

# **Aufnahme und Nachbearbeitung von Bahnen bei der Programmierung durch Vormachen von Industrierobotern**

Von der Fakultät für Konstruktions-, Produktions- und Fahrzeugtechnik  
der Universität Stuttgart zur Erlangung der Würde eines  
Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

Vorgelegt von  
Dipl.-Systemwiss. Christian Meyer  
aus Georgsmarienhütte

Hauptberichter: Prof. Dr.-Ing. Alexander Verl

Mitberichter: Prof. Dr.-Ing. Heinz Wörn

Tag der mündlichen Prüfung: 05. April 2011

Institut für Steuerungstechnik der Werkzeugmaschinen  
und Fertigungseinrichtungen der Universität Stuttgart

# IPA-IAO Forschung und Praxis

Berichte aus dem  
Fraunhofer-Institut für Produktionstechnik und  
Automatisierung (IPA), Stuttgart,  
Fraunhofer-Institut für Arbeitswirtschaft und  
Organisation (IAO), Stuttgart,  
Institut für Industrielle Fertigung und  
Fabrikbetrieb (IFF), Universität Stuttgart  
und Institut für Arbeitswissenschaft und  
Technologiemanagement (IAT), Universität Stuttgart und  
Institut für Steuerungstechnik der Werkzeugmaschinen  
und Fertigungseinrichtungen (ISW), Universität Stuttgart

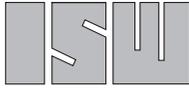
Herausgeber:

Univ.-Prof. Dr.-Ing. Prof. e.h. Dr.-Ing. e.h. Dr. h.c. mult. Engelbert Westkämper  
und

Univ.-Prof. Dr.-Ing. habil. Prof. E.h. mult. Dr. h.c. mult. Hans-Jörg Bullinger  
und

Univ.-Prof. Dr.-Ing. Dr.-Ing. E.h. Dieter Spath  
und

Univ.-Prof. Dr.-Ing. Dr. h.c. Alexander Verl



Institut für Steuerungstechnik  
der Werkzeugmaschinen und Fertigungseinrichtungen



Christian Meyer

# Aufnahme und Nachbear- beitung von Bahnen bei der Programmierung durch Vormachen von Industrie- robotern

Nr. 507

**JUST-JETTER VERLAG**  
Fachverlag · 71296 Heimsheim

Dr.-Ing. Dipl.-Systemwiss. Christian Meyer

Fraunhofer-Institut für Produktionstechnik und Automatisierung (IPA), Stuttgart

Univ.-Prof. Dr.-Ing. Prof. e.h. Dr.-Ing. e.h. Dr. h.c. mult. Engelbert Westkämper

ord. Professor an der Universität Stuttgart

Fraunhofer-Institut für Produktionstechnik und Automatisierung (IPA), Stuttgart

Univ.-Prof. Dr.-Ing. habil. Prof. E.h. mult. Dr. h.c. mult. Hans-Jörg Bullinger

ord. Professor an der Universität Stuttgart

Präsident der Fraunhofer-Gesellschaft, München

Univ.-Prof. Dr.-Ing. Dr.-Ing. E.h. Dieter Spath

ord. Professor an der Universität Stuttgart

Fraunhofer-Institut für Arbeitswirtschaft und Organisation (IAO), Stuttgart

Univ.-Prof. Dr.-Ing. Dr. h.c. Alexander Verl

ord. Professor an der Universität Stuttgart

Fraunhofer-Institut für Produktionstechnik und Automatisierung (IPA), Stuttgart

D 93

ISBN 978-3-939890-75-1

Jost Jetter Verlag, Heimsheim

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils gültigen Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

© Jost-Jetter Verlag, Heimsheim 2011.

Printed in Germany.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Sollte in diesem Werk direkt oder indirekt auf Gesetze, Vorschriften oder Richtlinien (z. B. DIN, VDI, VDE) Bezug genommen oder aus ihnen zitiert worden sein, so kann der Verlag keine Gewähr für die Richtigkeit, Vollständigkeit oder Aktualität übernehmen. Es empfiehlt sich, gegebenenfalls für die eigenen Arbeiten die vollständigen Vorschriften oder Richtlinien in der jeweils gültigen Fassung hinzuzuziehen.

Druck: printsystem GmbH, Heimsheim

## Geleitwort der Herausgeber

Über den Erfolg und das Bestehen von Unternehmen in einer marktwirtschaftlichen Ordnung entscheidet letztendlich der Absatzmarkt. Das bedeutet, möglichst frühzeitig absatzmarktorientierte Anforderungen sowie deren Veränderungen zu erkennen und darauf zu reagieren.

Neue Technologien und Werkstoffe ermöglichen neue Produkte und eröffnen neue Märkte. Die neuen Produktions- und Informationstechnologien verwandeln signifikant und nachhaltig unsere industrielle Arbeitswelt. Politische und gesellschaftliche Veränderungen signalisieren und begleiten dabei einen Wertewandel, der auch in unseren Industriebetrieben deutlichen Niederschlag findet.

Die Aufgaben des Produktionsmanagements sind vielfältiger und anspruchsvoller geworden. Die Integration des europäischen Marktes, die Globalisierung vieler Industrien, die zunehmende Innovationsgeschwindigkeit, die Entwicklung zur Freizeitgesellschaft und die übergreifenden ökologischen und sozialen Probleme, zu deren Lösung die Wirtschaft ihren Beitrag leisten muss, erfordern von den Führungskräften erweiterte Perspektiven und Antworten, die über den Fokus traditionellen Produktionsmanagements deutlich hinausgehen.

Neue Formen der Arbeitsorganisation im indirekten und direkten Bereich sind heute schon feste Bestandteile innovativer Unternehmen. Die Entkopplung der Arbeitszeit von der Betriebszeit, integrierte Planungsansätze sowie der Aufbau dezentraler Strukturen sind nur einige der Konzepte, welche die aktuellen Entwicklungsrichtungen kennzeichnen. Erfreulich ist der Trend, immer mehr den Menschen in den Mittelpunkt der Arbeitsgestaltung zu stellen - die traditionell eher technokratisch akzentuierten Ansätze weichen einer stärkeren Human- und Organisationsorientierung. Qualifizierungsprogramme, Training und andere Formen der Mitarbeiterentwicklung gewinnen als Differenzierungsmerkmal und als Zukunftsinvestition in *Human Resources* an strategischer Bedeutung.

Von wissenschaftlicher Seite muss dieses Bemühen durch die Entwicklung von Methoden und Vorgehensweisen zur systematischen Analyse und Verbesserung des Systems Produktionsbetrieb einschließlich der erforderlichen Dienstleistungsfunktionen unterstützt werden. Die Ingenieure sind hier gefordert, in enger Zusammenarbeit mit anderen Disziplinen, z. B. der Informatik, der Wirtschaftswissenschaften und der Arbeitswissenschaft, Lösungen zu erarbeiten, die den veränderten Randbedingungen Rechnung tragen.

Die von den Herausgebern langjährig geleiteten Institute, das

- Fraunhofer-Institut für Produktionstechnik und Automatisierung (IPA),
- Fraunhofer-Institut für Arbeitswirtschaft und Organisation (IAO),
- Institut für Industrielle Fertigung und Fabrikbetrieb (IFF),
- Institut für Arbeitswissenschaft und Technologiemanagement (IAT),
- Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen (ISW),

arbeiten in grundlegender und angewandter Forschung intensiv an den oben aufgezeigten Entwicklungen mit. Die Ausstattung der Labors und die Qualifikation der Mitarbeiter haben bereits in der Vergangenheit zu Forschungsergebnissen geführt, die für die Praxis von großem Wert waren. Zur Umsetzung gewonnener Erkenntnisse wird die Schriftenreihe „IPA-IAO - Forschung und Praxis“ herausgegeben. Der vorliegende Band setzt diese Reihe fort. Eine Übersicht über bisher erschienene Titel wird am Schluss dieses Buches gegeben.

Dem Verfasser sei für die geleistete Arbeit gedankt, dem Jost Jetter Verlag für die Aufnahme dieser Schriftenreihe in seine Angebotspalette und der Druckerei für saubere und zügige Ausführung. Möge das Buch von der Fachwelt gut aufgenommen werden.

Engelbert Westkämper    Hans-Jörg Bullinger    Dieter Spath    Alexander Verl

## **Vorwort**

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA in Stuttgart.

Herrn Prof. Dr.-Ing. Alexander Verl gilt mein großer Dank für die Übernahme des Hauptberichtes. Seine wohlwollende Unterstützung und seine wertvollen Hinweise haben zum Gelingen der Arbeit beigetragen. Ebenfalls gilt mein Dank Herrn Prof. Dr.-Ing. Heinz Wörn für die Übernahme des Mitberichts. Heide Kreuzburg danke ich für die Organisation des Promotionsprozesses.

Für die kritische Durchsicht der ersten Fassung danke ich insbesondere Dr.-Ing. Walter Schaaf und Dr.-Ing. Andreas Pott. Beide haben mir umfangreiches konstruktives Feedback gegeben, ebenso wie neue Motivation. Für die gründliche redaktionelle Unterstützung danke ich Luzia Schuhmacher M. A.

Ein Teil dieser Arbeit ist im Rahmen des EU-Projektes SMERobot entstanden. Für das immer gute und produktive Projektklima gilt mein Dank dem Koordinator Dipl.-Ing. Martin Hägele M. S., dem Projektmanager Thilo Zimmermann und allen Projektpartnern. Die Realisierung und die Durchführung der Versuche wäre ohne meine Kollegin Dipl.-Inf. Rebecca Hollmann nicht möglich gewesen. Ihr danke ich für die hervorragende Zusammenarbeit. Herrn Klaus Killmann und Tobias Herbst bin ich dankbar für die Unterstützung in allen Fragen des Aufbaus und Transports der Roboterzelle. Paul Treffler, Paul Treffler junior und den Mitarbeitern der Firma Treffler Maschinenbau möchte ich für die Möglichkeit der Arbeit im Betrieb, sowie das vielfältige und immer freundliche Feedback danken.

Für Abwechslung hat immer die Entwicklung der Kernthemen unserer Gruppe gesorgt. Dies war eine besondere Freude durch das Know How, das Engagement, die Kreativität und insbesondere die Freundlichkeit meiner Kollegen Dipl.-Math. techn. Susanne Oberer-Treitz, Dipl.-Inf. Rebecca Hollmann, Dr.-Ing. Andreas Pott, Dipl.-Ing. Arnold Puzik, Dipl.-Ing. Manuel Drust, Dipl.-Ing. Thomas Dietz und Dipl.-Ing. Ulrich Schneider. Ermöglicht wird dies durch die Freiheiten und die Unterstützung, die ich von meinem Abteilungsleiter Dipl.-Ing. Martin Hägele M. S. erhalten habe.

Dipl.-Inf. Christopher Parlitz M. A. danke ich für die gemeinsamen Bürojahre, den Palast und den Perspektivwechsel.

Den Grundstein für mein technisches Interesse und damit auch diese Arbeit habe ich meinen Eltern zu verdanken. Schule, Hobbies, Studium und Beruf wurden durch sie geprägt. Widmen möchte ich diese Arbeit meiner fantastischen Frau Uli. Ihre Geduld war groß, aber nicht endlos; genau richtig. Und meine beiden Kinder Thore und Paula sorgten immer für Entspannung.

Stuttgart im April 2011

Christian Meyer



# Inhaltsverzeichnis

<b>Formelzeichen .....</b>	<b>- 12 -</b>
Namen und Abkürzungen.....	- 12 -
Formelzeichen .....	- 12 -
<b>1 Einleitung .....</b>	<b>- 13 -</b>
1.1 Problemstellung .....	- 13 -
1.2 Zielsetzung und Vorgehensweise.....	- 13 -
<b>2 Grundlagen und Ausgangssituation.....</b>	<b>- 15 -</b>
2.1 Begriffe und Definitionen.....	- 15 -
2.2 Abgrenzung der intuitiven Programmierung zu anderen Programmiermethoden .....	- 17 -
2.2.1 Kategorisierung der Programmiermethoden .....	- 17 -
2.2.2 Einordnung der untersuchten Programmiermethode.....	- 18 -
2.3 Stand der Technik.....	- 18 -
2.3.1 Beispiele für intuitiv programmierbare Roboterzellen.....	- 18 -
2.3.2 Alternative Verfahren zur Aufnahme von Geometriedaten .....	- 19 -
2.3.3 Alternative Verfahren zur Nachbearbeitung von Trajektorien.....	- 21 -
2.3.4 Steuerungen als Basis der Programmierung.....	- 22 -
2.3.5 Personensicherheit .....	- 23 -
2.4 Folgerungen aus dem Stand der Technik und Forschung.....	- 25 -
<b>3 Analyse der Fertigungssituation bei KMU .....</b>	<b>- 26 -</b>
3.1 Analyse der Produktionssituation kleiner Losgrößen mit Industrierobotern.....	- 26 -
3.1.1 Ablauf der manuellen Fertigung bei kleinen und mittleren Losgrößen .....	- 28 -
3.1.2 Ablauf der robotergestützten Fertigung.....	- 29 -
3.2 Analyse potenzieller Anwendungsszenarien für die intuitive Roboterprogrammierung bei KMU .....	- 30 -
3.2.1 Klassifikation der mit dem Roboter durchgeführten Prozesse.....	- 30 -
3.2.2 Anwendungsszenario „MAG-Schweißen in der Kleinserie“.....	- 32 -
3.2.3 Anwendungsszenario „Handlingsaufgaben in der Prototypenfertigung“ .....	- 34 -
3.2.4 Anwendungsszenario „Positionierung von Messsystemen“ .....	- 35 -
3.3 Analyse der zur programmierenden Geometrieinformationen .....	- 37 -
3.4 Analyse der verfahrenstechnischen Anforderungen des Schweißprozesses .....	- 41 -
3.5 Analyse der konventionellen Programmierung von Schweißrobotern .....	- 43 -
3.6 Folgerungen aus den Analyseergebnissen und Ableitung von Untersuchungsschwerpunkten .....	- 45 -
3.7 Anforderungen an ein intuitiv programmierbares Robotersystem.....	- 46 -
3.7.1 Auswahl von Referenzwerkstücken .....	- 46 -
3.7.2 Definition von Teilfunktionen und Teilsystemen.....	- 46 -

