

Automatisierte Programmierung von Robotern für Montageaufgaben

Ulrike Thomas

Institut für Robotik und Prozessinformatik
Technische Universität Carolo-Wilhelmina zu Braunschweig
U.Thomas@tu-bs.de

Abstract: In dieser Arbeit wird eine automatisierte Prozesskette von der Spezifikation der Montageaufgaben über die Planung der Montagereihenfolgen bis hin zur Planung elementarer sensorbasierter feiner Roboterbewegungen vorgestellt. Als Eingaben dienen sehr genau modellierte zu montierende Objekte in Form von CAD-Daten und die durch den Anwender ausgewählten symbolisch-räumlichen Relationen. Aus dieser Spezifikation werden durch das hier vorgestellte neue Montageplanungssystem Montagereihenfolgen generiert und unter Berücksichtigung verschiedener Kriterien bewertet. Für die Ausführung der berechneten Pläne muss jede darin enthaltende Montageaufgabe in elementare Roboteranweisungen, so genannte Aktionsprimitive, umgesetzt werden. Hierfür wird ein neuer Algorithmus vorgestellt. Insgesamt ermöglichen die hier vorgestellten Verfahren eine automatisierte Programmierung von Robotern für Montageaufgaben.

1 Einleitung

Die in der Industrie in immer kleiner werdenden Zeitabständen veränderten Produkte erfordern eine sich schnell an die Produkte anpassende flexible Fertigung durch Industrieroboter. Die Installation einer Fertigungszelle und das Einlernen von Roboterposen nehmen üblicherweise einen sehr großen Zeit- und Arbeitsaufwand in Anspruch. Bei gleichzeitig steigenden Lohnkosten ist es erforderlich, Werkzeuge zu entwickeln, die es ermöglichen, Montageroboter auf einfache Art und Weise zu programmieren. Dabei soll die Montage möglichst autonom und fehlertolerant ablaufen. Eine flexible, robuste Montage erreicht man durch den Einsatz möglichst vieler Sensoren, was wiederum zu aufwändig zu programmierenden Systemen führt und kontinuierlich wachsende Personalkosten zur Folge hat. Mit der Anforderung, die „Time-to-Market“ gering zu halten, steigt die Anzahl neu installierter Roboter im Verhältnis zur Anzahl der Beschäftigten weltweit an. Diese Trends erzwingen die Entwicklung neuer Programmierwerkzeuge für Roboter, die die herkömmliche Roboterprogrammierung in Form von „Teach-In“-Verfahren ablösen. Viele komplizierte Montageaufgaben werden heute noch von Hand durchgeführt. Die Ursache liegt darin, dass für schwierige Montageaufgaben sensorbasierte intelligente Robotersysteme erforderlich sind. Erst in den letzten Jahren ließ sich die Steuerungstechnik für sensorbasierte Robotersysteme erheblich verbessern. Daher erschließen sich diesen Systemen

neue Anwendungsbereiche, wobei sich auch komplizierte Montagevorgänge weiter automatisieren lassen. Die Programmierung dieser Systeme ist sehr schwierig. Konzepte zur automatisierten Programmierung insbesondere für die Montage unter Berücksichtigung von Lage- und Bauteiltoleranzen fehlen, beziehungsweise wurden bisher nur für sehr einfache Aufgaben realisiert.

Die Dissertation beschreibt eine neue vollständige Prozesskette von der Spezifikation der Montageaufgaben über die Planung von Montagesequenzen bis hin zur sensorbasierten Ausführung von Montageaufgaben. Hierdurch soll die Roboterprogrammierung für Montageaufgaben erheblich vereinfacht werden. Dabei können Robotersysteme, die ihre nahe Umgebung durch Kraftrückkopplung wahrnehmen, automatisiert programmiert werden. Im nächsten Abschnitt wird der Stand der Technik erläutert und in den folgenden Abschnitten die einzelnen Teile dieser Prozesskette beschrieben.

2 Roboterprogrammierung: Stand der Technik

Zur automatisierten Roboterprogrammierung gibt es nur wenige Arbeiten, die eine vollständige Prozesskette von der Planung bis zur Ausführung von Montageaufgaben realisieren. Das wohl am weitesten entwickelte Werkzeug ist das System Archimedes 2 [KWJC96]. Dieses System verbindet die Montageplanung mit der Generierung von V+ Roboterprogrammen. Jedoch besteht das generierte Programm nur aus Positionsanweisungen für den Roboter. Lage- und Bauteiltoleranzen können dabei nicht berücksichtigt werden. Meistens ist ein aufwändiges Nachlernen der Roboterposen notwendig. Eine weitere Arbeit, die sich mit der gesamten Thematik beschäftigt, wurde von Mosemann [Mos00] vorgestellt. Sie enthält eine automatische Planung der Roboteraufgaben. Die zu erledigenden Roboteraufgaben werden erkannt und entsprechende Sequenzen von Aktionsprimitiven geladen. Die Aktionsprimitivsequenzen müssen jedoch von Hand festgelegt werden.

