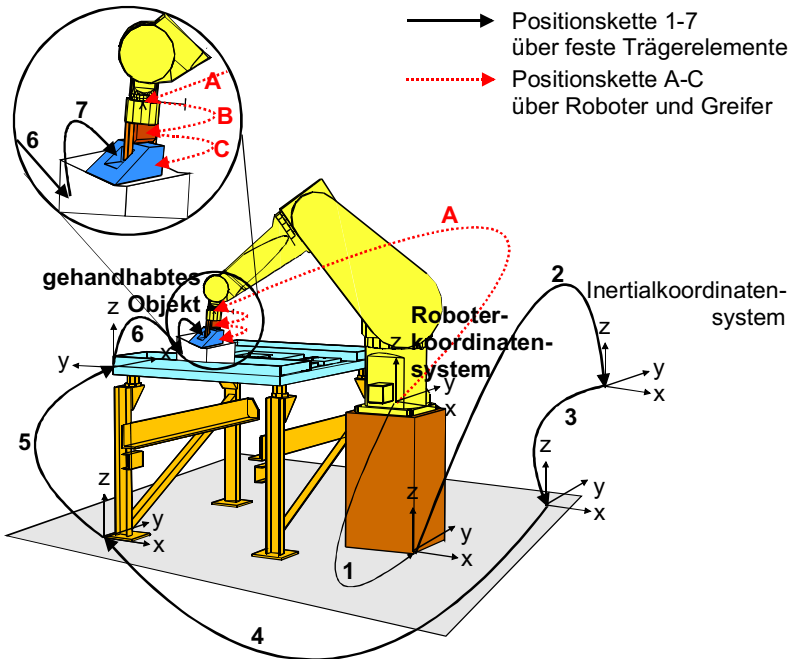


Bild 6-4: Zwei Positionsketten zur Berechnung der Objektposition im Roboterkoordinatensystem

Aus Greif- und Ablegestellungen des Roboters lassen sich die Positionen der Objekte im Layout relativ zum Roboter festlegen, falls die Lage des Objektes relativ zum Greifer und die Lage des Greifers relativ zum Werkzeugkoordinatensystem des Roboters bekannt ist. Zu der Lage des TCP des Roboters (Position A in Bild 6-4) muß dazu die Position des Greifers relativ zum TCP (Position B) und die Position des Objektes relativ zum Greifer (Position C) addiert werden. Die

Objektposition kann jedoch auch aus den Einzelpositionen seiner Trägerelemente im Inertialkoordinatensystem bestimmt werden (Position 3 bis 7 in Bild 6-4). Diese kann durch Berücksichtigung der Roboterposition im Inertialkoordinatensystem, die sich aus seiner Kette von Trägerelementen errechnen läßt (in Bild 6-4 also über Position 1 und 2), in das Roboterkoordinatensystem transformiert werden. Man hat nun zwei Objektpositionen und muß versuchen durch Variation der Einzelpositionen zu erreichen, daß die beiden errechneten Positionswerte identisch sind.

Da bei der Berechnung der Position des gegriffenen Objekts über die Positionen der Trägerelemente sich die resultierende Position aus den variierenden Positionen aller Trägerobjekte in der Positionskette bis zum eigentlich zu platzierenden Objekt zusammensetzt, sind wesentlich mehr Variable zu bestimmen, als Gleichungen zur Verfügung stehen. In seltenen Fällen könnte der Lösungsraum durch die Wahl der zulässigen Positionsvariationen wie oben beschrieben eingeschränkt werden. Dies wird aufgrund der relativ groben Variationswerte jedoch fast nie eintreten. Mit dem über die Positionsketten aufgestellten Gleichungssystem können die Variablen nicht eindeutig bestimmt werden. Es sind mehrere Lösungen möglich, so daß selbst bei der Berücksichtigung zulässiger Variationsbereiche eine Lösung für das Gleichungssystem existiert.

WOENCKHAUS (1994) stellt einen Ansatz für die Optimierung eines Zellenlayouts in der Phase der Layoutplanung zur Verkürzung der Taktzeiten und Roboterachswinkelwege vor (Bild 6-5), dessen Realisierung auf Basis des 3D-Simulationssystems USIS erfolgte. Bei dem von ihm vorgestellten System gibt der Benutzer über eine Bedienoberfläche zunächst die zulässigen Verschiebungen und Verdrehungen der einzelnen Bauteile vor. Ebenso muß er ein Zellenlayout und ein Roboterprogramm für die Handhabungs- oder Montageaufgabe zur Verfügung stellen.

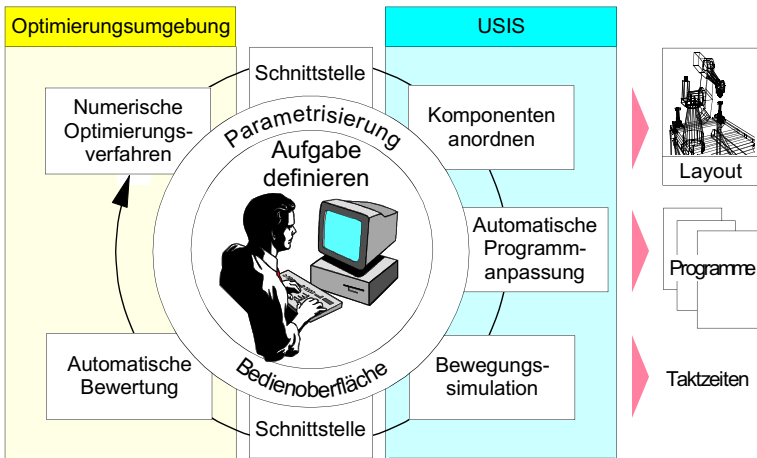


Bild 6-5: System zur Optimierung von Layoutentwürfen von WOENCKHAUS (1994)

Damit das Roboterprogramm bei einer Änderung des Layouts angepaßt werden kann, muß für die Greif- und Ablegepositionen ein Bezugsobjekt definiert sein, auf das sich die Roboterposition bezieht. Somit wird sichergestellt, daß Objekte immer gleich gegriffen und abgelegt werden (Bild 6-6).

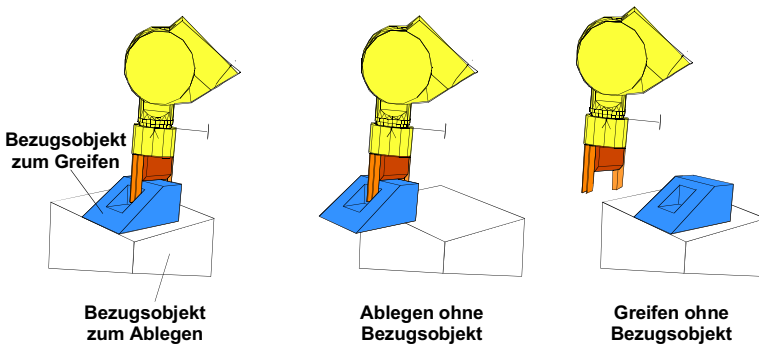


Bild 6-6: Definition von Bezugsobjekten zur Programmanpassung

Die Angaben über zulässige Verschiebungen und Verdrehungen der Objekte, das Zellenlayout und das Roboterprogramm sind zusammen mit dem zu minimierenden Zielkriterium (in der Regel die Taktzeit und die Achswinkelwege des Roboters) die Eingangsdaten für einen Optimierungslauf, bei dem zunächst die Komponenten innerhalb des vorgegebenen Rahmens automatisch anders angeordnet werden. Nach einer automatischen Anpassung des Roboterprogramms an das veränderte Zellenlayout wird das Programm zur Bestimmung der Taktzeit und der Achswinkelwege ausgeführt, und aus diesen Werten schließlich ein Gesamtgütwert für die Layoutvariante berechnet. Ein numerischer Optimierungsalgorithmus errechnet unter Berücksichtigung dieses Gesamtgütwertes (und eventueller Vorgängerwerte) die als nächste zu untersuchende Komponentenanzahl. Dieser Vorgang wird so lange wiederholt, bis der Gütwert sich bei einem lokalen Minimum einpendelt oder eine bestimmte zulässige Rechenzeit verstrichen ist.

Das Verfahren von WOENCKHAUS (1994) läßt sich für die Anpassung des Simulationsmodells an das reale Roboterprogramm übertragen. Es ist jedoch die geänderte Randbedingung zu beachten, daß das Roboterprogramm sich nicht verändern darf, sondern im Gegenteil das innerhalb des Toleranzbereichs veränderte Layout für das Roboterprogramm geeignet sein muß.

Die Layoutanpassung erfolgt dann in zwei Schritten, die in Kap. 6.2.1 und Kap. 6.2.2 ausführlicher behandelt werden:

1. Zuerst wird die Optimierungsaufgabe definiert.
 - Dazu werden aus dem eingefahrenen Roboterprogramm die Positionen der gehandhabten Objekte relativ zum Roboter extrahiert.
 - Aus dem alten Simulationsprogramm werden die zugehörigen Objektpositionen relativ zum (jeweiligen) Greifer ermittelt.
 - Zur vollständigen Definition der Aufgabe werden schließlich mit Hilfe des alten Simulationsprogramms und des zugehörigen Simulationslayouts die Objekte bestimmt, deren Position u. U. verändert werden muß, da sie Trägerobjekte des Roboters oder der zu handhabenden Objekte sind.

2. Im zweiten Schritt werden gültige Layoutvarianten erstellt. Hierzu werden Teile der Layoutoptimierung von Woenckhaus (1994) eingesetzt. Das Programm des Roboters wird jedoch nicht verändert und das Optimierungskriterium ist die Differenz der Positionen der gehandhabten Objekte bei den beiden in Bild 6-4 dargestellten Berechnungsverfahren. Das Optimierungskriterium muß minimiert werden.

6.2.1 Definition der Variationsaufgabe

Für eine einfache Anpassung des Simulationslayouts an das eingefahrene Roboterprogramm soll die Anpassung größtenteils automatisch erfolgen. Dazu müssen weitestgehend automatisch die Randbedingungen für die spätere Variation des Zellenlayouts ermittelt werden. Die einzelnen gehandhabten Objekte und die verwendeten Greifer können aus dem unveränderten Simulationslayout mit dem in der Simulation erstellten Roboterprogramm ermittelt werden. Für jede Greif- und Ablegeposition wird damit das gegriffene Objekt, der verwendete Greifer, die Relativposition zwischen Roboterhand und jeweiligem Greifer sowie die Relativposition zwischen Greifer und gegriffenem Objekt bestimmt.

Zur Bestimmung der Objekte, die zu Positionsungenauigkeiten beitragen, muß das Layout derart aufgebaut sein, daß Objekte, die aufeinander befestigt sind, z. B. eine Palette auf einem Grundgestell oder ein Werkstückträger auf einer Palette, im Simulationslayout entsprechend miteinander verbunden sind. Die Palette muß an dem Grundgestell hängen und sich mit diesem mitbewegen. Die Palette soll für diesen Fall als Sohn, das Grundgestell als Vater bezeichnet werden, wie bei SCHRÜFER (1992) für den Aufbau von Verbindungen in einem 3D-Simulationssystem beschrieben. Werden diese Verbindungen im Layout entsprechend abgebildet, so besitzt das Layout eine Baumstruktur (REINHART & ROBGODERER 1998), die z. B. zur Definition von Freiheitsgraden bei der Layoutplanung verwendet werden kann.