3.7.3	Ableitung von Anforderungen an das Handhabungssystem .....	- 48 -
3.7.4	Ableitung von Anforderungen an die Mensch-Maschine-Schnittstelle.....	- 48 -
3.7.5	Ableitung von Anforderungen an die Prozesseinheiten und die Sensorik .....	- 48 -
3.7.6	Ableitung von Anforderungen an die Sicherheitseinrichtungen .....	- 49 -
3.7.7	Ableitung von Anforderungen an den Verfahrensablauf und das Gesamtsystem .....	- 49 -
<b>4</b>	<b>Konzeption des intuitiven Programmierverfahrens und des Gesamtsystems zum MAG-Schweißen .....</b>	<b>- 51 -</b>
4.1	Konzeption des Verfahrensablaufes .....	- 51 -
4.1.1	Konzeption der geometrischen Grundelemente .....	- 51 -
4.1.2	Konzeption von Art und Umfang der Nutzerinteraktion .....	- 52 -
4.2	Konzeption der Mensch-Maschine-Interaktion.....	- 53 -
4.2.1	Konzeption der Interaktionsschnittstellen.....	- 54 -
4.2.2	Konzeption der Führungseinheit.....	- 56 -
4.2.3	Sicherheitskonzept zur Mensch-Roboter-Interaktion .....	- 57 -
<b>5</b>	<b>Entwicklung eines Verfahrens zur Adaption von Bahndaten für die intuitive Programmierung .....</b>	<b>- 60 -</b>
5.1	Erstellung eines Modells der Aufgabe .....	- 60 -
5.2	Formale Lösung.....	- 60 -
5.2.1	Testimplementierung der Algorithmen .....	- 61 -
5.2.2	Kompression der Bahndaten.....	- 62 -
5.2.3	Segmentierung der Bahn und Mapping von Geometrieprimitiven .....	- 66 -
5.3	Manuelle Wahl der optimalen Kompressionsparameter .....	- 69 -
5.4	Automatische Wahl der optimalen Kompressionsparameter .....	- 72 -
5.5	Zusammenfassung der Erkenntnisse und Schlussfolgerungen.....	- 73 -
<b>6</b>	<b>Entwicklung eines Verfahrens zur Bahnoptimierung auf der Basis der sensorisch erfassten lokalen Werkstückgeometrie .....</b>	<b>- 74 -</b>
6.1	Datenakquisition .....	- 75 -
6.2	Filterung der Sensordaten .....	- 76 -
6.3	Scanlinienbasierte Naherkennung.....	- 76 -
6.3.1	Kantendetektion an Hand des Krümmungskriteriums .....	- 78 -
6.3.2	Auswahl der Kantenpunkte an Hand heuristischer Kriterien .....	- 79 -
6.3.3	Erzeugung einer Bahn .....	- 80 -
6.3.4	Test und Verifikation.....	- 81 -
6.4	Scanlinienbasierte Flächenrückführung.....	- 84 -
6.4.1	Segmentierung mittels Region-Growth-Verfahren.....	- 85 -
6.4.2	Optimierung des Region-Growth-Verfahren .....	- 88 -
6.4.3	Ermittlung der Schnittgeraden, Interaktion.....	- 90 -

6.4.4	Reduzierung des geometrischen Fehlers .....	- 91 -
6.4.5	Reduzierung der Laufzeit .....	- 93 -
6.4.6	Use-Case- und Klassen-Entwurf .....	- 93 -
<b>7</b>	<b>Realisierung und Erprobung des Programmierverfahrens .....</b>	<b>- 95 -</b>
7.1	Gesamtaufbau der Versuchszelle .....	- 95 -
7.2	Eingesetzte Teilsysteme und Komponenten.....	- 96 -
7.2.1	Industrieroboter und Dreh-Kipp-Tisch.....	- 96 -
7.2.2	Schweißanlage .....	- 96 -
7.2.3	Sensorsystem .....	- 97 -
7.2.4	Mensch-Maschine-Schnittstelle.....	- 97 -
7.3	Steuerungskomponenten .....	- 98 -
7.3.1	Software-Architektur .....	- 98 -
7.3.2	Steuerungskomponenten zur manuellen Führung des Roboters.....	- 99 -
7.3.3	Steuerungskomponenten zur Nachbearbeitung des aufgenommenen Programmes .....	- 100 -
7.3.4	Format des erzeugten Robotercodes.....	- 102 -
7.4	Ablauf der Programmierung .....	- 103 -
7.5	Erprobung am Beispiel eines Galerieträgers.....	- 104 -
7.5.1	Spezifikation der Aufgabe .....	- 105 -
7.5.2	Test und Ergebnisse des Programmierens durch Vormachen.....	- 107 -
7.5.3	Test und Ergebnisse der grafischen Nachbearbeitung .....	- 108 -
7.5.4	Test und Ergebnisse der sensorbasierten Bahnoptimierung.....	- 108 -
7.5.5	Prozesszeiten der Programmierung.....	- 110 -
7.6	Schlussfolgerungen aus Realisierung und Erprobung.....	- 111 -
<b>8</b>	<b>Zusammenfassung und Ausblick .....</b>	<b>- 112 -</b>
<b>9</b>	<b>Abstract.....</b>	<b>- 114 -</b>
<b>10</b>	<b>Referenzen .....</b>	<b>- 116 -</b>

# Formelzeichen

## Namen und Abkürzungen

CFK	Carbonfaserverstärkter Kunststoff
CNC	Computerized Numerical Control, computergestützte Maschinensteuerung
COM	Component Object Model
GER	Geometric Reverse Engineering
IPO	Interpolationstakt, zentraler Rechentakt in Maschinen- und Robotersteuerungen
IP65	Industrieschutzklasse bzgl. Staub / Feuchtigkeit
KCP	KUKA Control Panel
KMS	Kraft-Momenten-Sensor
KMU	Kleine und mittlere Unternehmen: Unternehmen mit einer Mitarbeiterzahl < 250 und einem Jahresumsatz < 50 Mio. € bzw. Jahresbilanz höchstens 43 Mio. € (EU-KMU 2008).
MAG	Metall-Aktiv-Gas: Lichtbogen-Schweißverfahren mit Verwendung eines aktiven Schutzgases, etwa Corgon (82 % CO <sub>2</sub> , 18 % Argon)
MRK	Mensch-Roboter-Kooperation
PHG	Programmierhandgerät
RSI	Robot Sensor Interface, Sensorschnittstelle der Fa. KUKA Roboter
RTX	Echtzeiterweiterung der Fa. IntervalZero
SiNumerik	CNC-Steuerung der Fa. Siemens
VxWorks	Echtzeit-Betriebssystem der Fa. Wind River Systems

## Formelzeichen

$\kappa$	Krümmung
$n_b$	Anzahl der Klassen des Histogrammes
$\kappa_{max}$	Maximaler Krümmungswert
$\varphi$	Drehwinkel zweier Geraden zueinander
$\delta$	Abstand zweier Geraden
$\theta$	Shift zweier Geraden
$\Delta_M$	Abstand der Mittelpunkte zweier Geraden
$a$	Nahtdickenparameter
$\Delta_{DP}$	Kompressionsparameter
$\alpha_{dist}$	Segmentierungsparameter
$C^n$ -stetig	Kriterium zur Stetigkeit und Differenzierbarkeit von Funktionen

# 1 Einleitung

## 1.1 Problemstellung

Industrieroboter sind seit den 50er-Jahren ein Kernelement der Automatisierung und leisten einen wesentlichen Beitrag zur Stärkung der Wettbewerbsfähigkeit der Unternehmen. Beinahe eine Million Industrieroboter weltweit sorgen insbesondere in den Branchen der Automobilindustrie, chemischen Industrie und dem Maschinenbau für eine hohe und reproduzierbare Qualität, geringe Taktzeiten und die Verringerung der Fertigungskosten (IFR 2009).

Begünstigt wurde diese Entwicklung durch die Struktur der roboterintensiven Industrien. Im Gegensatz zum allgemeinen verarbeitenden Gewerbe zählen über 70 % der Unternehmen in der Automobilindustrie mehr als 1.000 Mitarbeiter (DeStatis 2008). Hohe Stückzahlen, Fertigung im Schichtbetrieb und Simultaneous Engineering begünstigen den Einsatz von Industrierobotern. Auf dieser Basis haben sich viele technologisch weit fortgeschrittene Lösungen für branchenspezifische Prozesse etabliert, etwa im Bereich der Verbindungstechnik (Reinhart 2008), der Montage (Zürn 2008) oder der Qualitätssicherung (Rawiel 2006).

Im Zuge der Marktveränderungen sind aber auch kleine und mittelständische Unternehmen (KMU) immer stärker darauf angewiesen, Innovationen umzusetzen, um wettbewerbsfähig zu bleiben. Dies kann in Form von innovativen Produkten oder Dienstleistungen geschehen, aber auch durch eine Rationalisierung der Produktion. Der Einsatz von Industrierobotern verspricht hier ebenfalls Verbesserungen in Qualität, Ausstoß und Kosten.

Der einfache Transfer der für die Automobilbranche, die chemische Industrie oder den Maschinenbau entwickelten Lösung schlägt aber oft fehl, zu unterschiedlich sind die Rahmenbedingungen. Häufig wird nicht in Großserie, sondern in kleineren Losgrößen produziert, die Betriebsgröße lässt keine Robotik-Abteilung für Projektierung, Umsetzung und Instandhaltung von Roboteranlagen zu, eine Produktion im 1-Schichtmodell verschärft Rentabilitätsfragen.

Insbesondere bezüglich der Programmierung der Roboter, die in den etablierten Branchen oft offline erfolgt, gibt es nur wenige Lösungen speziell für KMU. Oft ist die Programmierung der roboterbasierten Fertigungsanlagen zu komplex und zu langwierig (Kinkel 2009, Weck 2006).

## 1.2 Zielsetzung und Vorgehensweise

Ausgehend von der Problemstellung sind die Ziele der vorliegenden Arbeit:

- Anforderungen aus den KMU-spezifischen Aufgabenstellungen abzuleiten und daraus Konzepte für die Programmierung von Industrierobotern nach der Methode der Programmierung durch Vormachen zu erstellen,

- Verfahren zu entwickeln, die eine intuitive, schnelle und genaue Programmierung von Roboterzellen erlauben und
- die Realisierung einer Roboterzelle zur Verifikation der entwickelten Konzepte und Verfahren.

Die zur Erreichung der Ziele verfolgte Vorgehensweise beruht auf der wissenschaftlichen Methode der Kombination und Bewertung verschiedener Teilalgorithmen und gliedert sich in 6 wesentliche Teilaufgaben:

- *Kapitel 2: Grundlagen und Ausgangssituation:*  
Der aktuelle Stand bei der Programmierung von Industrierobotern und der sicheren Mensch-Roboter-Interaktion wird analysiert. Folgerungen aus dem Stand der Technik und Wissenschaft werden abgeleitet.
- *Kapitel 3: Analyse der Fertigungssituation bei KMU:*  
Die Situation des Robotereinsatzes in Bezug auf Branchen, Betriebsgrößenklassen und Anwendungen wird untersucht. Verschiedene Anwendungsszenarien und die daraus entstehenden Anforderungen an das Programmiersystem werden abgeleitet. Es werden die Teilkomponenten des Programmierverfahrens und der aufzubauenden Roboterzelle definiert.
- *Kapitel 4: Konzeption des intuitiven Programmierverfahrens und des Gesamtsystems zum MAG-Schweißen:*  
Das Programmiersystem und seine wesentlichen Teilkomponenten werden konzipiert.
- *Kapitel 5: Entwicklung eines Verfahrens zur Adaption von Bahndaten für die intuitive Programmierung:*  
Ein Verfahren zur Kompression aufgenommener Bahndaten wird entwickelt, bestehend aus den Teilverfahren Bahnkompression, Segmentierung und Mapping von Geometrieprimitiven. Verschiedene Algorithmen werden kombiniert und evaluiert.
- *Kapitel 6: Entwicklung eines Verfahrens zur Bahnoptimierung auf der Basis der sensorisch erfassten lokalen Werkstückgeometrie:*  
Zur Integration einer Genauigkeitsverbesserung auf der Basis lokaler Sensordaten werden zwei Verfahren entwickelt, die scanlinienbasierte Bahngenerierung und die scanlinienbasierte Flächenrückführung.
- *Kapitel 7: Realisierung und Erprobung des Programmierverfahrens:*  
Die Programmierumgebung wird in eine MAG-Schweißroboterzelle integriert und in einem Maschinenbauunternehmen getestet. Daten und Aussagen zu den Testläufen werden zusammengestellt.

## 2 Grundlagen und Ausgangssituation

### 2.1 Begriffe und Definitionen

Industrieroboter	„Automatisch gesteuerte, frei programmierbare Mehrzweck-Manipulatoren, die zur Handhabung von Werkzeugen oder Werkstücken in drei oder mehr Achsen programmierbar sind und zur Verwendung in der Automatisierungstechnik entweder statisch oder mobil angeordnet sein können“ (DIN EN ISO 8373).
Serviceroboter	„Ein Serviceroboter ist eine frei programmierbare Bewegungseinrichtung, die teil- oder vollautomatisch Dienstleistungen verrichtet. Dienstleistungen sind dabei Tätigkeiten, die nicht der direkten industriellen Erzeugung von Sachgütern, sondern der Verrichtung von Leistungen für Menschen und Einrichtungen dienen.“ (Schraft 1998)
Programmierung durch Vormachen	Der Begriff <i>Programmierung durch Vormachen</i> wird in dieser Arbeit für das direkte aktive oder passive Führen des Roboters mit gleichzeitiger Aufzeichnung der Trajektorie verwendet. Ein weiterer bekannter Begriff ist <i>Programming by Demonstration</i> , aber auch die Begriffe <i>Lead-Through-Programming</i> und <i>Guide-Through-Programming</i> werden häufig synonym benutzt (Vogl 2009).
Mensch-Roboter-Kooperation (MRK)	Bei der MRK interagieren Mensch und Roboter direkter, als es bei herkömmlichen Automatisierungsanlagen üblich ist. Diese Interaktion kann zu abwechselnden Zeiten, bei wechselnder Nutzung des Arbeitsraumes oder bei gemeinsamer Arbeit im Arbeitsraum stattfinden. In der DIN-EN-ISO-Norm 10218-1 wird für MRK der Begriff <i>Collaborative Operation</i> in Abgrenzung zu der <i>Cooperative Operation</i> von zwei interagierenden Industrierobotern benutzt.
Systemintegrator	Bei Projekten zur roboterbasierten Automatisierung interagieren im Wesentlichen drei Partner: der Endnutzer ist der Auftraggeber und verwendet das Robotersystem in seiner Fertigung; der Roboterhersteller liefert die Roboter; der Systemintegrator ist der Auftragnehmer des Endnutzers und ist für die Systemlösung und damit die produktive Fertigungsanlage ver-

antwortlich. Roboterhersteller und Systemintegrator können identisch sein.

MAG-Schweißen

Lichtbogen-Schweißverfahren mit Verwendung eines reaktiven Schutzgases (Metall-Aktiv-Gas), etwa Corgon (82 % CO<sub>2</sub>, 18 % Argon). Mit 46 % das verbreitetste Schweißverfahren (Schal 2002).

Intuitive  
Programmierung

Begriff für ein auf der Programmierung durch Vormachen basierendes Verfahren mit einer intuitiven Nutzerschnittstelle.

## 2.2 Abgrenzung der intuitiven Programmierung zu anderen Programmiermethoden

### 2.2.1 Kategorisierung der Programmiermethoden

Die Programmierung von Industrierobotern ist eine zeit- und damit kostenintensive Tätigkeit, die sich entscheidend auf die spätere Prozessqualität und Flexibilität der Fertigungsanlage auswirkt. Entsprechend vielfältig sind die Programmierverfahren, die zur Wahl stehen. Ihre systematische Unterteilung nehmen verschiedene Kategoriensysteme vor. Lozano-Pérez unterteilt die Verfahren in *Geführte Systeme*, *Roboter-Level Programmiersysteme* und *Aufgaben-Level Programmiersysteme* (Lozano 1983). Biggs und McDonald unterteilen in *manuelle* und *automatische Programmierung*. Dabei kann die manuelle Programmierung etwa textbasiert oder graphisch durchgeführt werden, die automatische Programmierung erfolgt durch Vormachen, instruktive Systeme oder lernende Systeme (Biggs 2003).