Generell lässt sich die Roboterprogrammierung in drei unterschiedliche Arten der Programmiermethodik unterteilen. Es gibt textuelle Programmiersprachen, die spezielle Roboteranweisungen interpretieren. Die meisten textuellen Roboterprogrammiersprachen sind heute noch „Basic-like“-Sprachen. Eine andere Möglichkeit stellt die Icon-basierte Roboterprogrammierung dar. Sie erinnert an das Zeichnen von Flussdiagrammen. Es werden beispielsweise Petri-Netze oder State-Charts verwendet [BC01]. Insgesamt gibt es in diesem Bereich erst sehr wenige Programmiersysteme, die es ermöglichen, auch Sensorereignisse in die Bewegungsbefehle einzukoppeln.

Die automatisierte Roboterprogrammierung befasst sich einerseits mit dem Generieren von textuellen Programmcode aus CAD-Daten und andererseits mit dem einfachen Erlernen von Montageoperationen durch Vormachen. Meistens werden hier Posen in einer virtuellen Umgebung gelernt und an die Robotersteuerung weitergegeben. Der Roboter wird weich geschaltet, das heißt, es wird eine Nullkraftregelung am Endeffektor eingestellt. Ein Mensch kann dann die Bewegung am Roboter demonstrieren, indem er am Endeffektor angreift und die Bewegung ausführt. Die Robotersteuerung speichert diese Bahnen ab und führt sie später während der Montage aus. Beispiele für diese Art der Roboter-

programmierung finden sich bei Fukuda und Slaets [FNUH01, SLR⁺07]. Eine weitere Variante der Roboterprogrammierung besteht darin, Anweisungen für Roboter aus Sprache und Gesten zu interpretieren. Knoll et al. entwickelten ein System, welches für die Montage eines Baufix-Flugzeug entsprechende Roboterbefehle erzeugt [KHZ97]. Vorarbeiten für eine automatisierte Roboterprogrammierung, die die Montageplanung betreffen, finden sich bei [KWJC96, Mos00]. Ein weiterer wichtiger Bestandteil für eine robuste Programmierung von Montageaufgaben ist die Entwicklung von Konzepten und Steuerungen zur sensorbasierten Bewegung von Robotern. Hasagawa startete mit der Entwicklung von „skill-primitives“ [HST92], bis schließlich in [Fin04] eine Aktionsprimitiv basierte Steuerung zur Verfügung stand. Jedoch war der Prozess der Roboterprogrammierung bis dato händisch und es konnten keine Montageaufgaben automatisiert programmiert werden.

3 Spezifikation von Montageaufgaben

Für die automatisierte Roboterprogrammierung soll der Anwender einfach und intuitiv Montageaufgaben spezifizieren können. In heutigen CAD-Systemen lassen sich Modelle der zu montierenden Baugruppen mit unterschiedlichen Modellierungstechniken erstellen. Damit erhält man geometrische Beschreibungen der in Roboterarbeitszellen vorhandenen Elemente: der Bauteile, des Roboters, der Zuführeinrichtungen, der am Roboterhandflansch befestigten Werkzeuge, usw. Neben der geometrischen Beschreibung sind kinematische Beschreibungen des Roboters und Modelle der Sensoren notwendig. In dieser Arbeit wird ein an der Roboterhand befestigter Kraft/Momentensensor eingesetzt. Neben diesen Beschreibungen nimmt die Spezifikation der relativen Lagen der Bauteile einen wesentlichen Teil der händischen Arbeit des Programmierers ein. Um diese zu reduzieren, ist hierfür ein graphisch-interaktives System geschaffen worden. Die relative Lage der Bauteile zueinander kann durch so genannte symbolisch-räumliche Relationen spezifiziert werden [AP75]. Hier wurde ein Lösungsverfahren entwickelt, das im Gegensatz zu bekannten Verfahren auch in der Lage ist, Widersprüche zu erkennen und dem Anwender verbleibende Freiheitsgrade anzuzeigen.