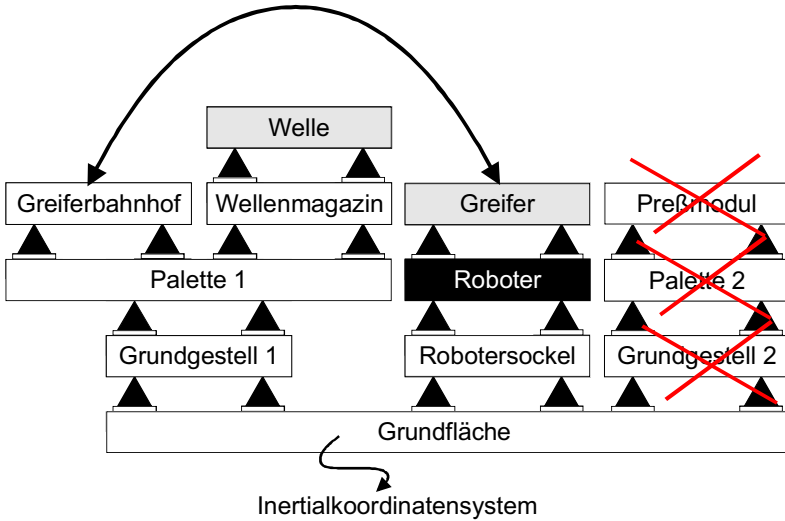
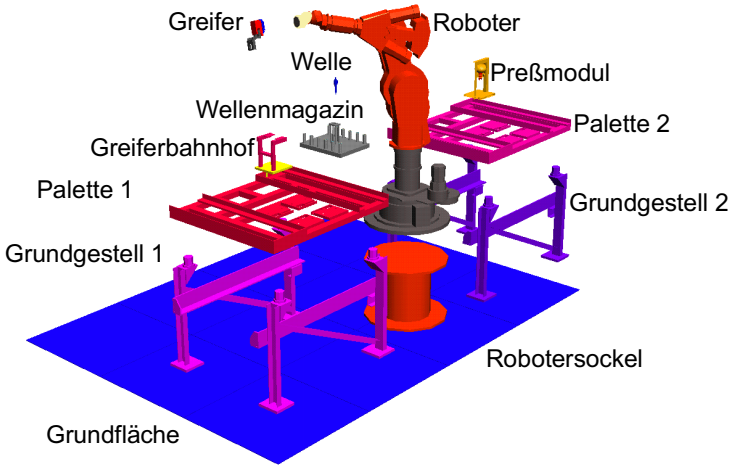


Bild 6-7: Verbindungsbaum nach REINHART & ROßGODERER (1998) und zu betrachtende Layoutkomponenten für den hier vorgestellten Ansatz zum Layoutabgleich

In Bild 6-7 sind die Komponenten einer Roboterzelle sowie die zugehörige Baumstruktur des Layouts abgebildet. Von der Grundfläche zweigen drei Äste ab, von denen einer den Roboter enthält, die beiden anderen je ein Grundgestell mit Palette und darauf befindlichen Komponenten. Gehandhabte Objekte in dieser Struktur sind der Greifer, der aus einem Greiferbahnhof entnommen und in diesem abgelegt wird, sowie eine Welle, die einem Wellenmagazin entnommen wird. In dem vollständigen Montageprogramm wird diese Welle in ein Gehäuse gefügt und das Gehäuse schließlich in das Preßmodul auf Palette 2 gestellt. Hier wird zur Vereinfachung jedoch nur der Ausschnitt betrachtet, der das Entnehmen des Greifers aus dem Bahnhof und das Greifen der Welle beinhaltet.

Auch im allgemeinen Fall werden für den Layoutabgleich die Äste des Strukturbaums vernachlässigt, in denen weder der Roboter noch ein zu handhabendes Objekt ist. Hier wird also der Ast, dessen Wurzel Grundgestell 2 ist, gestrichen. Würde man den Greiferwechsel nicht beachten, so müßte man auch den Ast mit der Wurzel Greiferbahnhof streichen. Für die Variationsaufgabe sind die beiden Äste mit den Wurzeln Robotersockel und Grundgestell 1 relevant. Die Position der Grundfläche der Roboterzelle muß nicht variiert werden, da sie als gemeinsame Wurzel von Roboter und zu handhabenden Objekten keinen Einfluß auf die Eignung des Layouts für das Roboterprogramm hat.

Zur Anpassung des Layouts an das eingefahrene Programm im abgebildeten Beispiel müssen also zwei Gesamtpositionsabweichungen relativ zum Roboter minimiert werden: die Position des Greifers in seinem Bahnhof und die Position der Welle im Wellenmagazin zu Beginn der Montage. Die Differenz wird jeweils durch den Vergleich der zugehörigen Roboterposition im eingefahrenen Programm mit der Roboterposition im zum Simulationsmodell passenden Programm bestimmt.

Beide Positionsabweichungen werden von den zu variierenden Relativpositionen des Robotersockels zur Grundfläche, des Roboters zum Robotersockel, des Greifers zur Roboterhand, von Palette 1 zu Grundgestell 1 und von Grundgestell 1 zur Grundfläche beeinflusst. Dies sind insgesamt 5 zu variierende Relativpositionen für beide Positionsabweichungen. Beim Entnehmen des Greifers aus seinem Bahn-

hof muß zusätzlich die Relativposition des Greiferbahnhofs zu Palette 1 betrachtet werden, beim Greifen der Welle die Relativposition der Welle zum Wellenmagazin sowie dessen Position relativ zu Palette 1. Zu den 5 gemeinsamen zu variierenden Relativpositionen kommen demnach noch 3 für jeweils nur eine der Positionsdifferenzen ausschlaggebende zu variierende Relativpositionen. Da jede Relativposition sechs Freiheitsgrade festlegt, müssen $48 = 8 \times 6$ Variablen so belegt werden, daß sie zu den $12 = 2 \times 6$ Koordinaten aus den Roboterstellungen passen und innerhalb ihrer Variationsbereiche liegen.

Zur Initialisierung der Variationsaufgabe werden die betroffenen Objekte deshalb mit ihren zulässigen Positionsvarianzen versehen und erhalten einen Verweis auf die Positionen von gehandhabten Objekten, die von ihrer Positionierung beeinflusst werden. Interaktive Eingaben des Benutzers sind beim Ablegen von Objekten erforderlich. Bei Ablegevorgängen kann nicht automatisch ermittelt werden, mit welchem anderen Objekt das gehandhabte Objekt nach dem Ablegen verbunden werden soll.

Nach der Definition der Variationsaufgabe stehen Informationen über diejenigen Objekte im Layout zur Verfügung, deren Position an die Randbedingungen des eingefahrenen Roboterprogramms angepaßt werden muß. Dies sind der Roboter und seine Trägerobjekte sowie alle gehandhabten Objekte und deren Trägerobjekte. Für diese ist der Variationsbereich gegeben, die ihre Position relativ zum jeweiligen Trägerobjekt erfahren darf.

6.2.2 Automatische Variation zur Ermittlung einer gültigen Layoutvariante für das eingefahrene Roboterprogramm

Die einzelnen Objektkoordinaten sind die Ein- und Ausgangsparameter für einen numerischen Optimierungsalgorithmus. Der Optimierungsalgorithmus verändert die Parameter im zulässigen Bereich, fordert für die aktuellen Parameterwerte eine Bewertung an und ermittelt aus den bisherigen Varianten mit ihren Bewertungsergebnissen die nächste zu untersuchende Variante. Das Bewertungskrite-

6.2.2 Automatische Variation zur Ermittlung einer gültigen Layoutvariante für das eingefahrene Roboterprogramm

rium ist die Differenz zwischen den betrachteten eingefahrenen Roboterpositionen an den Greif- und Ablegestellen und den entsprechenden Positionen für die aktuelle Layoutvariante. Stimmen beide Roboterpositionen überein, so ist die aktuelle Variante ein mögliches Layout der realen Zelle. Sie ist also eine für das eingefahrene Programm gültige Layoutvariante.

Bei der Ermittlung der Objekte, deren Positionen variiert werden müssen, wurde das Roboterprogramm des ursprünglichen Simulationslayouts in eine Form konvertiert, bei der die für die Optimierung relevanten Roboterpositionen relativ zu den ausschlaggebenden Bezugsobjekten (Bild 6-6) abgelegt sind. Wird ein Objekt gegriffen, so ist es selbst das Bezugsobjekt für die Roboterposition. Wird es abgelegt, so ist sein direktes Trägerobjekt das Bezugsobjekt. Es steht deshalb für jede Layoutvariante das angepaßte Roboterprogramm sofort zur Verfügung. Dieses Programm wird ausgeführt und für jede Greif- und Ablegstellung des Roboters werden die Achswinkelwerte des zu der aktuellen Layoutvariante gehörenden Programms mit den entsprechenden Achswinkelwerten des eingefahrenen Programms verglichen. Für jede Greif- und Ablegstellung i wird ein Gütewert G_i berechnet, der sich aus den quadratischen Abweichungen der einzelnen Achswinkel zusammensetzt:

$$G_i = \sum_{j=1}^{\text{Achszahl}} \Delta q_j^2 = \sum_{j=1}^{\text{Achszahl}} (q_{j,\text{eingefahren}} - q_{j,\text{Layoutvariante}})^2.$$

Der Gesamtgütewert der Layoutvariante berechnet sich dann zu

$$G = \sum_{i=1}^n G_i = \sum_{i=1}^n \sum_{j=1}^{\text{Achszahl}} \Delta q_j^2,$$

wobei n die Anzahl der betrachteten Roboterstellungen ist. Für den Fall $G = 0$ ist die aktuelle Layoutvariante eine gültige Layoutvariante und die Optimierungsaufgabe ist gelöst.

6.2.3 Konzeption des Gesamtsystems zum Layoutabgleich

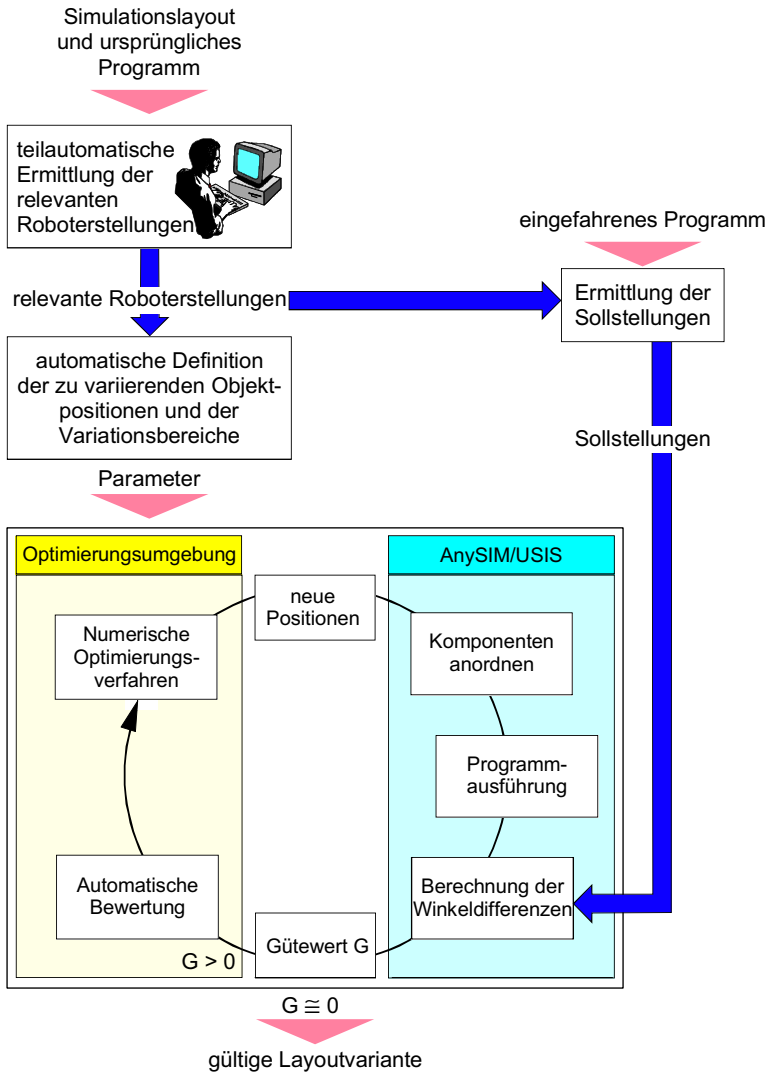


Bild 6-8: Konzeption für das Gesamtsystem zum Layoutabgleich

Basierend auf der Definition der Variationsaufgabe (Kap. 6.2.1) und dem Konzept zur Ermittlung einer gültigen Layoutvariante durch automatische Variation der Objektpositionen wird nun ein Konzept für das Gesamtsystem aufgestellt. Der Ablauf eines Layoutabgleichs für das Systemkonzept ist in Bild 6-8 dargestellt. Der Benutzer stellt ein Simulationslayout mit zugehörigem Roboterprogramm zur Verfügung. Zusätzlich gibt er das eingefahrene Roboterprogramm an. Nach einer teilautomatischen Definition der Variationsaufgabe, bei der der Benutzer nur fehlende Informationen ergänzt, wird schließlich der Optimierungsprozess zur Erstellung einer gültigen Layoutvariante gestartet. So lange die Achswinkel der Roboterbewegung der betrachteten Layoutvariante von den durch das eingefahrene Programm vorgegebenen Sollwinkeln abweichen, werden neue Layoutvarianten erstellt. Erst wenn keine Abweichung mehr vorliegt, ist die aktuelle Variante eine gültige Layoutvariante und die Anpassung des Simulationslayouts ist abgeschlossen.