Eine greifbarere Kategorisierung wird von der amerikanischen Occupational Safety and Health Agency OSHA vorgenommen (OSHA 2008). Wie in Bild 2.1 dargestellt unterteilt sie in die Programmierung mit dem *Programmierhandgerät (PHG)*, dem *Programmieren durch Vormachen* und in *Offline-Programmiersysteme*. Die Begrifflichkeiten wurden dabei an aktuelle Arbeiten angepasst (Schraft 2005, Meyer 2007-1, Vogl 2009).

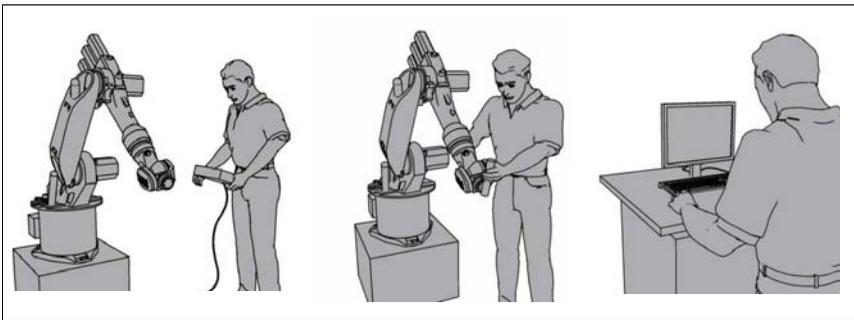


Bild 2.1: Programmierung mit dem Programmierhandgerät, Programmieren durch Vormachen und Offline-Programmierung (von links nach rechts).

Typische Beispiele für diese im Folgenden verwendete Kategorisierung sind in der Forschungslandschaft ebenso wie im industriellen Alltag zu finden.

- **Kategorie A: Programmierung mit dem Programmierhandgerät**  
Hersteller von Industrierobotern liefern Robotersysteme typischerweise mit einem Programmierhandgerät aus, bzw. bieten die Option dazu. Die Programmierung erfolgt durch die Bewegung des Roboters über Verfahrstasten, Joysticks oder 6-D-Mäuse. Die Programmerstellung erfolgt über Texteingabe, eine Menüstruktur oder einen Touchscreen. (Kazi 2005, Weck 2006, KUKA 2008)

- *Kategorie B: Programmierung durch Vormachen*  
Die zu programmierende Bewegung wird dem Roboter vorgemacht, aufgezeichnet und bei Bedarf wieder abgespielt. Entwickelt hat sich diese Programmiermethode insbesondere im Bereich der Lackierroboter, wo die Dynamik der Bewegung von großer Bedeutung ist (Paintmate 2009, Foitzik 2001). Viele weitere Ansätze wurden im Forschungs- und Entwicklungsbereich umgesetzt (Albu-Schäffer 2002, Heiligensetzer 2005, Som 2007).
- *Kategorie C: Offline-Programmierung*  
Meist auf der Basis von CAD-Daten der Roboterzelle und des Werkstücks können automatisch oder manuell Prozessbahnen und damit Roboterprogramme erstellt werden. Die Variantenbreite reicht hier von einfachen Systemen, die Bahnpunkte durch Anklicken eines Geometrie-Features erstellen bis hin zu vollautomatischen Codegeneratoren, die mehrere Portalroboter kollisionsüberwacht etwa im Bereich des Schweißens von Schiffskonstruktionen ansteuern (Rinas 2009). Wesentliche Voraussetzung ist die Verfügbarkeit von qualitativ hochwertigem CAD-Material (Denkena 2004).

Im relevanten Ausmaß industriell genutzt werden die Methoden der Programmierung durch das Handgerät sowie die Offline-Programmierung (Hein 2007, Dilthey 2005).

## **2.2.2 Einordnung der untersuchten Programmiermethode**

In dieser Arbeit werden intuitive Programmierverfahren auf der Basis der Kategorie B, des *Programmierens durch Vormachen*, untersucht.

Je nach der spezifischen Aufgabenstellung und der Ausprägung einer Reihe von Kriterien, unter anderem betreffend die Verfügbarkeit von CAD-Daten, die Programmierhäufigkeit oder die Notwendigkeit des Simultaneous Engineering, eignen sich Verfahren der Kategorien A und C möglicherweise besser für die Programmierung einer Fertigungsanlage.

## **2.3 Stand der Technik**

### **2.3.1 Beispiele für intuitiv programmierbare Roboterzellen**

In Bild 2.2 sind Beispiele für Roboterzellen dargestellt, die die Methode des Programmierens durch Vormachen verwenden. In der oberen Reihe ist ein KUKA-Roboter mit dem KUKA Guiding Device (Heiligensetzer 2005) dargestellt, daneben der Leichtbauroboter LBR3 des Deutschen Zentrums für Luft- und Raumfahrt DLR (Hirzinger 2002) und die am IWB der TU München entwickelte Augmented-Reality-unterstützte Programmierung (Zäh 2005, Vogl 2009). Das Guiding Device vereint Sicherheitsaspekte mit der Programmierung. Es ermöglicht die Bewegung des Roboters über einen 6-D-Joystick. Der LBR3 verfügt über sieben einzeln kraftgeregelter Achsen, die eine sehr feinfühligere Bewegung des Roboters durch Führen

erlauben. Dabei ist es möglich, entlang der gesamten Struktur des Roboters lenkend einzugreifen. Der Arm wird insbesondere für die Entwicklung von Anwendungen in der Servicerobotik verwendet.



Bild 2.2: Beispiele für durch Vormachen programmierbare Roboterzellen

Die Augmente-Reality-unterstützte Programmierung ermöglicht die Eingabe von Punkten und Bahnen mittels eines IR-getrackten Stiftes, der Programmierer wird durch situationsangepasste Laserprojektionen, etwa der bisher programmierten Bahn, unterstützt. In der unteren Reihe sind Produkte und Entwicklungen der Fa. Paintmate SA (Paintmate 2009) und des IPR der Universität Karlsruhe (Hein 2007) dargestellt. Der Lackierroboter Paintmate wird unter anderem durch Vormachen programmiert, dazu werden die Aktoren der ausbalancierten Kinematik passiv geschaltet. Die Roboterzelle des IPR ermöglicht ebenfalls eine projektionsbasierte Unterstützung des Nutzers. Zum Führen wird unter anderem das kameragetrackte Zeigergerät verwendet.

Diese für Implementierungsbeispiele der Kategorie B exemplarisch ausgewählten Programmierverfahren und Roboterzellen geben eine Impression der Vielfalt der Lösungen. Im folgenden Abschnitt werden diese Lösungen und Ansätze weiter klassifiziert.

### 2.3.2 Alternative Verfahren zur Aufnahme von Geometriedaten

Der Roboter muss für seine Aufgabenausführung die zu verfahrenen Trajektorien kennen, neben weiteren prozess- und systemrelevanten Informationen sind dies die notwendigen Kerndaten. Im Bereich des Programmierens durch Vormachen wird dazu die Roboterkinematik oder eine alternative Teach-Einrichtung an die zu programmierenden Punkte oder über die zu programmierenden Prozessbahnen geführt, dabei werden die Punkte und Bahnen

aufgezeichnet. Realisierungsalternativen sind in Tabelle 2.1 aufgeführt, im Wesentlichen zu unterscheiden in die drei Fälle *Roboter passiv* (A), *Roboter aktiv* (B, C, D) und *Teacheinrichtung* (E, F).

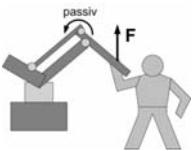
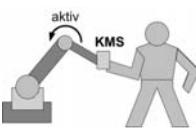
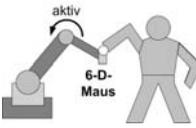
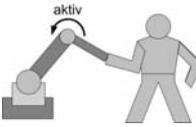
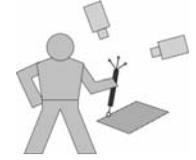
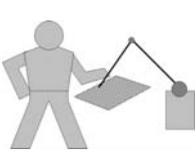
	Skizze	Beschreibung	Realisierungen / Prozesse
A		Roboter passiv: Bewegen der passiv geschalteten Struktur	Programmierung von Lackierrobotern, etwa (Paintmate 2009). Die Kinematik ist meist konstruktiv ausbalanciert, für den Prozess ist insbesondere die Dynamik von Belang.
B		Roboter aktiv: Aktive Steuerung und Regelung des Roboters über einen oder mehrere Kraft-Momenten-Sensoren	Eine Übersicht ist gegeben in (Albu-Schäffer 2002), als Produkt u.a. von ABB erhältlich (ABB 2008). Anwendungen sind unter anderem Lackieren (Drews 1997, Foitzik 2001), Entgraten (Schulte 2003) und Handling.
C		Roboter aktiv: Steuerung des Roboters über 6-D-Maus oder Joystick	Vielfältige Anwendungen im Handling (Som 2007), beim Entgraten (Heiligensetzer 2005) oder der kooperierenden Montage (Schrift 2005)
D		Roboter aktiv: Regelung über Motorströme (Halten der Ströme oder Vergleich mit Modell)	Anwendungen überwiegend für Kleinroboter, etwa Pick-and-Place und Montageaufgaben (MacGregor 2009, Neuronics 2009), für größere Kinematiken viele Störeinflüsse (Colombo 2006)
E		Teacheinrichtung: Programmierung mit getrackten Zeigeeinheiten	Mit Trackingsystemen etwa auf Infrarot-Basis werden Roboter beispielsweise für Laseranwendungen (BlackbirdRobotics 2010, Vogl 2009) und Schweißanwendungen programmiert.
F		Teacheinrichtung: Eine leichte Programmierkinematik wird an Stelle des Roboters aufgestellt und zum Programmieren verwendet.	Programmierkinematiken werden von ABB für Anwendungen im Gießerei-Bereich untersucht (Marshall 2004). Mit Messarmen etwa von Faro oder Nikon können ebenfalls Bahnen für Roboterprogramme aufgenommen werden (Krause 2009).

Tabelle 2.1: Stand der Technik für verschiedene Verfahren des Programmierens durch Vormachen

### 2.3.3 Alternative Verfahren zur Nachbearbeitung von Trajektorien

Zusätzlich zu den Geometriedaten, deren Ermittlung im vorherigen Abschnitt dargestellt wurde, sind weitere Informationen notwendig: Prozessparameter beschreiben, wie die Peripherieanlagen anzusteuern sind, komplexere Anlagen bedürfen der Synchronisation durch Kommunikation, Fehlerzustände und ihre Behandlung müssen definiert werden. Die heute verfügbaren Möglichkeiten zur Anpassung eines aufgenommenen Programmes sind in Tabelle 2.2 aufgelistet. Im Wesentlichen gibt es zwei Möglichkeiten, auf Basis des Programmquellcodes (A, B) und auf der Basis grafischer Schnittstellen (C, D).

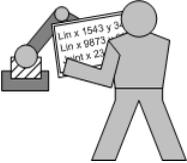
	Skizze	Beschreibung	Realisierungen / Prozesse
A		Quellcode durch PHG: Editierung des Programms direkt an der Roboterzelle	Standard-Lösung zur Programmanpassung in der Fertigung
B		Quellcode durch Editor: Editierung des Programms an einem PC	Standard-Lösung zur Programmanpassung bei allen Roboterherstellern, Down- und Upload der Programme über USB oder Netzverbindungen
C		Grafisch auf PHG: Anpassung mit Hilfe grafischer Interfaces auf dem PHG	Möglich bei Einsatz eines grafischen PHG, etwa des FlexPendant auf der ABB-Steuerung IRC-5 (ABB 2004)
D		Grafisch auf PC: Anpassung des Programms an einem externen PC mit Visualisierung	Möglich durch Verwendung eines Offline-Programmiersystems wie KUKASim, RobOffice, RobotStudio, IGRIP, 3DCreate, oder ähnliche

Tabelle 2.2: Stand der Technik für die Alternativen der Programmnachbearbeitung

Sensoren werden bisher nicht für die Nachbearbeitung aufgenommenen Trajektorien verwendet. Mit Hilfe von handgeführten Scannern (Nikon 2010-1) können Werkstücke digitalisiert und diese Daten als Grundlage einer Offline-Programmierung verwendet werden. Dieses Vorgehen ist aber der Kategorie C, siehe Kapitel 2.2.1, zuzuordnen.

### 2.3.4 Steuerungen als Basis der Programmierung

Um mit der Kinematik des Roboters eine Bewegung ausführen zu können, ist eine Robotersteuerung notwendig, die das Bewegungsprogramm verwaltet, interpoliert und aus den Positionsdaten die notwendigen Soll-Achsstellungen erzeugt. Weitere Aufgaben der Robotersteuerung sind die Kommunikation mit Sensoren, weiteren Aktoren und externen Kommunikationseinheiten, die etwa über Feldbussysteme verbunden sein können.

Robotersteuerungen sind grundsätzlich in der Lage, diese Aufgaben wahrzunehmen. Sie unterscheiden sich aber in Realisierungsdetails, etwa dem Interpolationstakt (IPO-Takt), der Breite der möglichen Kommunikationsschnittstellen, der Eingriffsmöglichkeiten und der Art der Geometrieprimitive (Wörn 2005 und Datenblätter der Hersteller).

Eine Grundunterscheidung kann zwischen roboterspezifischen Steuerungen und allgemeinen CNC-Steuerungen, die für Roboter angepasst wurden, getroffen werden. Erstere sind optimiert auf die Anforderungen typischer Roboter-Einsatzszenarien, letztere entstammen dem Werkzeugmaschinenbereich. Tabelle 2.3 vergleicht die Steuerungen einiger Roboterhersteller und Speziallösungen.

	Hersteller	Steuerungs-Basis	IPO-Takt	Geometrie-elemente	Handführung
Spezifische Robotersteuerung	A KUKA Roboter KRC2	PC-basiert, Windows mit VxWorks	12 ms	KRL-Programmiersprache: Joint-, Linear- und Zirkularinterpolation	u.a. mittels RSI-Schnittstelle, Anpassung der Sollwerte im IPO-Takt
	B ABB IRC5	PC-basiert WinCE	8 ms	RAPID-Programmiersprache: Joint-, Linear- und Zirkularinterpolation	u.a. Kraft/Handführung als Produktionsoption, Regelung über Achscontroller
	C Reis Robotics RSV	PC-basiert	12 ms	RSV-Programmiersprache: Joint-, Linear- und Zirkularinterpolation	u.a. Vorgabe von Sollwerten über XML-Kommunikation mit ca. 40 Hz
	D Denso RC7M	spezifisch	4 ms	PAC-Programmiersprache: Joint-, Linear- und Zirkular und Splineinterpolation	u.a. Kommunikation und Setzen von Variablen über ORIN-Schnittstelle mit ca. 30 Hz.
CNC-Steuerung	E Siemens SiNumeric 802D	spezifisch	n/a	CNC-Code	u.a. Bahnanpassung durch Offset-Addition
	G Beckhoff TwinCAT 2	PC-basiert, Windows mit RTX-Kern	ab 50 µs	CNC-Code	u.a. OPC-Echtzeitzugriff auf Soll- und Istwerte der Steuerung

Tabelle 2.3: Kenndaten der im Bereich der Robotik eingesetzten Steuerungen.