Die Abbildung 1 zeigt beispielhaft die zur Spezifikation verwendeten symbolisch-räumlichen Relationen für einige Bauteile der Baugruppe eines Ventils. Der Anwender selektiert für jede Relation zwei Merkmale, beispielsweise des Types „Shaft“ und „Hole“ und gibt den Typ der Relation vor. Das graphisch-interaktive System erzeugt für jede symbolisch-räumliche Relation eine Gleichung, siehe Abbildung 1. Daraus ergibt sich ein System von nicht-linearen Gleichungen, die trigonometrische Ausdrücke enthalten. Für das Lösen werden zunächst alle trigonometrischen Ausdrücke geeignet substituiert. Das so entstandene Gleichungssystem lässt sich zu einer Gröbner-Basis reduzieren und dann lösen. Damit können die relativen Lagen der Bauteile zueinander berechnet werden. Somit ist bekannt, wie die Baugruppe im zusammengesetzten Zustand aussehen muss und es lassen sich Montagesequenzen berechnen. Der nächste Abschnitt beschreibt den Planungsvorgang für ganze Baugruppenaggregate.

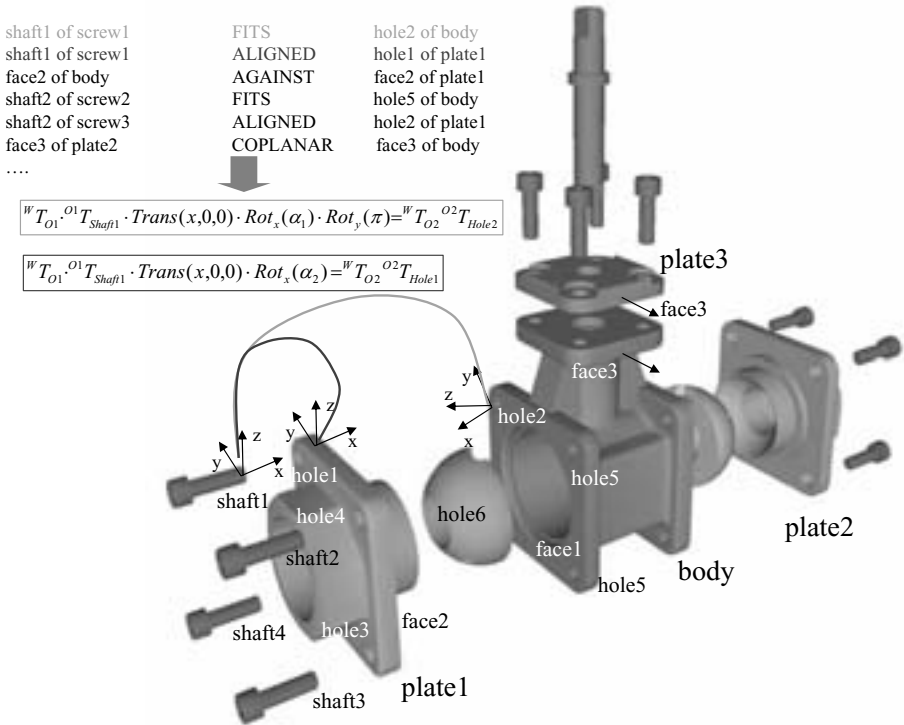


Abbildung 1: Bauteile eines Automobilscheinwerfers spezifiziert mit symbolisch-räumlichen Relationen; dargestellt sind zwei Relationen mit ihren Transformationen.

4 Montageplanung

Mit einer automatischen Montageplanung lassen sich Reihenfolgen für den Zusammenbau von Baugruppen berechnen. Die Herausforderung liegt darin, möglichst zuverlässig geometrisch und physikalisch durchführbare Montagereihenfolgen zu finden. Hierbei wird die so genannte *Assembly-by-Disassembly* Strategie angewendet. Man geht von der zusammengesetzten Baugruppe aus und versucht, unter Randbedingungen Bauteile oder Teilbaugruppen zu entfernen. Die umgekehrte Reihenfolge ergibt dann eine gültige Montagesequenz. Es ist leicht einzusehen, dass es sich bei der Montageplanung um ein exponentielles Problem handelt. So gibt es bei n Bauteilen $2^{n-1} - 1$ mögliche Zerlegungen in zwei disjunkte Bauteilgruppen. Im schlimmsten Fall müssen somit für Montagesequenzen insgesamt $\frac{3^{n+1}-1}{2} - 2^n$ Zerlegungen betrachtet werden. Damit kommt einem effizienten Algorithmus zur Berechnung der geometrischen Separierbarkeit eine entscheidende Bedeutung zu.