Dem Planer steht nun ein Zellenlayout zur Verfügung, für das das eingefahrene Programm für den Handhabungsvorgang eingesetzt werden kann. Bei diesem Zellenlayout können u. U. einzelne Objekte miteinander kollidieren, da die Layoutanpassung nur auf dem Niveau der Objektpositionen erfolgt, nicht aber die Ursachen für das veränderte Layout ermittelt werden, wie z. B. ein versetzter Führungsbolzen oder eine schiefe Auflagefläche. WOENCKHAUS (1994) hatte bei seiner Layoutoptimierung für die Planungsphase die Kollisionsfreiheit als Randbedingung eingeführt. Bei der hier durchgeführten Übertragung dieses Optimierungssystems zur Anpassung eines Simulationslayout an die realen Gegebenheiten wäre der Aufwand für eine kollisionsfreie Variante jedoch enorm hoch, da zusätzlich Rückschlüsse auf die Istgeometrie der einzelnen Objekte gezogen werden müssten. Ziel des hier vorgestellten Konzepts ist es jedoch, die Kalibration des Simulationslayouts wesentlich zu vereinfachen.

Beim Einsatz eines auf diesem Konzept basierenden Verfahrens wird es z.B. genügen, wenn der Benutzer eine reale Zelle nur grob vermisst und das Simulationsmodell dann mit einem in der realen Zelle eingefahrenen Programm mit möglichst vielen Roboterpositionen anpaßt. Dieser Ansatz verspricht eine Kompensation der durch die Positionstoleranzen der Zellenkomponenten hervorgerufenen

Abweichungen. Nicht berücksichtigt werden Ungenauigkeiten des Roboters bei der Bewegung. Diese sind aber in der Regel wesentlich geringer als die Positionungenauigkeiten der Zellenkomponenten. Für die weitere Ausarbeitung des Systemkonzepts, die Realisierung eines Systemprototypen und die Verbesserung des Systemkonzepts auf Basis der Testergebnisse mit dem Prototypen besteht jedoch weiterer Forschungsbedarf.

7 Anwendungsbeispiele

Das in den Kapiteln 3 bis 5 vorgestellte Konzept für ein System zur modellgenauigkeitsgerechten Offline-Programmierung wurde weitestgehend als Prototyp in das 3D-Bewegungssimulationssystem AnySIM/USIS implementiert. Die Anwendung der Programmgenerierung wird in diesem Kapitel anhand mehrerer Beispiele für verschiedene Aufgabenstellungen erläutert. Dabei werden insbesondere die Funktionalitäten in den Vordergrund gestellt, die andere aufgabenorientierte Programmiersysteme nicht bieten. Zunächst wird der Einsatz des Systems bei der Zellenplanung und anschließend bei der Programmgenerierung für real existente Zellen gezeigt.

7.1 Greifplanung mit tolerierten oder unterschiedlichen Greifobjekten

Im Planungsstadium bietet das System dem Benutzer die Möglichkeit, die Eignung seiner Betriebsmittel mit Hilfe der automatischen Programmgenerierung zu überprüfen.

Bild 7-1 zeigt zwei Flansche, die sich lediglich im Durchmesser ihrer Innenbohrung unterscheiden. (Es könnte sich auch um einen Flansch mit tolerierter Innenbohrung handeln.) Die Flansche sollen mit einem Zweifinger-Innengreifer an ihrer Innenbohrung aufgenommen werden. Der oben im Bild dargestellte Greifer ist für den Flansch mit der größeren Bohrung sehr gut geeignet (Abbildung links). Seine Finger haben aber selbst in der geschlossenen Stellung nur einen geringen Abstand zum Material (Abbildung oben rechts). Bei dem Flansch mit der engeren Bohrung können die Finger deshalb nicht in die Bohrung eingeführt werden bzw. es kommt bei der Simulation zu einer Kollision.

Der Planer gestaltet die Greiferfinger um und führt die Planung erneut durch (Bild 7-1 unten). Der Greifer ist nun für beide Teile geeignet, die erforderliche Umkonstruktion kann rechtzeitig weitergegeben werden.

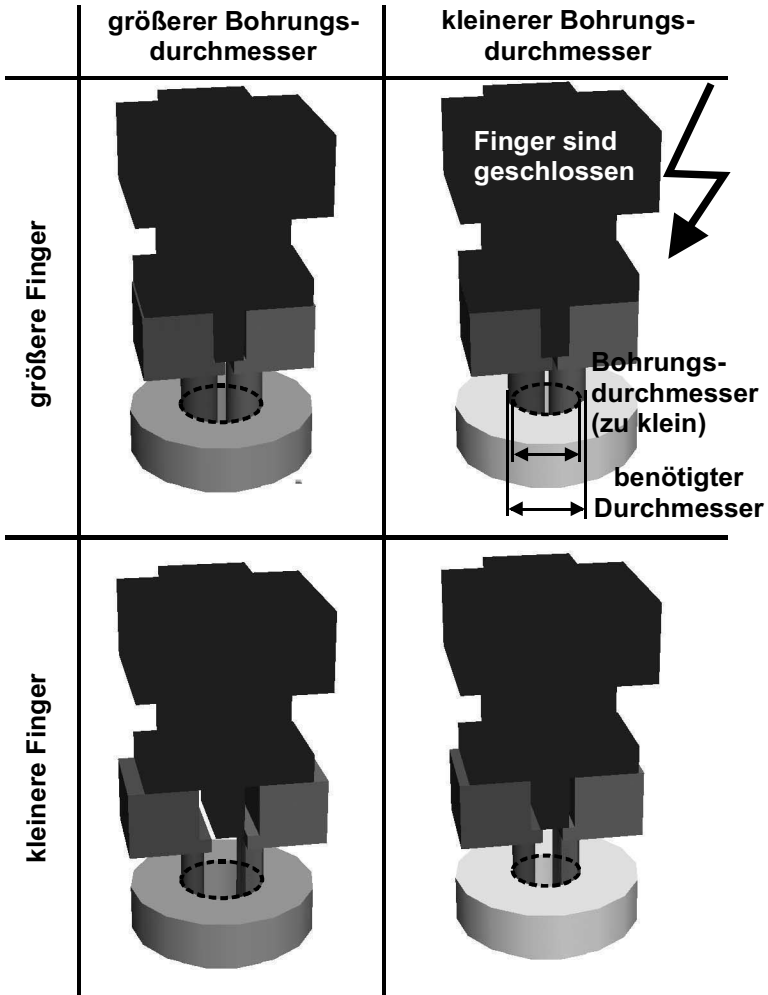


Bild 7-1: Greifer mit verschiedenen Fingern und zu greifendes Objekt mit enger und weiter Innenbohrung.

7.2 Planung von Feinbewegungen

Auch bei der Planung von Feinbewegungen kann Verbesserungsbedarf bei der Zellen- und Betriebsmittelgestaltung aufgedeckt werden. Oft kann ein Objekt nicht einfach mit einer einzelnen Linearbewegung vom Greifort entfernt werden wie z. B. bei der in Bild 7-2 dargestellten Batterie, die leicht mit ihrer Peripherie kollidiert. Mit Hilfe der Feinbewegungsplanung kann über mehrere Zwischenpositionen eine kollisionsfreie Feinbewegung aus Linearbewegungen des Roboters zusammengesetzt werden.

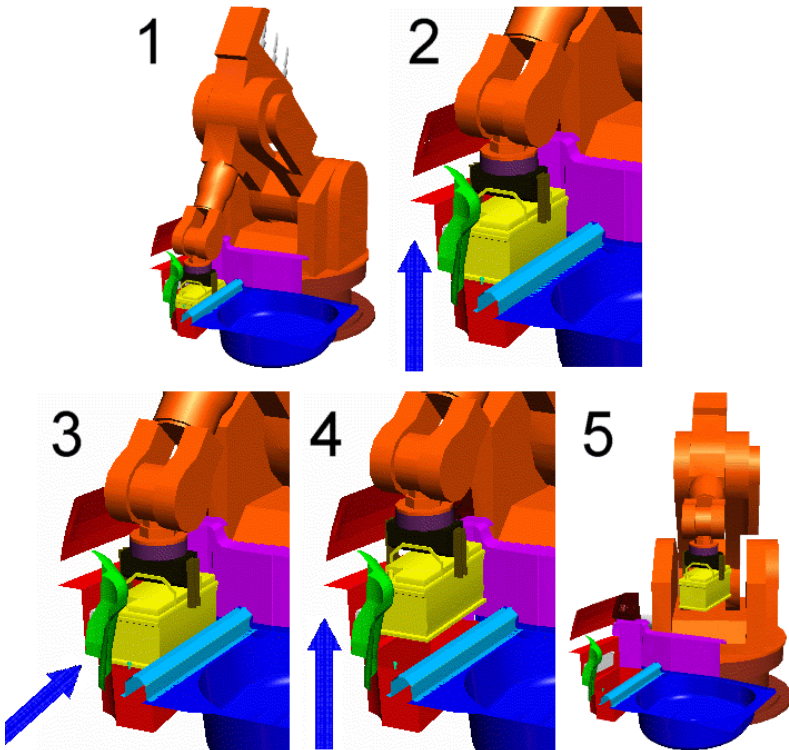


Bild 7-2: Feinbewegung beim Ausbau einer Batterie

Anhand dieser Feinbewegung kann der Planer dann entscheiden, ob diese Lösung akzeptabel ist oder ob die Gestaltung der Betriebsmittel und der Zellenaufbau überdacht werden muß, um eine günstigere Bewegung zu ermöglichen.

Die beiden bisher behandelten Beispiele zeigten die Anwendung des Systems vor allem im Planungsstadium einer Anlage, wenn Betriebsmittel ausgelegt werden müssen und Machbarkeitsstudien wie z. B. Einbausimulationen durchgeführt werden.

7.3 Störungsbehebung durch einen autonomen, mobilen Roboter

Die Programmerstellung für eine bereits real existente Umgebung wurde am Beispiel des autonomen, mobilen Roboters MOBROB des iwv (NABER 1991), der in Bild 7-3 dargestellt ist, erprobt.

Er verfügt über eine Kamera in seinem Handflansch und kann über eine videobasierte Objekterkennung Objekte in seinem Arbeitsraum lokalisieren (BLESSING ET AL. 1996). Die Position der Objekte kann dann im Layout des Simulationssystems angepaßt werden, so daß für die Generierung eines Handhabungsprogramms für das Objekt ein aktuelles Umgebungsmodell zur Verfügung steht. MOBROB wird für den Materialfluß in der Modellfabrik des iwv (SCHNEIDER & GALLASCH 1997) eingesetzt, wo er insbesondere bei der Störungsbehebung seine Fähigkeiten in vollem Umfang einsetzen kann (PISCHELTSRIEDER 1996).