### 2.3.5 Personensicherheit

Die direkte Mensch-Roboter-Kooperation (MRK) wird seit etwa 2001 intensiv diskutiert und vorangetrieben (Helms 2007, Krüger 2009). Durch das MRK-Konzept wird erwartet, bisher manuelle Prozesse zumindest teilautomatisieren zu können und dadurch die Produktion weiter zu rationalisieren. Die wesentlichen technischen Fragestellungen sind die Aufteilung der Prozessschritte zwischen Mensch und Roboter und die Sicherheit der Kooperation.

Beispiele für industriell relevante Mensch-Roboter-Kooperationszellen sind bis heute noch selten, eine Auswahl ist in der folgenden Tabelle 2.4 dargestellt.

Die Aufgabenteilung wurde insbesondere in Forschungsprojekten wie ASSISTOR (Helms 2006) oder PiSA (PiSA 2008) untersucht. Die Diskussion der Sicherheit mündete 2006 in der Erstellung der Norm für die Sicherheit von Industrierobotern, der (DIN EN ISO 10218-1, Umbreit 2007).

Die meisten in der Tabelle 2.4 aufgeführten Zellen dienen dem Zweck der kooperierenden Montage, etwa für Sitze, Kleinteile oder KfZ-Module. Hier sind steuerungstechnisch überschaubare Anforderungen an die notwendigen Sicherheitssysteme gestellt, nach der DIN EN ISO 10218-1 Abschnitt 5.10.3 *Collaborative Operation Requirements – Hand Guiding* muss die Sicherheitssteuerung der Kategorie 3 der (DIN EN ISO 13849-1) entsprechen und bei Betätigung eines Zustimmenschalters eine Maximalgeschwindigkeit von 250 mm/s überwachen, alternativ den sicheren Stillstand des Roboters.

Darüber hinausgehende Verfahren, etwa zur Detektion des Werkers mit geometrisch auflösenden Sensoren und einer positionsabhängigen Reaktion des Roboters sind industriell bisher nicht einsetzbar. Hier kommt das Produkt SafetyEye (Pilz 2009) den Wünschen der Anwender am nächsten, auf Forschungsebene erlauben Überwachungssysteme etwa auf Time-of-Flight-Basis (Winkler 2008) oder basierend auf Standard-Kamerasystemen (Henrichs 2006) neue Anwendungen.

In Bezug auf den hier dargestellten Anwendungsfall der Roboterprogrammierung bieten diese sicheren Roboteranlagen einen entscheidenden Vorteil: mit ihnen darf jeder Prozessexperte die Fertigungsanlage programmieren, nicht nur der ausgebildete Roboterprogrammierer. Dies vermindert den Bedarf an hochqualifiziertem Personal und damit ein wesentliches Einstiegshemmnis für KMU.

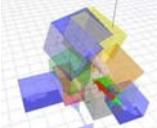
	Abbildung	Anwendung	Sicherheit
A		Assistierte Montage von Cockpitmodulen (Garcia 2007)	MRK nach DIN EN ISO 10218-1, Abschnitt 5.10.3: Zweihand-Zustimmschalter, Geschwindigkeits- und Stillstandsüberwachung basierend auf der KUKA SafeRobot Technologie
B		Positionierung von Schweißblechen (Som 2007)	MRK nach DIN EN ISO 10218-1, Abschnitt 5.10.3: Zustimmschalter, Geschwindigkeits- und Stillstandsüberwachung basierend auf Reis Safety Controller
C		Prozessbeobachtung beim Aufpanzern von Förderschnecken (Som 2007)	Prozessbeobachtung: Geschwindigkeits- und Bereichsüberwachung basierend auf Reis Safety Controller
D		Assistierte Montage von Hinterachs-differentialgetrieben (Schraft 2005, Meyer 2007-1).	MRK nach DIN EN ISO 10218-1, Abschnitt 5.10.3: Zweihand-Zustimmschalter, Geschwindigkeits-, Bereichs- und Stillstandsüberwachung basierend auf Reis Safety Controller
E		Arbeitsteilige Montage von Ölpumpen (Thiemermann 2005)	3-D-Erfassung von Körperteilen mittels Stereo-Kamerasystem. Abbremsen des Robotersystems. Keine sicherheitsgerichtete Hardware.
F		Assistierte Montage von Sitzeinheiten (Matthias 2008)	MRK nach DIN EN ISO 10218-1, Abschnitt 5.10.3: Zweihand-Zustimmschalter, Geschwindigkeits- und Stillstandsüberwachung basierend auf der ABB SafeMove Technologie
G		Interaktion von Mensch und Roboter mit 3-D-TOF-Sensorik (Winkler 2008).	Detektion des Abstandes des Menschen zum Roboter mittels 3-D-Kamera. Reduktion der Robotergeschwindigkeit entsprechend dem Restabstand. Bisher keine sicherheitsgerichtete Hardware. Kamerabasierte Verfahren zur frühen Detektion sind etwa (Henrichs 2006)

Tabelle 2.4: Beispiele zum Stand der Technik für Sicherheitskonzepte zur Mensch-Roboter-Kooperation

## 2.4 Folgerungen aus dem Stand der Technik und Forschung

Bezogen auf die Zielsetzung in Kapitel 1 lässt der Stand der Technik und Forschung folgende Schlussfolgerungen zu:

- Die Programmierung von Industrierobotern erfolgt heute meist über das Programmierhandgerät oder eine Offline-Programmiersoftware.
- Alternative Programmierverfahren werden entwickelt und in ersten Applikationen getestet, haben aber bisher keine breite Anwendung gefunden.
- Für das räumliche Verfahren des Roboters sind diverse technische Lösungen vorhanden, sei es über kraftgeregelte Einzelachsen, über den Einsatz eines KMS oder mit Hilfe von 6-D-Mäusen bzw. Joysticks.
- Bisher realisierte Programmierverfahren bieten die Möglichkeit der Programmanpassung über das Programmierhandgerät oder über Offline-Programmiersysteme. Für die Programmierung durch Prozessexperten sind sie wenig geeignet.
- Die Nutzung von Sensorik für die Nachbearbeitung der Trajektorien ist nicht bekannt.
- Sowohl die technologische Entwicklung als auch die Normungslage erlauben die Umsetzung von Roboteranlagen, in denen Prozessexperten den Roboter führen bzw. programmieren.

Programmiersysteme, die ein Programmieren durch Vormachen erlauben, können auf der Basis bekannter Technologien aufgebaut werden. Vielfältige Beispiele in der Forschungscommunity und der industriellen Anwendung belegen dies. Neben der häufigen Spezialisierung dieser Programmiersysteme ist eine multimodale und intuitive Interaktion aber selten realisiert. Insbesondere die Nachbearbeitung der aufgenommenen Bahnen lässt sich bisher nicht für den Prozessexperten offensichtlich realisieren. Weiterhin ist es bisher nicht möglich, die programmierte Bahn durch Geometriesensoren an die Werkstücke anzupassen, also Toleranzen zu minimieren.

Das Konzept und die Realisierung einer Programmierumgebung, die die intuitive und multimodale Programmierung erlaubt, die Nachbearbeitung auf eine für den Prozessexperten offensichtliche Art und Weise ermöglicht und Sensordaten zur Verminderung von Toleranzen nutzen kann, ist bisher nicht bekannt. Als Konkretisierung von Abschnitt 1.2 ist es das Ziel dieser Arbeit, eine solche Programmierumgebung zur Verfügung zu stellen. Neben der Definition eines Anforderungskataloges erfordert dies die Entwicklung geeigneter Konzepte und Realisierungsmöglichkeiten unter Beachtung der technologischen und prozessrelevanten Randbedingungen.

### 3 Analyse der Fertigungssituation bei KMU

Die Analyse der manuellen und automatisierten Fertigung bei kleinen und mittelständischen Unternehmen stellt die Basis für die weiteren Konzeptions-, Entwicklungs- und Realisierungsschritte dar. Untersucht wird die aktuelle Verbreitung von Industrierobotern, mögliche Prozesse und die relevanten betrieblichen Rahmenbedingungen. Betrachtet werden technologische und ökonomische Randbedingungen. Es werden Szenarien und Anforderungen abgeleitet, an denen sich die weitere Entwicklung orientiert.

#### 3.1 Analyse der Produktionssituation kleiner Losgrößen mit Industrierobotern

Industrieroboter werden in vielen Branchen, Betrieben und Prozessen eingesetzt, Ende 2008 sind nach den Statistiken der International Federation of Robotics über 132.000 Robotersysteme in Deutschland im Einsatz, weltweit mehr als 950.000 (IFR 2009). Auf jeweils 10.000 Werk-tätige errechnen sich damit 186 Industrieroboter, wobei die Verteilung stark fokussiert ist: in der Automobilindustrie sind es 1.219 Roboter je 10.000 Werk-tätige, im allgemeinen Fertigungsbereich sind es noch 288. (alle Werte Deutschland, IFR 2009).

In Bild 3.1 ist dargestellt, in welchen Branchen Robotersysteme eingesetzt werden.

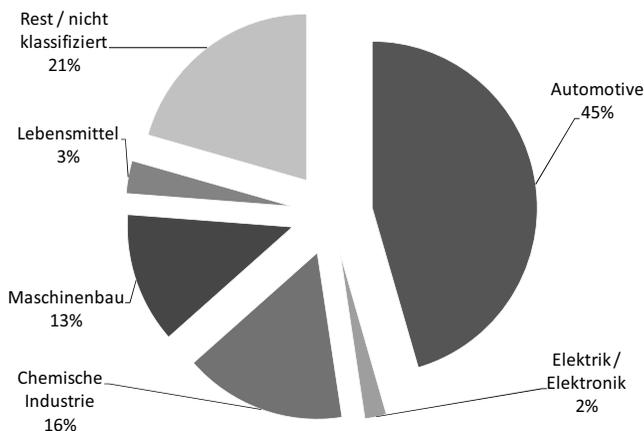


Bild 3.1: Prozentuale Verteilung der Industrieroboter nach Branchen (IFR 2009)

Aus der Zuordnung zu den Branchen ergibt sich eine starke Fokussierung auf den Automotiv-Bereich, dort herrschen hohe Stückzahlen, Mehrschichtbetrieb und eine lange Tradition des Robotereinsatzes vor. Präzisieren lässt sich die Aussage durch die Kombination mit der in

Bild 3.2 dargestellten Verteilung der Betriebsgrößen der Unternehmen (DeStatis 2008): Während im allgemeinen verarbeitenden Gewerbe eine etwa gleiche Verteilung der Werk­ tätigen auf die verschiedenen Betriebsgrößen gegeben ist, sind insbesondere im Automobilbau und in der chemischen Industrie Großunternehmen vorherrschend (73 % bzw. 43 % aller Werk­ tätigen in Unternehmen mit mehr als 1.000 Mitarbeitern).

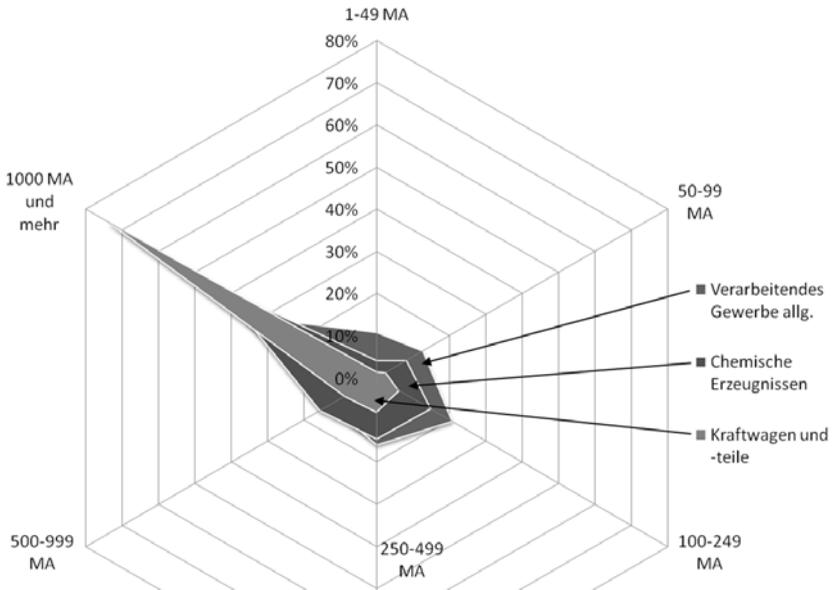


Bild 3.2: Prozentuale Verteilung der Werk­ tätigen nach Betriebsgröße (DeStatis 2008)

Ein Großteil der Industrieroboter ist also in Betrieben mit mehr als 1.000 Mitarbeitern installiert. Diese Unternehmen haben alle notwendigen Ressourcen, um die Investitionskosten, die Installation und den Betrieb zu realisieren.

Im allgemeinen verarbeitenden Gewerbe aber sind die kleinen und mittleren Unternehmen sowohl nach ihrer Zahl, als auch nach der Anzahl der Mitarbeiter, weitaus zahlreicher als die Großunternehmen, siehe Bild 3.3.

Diese statistikgetriebene Analyse lässt den Schluss zu, dass der wesentliche Anteil der Industrieroboter in Großunternehmen installiert ist. Das Potenzial, das ein Einsatz in den weit­ aus zahlreicheren Unternehmen der mittleren und kleinen Größenklassen bildet, ist weit­ gehend ungenutzt: die KMU profitieren im Wesentlichen nicht von dem Rationalisierungspotenzial der Industrieroboter. Die Roboterhersteller und Systemintegratoren konnten diesen Markt bisher kaum erschließen.

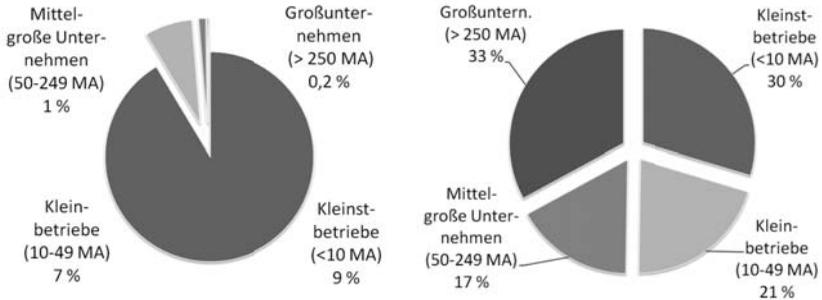


Bild 3.3: Prozentuale Verteilung der Betriebsgrößenklassen in der EU (links), Beschäftigtenzahlen nach Betriebsgrößenklassen (EU-weit, Nicht-Finanzsektor, Eurostat 2008)

Diese von technischen Randbedingungen bisher freie Aussage wird in den folgenden Abschnitten präzisiert.

### 3.1.1 Ablauf der manuellen Fertigung bei kleinen und mittleren Losgrößen

Die Fertigung, also die Transformation von Ausgangsstoffen unter dem Einsatz von Energie, Arbeitskraft und Produktionsmitteln in ökonomische Güter, umfasst im hier relevanten Sinne die Teilefertigung, die Montage und die Prüfung von Teilen (Westkämper 2006-1, Fischer 1992). Bei kleinen und mittleren Losgrößen wird häufig eine manuelle Fertigung vorgenommen, Prozesse und Transport werden von Arbeitskräften durchgeführt. Einzelne Prozesse können, etwa durch CNC-Stationen oder Roboter, automatisiert sein. Zur Organisation der Fertigung in Einmal-, Wiederhol- und Variantenfertigung sind insbesondere die Baustellenfertigung, das Werkstättenprinzip und Fertigungszellen zweckmäßig (Westkämper 2006-2). Ein für die Teilefertigung mit dem MAG-Schweißprozess einsetzbarer manueller Fertigungsablauf beinhaltet die Schritte vom Vorrichtungsbau bis hin zur Produktion (Arbeitsvorbereitung und Logistik nicht berücksichtigt), siehe Bild 3.4.