Bei dem in dieser Arbeit entwickelten Algorithmus werden von allen Bauteilpaaren (c_i, c_j) die Konfigurationsraumhindernisse bestimmt, kurz CO_{c_i, c_j} . Auf Grund der aus

vielen Dreiecken bestehenden resultierenden Konfigurationsraumhindernisse können diese nicht für alle Bauteilpaare im Hauptspeicher heutiger Rechner gehalten werden. Daher werden die Ergebnisse geeignet in 2d projiziert und gespeichert. Die Abbildung 2 zeigt die verschiedenen Schritte zur Berechnung der geometrischen Separierbarkeit für die Fernlichtlampe bezüglich des Projektionsglases eines Automobilscheinwerfers.

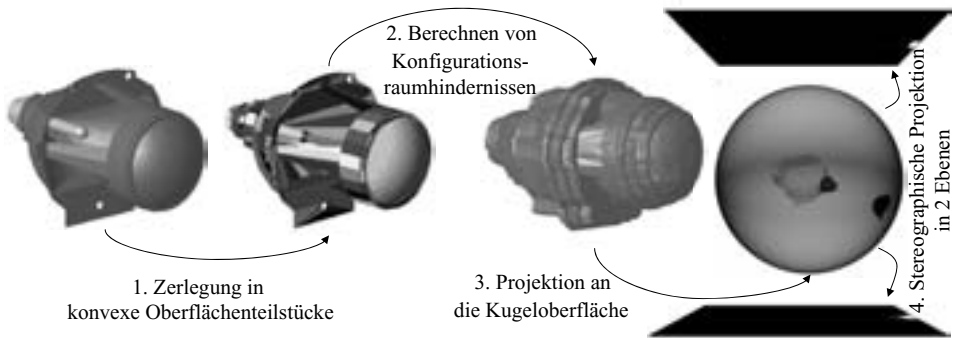


Abbildung 2: Schritte zur Berechnung der translatorischen geometrischen Separierbarkeit von Bauteilen.

Erst durch die Existenz von Algorithmen, die die Oberflächen von komplexen Objekten ausschließlich in konvexe Patches (Abbildung 2 erster Schritt) zerlegen, ist es möglich, effizient Konfigurationsraumhindernisse in 3d zu berechnen. Nach dieser Zerlegung werden für alle Paare von konvexen Oberflächenpatches die Minkowski-Punktmengen bestimmt. Die Addition aller konvexen Hüllen dieser Punktmengenpaare ergibt das Konfigurationsraumhindernis CO_{c_i, c_j} , siehe Abbildung 2 nach dem zweiten Schritt. Dieses Hindernis beschreibt alle Konfigurationen des Objekts c_i bei denen eine Kollision mit Objekt c_j auftritt. Der Rand des Objekts beschreibt die Konfigurationen, bei denen die Bauteile sich berühren. Das Berechnen der Konfigurationsraumhindernisse aus der Vereinigung der konvexen Hüllen ist sehr rechenintensiv. Mit dem Trick der Projektion der CO_{c_i, c_j} an die Kugeloberfläche lässt sich dieser Prozess umgehen und es können alle Montagerichtungen nach beispielsweise einer stereographischen Projektion in 2d mit einer sehr hohen Auflösung gespeichert werden.

Für das Berechnen der geometrischen Separierbarkeit von je zwei Teilbaugruppen $\mathcal{B}_n$ und $\mathcal{B}_m$, die jeweils mehrere Bauteile enthalten, müssen alle Tupel (c_i, c_j) mit $c_i \in \mathcal{B}_n$ und $c_j \in \mathcal{B}_m$ getestet werden. Für jedes Tupel stehen die 2d Projektionen zur Verfügung. Sie werden für alle Kombinationen von Bauteilpaaren in einer *Look-up-Tabelle* vorgehalten und in einem Vorverarbeitungsschritt vor der eigentlichen Montageplanung berechnet. Durch eine effizient zu berechnende Und-Verknüpfung der Bilder, lässt sich während der Planungsphase schnell entscheiden, ob eine Zerlegung geometrisch möglich ist, siehe auch Abbildung 3.