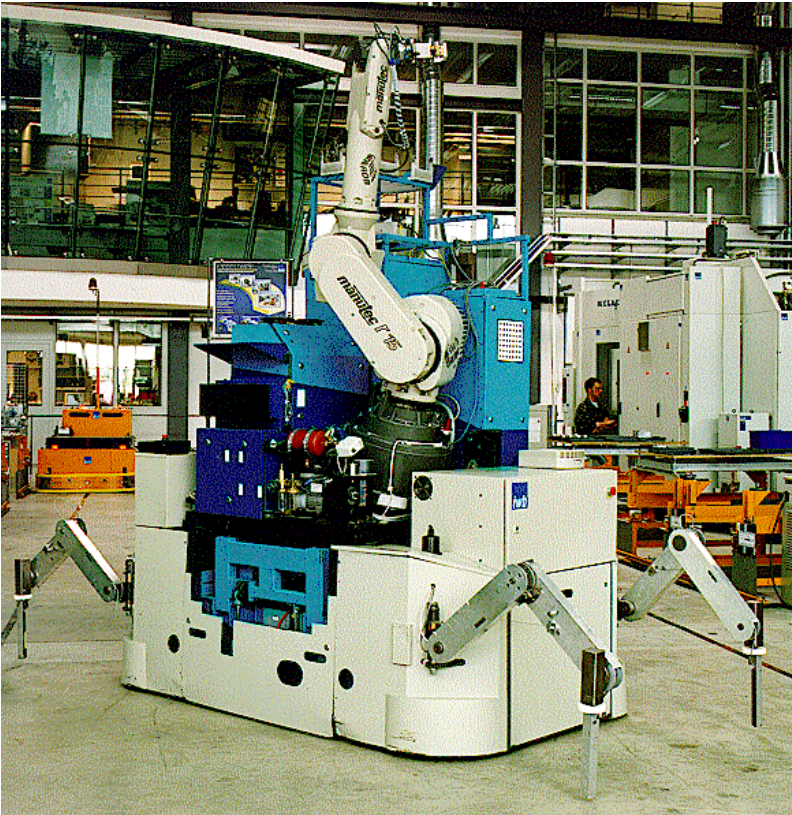


Bild 7-3: Der autonome, mobile Roboter MOBROB

MOBROB kann entweder von einem Fahrzeug aus agieren, z. B. dem in Bild 7-3 vorne abgebildeten Fahrzeug FLEXL II, oder aber stationär, da er vom Fahrzeug auf einem Grundgestell abgesetzt und später von diesem wieder aufgenommen werden kann. Dies wird durch die Montage des Roboters und seiner Peripherie auf einer Standard-Industriepalette möglich. Bild 7-4 zeigt den Roboter neben einem fahrerlosen Transportsystem (FTS) vor einer Drehmaschine, für deren Beschickung mit Material er eingesetzt wird.

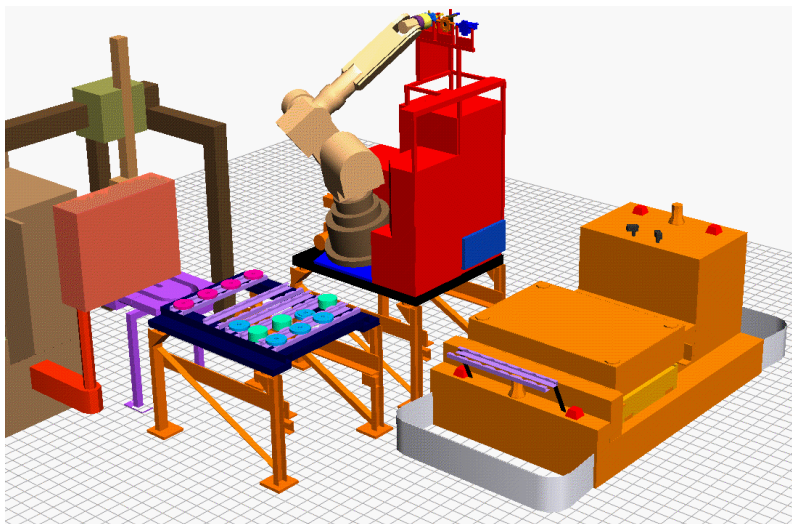


Bild 7-4: MOBROB in einer Drehzelle

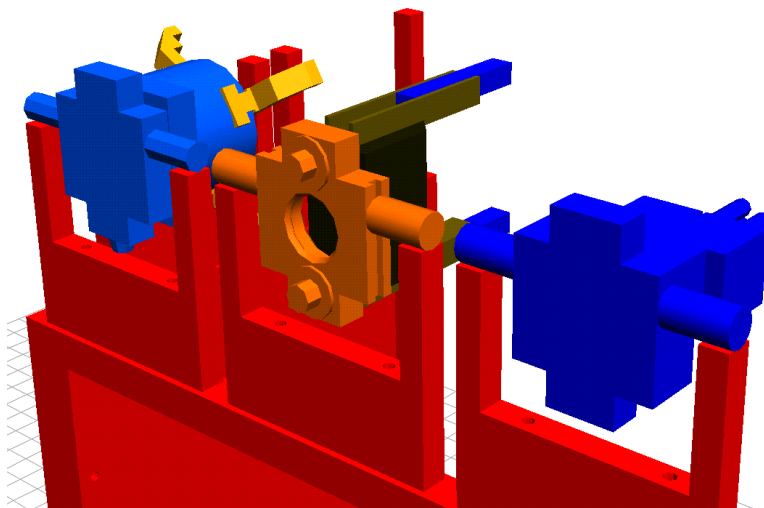


Bild 7-5: verfügbare Greifer

Zur Handhabung kann er drei verschiedene Greifer (Bild 7-5) verwenden, deren Bahnhöfe über seinem Steuerungsschrank montiert sind. In diesem Szenario wird nun ein Beispiel zur Störungsbehebung betrachtet, das mit Hilfe der übergeordneten Handhabungsplanung gelöst wird.

Auf der Materialpalette in Bild 7-6 a liegt ein Fertigteil falsch auf einem Rohteil und muß in die in Bild 7-6 d dargestellte Sollage gebracht werden. Da es sich bei dem zu handhabenden Objekt um ein Drehteil handelt, ist eigentlich der Drei-Finger-Außengreifer (Bild 7-5 links) am besten zum Greifen geeignet. Das Fertigteil hat jedoch einen zu geringen Außendurchmesser und muß zudem für seine in Bild 7-6 d dargestellte Sollposition um 180° gedreht werden. Deshalb kann es weder mit dem Drei-Finger-Außengreifer noch mit dem Innengreifer gegriffen werden.

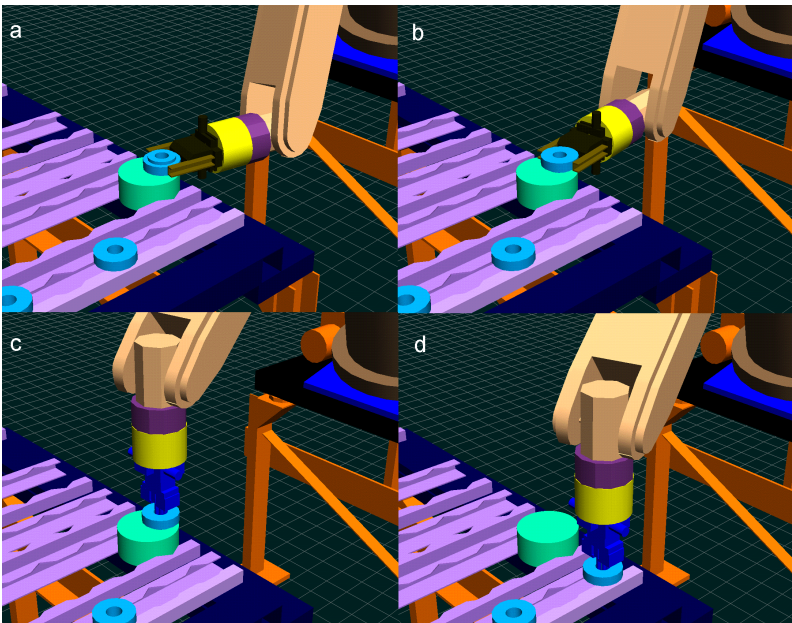


Bild 7-6: Lösung der Handhabungsaufgabe durch Umgreifen an einer Zwischenposition

Ein Wenden ist nur mit dem Zweifingeraußengreifer möglich. Allerdings kann das Teil mit diesem Greifer nicht in der Endlage abgelegt werden, da sonst das Greiferjoch bzw. die Schnittstelle zum Roboter mit der Materialpalette kollidieren würde. Die Handhabungsplanung unterteilt den Handhabungsvorgang deshalb mit einer Zwischenposition in zwei Handhabungsvorgänge (Bild 7-6 b, c). Das Objekt wird zunächst mit dem Zweifingeraußengreifer gegriffen und um 180° gewendet an der Zwischenposition abgelegt. Das Wenden des Fertigteils zwischen der Ausgangsposition und der Zwischenposition ist nur mit dem Zweifingeraußengreifer möglich, da er seine Finger am Zylindermantel des Drehteils sowohl parallel zur Drehachse als auch senkrecht (Bild 7-6 a, b) zur Drehachse ausrichten kann. Das Fertigteil kann mit dem Innengreifer an der Zwischenposition gegriffen werden (Bild 7-6 c) und an der Endposition abgelegt werden (Bild 7-6 d). Die Feinbewegungen können hier als Linearbewegungen in vertikaler Richtung erfolgen. Die Transferbahn wird mit der automatischen Bahnplanung überprüft und muß hier nicht durch Zwischenpositionen ergänzt werden, um Kollisionen zu vermeiden. Die Planungsschritte nach der Greif- und Ablegeplanung sind in dem hier vorgestellten Beispiel also sehr einfach durchzuführen. Ohne die Funktionalität des Umgreifens hätte der autonome Roboter jedoch die Störung in Form des falsch plazierten Fertigteils nicht beheben können.

7.4 Maschinenentladung

Bei dem in Bild 7-7 dargestellten Handhabungsvorgang für das Entnehmen eines Fertigteils aus einem Bearbeitungszentrums ist der direkte Weg zwischen Start- und Zielposition der weiträumigen Transferbewegung nicht kollisionsfrei. Sie wird von der weißen Linie zwischen den beiden die Start- und die Zielposition repräsentierenden großen Würfeln markiert. Diese Linie schneidet die heruntergefahrte Schutztür. Es muß eine Bahnplanung durchgeführt werden.

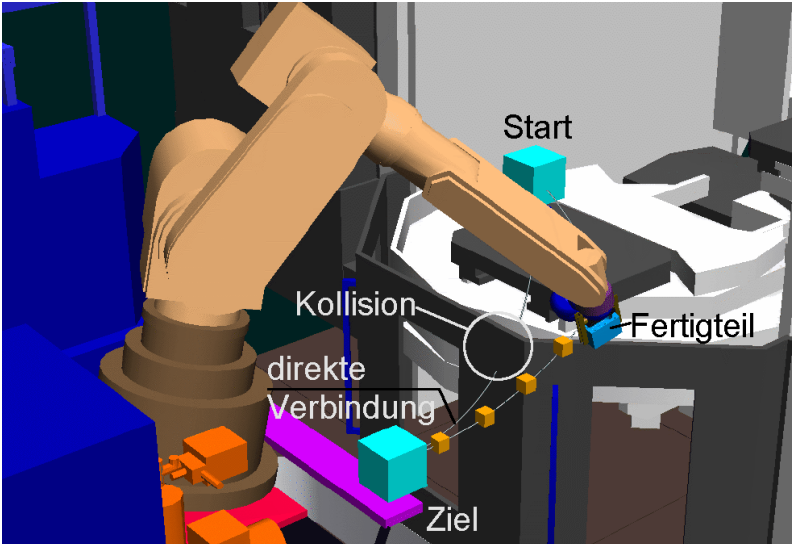


Bild 7-7: Entnahme von Fertigteilen

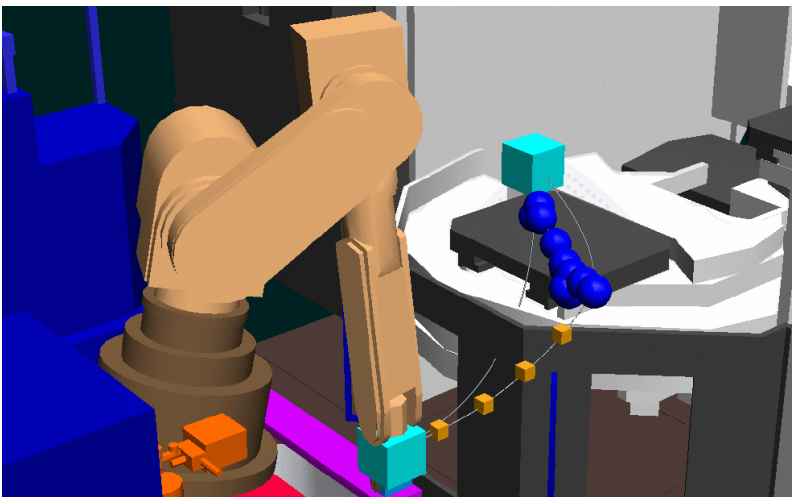


Bild 7-8: Endstellung der Transferbewegung

Der Roboter fährt schließlich etwas seitlich aus der Maschine heraus, bevor er in die Feinbewegung zum Ablegen übergeht, deren Anfangsstellung in Bild 7-8 dargestellt ist. Diese Stellung ist gleichzeitig die Zielstellung der Transferbahn. In Bild 7-8 ist bei dieser Roboterstellung das Fertigteil durch den die Zielstellung der Bahn markierenden großen Würfel verdeckt.