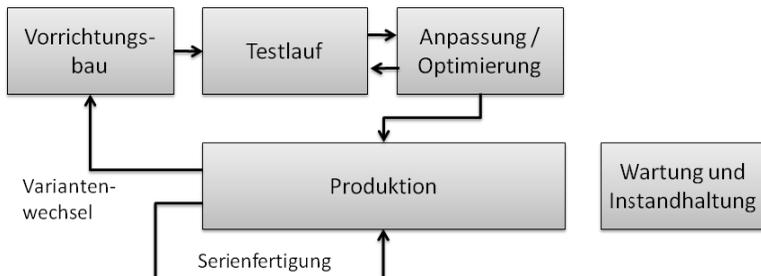


Bild 3.4: Ablauf der manuellen Fertigung für einen Schweißprozess.

Alle für die Produktion notwendigen Schritte liegen in der Hand des Fertigungsbetriebes. Zulieferfirmen sind häufig nur für den Ankauf von Halbzeugen, Verbrauchsstoffen und Geräten notwendig, das komplette Wissen um die Fertigung liegt im Betrieb.

### 3.1.2 Ablauf der robotergestützten Fertigung

Um die Produktivität und die Fertigungsqualität zu erhöhen, werden Industrierobotersysteme eingesetzt. Sie führen als frei programmierbare Manipulatoren einzelne Vorgänge oder Kombinationen von Prozessschritten schnell, automatisch und in hoher Qualität aus. Das Ablaufdiagramm in Bild 3.5 zeigt den Arbeitsablauf der robotergestützten Fertigung in einem Schweißszenario, nach Verantwortlichkeiten zwischen Systemintegrator (VS) und Nutzer (VN) unterschieden.

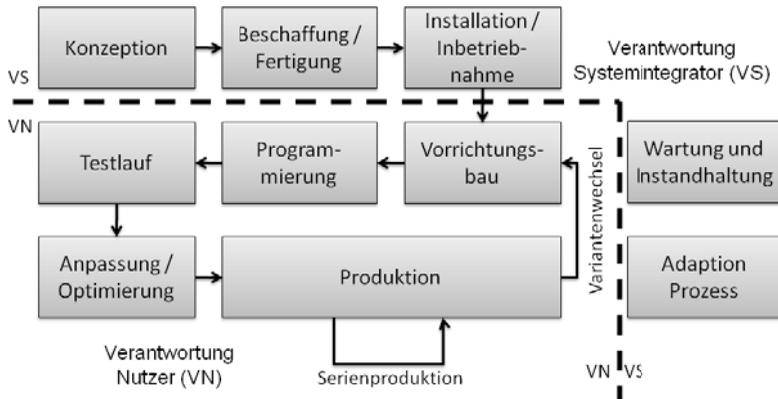


Bild 3.5: Ablaufdiagramm in einer robotergestützten Fertigung in einem Schweißszenario. Dargestellt ist eine typische Aufgabenteilung zwischen dem Systemintegrator und dem den Roboter einsetzenden Fertigungsbetrieb, dem Nutzer.

Dieser Ablauf ist in der industriellen Serienfertigung etabliert, er lässt sich aber schwer auf die Fertigung von kleinen und mittleren Losgrößen übertragen. Dies liegt insbesondere an der Investitionshöhe, dem Zeitaufwand der Programmierung und der notwendigen Qualifikation der Bediener (Haegele 2008).

**Höhe der Investition:** Die Investitionskosten von typischerweise >150 T€ können nicht auf eine hohe Zahl gleicher Bauteile über einen langen Zeitraum verteilt werden. Typisch für KMU sind hohe Variantenzahlen und geringe Auftragsvolumina.

**Zeitaufwand:** Die Einrichtung und Programmierung der Roboter auf das Bauteil ist zeitaufwändig, bei kleinen Losgrößen ergibt sich ein sehr ungünstiges Verhältnis der nicht wertschöpfenden Programmierzeit zur Prozesszeit.

**Qualifikation:** Einrichtung und Programmierung müssen von hoch qualifiziertem Personal vorgenommen werden. Während des Betriebes muss geschultes Personal zur Behebung von Fehlern verfügbar sein.

Lösungen für den Einsatz von Industrierobotern in der Fertigung von KMU, insbesondere also zur Überwindung der oben dargestellten Transferprobleme, werden aktuell sowohl in öffentlichen Forschungsprojekten (SMERobot 2008, PISA 2008) als auch von den Herstellern angegangen (Huggenberger 2007, Som 2007, Schäfer 2007). Eine marktabdeckende Lösung ist bisher allerdings nicht vorhanden.

## 3.2 Analyse potenzieller Anwendungsszenarien für die intuitive Roboterprogrammierung bei KMU

### 3.2.1 Klassifikation der mit dem Roboter durchgeführten Prozesse

Nach (IFR 2009) werden Roboterapplikationen in 6 Haupt- und 29 Unterprozesse unterschieden, Bild 3.6 zeigt eine Aufteilung mit der entsprechenden prozentualen Verteilung.

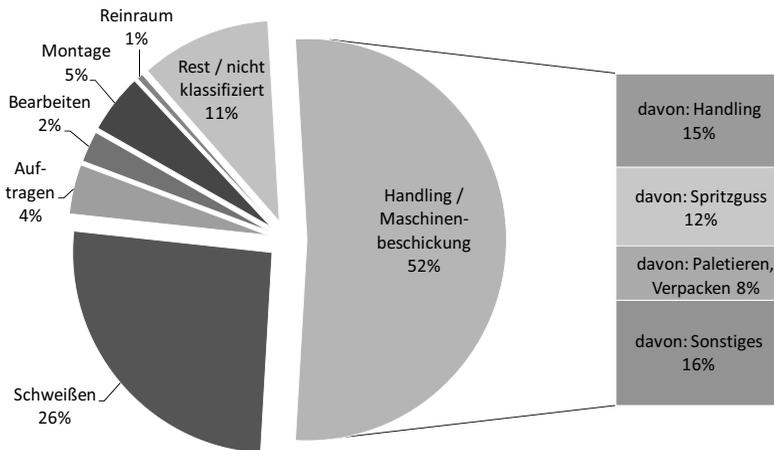


Bild 3.6: Prozentuale Verteilung der Industrieroboter nach Anwendungen (IFR 2009)

Diese Anwendungsklassen stellen jeweils spezielle Anforderungen an den Roboter, aber auch an die Programmierung. Die Roboterhersteller kommen diesen Anforderungen entgegen, indem sie angepasste Modelle und Softwareversionen mit entsprechenden Technologiefunktionen entwickeln. Klar erkennbar sind die Wurzeln der Industrierobotik im Handlingsbereich, mit 52 % ist jeder zweite Roboter mit so genannten Pick-and-Place Aufgaben beschäftigt. Als nächster großer Anwendungsfall kommen die Schweißaufgaben mit 26 %.

Hier muss zwar ein komplexer Prozess beherrscht werden, es treten aber nahezu keine Prozesskräfte auf, die notwendigen Positioniergenauigkeiten sind gut erreichbar. Für beide Anwendungsfälle, Handling und Schweißen, gibt es von den Roboterherstellern vielfältige Software- und Servicepakete zur Anwendungserstellung.

Trotz der großen Erfahrung durch die Roboter im Einsatz lassen sich nicht alle Anwendungen automatisieren, was einerseits an der technischen Umsetzbarkeit, andererseits an der Wirtschaftlichkeit liegt (Kinkel 2009, Kleine 2007). So werden 13 % aller Aufgaben ohne den Einsatz von Industrierobotern bewältigt, da keine wirtschaftlichen oder technischen Lösungen vorhanden sind, siehe Bild 3.7, für weitere 44 % fehlt noch das Anwendungskonzept.

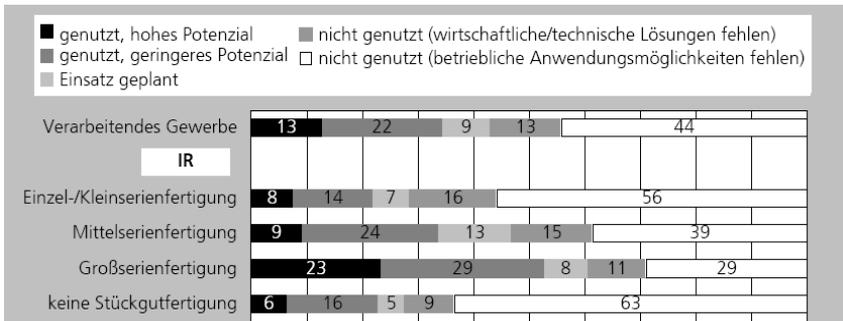


Bild 3.7: Einsetzbarkeit von Industrierobotern nach Losgröße aus (Kleine 2007).

Gerade für die Einzel- und Kleinserienfertigung ist dieses Potenzial mit 16 % am größten, einem bisherigen Robotereinsatz in 22 % der Prozesse steht ein Potenzial von weiteren 23 % gegenüber.

Diese weiteren Untersuchungen haben einige Gründe für die bisherige Konzentration des Robotereinsatzes auf Großbetriebe dargestellt, insbesondere die Untersuchungen von (Kleine 2007) und (Kinkel 2009) belegen aber noch einmal das große Potenzial, das eine Überwindung der technologischen Barrieren erschließt: beinahe eine Verdopplung des Robotereinsatzes wird von den Betrieben bei der Schaffung der entsprechenden Rahmenbedingungen schon heute als realistisch angesehen, etwa im Bereich der Einzel- und Kleinserienfertigung (105 %) bzw. der Mittelserienfertigung (85 %).

### 3.2.2 Anwendungsszenario „MAG-Schweißen in der Kleinserie“

Von den verschiedenen üblichen Schweißverfahren ist das Lichtbogenschmelzschweißen und insbesondere das MAG-Schweißen (Schutzgas, Metall-Aktiv-Gas) der industriell relevanteste Schweißprozess, siehe Bild 3.8. Der Prozess ist seit langem auf dem Industrieroboter etabliert, vielfach implementiert in mittleren bis großen Serien (Verl 2007). Es gibt diverse Technologiepakete der Roboterhersteller als Unterstützung bei der Programmierung.

Die Übertragbarkeit auf kleine Losgrößen ist aber nicht immer gegeben. Unternehmen, die an solchen Schweißsystemen für einen kleinen Losgrößenbereich einen Bedarf haben, sind etwa Metallbauer und Stahlbau- und Maschinenbau-Unternehmen mit 20 bis 250 Mitarbeitern. Hier fallen häufig so viele Schweißstunden an, dass das wirtschaftliche Potenzial einer Automatisierung gegeben ist, allerdings nur, wenn sie mit einer großen Variantenzahl und wenigen Exemplaren je Variante effektiv umgehen kann.

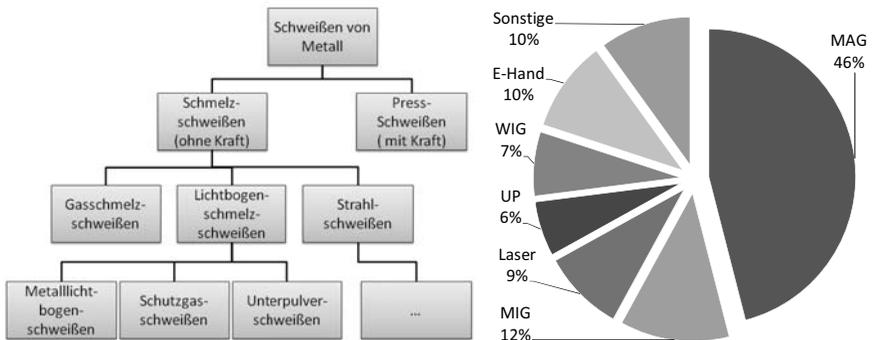


Bild 3.8: Übersicht über die Schweißverfahren (links, Fischer 1992), Verbreitung der Schweißverfahren (rechts, Schal 2002)

Produkte, bei denen der Wunsch nach einer Automatisierungslösung vorhanden ist, sind Stahlkonstruktionen, die in Serien von 10 Teilen aufwärts gefertigt werden. Einige Beispiele sind in Bild 3.9 gegeben. Untersuchungen zu der Fertigungssituation speziell bei kleinen und mittelständischen Unternehmen wurden im Projekt SMERobot durchgeführt (SMERobot 2008).

Die Werkstücke werden zugeschnitten und gepunktet, nach einer Oberflächenbearbeitung etwa durch Sandstrahlen werden sie vollständig geschweißt. Der Gallerieträger in Bild 3.9, oben links, etwa benötigt eine manuelle Schweißzeit von etwa 45 min. Der Transfer der Fertigung auf ein Robotersystem muss unter den folgenden Randbedingungen erfolgen:



Bild 3.9: Geeignete Produkte für das Anwendungsszenario „MAG-Schweißen in Kleinserie“: Gallerieträger für Großmotoren, Strukturelement, Bodenbearbeitungsgerät, Lagerböcke (von links oben nach rechts unten).

Randbedingung	Ausprägung
Vorrichtung	Im Gegensatz zur manuellen Fertigung muss eine Vorrichtung vorhanden sein, die eine reproduzierbare Positionierung des Werkstücks in allen zum Schweißen notwendigen Orientierungen (Wannenlage) erlaubt.
Bahngeometrie	Der Hauptanwendungsfall ist die Verbindung ebener Bleche. Weniger häufig sind Verbindungen gebogener Bleche mit ebenen Blechen (Kurven in einer Ebene) oder die Verbindung zweier gebogener Bleche (Raumkurven).
Nahtformen	Die betrachteten Werkstücke bestehen aus Blechen einer Dicke von 3 bis 20 mm. Sie werden mit einer Naht verbunden. Die Kanten sind mit entsprechenden Fasen vorbereitet.
Genauigkeit	Im eingespielten Prozess sollte die Wiederholgenauigkeit zwischen 0,5 und 0,1 mm liegen, je nach dem das aufgebrauchte Materialvolumen definierenden Nahtdickenparameter $\alpha$ . Anforderungen an die Absolutgenauigkeit des Roboters hängen vom gewählten Programmierverfahren ab, das heißt ob die Wiederhol- oder die Absolutgenauigkeit die Prozessgenauigkeit bestimmt.

Bedienbarkeit	Das Robotersystem muss einfach zu Bedienen und zu Programmieren sein, dies betrifft die Qualifikation der Bediener und die Zeitdauer der Programmierung.
Fehlerstrategien	Um wirtschaftlich zu sein, muss das Schweißrobotersystem möglichst lange ohne Interaktion arbeiten. Notwendige Interaktionen sollten sich einfach durchführen lassen. Dies muss durch intelligente Fehlerstrategien und entsprechende Mensch-Maschine-Schnittstellen umgesetzt werden.