Für das Generieren von Montagesequenzen wird von der zusammengesetzten Baugruppe ausgegangen und es werden während der Planung geschickt Bauteile oder Bauteilgruppen entfernt, so dass die Teilbaugruppen wiederum zusammenhängend sind. Für die Be-

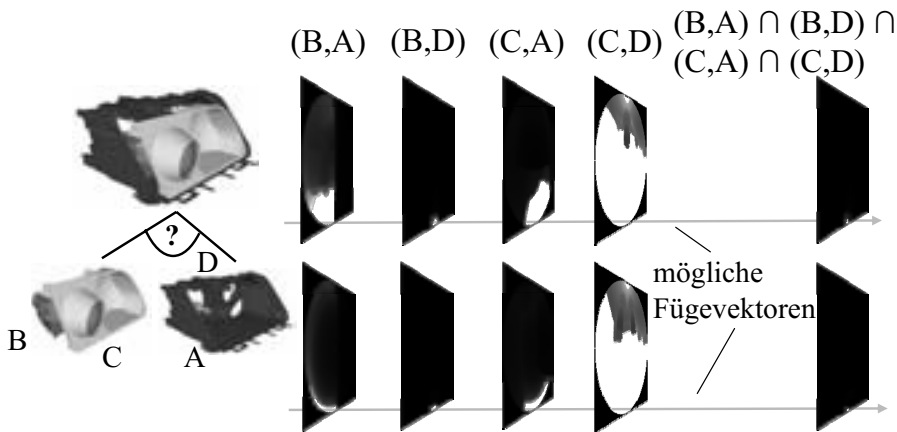


Abbildung 3: Effizienter Separierbarkeitstest beim Zerlegen von Baugruppen.

rechnung von Zusammenhangskomponenten eignet sich ein Verfahren aus der Graphentheorie. Zunächst wird ein Verbindungsgraph aufgestellt, der dort Kanten enthält, wo eine Berührung der Bauteile vorliegt. Mit Hilfe der Analyse von Eigenwerten und Eigenvektoren der zugehörigen Laplace-Matrix des Verbindungsgraphen lassen sich Vorschläge zum Trennen der Baugruppe berechnen. Dabei werden beispielsweise zu erst Zerlegungen untersucht, die möglichst eine ausgeglichene Anzahl von Bauteilen enthalten. Ein weiteres auf einem Schalenmodell basierendes Verfahren schlägt das Abtrennen von außen liegenden Bauteilen oder Teilbaugruppen vor. Die generierten Zerlegungen werden bezüglich verschiedenen Kriterien bewertet oder wenn bestimmte Bedingungen, wie beispielsweise die geometrische Separierbarkeit nicht erfüllt ist, verworfen. Dabei ist es wichtig, Zerlegungen, die für den Roboter schwierig auszuführende Montagesequenzen zur Folge haben, möglichst frühzeitig zu verwerfen. Die Bewertungskriterien sind folgende:

- **Geometrische Zugänglichkeit:** Für die geometrische Zugänglichkeit sei hier die Größe des Bereichs maßgebend, der für die Demontagerichtungen zu Verfügung steht.
- **Parallelität:** Lässt man eine Baugruppe parallel montieren, so ist dies zeitlich betrachtet am günstigsten, wenn die Teilbaugruppen ungefähr gleich groß sind.
- **Mechanische Stabilität:** Jede Teilbaugruppe sollte möglichst mechanisch stabil sein. Hier wird eine Heuristik angewendet, die auf Basis der differentiellen potenziellen Energie unter gegebenen Gravitationsvektoren die Stabilität sehr schnell berechnet.
- **Handhabbarkeit:** Zur Bewertung der Handhabbarkeit wird die Menge der möglichen geometrischen Griffe herangezogen.
- **Umorientierungen:** Das Umorientieren von Baugruppen während der Montage fließt mit hohen Kosten in die Planung ein.

- **Greiferwechsel:** Für jede Teilbaugruppe ist mindestens ein Griff definiert, der aus Pose und Greifertyp besteht. Muss der Greifer zwischen der Montageoperation gewechselt werden, so steigen hierfür die Kosten.
- **Wechsel der Montierichtung:** Falls die Montierichtung von einer Montageaufgabe zur nächsten Montageaufgabe innerhalb einer Sequenz geändert werden muss, sind die Kosten sehr hoch.

Die verschiedenen Zerlegungen werden generiert, bewertet und in einen so genannten AND/OR-Graphen einsortiert. Die Auswahl einer hinreichend guten Montagesequenz erfolgt schließlich mit einer „Branch-and-Bound“-Suche durch den AND/OR-Graphen. Damit wird eine Montagesequenz selektiert, die hinreichend „gut“ ist. Die Abbildung 4 zeigt einige Ergebnisse des Montageplanungssystems.