Da die direkte Verbindung zwischen Start- und Zielposition kollisionsbehaftet ist, wird versucht durch das in den Kapiteln 2 und 5 beschriebene Gleiten eine kollisionsfreie Bewegung zu erzeugen. Durch das Gleiten wird eine kollisionsfreie Bewegung von der Startposition über die durch die dunklen Kugeln dargestellten Positionen zur Zielposition ermittelt. In einem anschließenden Optimierungsschritt wird untersucht, ob bei dieser Bewegung auf Zwischenpositionen verzichtet werden kann, so daß schließlich eine Bahn ermittelt wird, die von der Startposition über die in Bild 7-7 dargestellte Position zur Zielposition führt. Längere Bahnstücke werden beim Optimierungsschritt durch zusätzliche Positionen unterteilt, um so weitere Abkürzungen zu finden. Diese Zwischenpositionen zur Unterteilung der längeren Bahnstücke für die Optimierung sind in Bild 7-7 und Bild 7-8 durch kleine Würfel markiert.

7.5 Kommissionierung einer Montagepalette

Bei der in Bild 7-9 gezeigten Situation ist die Bewegung von der Start- zur Zielposition ohne Zwischenposition möglich, so daß sehr weite Winkelwege auftreten. Bei weiten Winkelwegen müssen aber große Hüllkörper bei der Bahnplanung verwendet werden, um eine kollisionsfreie PTP-Bewegung des Roboters zu gewährleisten. MOBROB soll einen Locherdeckel von seiner Ablage, auf der noch ein zweiter Deckel liegt, auf die Montagepalette legen. Die Transferbewegung ist für den dargestellten Fall, bei dem der Oberarm des Roboters vor dem Rumpf bleibt, kollisionsfrei (Bild 7-9 und Bild 7-10 rechts).

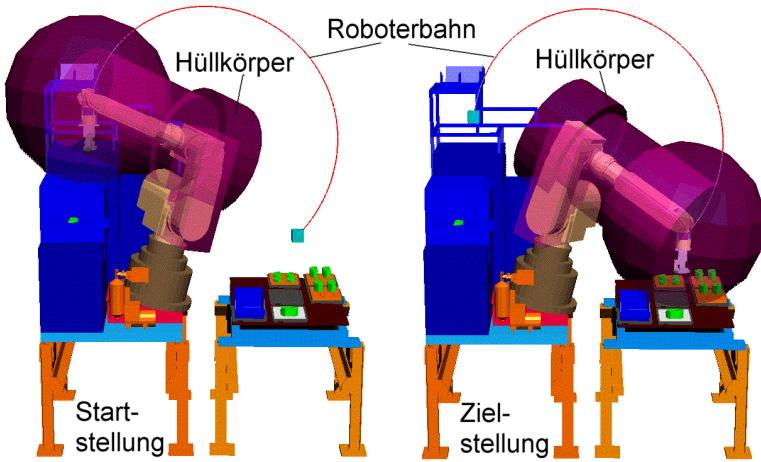


Bild 7-9: Bahnplanung mit Hüllkörpern

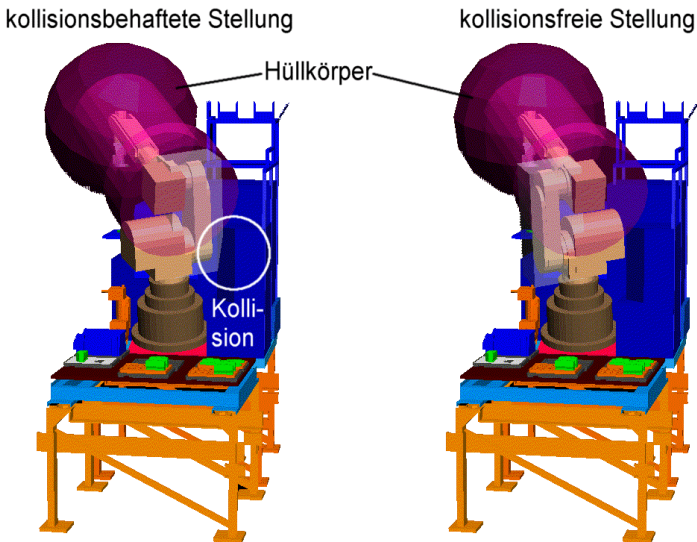


Bild 7-10: Kollision in Abhängigkeit vom Stellungsparameter

Die Greifstellung, bei der der Oberarm sich auf der anderen Seite des Rumpfes befindet, ist nicht für den Handhabungsvorgang geeignet, da hier selbst in der Endstellung der Feinbewegung der Oberarm noch mit dem Steuerschrank des Roboters kollidiert (Bild 7-10 links). Auch eine Herabsetzung des zulässigen Verfahrbereichs führt nicht zu einer Lösung mit dieser Stellung. Die in Bild 7-10 rechts dargestellte Stellung aus Bild 7-9 links dagegen ist nicht kollisionsbehaftet.

8 Zusammenfassung und Ausblick

Der Einsatz der 3D-Simulation zur Offline-Programmierung beschränkt sich in vielen Fällen auf die einmalige Erstellung eines Zellenlayouts mit zugehörigem Roboterprogramm. Ein Einfahren des Programms gestaltet sich oft schwierig, da Ungenauigkeiten im Zellaufbau und Toleranzen der verwendeten Betriebsmittel in der Simulation nicht berücksichtigt werden. Dadurch kommt es mitunter sogar zur kompletten Neuerstellung des Roboterprogramms online. Damit ist der Gesamtaufwand mit Offline-Programmierung nicht mehr niedriger als der Gesamtaufwand mit Programmierung am realen Roboter.

In Bild 8-1 links ist qualitativ der Aufwand mit Programmierung am realen Roboter, in der Mitte der Aufwand für den Idealfall einer Offline-Programmierung dargestellt. Aufgrund der fehlenden Berücksichtigung von Ungenauigkeiten bei der Offline-Programmierung (OLP) bzw. der 3D-Simulation fällt jedoch oft der rechts im Bild dargestellte Aufwand bis zur Begrenzung "OLP mit modellbedingten Fehlern" an. Diese Situation kann sich in der Phase einer Zellenplanung noch verschlimmern, wenn die Toleranzen der einzelnen Zellenkomponenten und der zu handhabenden Produkte sowie der Zellaufbau nicht harmonisieren und eine Umkonstruktion erforderlich ist. Der Zellenbediener beginnt dann zunächst mit dem Einfahren des offline erstellten Programms, erkennt dessen Unzulänglichkeiten, versucht ein besseres Programm am realen Roboter zu erstellen, bis er schließlich bemerkt, daß eine Umkonstruktion oder Umplanung vorgenommen werden muß. Zu der verlorenen Zeit und den damit verbundenen Kosten bei den Versuchen des Zellenbedieners addieren sich die Kosten, die durch die Verzögerung des Produktionsanlaufs in der Roboterzelle entstehen. Außerdem ist nach der Änderung des Zellaufbaus, der Produkts oder einzelner Zellenkomponenten erneut eine Programmanpassung erforderlich.

Zudem werden auch bei dem in der Mitte dargestellten Idealfall mit Offline-Programmierung die eingefahrenen Programme meist nicht in das Simulationsmodell zurückgeführt und das Modell nicht an die realen Gegebenheiten angepaßt. Damit steigt aber der Kostenanteil für die Modellerstellung, da sich die Kosten nicht auf

mehrere Programmiervorgänge, z. B. für das ursprüngliche Produkt und eine spätere Variante, verteilen.

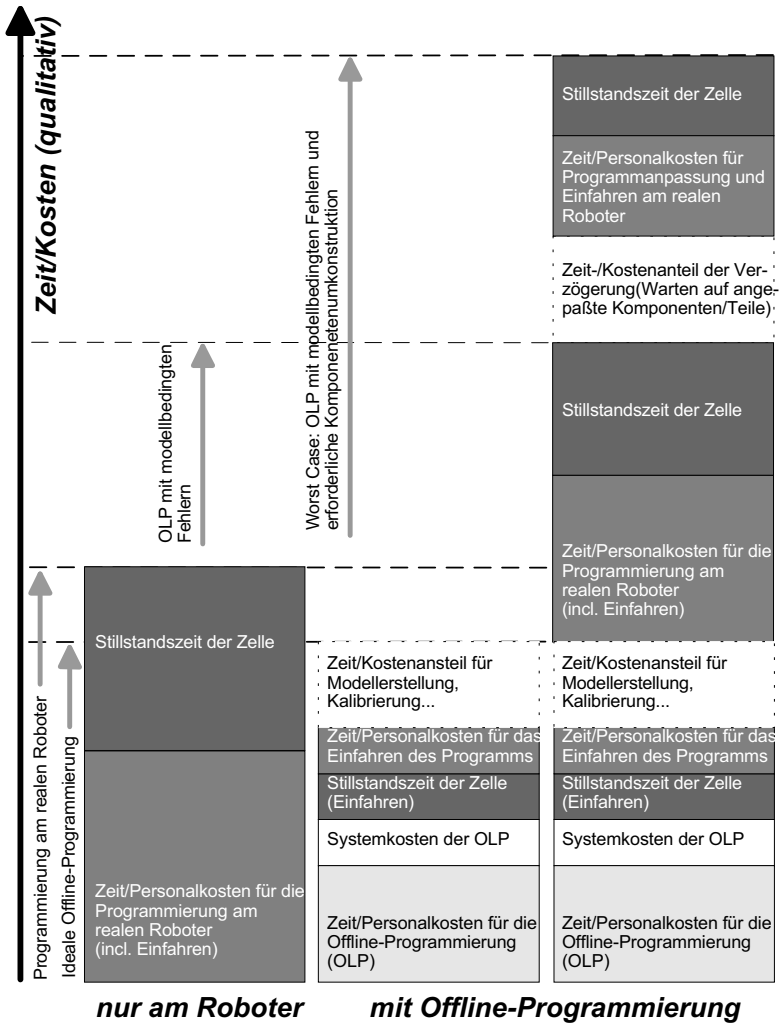


Bild 8-1: qualitativer Vergleich des Aufwands für Programmierung am realen Roboter und Offline-Programmierung

Ursachen für die nicht durchgeführte Rückführung des eingefahrenen Programms ins Modell sind der hohe Aufwand und die unzureichenden Benutzerunterstützung durch das System dabei. Die Benutzerunterstützung läßt auch bei der Programmerstellung zu wünschen übrig, bei der die meisten Systeme nicht wesentlich mehr als die Funktionalitäten des Teach-In-Verfahrens bieten, obwohl gerade ein computergestütztes System die Möglichkeiten zur Bereitstellung aufgabenorientierter, benutzerfreundlicher Planungsmodule bietet.