Tabelle 3.1: Randbedingungen zum Transfer von Schweißprozessen auf eine robotergestützte Fertigung

### 3.2.3 Anwendungsszenario „Handlingsaufgaben in der Prototypenfertigung“

Ein dem Handlingsbereich entnommener Aufgabentyp ist die roboterunterstützte Fertigung von Prototypen. Sowohl im Automobilbereich, wo in der Vorserie kleine Stückzahlen von Fahrzeugen gefertigt werden müssen, als auch im CFK-Bereich (Carbonfaser-Komposites), wo beispielsweise die Wicklungen der Kohlefaser um einen Positiv reproduzierbar aufgebracht werden müssen (Bild 3.10, rechts), ist die Fertigung mit dem Roboter notwendig oder zumindest sinnvoll, die konventionelle Programmierung mit Offline-System oder Programmierhandgerät aber ist zu zeitaufwändig.

Während die Automobilindustrie die Einsatzpotenziale und -hindernisse des Robotereinsatzes kennt und die notwendige Infrastruktur verfügbar hat, steht in der CFK-Entwicklung und -fertigung bisher der manuelle Fertigungsprozess im Vordergrund. Dies liegt auch daran, dass an breit einsetzbaren automatisierten Prozessen für die Faserzusammenstellung, die Imprägnierung und das Aushärten aktuell noch geforscht wird (Drechsler 2009).



Bild 3.10: Kleinserien- und Prototypenfertigung mit Industrierobotern (links: Bolzenschweißen, rechts: Wicklung von CFK-Rohlingen)

Ein Beispiel ist die Fertigung von CFK-Bauteilen mit einer Radialflechtmaschine. Die Flechtmaschine umgibt einen durch sie hindurchgezogenen Positiv mit einem Kohlefaser-Geflecht,

dieses kann im Anschluss die weiteren Prozessstufen Imprägnierung und Aushärtung durchlaufen (Breckweg 07). Durch die beim Flechten aufgebrachte Spannung der Fäden erfolgt eine Selbstzentrierung des Werkstücks im Flechtzentrum, die Programmierung muss also nicht genauer als ca. 5 mm sein. Wichtiger ist eine weiche Definition der Kurven- und Geschwindigkeitsverläufe, um das gewünschte Fasermuster zu erreichen.

Randbedingung	Ausprägung
Werkstücke	Positive aus diversen Materialien, bis 2 m Länge, 50 kg Gewicht
Genauigkeit	< 5 mm. Durch die Vorspannung der Fasern wird das Positiv ins Zentrum der Flechtmaschine gezogen.
Bedienbarkeit	Anforderung im Forschungsbetrieb: Leicht bedienbar für häufige Programmierung mit wechselnden Personen
Anbindung	Eine Anbindung an die Prozesstechnik der Fertigungszelle ist notwendig.

Tabelle 3.2: Randbedingungen zur Programmierung von Prozessen in der Prototypenfertigung.

### 3.2.4 Anwendungsszenario „Positionierung von Messsystemen“

Für die Vermessung von Produkten oder Teilprodukten im Fertigungsprozess werden vielfältige Messsysteme eingesetzt. Neben der Geometrie der Produkte können Vibrationseigenschaften, Farbmerkmale oder relative Geometriedaten wie Spaltmaße erfasst werden. Die Daten werden unter anderem in der Fertigung zur Qualitätssicherung (Nikon 2010-2, KUKA 2009, Verl 2009) oder in der Entwicklung zur Analyse von Designstudien und Prototypen (Oberer-Treitz 2008-2) verwendet, siehe Bild 3.11.

Gerade bei größeren Produkten wie Automobilen ist eine vollständige Erfassung nur über mehrmalige Messungen realisierbar. Dies kann über das Umstellen und Einmessen des Sensors passieren oder über eine automatisierte Handhabung. Dies hat die Vorteile, dass die Position des Sensors im Raum bekannt ist und der Messablauf schneller und unbeaufsichtigt ablaufen kann.

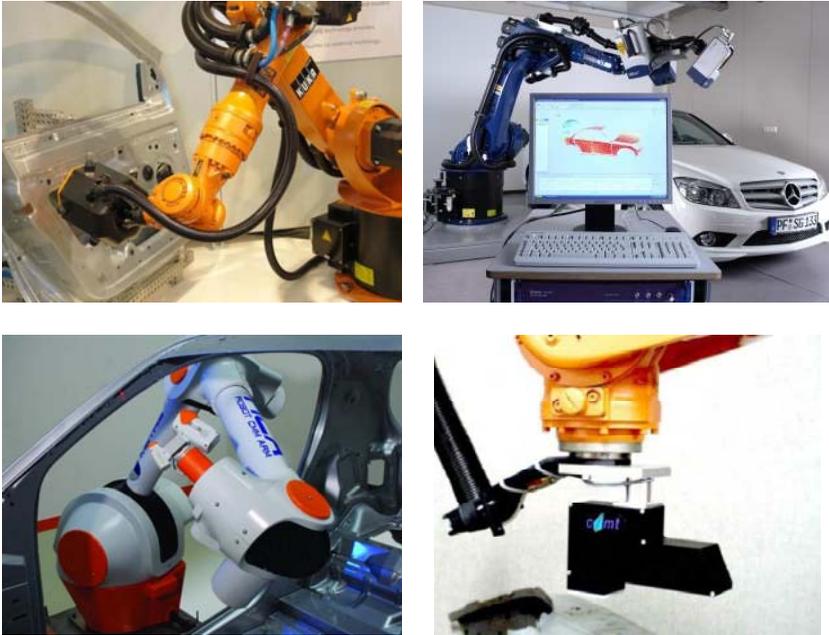


Bild 3.11: Von Industrierobotern geführte Messgeräte (oben links nach unten rechts): Nikon Metrology K-Rob (Nikon 2010-1), Polytec RoboVib (Oberer-Treitz 2008-2), Nikon Metrology Robot CMM Arm (Nikon 2010-2), KUKA X-Rob (KUKA 2009).

Auch hier stellt sich aber die Frage der Programmierung. Die Definition der Messpositionen und daraus abgeleitet der Roboterbahn aus dem CAD ist möglich, erfordert aber eine hohe Spezialisierung. Die Programmierung mit dem Handgerät ist zeitaufwändig. In vielen Fällen bietet das Verfahren des Programmierens durch Vormachen eine schnelle und einfach erlernbare Alternative, insbesondere wenn - wie im Fall der RoboVib-Vibrationsmessung - die Absolutposition des Sensors nicht besonders genau sein muss.

Die Anforderungen an die Programmierung sind in Tabelle 3.3 zusammengefasst.

Randbedingung	Ausprägung
Vorrichtung	Sensorsystem wird am Roboter befestigt. Messobjekt auf getakteter Fördereinrichtung oder in Einzelvorrichtung. Weiterleitung der Taktinformation.
Genauigkeit	Das am Roboter angebrachte Sensorsystem muss mit der Absolutgenauigkeit des Roboters arbeiten können: Bspw. Fehler $\mu < 0,7$ mm für absolutvermessene Roboter (Koeppel 2007). Wenn dies nicht ausreicht, kann ein zusätzliches Messsystem die TCP-Position des Roboters hochgenau ermitteln (Rawiel 2006).

Bedienbarkeit	Ein manuelles Verfahren und Programmieren des Roboters kann den Prozess beschleunigen, da häufig Einzelstücke gemessen werden.
Anbindung	Insbesondere für den Fall der Inline-Messung ist eine Anbindung an die Produktionsanlage und an das Qualitätssicherungssystem notwendig.

Tabelle 3.3: Randbedingungen zur Programmierung durch Vormachen von robotergeführten Messsystemen.

Für die weiteren Arbeiten wird der Anwendungsfall des MAG-Schweißens wie in 3.2.2 dargestellt angenommen.

### 3.3 Analyse der zur programmierenden Geometrieinformationen

Für die Aufnahme der Trajektorie sowie deren Nachbearbeitung ist es wesentlich, welche Geometrieprimitive zur Approximation des realen Werkstücks erzeugt werden. Generell können bei Schweißprozessen frei geformte Oberflächen auftreten, denen die Schweißbahn folgen muss. Diese im Raum definierte Bahn muss unter anderem drei Randbedingungen genügen: ihre Struktur muss technisch auf der Robotersteuerung umsetzbar sein, sie muss die Naht mit möglichst geringem Fehler abbilden und sie muss mit konstanten Prozessparametern abzufahren sein, etwa mit konstanter Geschwindigkeit.

Die für die weiteren Schritte relevanten abzubildenden Nahtformen ergeben sich einerseits durch die Klassifizierungen in der Literatur, andererseits durch analysierte Werkstücke. So klassifizieren (Schal 2002) und (Fischer 1992) nach Art (I-Naht, V-Naht, Überlappnaht, Kehlnaht, Ecknaht, ...) und Position (Überkopfschweißen, Steignaht, Fallnaht, Wannanlage, ...) der Schweißnähte. Ohne eine allgemeine Eingrenzung vorzunehmen, wiesen die in Maschinenbauunternehmen untersuchten Werkstücke, siehe Bild 3.9, im Wesentlichen Keh-, Eck- und V-Nähte auf. Die Schweißposition war, wenn möglich, immer die Wannanlage bei einer stechenden Orientierung des Brenners.

Diese Eingrenzungen gelten nicht für alle Anwendungsfälle, sondern spiegeln die Aufgaben in den untersuchten Betrieben wider. Größere Strukturen beispielsweise lassen sich nur mit hohem anlagentechnischem Aufwand so frei positionieren, dass immer in Wannanlage geschweißt werden kann.

Innerhalb dieser Eingrenzung lassen sich fünf wesentliche Nahtformen identifizieren, deren Eigenschaften bei der Konzeption des Programmiersystems berücksichtigt werden müssen, siehe Tabelle 3.4. Insbesondere in Bezug auf die Orientierung müssen die Eckpunkte in Betracht gezogen werden, da dort eine schnelle Umorientierung aus der schleppenden in die stechende Lage vollzogen werden muss.

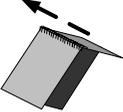
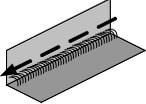
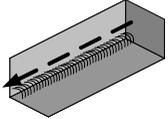
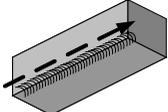
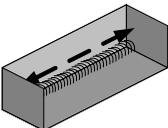
	<p>Ecknaht, Lage PA nach (DIN EN ISO 6947)</p>	<p>Problemlose Nahtform</p>
	<p>Kehlnaht, Lage PA nach (DIN EN ISO 6947) Anfang und Ende frei.</p>	<p>Problemlose Nahtform sofern kollisionsfreie Erreichbarkeit gegeben</p>
	<p>Kehlnaht, Lage PA nach (DIN EN ISO 6947) Nahtbeginn an Abschlussblech, Ende frei.</p>	<p>Orientierungsänderung von schleppend nach stechend notwendig</p>
	<p>Kehlnaht, Lage PA nach (DIN EN ISO 6947) Anfang frei, Nahtende an Abschlussblech.</p>	<p>Keine Orientierungsänderung notwendig, problemlose Naht</p>
	<p>Kehlnaht, Lage PA nach (DIN EN ISO 6947) Nahtbeginn und -ende an Abschlussblech.</p>	<p>Orientierungsänderung zu Nahtbeginn von schleppend zu stechend notwendig.</p>

Tabelle 3.4: Sektionen des Werkstücks mit jeweils eigenen Anforderungen: Ecknaht, offene Kehlnaht, am Nahtanfang und am Nahtende geschlossene Kehlnähte, beidseitig geschlossene Kehlnaht (von oben nach unten)

Besondere Beachtung bei der Programmierung der Naht müssen die folgenden Nahtformen finden:

- Anfangs geschlossene Kehlnähte, dort muss sehr schnell eine Umorientierung aus der schleppenden in die stechende Orientierung erfolgen
- Verdeckte Positionen oder solche mit schwerer Zugänglichkeit, etwa hinter Bauteilen oder der Vorrichtung
- Positionen mit schwerer Zugänglichkeit für das Robotersystem, etwa durch eine vom Roboter abgewandte Orientierung

In der Tabelle 3.4 sind die Nahtformen für ebene Bauteile dargestellt, die Anfangs- und Endbedingungen der Nahtformen gelten ebenso für ebene, auf der Ebene geschwungene und im Raum geschwungene Nähte, siehe Tabelle 3.5.

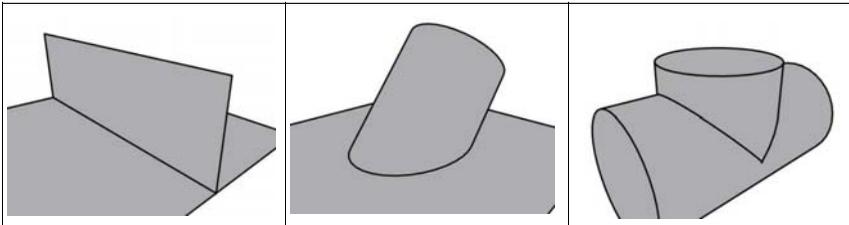


Tabelle 3.5: Drei Klassen von Raumkurven: Nicht gekrümmt, auf einer Ebene gekrümmt, im Raum gekrümmt (von links nach rechts).

Auf der Basis dieser nachzubildenden Nahtformen werden Geometrieprimitive auf ihre Eignung zur Abbildung der Formen untersucht. Neben Geraden und Kreisabschnitten sind hier insbesondere verrundete Geraden und Splines von Interesse. Sie werden untersucht im Hinblick auf ihre Abbildungsfähigkeit (wie einfach und genau kann die Naht abgebildet werden?), auf ihre Prozesseignung (können Bahn und Geschwindigkeit konstant gehalten werden?) und auf ihre Umsetzbarkeit auf den Robotersteuerungen (Welche Primitive lassen sich direkt auf der Steuerung abbilden?).

Nahttyp / Bahntyp	Nicht gekrümmt	Auf Ebene gekrümmt	Im Raum gekrümmt
Gerade	Gut geeignet	Nicht geeignet. Approximation über Teilstrecken notwendig.	Nicht geeignet. Approximation über Teilstrecken notwendig.
Verrundete Gerade	Gut geeignet	Wenig geeignet. Approximation notwendig.	Wenig geeignet. Approximation notwendig.
Kreisabschnitt	Nicht geeignet.	Wenig geeignet. Meist Approximation notwendig.	d die Schweißbarkeit der Bo- ximation notwendig.
Spline	Wenig geeignet. Besondere Behandlung notwendig.	Gut geeignet.	Gut geeignet.

Tabelle 3.6: Eignung der Geometrieprimitive Gerade, Kreisbogen, Spline und verrundete Gerade für aufgabenspezifische Werkstückgeometrien

Bezüglich der Abbildungsfähigkeit sind die Geometrieprimitive „Gerade“ oder „Verrundete Gerade“ und „Spline“ geeignet. Die gekrümmten Nähte werden dabei durch Splines approximiert. So erhält man einerseits eine gute Abbildungstreue, andererseits eine übersichtliche Zahl an Primitiven.

Bei kombinierten Teilstücken ist die Stetigkeit und Differenzierbarkeit der Gesamtbahn von Bedeutung. Nach dem Epsilon-Delta-Kriterium (Heuser 1990) gilt:

$$\begin{aligned} f: D \rightarrow \mathbb{R} \text{ ist stetig in } x_0 \in D, \text{ wenn zu jedem } \varepsilon > 0 \text{ ein} \\ \delta > 0 \text{ existiert, so dass für alle } x \in D \text{ mit } |x - x_0| < \delta \text{ gilt:} \end{aligned} \quad (3.1)$$
$$|f(x) - f(x_0)| < \varepsilon.$$

Die Einzelfunktionen der Teilstücke sind, übertragen auf den  $\mathbb{R}^3$ , stetig. Zusammengesetzte Funktionen sind ebenfalls stetig, sofern jeweils Anfangs- und Endpunkte identisch sind. Fraglich ist im Weiteren die Differenzierbarkeit dieser zusammengesetzten Funktionen, nachzuweisen etwa über das Grenzwertkriterium (Walter 1997):

$f: D \rightarrow \mathbb{R}$  ist genau dann differenzierbar an einer Stelle  $x_0 \in D$ , wenn der Grenzwert

$$\lim_{x \rightarrow x_0} \left( \frac{f(x) - f(x_0)}{x - x_0} \right) \quad (3.2)$$

existiert.