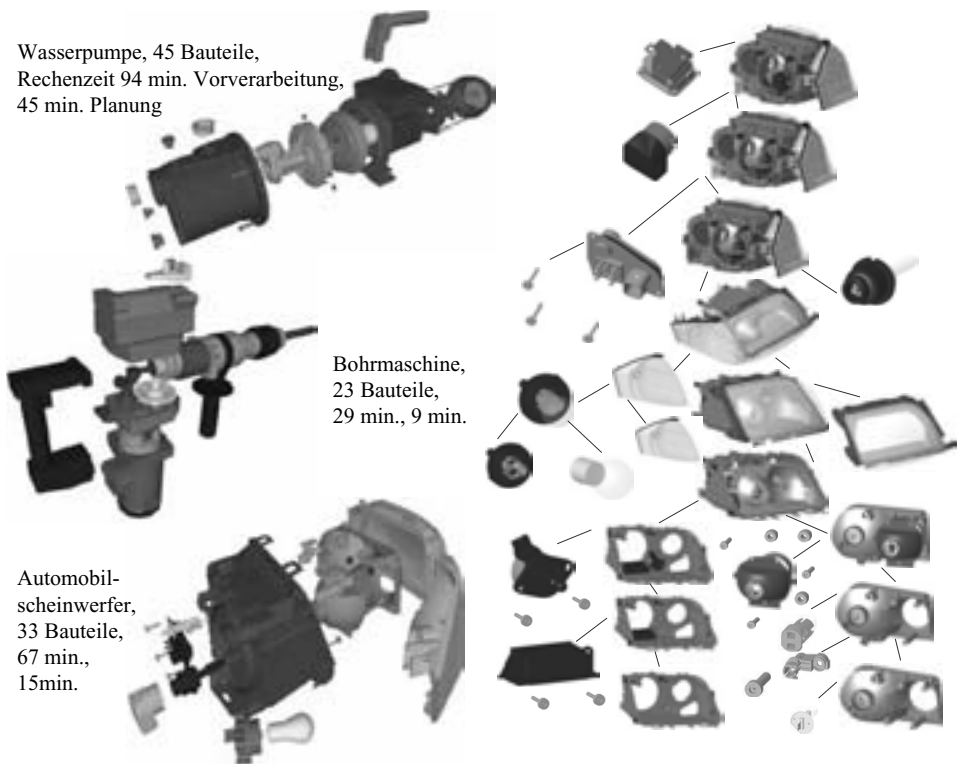


Abbildung 4: Links: Verschiedene Baugruppen. Rechts: Berechneter Montageplan für die Baugruppe eines Automobilscheinwerfers.

5 Automatisierte Programmierung durch Aktionsprimitivnetze

Aktionsprimitive sind elementare sensorbasierte Roboteraktionen, die es ermöglichen, sowohl Sensor geführte als auch Sensor überwachte Roboterbewegungen zu definieren, siehe Abbildung 5. Komplexe Montageaufgaben lassen sich durch Netze von Aktionsprimitiven beschreiben. Ein geeignete Robotersteuerung kann jedes Aktionsprimitiv ausführen. An den Kanten des Aktionsprimitivnetzes stehen Sensorbedingungen, an Hand derer ein Interpreter entscheidet, welches Aktionsprimitiv das Nächste ist. Somit kann auf verschiedene Sensorereignisse reagiert werden.

AP	HM	TF	Ref _{Motion} = Hand
			Anchor = true
			Hand T _{Task} = (0, 0, 113 mm, 0, 0, 0) ^T
		CM = L1: (0 POS, 0 POS, -20 CON, -0.03 VEL, 0 POS, 0 POS) L2: (- , - , - , - , - , -)	
	$\lambda = ((\text{Average} (T_x^{20ms}) > 0.5 \text{ Nm}) \mid (\text{Time} > 2s))$ $\rho = (\emptyset)$ $\sigma = (\emptyset)$		

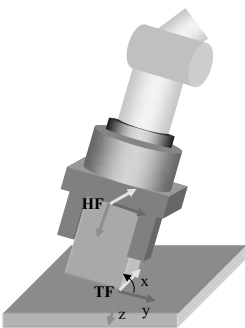


Abbildung 5: Aktionsprimitiv: Drehe um die x-Achse des Task-Frames (negative Richtung) und halte in z-Richtung eine Kontaktkraft von -20 N, bis ein Drehmoment von 0.5 Nm entgegen wirkt.