In der vorliegenden Arbeit wurde ein Konzept für eine modellgenauigkeitsgerechte, aufgabenorientierte und benutzerfreundliche Offline-Programmierung mit der 3D-Simulation aufgestellt. Der realen Umgebung wird dabei mit der Berücksichtigung von Ungenauigkeiten in der Simulation Rechnung getragen. Dadurch wird die Programmierung von Bewegungen, die nicht im gesamten zulässigen Ungenauigkeitsbereich gelten, bereits in der Phase der Offline-Programmierung erkannt, und es kann eine geeignete Umplanung offline erfolgen. Für Montage- und Handhabungsvorgänge wurde ein System zur automatischen, aufgabenorientierten Generierung kollisionsfreier Bewegungsprogramme vorgestellt. Es besteht aus je einem Modul zur Greifplanung, Planung von Feinbewegung, Planung weiträumiger Transferbewegung und übergeordneten Handhabungsplanung. Das System zeichnet sich gegenüber anderen Arbeiten durch die Berücksichtigung von Ungenauigkeiten und eine höhere Lösungswahrscheinlichkeit durch ein Modul zur automatischen Umplanung von Handhabungsvorgängen z. B. durch einen Greiferwechsel oder eine Zwischenposition in der übergeordneten Handhabungsplanung aus. Bei der Bewegungsplanung wird nicht eine mathematisch ideale Bahn hinsichtlich Kollision überprüft, sondern Kollisionsfreiheit für die real ausgeführten PTP-Bewegungen des Roboters gewährleistet. Die Eignung der resultierenden Roboterprogramme zum Einfahren in der realen Zelle wurde damit wesentlich verbessert. Die Berücksichtigung von Ungenauigkeiten erfordert zwar etwas mehr Rechenaufwand für das System, durch die Automatisierung der Planung kann der Benutzer jedoch nach Vorgabe der Planungsaufgabe einer anderen Tätigkeit nachgehen und muß schließlich nur noch das Planungsergebnis beurteilen. Dieser Aufwand fällt zudem auch bei einer manuellen Überprüfung der Toleranzbereiche an. Erst durch die Berücksichtigung von Ungenauigkeiten wird

ein geringerer Aufwand bei Verwendung der Offline-Programmierung als bei der Programmierung am realen Roboter sichergestellt. Nur so rechnet sich der Einsatz der Offline-Programmierung wirklich, da die rechts in Bild 8-1 dargestellten Situationen nicht mehr auftreten bzw. erforderliche Umplanungen und Umkonstruktionen bereits in der Planungsphase erkannt werden und die Kostensituation so wesentlich entschärft wird.

Der Nachweis für die Funktionsfähigkeit des Systems wurde anhand eines Prototypen, dessen Implementierung weitestgehend abgeschlossen ist, für mehrere Anwendungsfälle erbracht. Der Beitrag zur Steigerung der Akzeptanz bei den Systemanwendern - den Planern und Roboterprogrammierern - wird sich jedoch erst bei der Integration derartiger Funktionalitäten in die auf dem Markt verfügbaren Systeme messen lassen.

Zur Erhöhung der Wiederverwendbarkeit der Simulationsmodelle bei einer späteren Umprogrammierung der Roboter wurde ein Konzept zur einfachen Übernahme der realen Gegebenheiten in das Modell der Zelle erstellt, bei dem lediglich das eingefahrene Programm und das ursprüngliche Simulationsmodell benötigt werden. Die Layoutanpassung erfolgt weitestgehend automatisch, Rückfragen an den Benutzer sind nur bei der Definition der Randbedingungen für den Abgleich erforderlich. Dabei werden lediglich die gehandhabten Objekte und deren Trägerobjekte in der Ablageposition erfragt.

Die Berücksichtigung von Ungenauigkeiten bei der Offline-Programmierung und Modellanpassung wird sich zukünftig einfacher implementieren lassen, wenn CAD-Systeme leistungsfähigere Toleranzrechnungsmodule bieten, wozu mit den derzeit verfügbaren Funktionalitäten bereits ein erster Schritt getan ist. Ein weiterer vereinfachender Faktor ist ebenfalls die zunehmende Anzahl von Direktschnittstellen zwischen CAD-Systemen und 3D-Simulationssystemen sowie die wieder häufiger verfolgte Integration von Simulationsprogrammen in ein CAD-System.

Die Planung von Unwägbarkeiten bei Roboterbewegungen gewinnt auch mit der weiteren Verbreitung von Virtual-Reality-Techniken und Servicerobotern an Bedeutung, weil so in Zukunft vielleicht die Gefahr, die für einen Benutzer vom

Serviceroboter ausgeht, durch die Immersion des Nutzers in eine VR-Umgebung bereits in der Planungsphase richtig eingeschätzt werden kann. Damit können dann auch Situationen berücksichtigt werden, in denen z. B. die Sicherheitssensorik des Roboters ausgefallen ist.

Literaturverzeichnis

ADOLPHS & NAFZIGER 1990

Adolphs, P.; Nafziger, D.: Schnelle kollisionsvermeidende Bahnplanung im Konfigurationsraum mit Entfernungsfeldern. Robotersysteme 6 (1990) 3/4, S. 236-244.

ANGERBAUER ET AL. 1996

Angerbauer, R.; Gronbach, H.; Hohenadel, J.: Schneller zur betriebsbereiten Maschine - Maschinen- und Steuerungssysteme effizient entwickeln und simulieren. wt-Produktion und Management 86 (1996), S. 32-36.

BAGINSKI 1996

Baginski, B.: Local Motion Planning for Manipulators based on Shrinking and Growing Geometry Models. In: Proceedings of the 1996 IEEE International Conference on Robotics and Automation. Minneapolis, Minnesota, April 1996, S. 3303-3308.

BAGINSKI 1997

Baginski, B.: Efficient Dynamic Collision Detection Using Expanded Geometry Models. In: Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems IROS '97. Grenoble, September 1997, S. 1714-1719.

BARRAQUAND & LATOMBE 1991

Barraquand, J.; Latombe, J.-C.: Robot Motion Planning: A Distributed representation Approach. International Journal of Robotics Research 10 (1991) 6, S. 628-649.

BARTHOLD 1998

Barthold, H. M.: Engineering aus der Reserve locken. Automobil Industrie (1998) 4, S. 78-80.

BAUER 1998

Bauer, L.: Strategien zur rechnergestützten Offline-Programmierung von 3D-Laseranlagen. Dissertation TU München 1998.

BEHRENS & THATER 1992

Behrens, A.; Thater, G.: Off-line-programmierte Bewegungsplanung für Industrieroboter auf der Grundlage eines fehlerkompensierenden Identifikationsverfahrens. Robotersysteme 8 (1992) 3, S. 209-219.

BERNHARDT ET AL. 1994

Bernhardt, R.; Jacobi, A.; Schreck, G.; Willnow, C.: Realistische Simulation von Industrierobotern. Zwf 89 (1994) 4, S. 159-162.

BLESSING ET AL. 1996

Blessing, S.; Kugelmann, D.; Reinhart, G.: Sichere Handhabung mit 3D-Simulation und visueller Sensorik. In: Schmidt, G.; Freyberger, F. (Hrsg.): Autonome mobile Systeme 1996, München. München: Springer 1996, S. 204-213. (Informatik aktuell)

BÖTTCHER 1986

Böttcher, P.: Technisches Zeichnen. Stuttgart: Teubner, 1986.

BORGOLTE 1993

Borgolte, U.: IRL - Die deutsche Norm für explizite Roboterprogrammierung. In: Intelligente Steuerung und Regelung von Robotern. Düsseldorf: VDI, 1993, S. 535-544. (VDI Berichte Nr. 1094)

BORGOLTE 1994

Borgolte, U.: Real time, rule based collision avoidance for two robots. In: Proceedings of the IFAC Robot Control. Capri, Italy, 1994, S. 693-698.

CACCAVALE & CHIACCHIO 1994

Caccavale, F.; Chiacchio, P.: Energy-Based Identification of Dynamic Parameters for a Conventional Industrial Manipulator. In: Proceedings of the IFAC Robot Control. Capri, Italy, 1994, S. 547-552.

CUIPER & ROßGODERER 1997

Cuiper, R.; Roßgoderer, U.: Montageanlagen planen und steuern. In: Die neue Fabrik - Modellfabrik als Denkmodell. Landsberg: moderne industrie 1997, S. 58-60.

DENEB ROBOTICS, INC. 1997

Deneb Robotics, Inc. (Hrsg.): Integrated Solutions for your entire organization (Produktbeschreibung). Deneb Robotics 1997.

DESROCHERS & RIVIÈRE 1997

Desrochers, A.; Rivière, A.: A Matrix Approach to the Representation of Tolerance Zones and Clearances. International Journal of Advanced Manufacturing Technology 13 (1997), S. 630-636.

DIN 7168 1970

DIN 7168: Allgemeintoleranzen (Maße ohne Toleranzangabe). Berlin: Beuth 1970.

DIN EN 29283 1993

DIN EN 29283: Industrieroboter: Leistungskriterien und zugehörige Testmethoden. Berlin: Beuth 1993.

DILLMANN 1998

Dillmann, R.: Robotik ein multidisziplinäres Forschungsgebiet. In: Prassler, E.; Dillmann, R.; Kuntze, H.-B.: Robotik in Deutschland - Lehre, Forschung und Entwicklung. Aachen: Shaker 1998, S. 9-11.

DREWS & WEYRICH 1997

Drews, P.; Weyrich, M.: A System for Digital Mock-up's and Virtual Prototype Design in Industry: 'The Virtual Workbench'. In: ISIE '97 - Proceedings of the IEEE International Symposium on Industrial Electronics, Guimarães, Portugal 1997, S. 1292-1296.

DOMSCHKE 1972

Domschke, W.: Kürzeste Wege in Graphen: Algorithmen, Verfahrensvergleiche. Meisenheim am Glan: Hain, 1972.

DUELEN & SCHRÖER 1991

Duelen, G.; Schröer, K.: Roboter-Kalibration durch Abstandsmessungen. Robotersysteme 7 (1991) 1, S. 33-36.

FERRETTI ET AL. 1994

Ferretti, G.; Magnani, G.; Rocco, P.: Stiffness Constants Identification Methods for Industrial Robot Joints. In: Proceedings of the IFAC Robot Control. Capri, Italy, 1994, S. 541-546.

FISCHER & JACOB 1998

Fischer, R.; Jacob, R.: Automatischer Tischler. Flexible Automation (1998) 2, S. 50-51.

FOITZIK 1990

Foitzik, B. (Hrsg.): Marktübersicht: Roboter Off-Line-Programmiersysteme. Roboter 7 (1990) 5, S. 30-31

FREUND ET AL. 1993

Freund, E.; Uthoff, J.; Hypki, A.; van der Valk, U.: COSIMIR und PCROB: Integration von Zellensimulation und Robotersteuerung auf PCs. In: Intelligente Steuerung und Regelung von Robotern. Düsseldorf: VDI, 1993, S. 823-833. (VDI Berichte Nr. 1094)

FREUND ET AL. 1994

Freund, E.; Hypki, A.; Uthoff, J.: COSIMIR: Simulation of Complete Robotic Workcells. In: Gill, R.; Syan, C. S. (Hrsg.): Proceedings of the Twelfth International Conference on CAD/CAM Robotics and Factories of the Future. London, England: Middlesex University Press 1996, S. 37-42.

FREUND & ROßMANN 1997

Freund, E.; Roßmann, J.: The Projective Virtual Reality Approach to Smart Man-Machine Interfaces. In: Proceedings of the International Conference

on Field & Service Robotics (FSR '97), Canberra, Australia, Dec. 8-10., 1997.

GAUTIER ET AL. 1994

Gautier, M.; Khali, W.; Presse, C.; Restrepo, P. P.: Experimental Identification of Dynamic Parameters of Robot. In: Proceedings of the IFAC Robot Control. Capri, Italy, 1994, S. 553-558

GEIGER ET AL. 1998

Geiger, M.; Backes, F.; Menzel, T.: Technology Oriented Offline-Programming of 3D Laser Machines. Production Engineering 5 (1998) 1, S. 121-126.

GLAVINA 1991

Glavina, B.: Planung kollisionsfreier Bewegungen für Manipulatoren durch Kombination von zielgerichteter Suche und zufallsgesteuerter Zwischenzielerzeugung. Dissertation TU München. Institut für Informatik 1991.

GLAVINA & HEIB 1993

Glavina, B.; Heiß, H.: CAD-gestützte Bewegungsplanung in der industriellen Einbausimulation. In: Intelligente Steuerung und Regelung von Robotern. Düsseldorf: VDI, 1993, S. 579-586. (VDI Berichte Nr. 1094)

GUILFORD & TURNER 1993

Guilford, J.; Turner, J.: Representational primitives for geometric tolerancing. Computer-Aided Design 25 (1993) 9, S. 577-586.

GUPTA ET AL. 1995

Gupta, S. K.; Regli, W. C.; Das, D.; Nau, D. S.: Automated Manufacturability Analysis: A Survey. In den USA erhältlich als NIST IR 5713, CS-TR-3404, UMIACS-TR-95-08, ISR-TR-95-14.

HECK 1995

Heck, H.: Ein frei konfigurierbares System zur aufgabenorientierten Programmierung von Robotern - Konzept und Realisierung. Dissertation Fern-Universität Hagen. Wetter: Evangelische Stiftung Vollmarstein - Forschungsinstitut Technologie-Behindertenhilfe, 1995.