Je nach Kombination der Geometrieprimitive kann ein differenzierbarer oder nicht differenzierbarer Übergang realisiert werden. Die Relevanz in Bezug auf die Aufgabenstellung liegt darin, dass sich mit differenzierbaren Funktionen eine glatte Bewegung erzeugen lässt. Stetige, aber nicht differenzierbare Kombinationen von Geometrieprimitiven führen zu einem nicht konstanten Geschwindigkeitsverlauf in der Abarbeitung der Bahn oder zu Positionsfehlern durch ein Überschleifen der Kantenpunkte.

Eine Kombination aus Geraden und Kreisbögen lässt sich in der Regel nicht differenzierbar erzeugen, es sei denn die Geraden bilden immer Tangenten der sich anschließenden Kreisbögen. Splines und Geraden lassen sich  $C^1$ -stetig, also differenzierbar, zusammenfügen. Durch die Verrundung der Anfangs- und Endpunkte in der Robotersteuerung lässt sich ein der Differenzierbarkeit ähnliches Verhalten erzeugen, die Verrundung der Robotersteuerung unterliegt allerdings Randbedingungen, die nicht immer extern zugänglich sind.

Ein weiterer für die Bewertung notwendiger Aspekt ist die Handhabbarkeit der Geometrieprimitiven, insbesondere der Splines. Hier gibt es verschiedene Klassen mit spezifischen Vor- und Nachteilen in Bezug auf die Aufgabenstellung. Tabelle 3.7 gibt einen exemplarischen Überblick, siehe auch (Meißner 2000).

Splinetyp	Formel	Zeichnung	Einschätzung
Kubisch	$Y_i(t) = a_i + b_i t + c_i t^2 + d_i t^3$ mit $Y_i(0) = y_i = a_i$ und $Y_i(1) = y_{i+1}$ $= a_i + b_i + c_i + d_i$		Verbindet direkt die Stützpunkte. Randbedingungen definierbar.
Bezier	$B(t) = \sum_{i=0}^n \binom{n}{i} (1-t)^{n-i} t^i P_i$ Für die Punkte $P_0, \dots, P_n$ .		Stützpunkte liegen nicht auf der Kurve.
NURBS	$C(u) = \sum_{i=0}^n R_{i,p}(u) P_i$ mit den B-Spline Basisfunktionen $R_{i,p}$		Stützpunkte liegen nicht auf der Kurve. Mathematische Basis ist geschlossen.

Tabelle 3.7: Vorstellung und Eignung verschiedener Spline-Varianten in Bezug auf die Aufgabenstellung.

Für die Abbildung der relevanten Nahtformen eignen sich Kombinationen aus Geraden („Gerade“ oder „Verrundete Gerade“) und Splines („Kubischer Spline“, „Bezier-Spline“ oder „NURBS“).

### 3.4 Analyse der verfahrenstechnischen Anforderungen des Schweißprozesses

Voraussetzungen der Automatisierbarkeit eines Schweißprozesses sind die Schweißbarkeit der Bauteile und die Wirtschaftlichkeit gegenüber dem manuellen Schweißprozess. Die Schweißbarkeit, nach (Dilthey 1995) bestimmt durch die Schweißeignung des Werkstoffs, die Schweißsicherheit der Konstruktion und die Schweißmöglichkeiten der Fertigung, unterscheidet sich vom manuellen Prozess nur im Bereich der Fertigung.

Um die Qualität der Werkstücke zu gewährleisten, sind Fehler und Schäden an den Schweißverbindungen zu vermeiden. Hier sind die Fehler durch unsachgemäße Fertigung zu beachten, zu unterscheiden in „innere“ und „äußere“ Nahtfehler (Dilthey 1995). Äußere Nahtfehler sind Schweißspritzer und Zündstellen, Einbrand- und Randkerben und andere Nahtformfehler etwa geometrischer Natur. Innere Nahtfehler sind Bindefehler und unverschweißte Stellen, Schlacken- und andere Feststoffeinschlüsse sowie mechanische Porenbildung. Zwei Fehler sind in Bild 3.12 dargestellt.

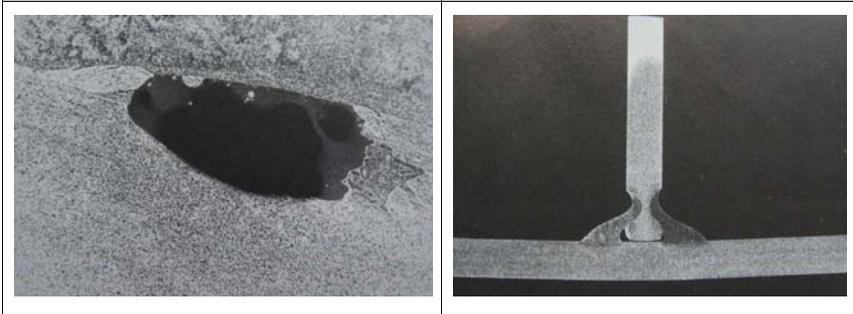


Bild 3.12: Beispiele für innere und äußere Nahtfehler aus (Dilthey 1995): Schlackeeinschluss (links), Einbrandkerben an T-Stoß (rechts).

Relevanz in Bezug auf die roboterbasierte Fertigung und speziell die Roboterprogrammierung haben die in Tabelle 3.8 aufgeführten fertigungsverursachten Nahtfehler.

Fehler	Relevanz	Vermeidungsmöglichkeiten	Klasse
Zündstellen außerhalb der Naht	Gefahr von Gefügeänderungen, Materialverhärtungen	Vermeidung durch Programmierung und Wiederholbarkeit gegeben	V
Einbrand- und Randkerben	Gefahr des Auftretens von Rissen	Nahtvorbereitung, Programmierung	./.
Nahtformfehler	Je nach Bewertungsgruppe verschiedene Grenzwerte, Gefahr der Bauteilschwächung	Vermeidung durch Programmierung und Wiederholbarkeit teils gegeben, teils Sensoreinsatz notwendig.	V / F
Bindefehler	Kalte Stellen, schwer auffindbar, verringern die Festigkeit. Der Grund ist häufig das Überschweißen von Schlackeresten oder ein vorlaufendes Schmelzbad.	Nahtvorbereitung, sorgfältige Parameterwahl und Programmierung. Robotereinsatz kann hier durch eine breitere Parameterwahl Vorteile bringen.	V / F
Schlacken / Festkörpereinschlüsse	Einschluss von nichtmetallischen Feststoffen, dadurch Verringerung der Festigkeit	Nahtvorbereitung, Parameterwahl. Vorteil durch Reproduzierbarkeit.	V
Mechanische Porenbildung	Einschluss expandierender Gase in der Schmelze, dadurch Schwächung der Schweißnaht.	Insbesondere Nahtvorbereitung, auch Parameterwahl und Wahl der Orientierung	./.

Tabelle 3.8: Relevanz von Nahtfehlern und deren Vermeidungsmöglichkeiten bei der Programmierung von Schweißrobotern. Klassifikation als Vorteil (V), Forderung (F) und indifferent (./.)

Eine automatisierte Schweißung hat den wesentlichen Vorteil der Reproduzierbarkeit in Bezug auf die Brennerbewegung, aber auch auf die Wiederholbarkeit und Bandbreite der Prozessparameter. Hierdurch lassen sich viele Fehler beim einmal eingestellten Prozess vermeiden. Zu untersuchen ist allerdings, ob der Einsatz von Sensoren zur Nahtverfolgung im Prozess notwendig ist. Von großem Vorteil ist die größere Bandbreite der Prozessparameter, etwa der höheren Geschwindigkeit oder der höheren einstellbaren Ströme gegenüber der manuellen Schweißung. Dies erlaubt zum Beispiel die Vermeidung von Bindefehlern etwa über Schlackeresen der Heftpunkte oder der Wurzelnaht. Arbeiten wie das Ausschleifen der Heftpunkte können entfallen.

Für die Fehlervermeidung ist es notwendig, alle relevanten Parameter und Fehlervermeidungsstrategien auch programmieren zu können, die Programmierumgebung muss dies erlauben. Die einzelnen Anforderungen sind in Abschnitt 3.7 aufgeführt.

### 3.5 Analyse der konventionellen Programmierung von Schweißrobotern

Die robotergestützte Fertigung von Bauteilen - wie sie in Abschnitt 3.1.2 dargestellt wurde - erfolgt bei KMU meist mit Hilfe des Programmierhandgerätes (PHG). In diesem Fall unterteilt sich der Programmiervorgang in die drei Hauptabschnitte Vorbereitung, Programmierung und Test / Optimierung, siehe Tabelle 3.9.

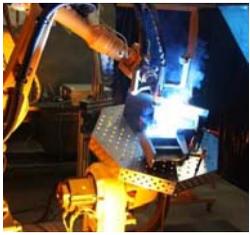
		
<b>Vorbereitung</b> Planung und Aufbau der Vorrichtung, Planung der Bearbeitungsbahnen	<b>Programmierung</b> Verfahren von Roboter und Dreh-Kipp-Tisch, Aufnahme der Prozess- und Verfahrenpunkte in das Roboterprogramm	<b>Test und Optimierung</b> Kalter und heißer Test mit Optimierung

Tabelle 3.9: Ablauf der PHG-basierten Programmierung von Schweißrobotern

Bei der Vorbereitung wird die Bearbeitung durch den Roboter geplant und die Vorrichtung entsprechend installiert, häufig mit Hilfe von Baukastensystemen. Hier ist einerseits Prozess-erfahrung notwendig, um Effekte von Hefttoleranzen, Temperatureintrag und Oberflächen-variationen einschätzen zu können, andererseits ist es wichtig, die Erreichbarkeit der einzel-

nen Bahnen abschätzen zu können. Hierzu ist Routine in der Programmierung der speziellen Roboteranlage notwendig.

In der Programmierphase werden die zu programmierenden Punkte mit dem PHG per Verfahrtasten, mit dem Joystick oder mit der 6-D-Maus angefahren und abgespeichert. Sie werden in der Programmlogik zu Schweißnähten zusammengefasst, die entsprechenden Parameter werden beim Programmieren mittels der PHG-Tasten gesetzt. Der Programmierer ist für die kollisionsfreie Erreichbarkeit der von ihm programmierten Punkte und Bahnen verantwortlich. Wie in Tabelle 3.10 dargestellt wechseln sich bei der Programmierung zwei Arbeitspunkte zyklisch ab, die Bewegung des Roboters zur Überbrückung von Distanzen und zur Feinpositionierung unterstützt durch haptische Kontrolle und die Erfassung der Punkte in der Programmstruktur über das PHG.

In der Test- und Optimierungsphase wird ein kalter Programmdurchlauf vorgenommen, bei dem Fehler erkannt und entfernt werden und die Präzision der Programmierung überprüft wird. Im Anschluss wird ein so genannter „heißer“ Testlauf mit aktivem Schweißprozess durchgeführt. Je nach Ergebnis müssen Positionen, Orientierungen oder Parameter angepasst werden, um die gewünschte Schweißqualität zu erreichen.

	
<p><b>Bewegung des Roboters</b> Verfahren des Roboters und des Dreh-Kipp-Tisches in die korrektes Position. Verfahren mittels Tasten, Joystick oder 6-D-Maus des PHG.</p>	<p><b>Aufnahme des Punktes in das Roboterprogramm</b> Aufnehmen der neuen Zeile, Touch-Up des aktuellen Punktes, Konfiguration von Parametern</p>

Tabelle 3.10: Detailablauf der Programmierung

Die Programmierung des Roboters mit dem Programmierhandgerät bietet viele Möglichkeiten und ist seit langem Stand der Technik. Ein wesentlicher Vorteil ist, dass die Benutzerschnittstelle alle Funktionen des Roboters zur Verfügung stellt.

Insbesondere aus dem Blick der Fertigungssituation kleiner und mittlerer Unternehmen aber lassen sich auch Defizite identifizieren. Diese sind in der Tabelle 3.11 zusammengefasst.

Wirtschaftlich	Technisch
<ul style="list-style-type: none"><li>• Lange Einlernzeit notwendig</li><li>• Erfahrung in der Programmierung notwendig</li><li>• Programmierung zeitintensiv, dadurch Wirtschaftlichkeit bei kleinen Losgrößen schwierig</li></ul>	<ul style="list-style-type: none"><li>• Bewegung des Roboters über Tasten des PHG ist umständlich</li><li>• Für das Verfahren mit der 6-D-Maus oder Joystick ist viel Übung notwendig</li><li>• Komplexe Fehlerbehebung</li></ul>

Tabelle 3.11: Defizite der PHG-orientierten Programmierung von Schweißrobotern

### 3.6 Folgerungen aus den Analyseergebnissen und Ableitung von Untersuchungsschwerpunkten

Aus der Analyse der roboterbasierten Fertigung bei kleinen und mittleren Unternehmen ergeben sich die Folgerungen:

- Industrieroboter werden zum überwiegenden Teil in Großbetrieben eingesetzt, insbesondere in der Automobilindustrie und in chemischen Betrieben.
- Industrieroboter, die in kleinen und mittleren Unternehmen eingesetzt werden können, besitzen auf Grund der hohen Anzahl möglicher Nutzer ein enormes Potenzial.
- Bisher gibt es keine breit einsetzbare Automatisierungslösung für den Einsatz von Industrierobotern in der Fertigung von KMU.
- Schweißroboter stellen mit 26 Prozent die nach dem Handling wichtigste Gattung von Industrierobotern dar.
- Die Anforderungen an die Programmierung sind sehr unterschiedlich, die notwendige Positionstreuung etwa schwankt zwischen 0,5 und 0,1 mm bzgl. der Wiederholgenauigkeit.
- Die Programmierung von Schweißrobotersystemen ist zeitaufwändig, 4-10 Stunden Programmierzeit werden etwa für Bauteile wie in Bild 3.9 kalkuliert.
- Für eine zügige Programmierung über das PHG muss der Mitarbeiter gut ausgebildet und in der Roboterprogrammierung erfahren sein.

Aus der Analyse und den oben dargestellten Folgerungen ergeben sich drei wesentliche Untersuchungsschwerpunkte:

- Fokussierung auf die Programmierung von MAG-Schweißaufgaben speziell für die Fertigung von KMU
- Verringerung der Komplexität und des Zeitaufwandes der Programmierung
- Entwicklung von Methoden zur Erreichung ausreichend hoher Genauigkeiten

## 3.7 Anforderungen an ein intuitiv programmierbares Robotersystem

### 3.7.1 Auswahl von Referenzwerkstücken

Basierend auf der vorangegangenen Analyse werden zwei Referenzwerkstücke unterschiedlicher Komplexität definiert, zum einen als Leitlinie bei der Entwicklung, zum anderen zur Verifikation der entwickelten Funktionalitäten. Bei den Referenzwerkstücken handelt es sich um Schweißbaugruppen, die für die Fertigung in KMU-Umgebungen typisch sind, die Verbindung eines I-Trägers auf einer Grundplatte (manuelle Prozesszeit ca. 5 Minuten), und die Fertigung eines Galerieträgers mit ca. 45 Minuten manueller Prozesszeit.