Das Generieren solcher Aktionsprimitivnetze läuft weitestgehend automatisch ab. Mit den berechneten Fügerichtungen oder Pfaden und den bekannten zu erwartenden Kontaktkräften wird zunächst ein Kontaktformationsgraph erzeugt. Hierfür werden die Bauteiloberflächen segmentiert, um ihre topologischen Elemente (Ecke, Kante, ebene Fläche, sphärische Fläche usw.) zu bestimmen. Alle Kontaktmöglichkeiten zwischen den topologischen Elementen zweier Bauteile werden in den Graphen einsortiert. Die Abbildung 6 zeigt einen Auszug aus dem Kontaktformationsgraphen für die oben genannte Montageaufgabe und die vorab segmentierten Bauteile.

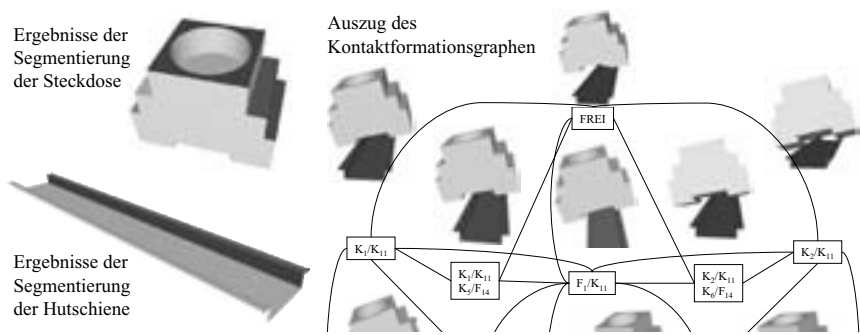


Abbildung 6: Links: Segmentierte Oberflächen; Rechts: Auszug aus Kontaktformationsgraphen.

Anschließend werden aus Kontaktformationen Aktionsprimitive generiert, um einen Kontaktzustand in den nächsten zu überführen. Das resultierende Aktionsprimitivnetz für die komplexe Aufgabe des Aufsetztes einer Steckdose auf eine Hutschiene ist in Abbildung 7 zu sehen. Mit dem entwickelten Algorithmus konnten Aktionsprimitivnetze für verschiedene Montageaufgaben generiert werden. Dies Montageaufgaben wurden mit einem Hexa-Parallelroboter erfolgreich ausgeführt, wie es in der Abbildung 7 links auch zu sehen ist.

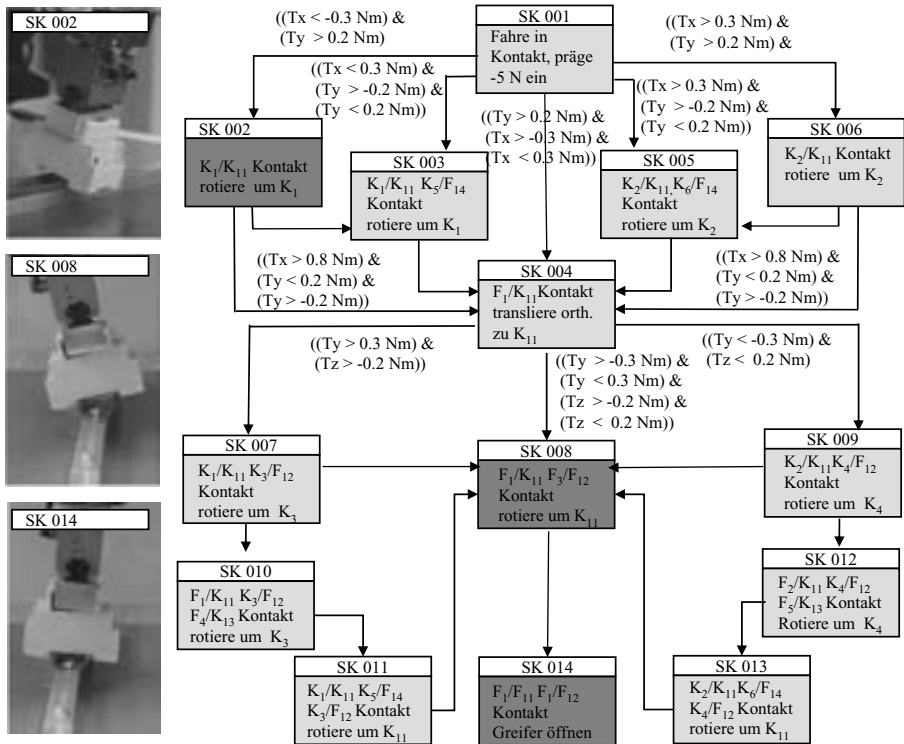


Abbildung 7: Aktionsprimitivnetz für die Montageaufgabe: Steckdose auf Hutschiene.