HEINE & SCHNARE 1991

Heine, R.; Schnare, T.: Kollisionsfreie Bahnplanung für Roboter. Robotersysteme 7 (1991) 1, S. 17-22.

HEINEMEIER 1993

Heinemeier, H.-J.: Programmiergerechte Betriebsmittelgestaltung von roboterintegrierten Montagezellen. Dissertation TU Berlin. München: Hanser, 1993. (Produktionstechnik - Berlin 109).

HEISEL ET AL. 1997

Heisel, U.; Richter, F.; Wurst, K.-H.: Thermal Behaviour of Industrial Robots and Possibilities for Error Compensation. *Annals of the CIRP* 46 (1997) 1, S. 283-286.

HEß 1994

Heß, P.: Programmierverfahren für Industrieroboter. *wt-Produktion und Management* 84 (1996), S. 474-478.

HIDALGO & BRUNN 1998

Hidalgo, F.; Brunn, P.: Robot metrology and calibration systems - a market review. *Industrial Robot* 25 (1998) 1, S. 42-47.

HÖRMANN 1987

Hörmann, K.: Kollisionsfreie Bahnen für Industrieroboter. Dissertation Universität Karlsruhe. Berlin: Springer, 1987. (Informatik-Fachbericht Nr. 166).

HÖRMANN & HÖRMANN 1990

Hörmann, A.; Hörmann, K.: Planung kollisionsfreier Greifoperationen: Analyse der Objektgeometrie. *Robotersysteme* 6 (1990) 1, S. 39-50.

HÖRMANN & WERLING 1989

Hörmann, K.; Werling, V.: Ein Verfahren zur Planung von Feinbewegungen für Montageoperationen. *Robotersysteme* 5 (1989) 1, S. 17-28.

HÖRMANN & WERLING 1990

Hörmann, K.; Werling, V.: Planung kollisionsfreier Greifoperationen: Kollisionsfreie Bahnplanung für Greifer und Manipulator. *Robotersysteme* 6 (1990) 2, S. 119-125.

HOLLENBERG 1995

Hollenberg, F.: CAD-basierte Off-line Programmierung von Lichtbogen-schweißrobotern. Dissertation RWTH Aachen. Aachen: Shaker, 1995. (wzl / IPT - Berichte aus der Produktionstechnik 14/95).

HSU ET AL. 1993

Hsu, W.; Lee, C. S. G.; Su, S. F.: Feedback approach to design for assembly by evaluation of assembly plan. *Computer-Aided Design* 25 (1993) 7, S. 395-410

HUCK 1990

Huck, M.: Produktorientierte Montageablauf- und Layoutplanung für die Roboter montage. Dissertation Universität Karlsruhe. Düsseldorf: VDI, 1990.

JAMES & GUPTILL 1993

James, P. A.; Guptill, R.: Using Simulation to Evaluate Robot Programming Decision Making. *Industrial Robot* 20 (1993) 4, S. 26-31.

JUSTER 1992

Juster, N.P.: Modelling and representation of dimensions and tolerances: a survey. *Computer-Aided Design* 24 (1992) 1, S. 3-17.

KLEIN 1994

Klein, B.: Mit statistischer Tolerierung die Herstellkosten senken. *Konstruktion* 46 (1994) 12, S.405-410.

KROTH 1998

Kroth, E.: Roboter mit CCD-Kamera handhabt Ziegel-Rohlinge. *Maschinenmarkt* 104 (1998) 14, S. 52-55.

KUBOTA ET AL. 1996

Kubota, N.; Fukuda, T.; Shimojima, K.: Trajectory planning of cellular manipulator system using virus-evolutionary genetic algorithm. *Robotics and Autonomous Systems* 19 (1996), S. 85-94.

KÜHN & VISSER 1992

Kühn, W.; Visser, A.: Identifikation der Systemparameter 6-achsiger Gelenkarmroboter mit Hilfe der Evolutionsstrategie. *Robotersysteme* 8 (1992) 3, S. 123-133.

KUGELMANN 1993

Kugelmann, F.: Einsatz nachgiebiger Elemente zur wirtschaftlichen Automatisierung von Produktionssystemen. Dissertation TU München. München: Springer, 1993. (iwb-Forschungsberichte 67).

KUGELMANN 1994A

Kugelmann, D.: Autonomous Robotic Handling Applying Sensor Systems and 3D Simulation. In: Proceedings of the 1994 IEEE International Conference on Robotics and Automation, San Diego, USA. Los Alamitos: IEEE Computer Society Press 1994, S. 196-201. (Vol. 1)

KUGELMANN 1994B

Kugelmann, D.: Automatische Online-Generierung von Handhabungsprogrammen mit der 3D-Simulation. In: Levi, P.; Bräunl, Th. (Hrsg.): *Autonome Mobile Systeme* 1994, Stuttgart. Berlin: Springer 1994, S. 349-360. (Informatik aktuell)

KUKA ROBOTER GMBH 1997

KUKA Roboter GmbH (Hrsg.): Offline-Programmierung: KR C1 Office / PC-Simulation: KR SIM (Produktbeschreibung). KUKA Roboter GmbH 1997.

LATOMBE ET AL. 1997

Latombe, J.-C.; Wilson, R. H.; Cazals, F.: Assembly Sequencing with tolerated parts. *Computer-Aided Design* 29 (1997) 2, S. 159-174.

LEE & JOHNSON 1993

Lee, J. and Johnson, G.E.: Optimal tolerance allotment using a genetic algorithm and truncated Monte Carlo Simulation. *Computer-Aided Design* 25 (1993) 9, S. 601-611.

LEE & WEI 1998

Lee, Y.-C.; Wei, C.-C.: Process Capability-Based Tolerance Design to Minimise Manufacturing Loss. *International Journal of Advanced Manufacturing Technology* 14 (1998), S. 33-37.

LI 1989

Li, W.: Automatische Bestimmung kollisionsfreier Bewegungsbahnen für Industrieroboter. *Robotersysteme* 6 (1990) 3/4, S. 218-224.

LIN & LU 1997

Lin, G. C. I.; Lu, T.-F.: Vision and force/torque sensing for calibration of industrial robots. *Industrial Robot* 24 (1997) 6, S. 440-445.

LINDEMANN & IRLINGER 1996

Lindemann, U.; Irlinger, R.: Documentation of Product Logic. In: Proceedings of the Second World Conference on Integrated Design and Process Technology, Austin, Texas, 1996, S. 91-96 (Vol. 3).

LINDEMANN ET AL. 1997

Lindemann, U.; Irlinger, R.; Gaul, H.-D.: Virtual Reality im Praxiseinsatz - Ergebnisse einer Umfrage. *ZwF* 92 (1997) 4, S. 172-174.

LINNER 1997

Linner, S.: Planungstool der Zukunft. In: Die neue Fabrik - Modellfabrik als Denkmodell. Landsberg: moderne industrie 1997, S. 34-36.

LINNER & WUNSCH 1998

Linner, S.; Wunsch, A.: Durchgängige Verifikation von Produkt und Produktion mit prozesskettenorientierten Simulationswerkzeugen. In: Prozessketten für die virtuelle Produktentwicklung in verteilter Umgebung. Düsseldorf: VDI, 1998, S. 395-409. (VDI Berichte Nr. 1435)

LIU & HU 1995

Liu, S.C.; Hu, S.J.: An Offset Finite Element Model and its Applications in Predicting Sheet Metal Assembly Variation. *International Journal of Machine Tools & Manufacture* 35 (1995) 11, S. 1545-1557.

LIU ET AL. 1996

Liu, S. C.; Hu, S. J.; Woo, T. C.: Tolerance Analysis for Sheet Metal Assemblies. Journal of Mechanical Design - Transactions of the ASME, 118 (1996) 3, S. 62-67.

LOZANO-PÉREZ 1987

Lozano-Pérez, T.: A Simple Motion-Planning Algorithm for General Robot Manipulators. IEEE Journal of Robotics and Automation 3 (1987) 3, S. 224-238

LOZANO-PÉREZ ET AL. 1989

Lozano-Pérez, T.; Jones, J. L.; Mazer, E.; O'Donnell, P. A.: Task-Level Planning of Pick-and-Place Robot Motions. Computer 22 (1989) 3, S. 21-29.

LÜTH 1993

Lüth, T.: Automatisierte Layoutplanung von Roboter-Fertigungszellen. Dissertation Universität Karlsruhe 1993.

MILBERG 1997

Milberg, J.: Produktion - Eine treibende Kraft unserer Volkswirtschaft. In: Reinhart, G.; Milberg, J. (Hrsg.): Münchener Kolloquium '97: Mit Schwung zum Aufschwung - Information, Inspiration, Innovation. Landsberg/Lech: moderne industrie 1997, S. 17-39.

MILBERG ET AL. 1997

Milberg, J.; Reinhart, G.; Trunzer, W.: Strategies for On-line Path Planning for Robots with 3D Contour-Following Sensors. Production Engineering 4 (1997) 1, S. 45-50.

MOCTEZUMA 1996

Moctezuma de la Barrera, J. L.: Ein durchgängiges System zur computer- und roboterunterstützten Chirurgie. Dissertation TU München. Berlin: Springer, 1996. (iwb-Forschungsberichte 99).

NABER 1991

Naber, H.: Aufbau und Einsatz eines mobilen Roboters mit unabhängiger Lokomotions- und Manipulationskomponente. Dissertation TU München. Berlin: Springer, 1991. (iwb-Forschungsberichte 36).

NEUGEBAUER ET AL. 1994

Neugebauer, J.; Flaig, T.; Wapler, M.; Däinghaus, R: Virtuelle Realität am IPA - Demonstrationszentrum Virtuelle Realität. Stuttgart-Vaihingen: Fraunhofer-Institut für Produktionstechnik und Automatisierung (IPA), 1994.

NNAJI 1993

Nnaji, B. O.: Theory of Automatic Robot Assembly and Programming. London: Chapman & Hall, 1993.

OSTERWINTER 1992

Osterwinter, M.: Steuerungsorientierte Robotersimulation. Dissertation RWTH Aachen. Braunschweig: Vieweg, 1992. (Fortschritte der Robotik 10).

OWENS 1996

Owens, J.: Task planning in robot simulation. Industrial Robot 23 (1996) 5, S. 21-24.

PAPANIKOLOPOULOS & SMITH 1998

Papanikolopoulos, N.; Smith., C. E.: Issues and experimental results in vision-guided robotic grasping of static or moving objects. Industrial Robot 25 (1998) 2, S. 134-140.

PAHL & FLEMMING 1997

Pahl, G.; Flemming, M.: Grundlagen der Konstruktionstechnik. In: Beitz, W.; Grote, K.-H. (Hrsg.): Dubbel - Taschenbuch für den Maschinenbau. Berlin: Springer 1997, Kapitel F.

PEPER 1998

Peper, S.: Schweißstruktur-orientierte Offline-Programmierung von Lichtbogenschweißrobotern. Dissertation RWTH Aachen. Aachen: Shaker, 1998. (wzl / IPT - Berichte aus der Produktionstechnik 15/98).

PFEIFFER 1996

Pfeiffer, F.: Assembly processes with robotic systems. Robotics and Autonomous Systems 19 (1996), S. 151-166.

PING ET AL. 1995

Ping, J.; Mingyang, K.; Ahluwalia, R.S.: Computer-Aided Operational Dimensioning for Process Planning. International Journal of Machine Tools & Manufacture 35 (1995) 10, S.1353-1362.

PISCHELTSRIEDER 1996

Pischeltsrieder, K.: Steuerung autonomer mobiler Roboter in der Produktion. Dissertation TU München. Berlin: Springer, 1996. (iwb-Forschungsberichte 102).

PRASSLER ET AL. 1998

Prassler, E.; Dillmann, R.; Kuntze, H.-B.: Robotik in Deutschland - Lehre, Forschung und Entwicklung. Aachen: Shaker 1998.

PRITSCHOW ET AL. 1993

Pritschow, G.; Gronbach, H.; Uhl, A.: Dynamiksimulation als Komponente einer umfassenden Roboterentwicklungsumgebung. In: Intelligente Steuerung und Regelung von Robotern. Düsseldorf: VDI, 1993, S. 545-556. (VDI Berichte Nr. 1094)

QAMHIYAH ET AL 1996

Qamhiyah, A. Z.; Venter, R. D.; Benhabib, B.: Geometric reasoning for the extraction of form features. Computer-Aided Design 28 (1996) 11, S. 887-903.