6 Zusammenfassung und Ausblick

In dieser Arbeit wurde eine Prozesskette zur automatisierten Roboterprogrammierung vorgestellt, die von der Spezifikation über die Montageplanung bis zum Generieren von Aktionsprimitivnetzen reicht. Die Aktionsprimitivnetze wurden interpretiert und von einer geeigneten Aktionsprimitivbasierten Robotersteuerung ausgeführt. Es konnte gezeigt werden, dass sich komplexe Montageaufgaben wie das Aufstecken einer Steckdose auf eine Hutschiene flexibel und sensorbasiert ausführen lassen. Dabei findet die Programmierung weitgehend automatisch statt. Weiterer Forschungsbedarf besteht bei der Handhabung deformierbarer Objekte und bei der Planung des Einsatzes von Sensoren. Neben

Kraft-/Momenten Sensoren gibt es bildgebende Sensoren. Die Entscheidung, welcher Sensor mit welcher Filteroperation zum Einsatz kommt und wo in der Roboterarbeitszelle die Sensoren angebracht werden sollten, könnte ein weiteres Planungssystem treffen.

Literatur

- [AP75] A. P. Ambler und R. J. Popplestone. Inferring the position of bodies from specified spatial relationships. *Artificial Intelligence*, 6, 1975.
- [BC01] M. S. Branicky und S. R. Chhatpar. A computational framework for the simulation, verification, and synthesis of force-guided robotic assembly strategies. In *2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Seiten 1471–1476, 2001.
- [Fin04] B. Finkemeyer. *Robotersteuerungsarchitektur auf der Basis von Aktionsprimitiven*. Dissertation, Technische Universität Braunschweig, 2004.
- [FNUH01] T. Fukuda, M. Nakaoka, T. Ueyama und Y. Hasegawa. Direct teaching and error recovery method for assembly task based on a transition process of a constraint condition. In *IEEE International Conference on Robotics and Automation*, Seiten 1518–1523, 2001.
- [HST92] T. Hasegawa, T. Suehiro und K. Takase. A Model-Based Manipulation System with Skill-Based Execution. *IEEE Transactions on Robotics and Automation*, 8(5):535–544, 1992.
- [KHZ97] A. Knoll, B. Hildebrandt und J. Zhang. Instructing cooperating assembly robots through situated dialogues in natural languages. In *IEEE International Conference on Robotics and Automation*, Seiten 888–894, 1997.
- [KWJC96] S. G. Kaufmann, R. H. Wilson, R. E. Jones und T. L. Calton. The Archimedes 2 Mechanical Assembly Planning System. In *IEEE International Conference on Robotics and Automation*, Seiten 3361–3368, 1996.
- [Mos00] H. Mosemann. *Beiträge zur Planung, Dekomposition und Ausführung von automatisch generierten Roboteraufgaben*. Dissertation, Technische Universität Braunschweig, 2000.
- [SLR⁺07] P. Slaets, T. Lefebvre, J. Rutgeerts, H. Bruyninckx und J. De Schutter. Incremental Building of a Polyhedral Feature Model for Programming by Human Demonstration of Force-Controlled Tasks. *IEEE Transactions on Robotics*, 23(2):20–33, 2007.



Ulrike Thomas studierte Informatik an der TU Braunschweig und an der University of Edinburgh. Nach ihrem Studium, das sie 2000 mit dem Diplom abschloss, nahm sie die Tätigkeit als wissenschaftliche Mitarbeiterin am Institut für Robotik und Prozessinformatik der TU Braunschweig auf. In 2003 wurde ihr Montageplanungssystem auf dem IEEE Symposium on Assembly and Task Planning mit einem Best Paper Award ausgezeichnet. 2008 promovierte sie am Lehrstuhl von Prof. Dr. F. M. Wahl. Seit September 2007 arbeitet sie als Wissenschaftlerin am Deutschen Zentrum für Luft- und Raumfahrt in Oberpfaffenhofen, dort beschäftigt sie sich mit Echtzeitbildverarbeitung. In 2008 erhielt sie

einen weiteren Outstanding Paper Award des European Journal of Transportation Systems. Einer ihrer Beiträge ist vorgeschlagen für den Emerald Literati Network Award 2009.