RALLI & HIRZINGER 1994

Ralli, E.; Hirzinger, G.: Fast Path Planning for Robot Manipulators Using Numerical Potential Fields in the Configuration Space. In: Proceedings of the 1994 IEEE/RSJ International Conference on Intelligent Robots and Systems, S. 1922-1929.

RAO & GOLDBERG 1993

Rao, A. S.; Goldberg, K. Y.: On the Relation between Friction and Part Shape in Parallel Jaw Grasping. In: 1993 IEEE International Conference on Robotics and Automation, Atlanta, Georgia, USA. Los Alamitos, CA: IEEE Computer Society Press 1993, S. 461-466.

REICHENEDER 1997

Reicheneder, J.: A Pro/DEVELOP module for dynamic simulation with consideration of tolerances. In: Proceedings of the 1997 Pro/USER International Conference. Orlando, USA 1997.

REINHART 1997

Reinhart, G.: Innovative Prozesse und Systeme - Der Weg zu Flexibilität und Wandlungsfähigkeit. In: Reinhart, G.; Milberg, J. (Hrsg.): Münchener Kolloquium '97: Mit Schwung zum Aufschwung - Information, Inspiration, Innovation. Landsberg/Lech: moderne industrie 1997, S. 173-202.

REINHART & FELDMANN 1997

Reinhart, G.; Feldmann, K.: Simulation - Schlüsseltechnologie der Zukunft? Stand und Perspektiven. München: Herbert Utz 1997.

REINHART ET AL. 1998

Reinhart, G.; Gräser, R.-G.; Klingel, R.: Automatisierung von Meßabläufen mit Standardrobotern. Maschinenmarkt 104 (1998) 14, S. 48-51.

REINHART & VON PRAUN 1998

Reinhart, G.; von Praun, S.: Digital Mock-Up Process Simulation - Next Generation Product Development. In: Proceedings of the 31st CIRP International Seminar on Manufacturing Systems, 26.-27. Mai 1998. Berkeley, USA: 1998, S. 271-276.

REINHART & ROßGODERER 1998

Reinhart, G.; Roßgoderer, U.: Spatial Planning of Manual and Automated Work Stations Utilising Mathematic Algorithms and Virtual Reality Devices. In: Juslin, K. (Hrsg.): In: Proceedings of the EUROSIM 98 Simulation Congress, Helsinki (1998), S. 499-505. (Vol. 3)

REINISCH 1992

Reinisch, H.: Planungs- und Steuerungswerkzeuge zur impliziten Geräteprogrammierung. Dissertation Friedrich-Alexander-Universität Erlangen-Nürnberg. München: Carl Hanser, 1992. (Fertigungstechnik - Erlangen 31).

RÖHRDANZ 1998

Röhrdanz, F.: Modellbasierte automatisierte Greifplanung. Dissertation TU Braunschweig. Aachen: Shaker, 1998. (Fortschritte in der Robotik 3).

ROOS ET AL 1997

Roos, E.; Behrens, A.; Anton, S.: RDS - Realistic Dynamic Simulation of Robots. In: Proceedings of the 28th International Symposium on Robotics, Detroit 1997, S. 17-27.

ROßGODERER ET AL. 1995

Roßgoderer, U.; Bauer, L.; Kugelmann, D.: Aufbau und Programmierung von Roboterzellen mit USIS. In: Reinhart, G.; Milberg, J. (Hrsg.): 3D-Simulation - schneller, sicherer und kostengünstiger zum Ziel, München. München: Utz 1995, S. 51-66.

SAKURAI & DAVE 1996

Sakurai, H.; Dave, P.: Volume decomposition and feature recognition, Part II: curved objects. Computer-Aided Design 28 (1996) 6/7, S. 519-537.

SATO ET AL. 1995

Sato, S.; Nakano, M.; Koyama, T.: Task-Level Teaching System with Trajectory Planning for Industrial Robots. In: Proceedings of the 1995 IEEE International Conference on Robotics and Automation, Nagoya, 1995, S. 2394-2400. (Vol. 3)

SCHELL 1997

Schell, H.: Bewertung alternativer Handhabungs- und Fertigungsfolgen. Dissertation RWTH Aachen. Aachen: Shaker, 1997. (wzl / IPT - Berichte aus der Produktionstechnik 3/97).

SCHNEIDER & GALLASCH 1997

Schneider, B.; Gallasch, A.: Modellfabrik - viel mehr als ein Spielzeug. In: Die neue Fabrik - Modellfabrik als Denkmodell. Landsberg: moderne industrie 1997, S. 18-19.

SCHRÖER 1993

Schröer, K.: Identifikation von Kalibrierungsparametern kinematischer Ketten. Dissertation TU Berlin. München: Hanser, 1993. (Produktionstechnik - Berlin 126).

SCHRÜFER 1992

Schrüfer, N.: Erstellung eines 3D-Simulationssystems zur Reduzierung von Rüstzeiten bei der NC-Bearbeitung. Dissertation TU München. München: Springer, 1992. (iwb-Forschungsberichte 48).

SCHWEIZER 1996

Schweizer, M.: Über allen Erwartungen. Roboter+Automation (1996) 6, S. 12-14.

SEDGEWICK 1992

Sedgewick, R.: Algorithmen in C. Bonn: Addison Wesley, 1992.

SHIRINZADEH ET AL. 1994

Shirinzadeh, B.; Paragreen, C.; Lee, W.: Calibration of Robot Cell for CAD-Based Planning Using Machine Vision Techniques. In: Gill, R.; Syan, C. S. (Hrsg.): Proceedings of the Twelfth International Conference on CAD/CAM Robotics and Factories of the Future. London, England: Middlesex University Press 1996, S. 76-81.

SIEGLER 1988

Siegler, A.: Fehleranalyse von Roboter manipulatoren als Teil der Bahnbe-
rechnung. Robotersysteme 4 (1988) 4, S. 233-239.

SKOWRONSKI & TURNER 1997

Skowronski, V. J.; Turner J. U.: Using Monte-Carlo variance reduction in statistical tolerance synthesis. Computer-Aided Design 29 (1997) 1, S. 63-69.

SORENTI 1996

Sorenti, P.: Rapid palletizing: simulation in the fast lane. Industrial Robot 23 (1996) 3, S. 16-19.

SRINIVASAN ET AL. 1996

Srinivasan, R. S.; Wood, K.L.; MacAdams, D. A.: Functional Tolerancing: A Design for Manufacturing Methodology. Research in Engineering Design (1996) 2, S. 99-115.

STETTER 1993

Stetter, R.: Rechnergestützte Werkzeuge zur Effizienzsteigerung des Industrierobotereinsatzes. Dissertation TU München. München: Springer, 1993. (iwb-Forschungsberichte 62).

STEWART 1993

Stewart, N.F.: Sufficient condition for correct topological form in tolerance specification. *Computer-Aided Design* 25 (1993) 1, S. 39-48.

SUN 1998

Sun, H.: Development of a Robot Assembly Task Planning System. In: *Preprints of the 1st IFAC-Workshop on Intelligent Assembly and Disassembly*, Bled, Slowenien. International Federation of Automatic Control 1998, S. 31-36. (Vol. 1)

TAUBER 1990

Tauber, A.: Modellbildung kinematischer Strukturen als Komponente der Montageplanung. Dissertation TU München. München: Springer, 1990. (iwb-Forschungsberichte 30).

TECNOMATIX TECHNOLOGIES LTD 1998

Tecnomatix Technologies Ltd. (Hrsg.): *ROBCAD - Die virtuelle Fertigungsumgebung zur Planung, Optimierung, Prüfung und Off-Line-Programmierung von Produktionsprozessen (Produktbeschreibung)*. Tecnomatix 1998.

THATER & BEHRENS 1993

Thater, G.; Behrens, A.: Minimierung der Poseabweichungen von Industrierobotern durch eine fehlerkompensierende Modellierung und Simulation der kinematischen Kette. In: *Intelligente Steuerung und Regelung von Robotern*. Düsseldorf: VDI, 1993, S. 501-510. (VDI Berichte Nr. 1094)

TÖNSHOFF ET AL. 1993

Tönshoff, H.-K.; Kalender, T.; Livotov, P.: Statistische Modellierung des nichtlinearen Getriebeeinflusses auf das statische und dynamische Roboterverhalten. Düsseldorf: VDI, 1993, S. 656-668. (VDI Berichte Nr. 1094)

VAN BRUGGEN ET AL. 1993

van Bruggen, M.; Baartmann, J. P.; Bronsvort, W. F.: Grips on Parts. In: *1993 IEEE International Conference on Robotics and Automation*, Atlanta, Georgia, USA. Los Alamitos, CA: IEEE Computer Society Press 1993, S. 828-833.

VAN DE LOGT ET AL. 1996

van de Logt, G., Inkmann, T.; Hunold, U.: Numerische kollisionsvermeidende Planung von Trajektorien für Roboter. *at - Automatisierungstechnik* 44 (1996) 4, S. 151-159.

VARGHESE ET AL. 1996

Varghese, P.; Braswell, R. N.; Wang, B.; Zhang, C.: Statistical tolerance analysis using FRPDF and numerical convolution. *Computer-Aided Design* 28 (1996) 9, S. 723-732.

VISSER ET AL. 1998

Visser, A.; Kohlhaas, E.; Steffen, K.: Vereinfachtes Kalibrieren von Industrierobotern. Werkstattstechnik 88 (1998) 1/2, S. 49-52.

VORMANN 1995

Vormann, B.: Beitrag zur automatisierten Handhabungsplanung komplexer Biegeteile. Dissertation Universität Erlangen-Nürnberg 1995.

WANNER 1989

Wanner, M.-C.: Rechnergestützte Verfahren zur Auslegung der Mechanik von Industrierobotern. Dissertation Universität Stuttgart. Berlin: Springer, 1989. (IPA-IAO Forschung und Praxis 130)

WEBER & STARK 1994

Weber, C.; Stark, R.: Stand und Perspektiven der rechnerunterstützten Toleranzanalyse und -synthese. In: VDI-Fachtagung Wege zum erfolgreichen Qualitätsmanagement in der Produktentwicklung. Düsseldorf: VDI, 1994, S. 203-221. (VDI Berichte Nr. 1106)

WEEKS 1997

Weeks, J.: Entwicklung eines aufgabenorientierten Greif- und Bahnplanungssystems für die automatisierte Montage mit SCARA-Robotern. Dissertation RWTH Aachen. Aachen: Shaker, 1997. (wzl / IPT - Berichte aus der Produktionstechnik 1/97).

WEISEL 1996

Weisel, W. K.: PC simulation: through the barricade. Industrial Robot 23 (1996) 1, S. 5-8.

WILLERT 1996

Willert, R.: NC-Programmiersysteme - eine Marktübersicht. Zwf 91 (1996) 12, S. 624-630.

WLOKA 1991

Wloka, D.: Robotersimulation. Berlin: Springer, 1991.

WOENCKHAUS 1994

Woenckhaus, C.: Rechnergestütztes System zur automatisierten Layoutoptimierung. Dissertation TU München. München: Springer, 1994. (iwb-Forschungsberichte 65).

WOENCKHAUS ET AL. 1994

Woenckhaus, C.; Kugelmann, D.; Bauer, L.: USIS - ein Simulationswerkzeug für Layoutplanung und Offline-Programmierung. CIM-Management 10 (1994) 3, S. 20-23.

WRBA 1990

Wrba, P.: Simulation als Werkzeug in der Handhabungstechnik. Dissertation TU München. Berlin: Springer, 1990. (iwb-Forschungsberichte 25).

XU & WURST 1992

Xu, W. L.; Wurst, K. H.: Relative calibration method for improving robot accuracy. Robotersysteme 8 (1992) 2, S. 107-113.

ZELLER 1995

Zeller, F.-J.: Sensorplanung und schnelle Sensorregelung für Industrieroboter. Dissertation Friedrich-Alexander-Universität Erlangen-Nürnberg. München: Hanser, 1996. (Fertigungstechnik - Erlangen 51).

ZHANG ET AL. 1997

Zhang, C.; Chan, K. W.; Chen, Y. H.: A Method for Recognising Feature Interactions and Feature Components within the Interactions. International Journal of Advanced Manufacturing Technology 13 (1997), S. 713-722.