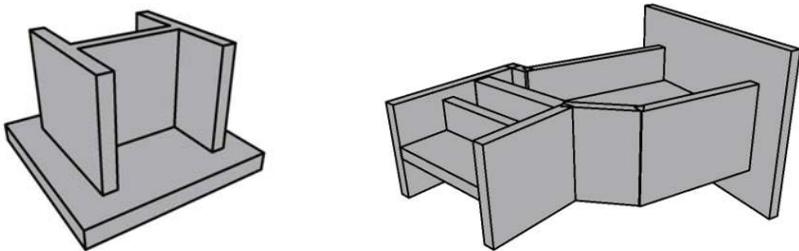


Bild 3.13: Zeichnungen der Referenzwerkstücke I-Träger (links) und Galerieträger (rechts)

Für die Fertigung des I-Trägers müssen 8 Schweißnähte programmiert werden, das gesamte Roboterprogramm inklusive Umorientierungen besteht aus etwa 40 Punkten. Für den Galerieträger müssen 73 Schweißnähte programmiert werden, die spätere Programmgröße liegt bei etwa 600 Punkten. Der Zeitaufwand für die konventionelle Programmierung mit dem PHG liegt bei etwa 30 Minuten bzw 7 Stunden. Nähere Details sind in Abschnitt 7.5.5 dargestellt.

### 3.7.2 Definition von Teilfunktionen und Teilsystemen

Aus den Ergebnissen des Standes der Technik und der Analyse ergeben sich die in Bild 3.14 dargestellten Teilfunktionen und Teilsysteme einer Programmierumgebung: Handhabungssystem, Sicherheitssystem, Prozesstechnik, Nachbearbeitung, Programmierereinrichtung und Sensorik.

Entsprechend den Zielen in Abschnitt 1.2 und der Ableitung von Untersuchungsschwerpunkten in Abschnitt 2.4 wird dabei ein Schweißroboter als Zielsystem angenommen. Mit einem entsprechenden Wechsel der Prozesstechnik können diese Unterteilung und die weiteren Überlegungen auf andere Szenarien übertragbar sein.

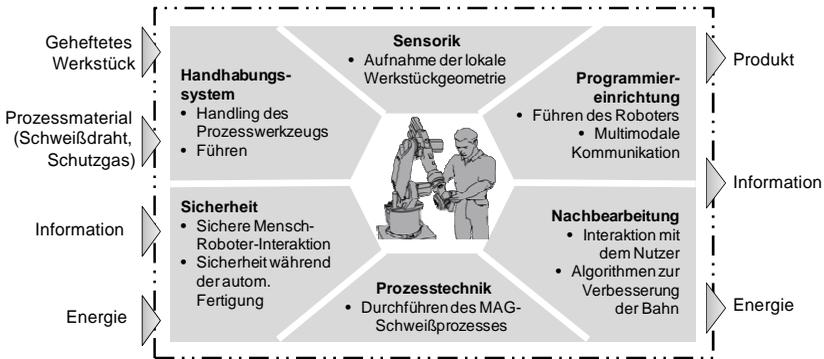


Bild 3.14: Definition von Teilfunktionen und -systemen

Die hier dargestellte Untergliederung in Teilsysteme ergibt sich aus der Analyse des Prozesses und der konventionellen Roboterprogrammierung, ebenso aus dem Stand der Technik. Die Sichtweise spiegelt zuerst den physischen Arbeitsablauf und die dafür notwendigen Komponenten wider.

Das Handhabungssystem besteht aus dem Industrieroboter und, sofern notwendig, weiteren Komponenten wie einem Dreh-Kipp-Tisch. Es trägt und bewegt entweder das Prozesswerkzeug oder das Werkstück. Die Sensorik übernimmt die Optimierung der programmierten Bahn, bis die gewünschten Toleranzen erreicht sind. Die Programmier-einrichtung dient der Führung des Roboters und der multimodalen Kommunikation mit dem Nutzer. Sie nimmt die Bahnen auf. Das Teilsystem der Nachbearbeitung enthält Software und Hardware, die ermöglichen, die aufgenommene Bahn zu korrigieren und mit weiteren notwendigen Informationen zu versehen. Das Teilsystem der Prozesstechnik enthält die spezifischen Komponenten des jeweiligen Prozesses, etwa Stromquelle und Drahtförderer beim MAG-Schweißen. Das Teilsystem der Sicherheit ist verantwortlich für die Personensicherheit des Nutzers, aber auch weiterer Personen.

Eine weitere Sichtweise auf die Programmierumgebung stellt die funktionale Sicht dar. Hier ergeben sich weniger Fragestellungen zu der physischen Realisierung des Gesamtsystems als vielmehr zu der Interaktion der Teilabläufe in Verbindung mit dem Nutzer. So ergeben sich die Anforderungen an die Mensch-Maschine Schnittstelle aus den Teilfunktionen mehrerer Teilsysteme, dem Handhabungssystem, der Programmierung und der Nachbearbeitung.

In den folgenden Abschnitten sind die Anforderungen an die Teilsysteme strukturiert erfasst. Diese Anforderungen bilden zusammen mit der in 3.7.1 erfolgten Auswahl von Referenzwerkstücken die Grundlage für die Bewertung alternativer Konzepte im folgenden Kapitel.

### **3.7.3 Ableitung von Anforderungen an das Handhabungssystem**

Das Handhabungssystem muss in der Lage sein, den Fertigungsbetrieb durchführen zu können, in dem hier relevanten Szenario also die Durchführung der Schweißungen. Die dazu notwendigen Traglasten und Genauigkeiten werden von allen typischen Schweißrobotern erfüllt. Ebenso muss der Roboter den Umgebungsbedingungen standhalten.

Zusätzlich müssen die für die Programmierung durch Vormachen notwendigen Eigenschaften gegeben sein, insbesondere in Bezug auf die Steuerung. Die Steuerung muss ein interaktives Handverfahren erlauben und eine Konnektivität zu der übergeordneten Steuerung bieten, etwa ein Ethernet- oder Bussystem zur Übertragung von digitalen Ein- und Ausgängen. Weiterhin müssen erzeugte Programme ohne ein komplexes manuelles Vorgehen auf die Steuerung geladen werden können, hierzu müssen die notwendigen Schnittstellen vorhanden sein.

Als wichtiger Punkt für die Umsetzung der Programme ist das Abfahren von Geraden und Kreisbahnen als Minimalanforderung notwendig.

### **3.7.4 Ableitung von Anforderungen an die Mensch-Maschine-Schnittstelle**

In diesem Punkt sind die Anforderungen an die Teilsysteme Programmierereinrichtung und Nachbearbeitung zusammengefasst.

Je nach späterer Konzeption muss das Programmiersystem verschiedenste Arten von Kommandos verstehen. Mit diesen muss der Nutzer in der Lage sein, eine Bahn zu programmieren, sie nachzubearbeiten und durch Sensorinformationen zu verbessern. Die sich daraus ergebenden Anforderungen an die Mensch-Maschine-Schnittstelle sind insbesondere die Verwendung intuitiver und für den Prozessexperten klarer Schnittstellen, die Verwendung von Schnittstellen, die die Aufgabenausführung nicht behindern, und die generelle Multimodalität der Schnittstellen.

Zusätzlich aber müssen die Schnittstellen den Anforderungen der Einsatzumgebung und des Prozesses gerecht werden, insbesondere in Bezug auf Lärm und Verschmutzung. Sprachsysteme etwa sind so auszulegen, dass Erkennungsfehler minimiert und Bewegungen auslösende Fehler unmöglich sind.

### **3.7.5 Ableitung von Anforderungen an die Prozesseinheiten und die Sensorik**

Prozesseinheit und Sensorik müssen den Umgebungsbedingungen standhalten. Dies ist für die eigentliche Prozesseinheit, also die Schweißquelle, eine unerlässliche Anforderung. Für die Sensorik sind die Fähigkeit zur Ableitung von Wärme und der Schutz vor Staub und Schweißspritzern relevant. Weiterhin muss die Sensorik in der Lage sein, die lokalen Geometriedaten des Werkstücks mit der notwendigen Genauigkeit aufzunehmen.

### **3.7.6 Ableitung von Anforderungen an die Sicherheitseinrichtungen**

Die Mensch-Roboter-Interaktion des Robotersystems muss den gültigen Sicherheitsrichtlinien, also insbesondere der Maschinenrichtlinie (MRL 2006) und den relevanten harmonisierten europäischen Normen entsprechen. Die wesentliche Norm ist hier die (DIN EN ISO 10218-1) für die Sicherheit von Industrierobotern, weiterhin relevant sind die Normen bezüglich Sicherheitsausrüstung und Sicherheitsabständen (DIN EN ISO 13849, DIN EN 999).

Das Robotersystem wird im Allgemeinen in zwei Zuständen betrieben, beide müssen sicher umgesetzt werden:

- Programmierung des Robotersystems in direkter Mensch-Roboter-Kooperation durch einen eingewiesenen Werker, aber nicht notwendigerweise durch einen erfahrenen Roboterprogrammierer.
- Fertigungsbetrieb im Automatikmodus bei aktivem Prozess. Personen sind vor Gefahren durch die Bewegungen des Roboters, aber auch vor Gefahren durch UV- und Hitzeabstrahlung des Prozesses zu schützen. In dieser Phase erfolgt keine Mensch-Roboter-Kooperation.

Um eine hohe Verfügbarkeit zu gewährleisten und die Gefahr von Umgehungen zu minimieren, muss bei der Auslegung des Sicherheitssystems auf die Umgebungsbedingungen geachtet werden. Sensoren dürfen nicht durch sich absetzenden Schleifstaub beeinträchtigt werden, ebenso darf sich die UV-Strahlung nicht auf ihre Funktionalität auswirken.

### **3.7.7 Ableitung von Anforderungen an den Verfahrensablauf und das Gesamtsystem**

Zusätzlich zu den Anforderungen an die Teilsysteme müssen Rahmenbedingungen für das Gesamtsystem definiert werden.

Wesentliche Anforderungen an das Gesamtsystem beziehen sich auf das Kosten-Nutzen-Verhältnis. Die zusätzlich notwendigen Investitionen für das Programmiersystem dürfen nicht zu hoch sein, ebenfalls muss sich der Einsatz in einer Vereinfachung der Programmierung, also auch einer Verkürzung der Programmierzeit von Schweißbauteilen zeigen.

Alle Komponenten müssen mindestens nach IP65 ausgelegt sein. Wartungsfreundlichkeit und Verfügbarkeit von Ersatzteilen muss gegeben sein.

Die Anforderungen an die Teilsysteme sind in Tabelle 3.12 zusammengefasst.

<b>MMI</b>	MMI-1	Haptische Bahnprogrammierung (Führen)
	MMI-2	Rein grafische Bahnnachbearbeitung
	MMI-3	Erzeugung nativen Robotercode
	MMI-4	Intuitives Programmierinterface
<b>Sensorik</b>	SEN-1	Sensorauflösung < 0,1 mm
	SEN-2	Eignung für Schweißumgebungen: IP65, Kühlung, Scheibenaustausch
	SEN-3	Verwertung von mehreren Scans: $0 \leq \#Scans \leq 10$
<b>Sicherheit</b>	SI-1	Programmierung entsprechend (DIN EN ISO 10218-1), Abschnitt 5.10.3
	SI-2	Fertigung entsprechend (DIN EN ISO 10218-1)
	SI-3	Verwendung markterhältlicher Komponenten
	SI-4	Schutz vor Prozessgefährdungen (Schweißspritzer, UV-Strahlung)
<b>Handhabungssystem und Steuerung</b>	ST-1	Eignung für Schweißprozess
	ST-2	Verwendbarkeit auf verschiedenen Robotersteuerungen
	ST-3	Frequenz Handführung > 50 Hz
	ST-4	Automatisierbarkeit von Programmupload und Programmstart
	ST-5	Mögliche Programmgröße > 100 Geometriefeatures, > 500 Wegpunkte
<b>Gesamtsystem</b>	GS-1	Kosten des Programmiersystems < 15 T€
	GS-2	Einsparung der Programmierzeit 30 % bis 50 %
	GS-3	Auslegung entsprechend Umgebungsbedingungen mind. IP65 im Prozessbereich
	GS-4	Toleranzoptimierung der Bahn bis 0,3 mm
	GS-5	Wartungsfreundlichkeit
	GS-6	Verwendung von Standard-Industriekomponenten
	GS-7	Programmierung ohne vorhandene CAD-Daten

Tabelle 3.12: Anforderungen an die Programmierumgebung

# 4 Konzeption des intuitiven Programmierverfahrens und des Gesamtsystems zum MAG-Schweißen

## 4.1 Konzeption des Verfahrensablaufes

In diesem Abschnitt werden auf der Basis der in der Analyse ermittelten Anforderungen möglichst optimale Verfahrensabläufe zur Bahnprogrammierung und Nachbearbeitung konzipiert. Betrachtet wird dabei das Zusammenspiel der geometrischen Grundelemente mit der Art der Nutzerinteraktion und der Datenverarbeitung.

### 4.1.1 Konzeption der geometrischen Grundelemente

In Unterkapitel 3.7.7 werden die verfahrenstechnischen Anforderungen an die geometrischen Grundelemente genannt, insbesondere die leichte Handhabbarkeit durch den Nutzer. In Unterkapitel 3.7.3 wird von der Steuerung eine Abarbeitungsmöglichkeit mindestens für Geraden und Kreisbögen verlangt. Tabelle 4.1 vergleicht die Alternativen für Grundgeometrien GG-A und GG-B.

	Lösung GG-A	Lösung GG-B
Schaubild	<p>1. Aufgenommene Punktwolke</p> <p>2. Bahnabstraktion auf Geraden und roboterspezifische Splines</p>	<p>1. Aufgenommene Punktwolke</p> <p>2. Bahnabstraktion auf Geraden und Splines</p> <p>3. Linienzug</p>
Verwendete Geometrielemente	Geraden und herstellerspezifische Splines	Geraden und anwendungsoptimale Splines
Transfer auf Robotersteuerung	Übersetzen der Geraden und Splines in Robotercode	Erzeugung eines Linienzuges, Übersetzen in Robotercode
Vor- und Nachteile	+ Anforderungen erfüllt + Durchgängige Repräsentation - Herstellerspezifisch	+ Anforderungen erfüllt + keine Herstellerbindung - weniger elegante Lösung

Tabelle 4.1: Vergleich der Möglichkeiten der Kombination von Geometrielementen

Aus der Anforderung der leichten Handhabbarkeit ergibt sich die Nutzung von Geraden als Basis der Programmierung. Diese stellen die überwiegend vorkommenden Geometrieelemente dar. Eine Approximation durch Splines wäre möglich, für den ‚Bediener aber schwer verständlich.

Die Kombination von Geraden und Splines oder Splines und Splines muss unter der Möglichkeit der Wahl der Stetigkeit vorgenommen werden; verlangt der Nutzer eine scharfe Kante oder einen differenzierbaren Übergang? Hierfür müssen entsprechende Wahlverfahren, etwa über die Nutzeroberfläche, vorgesehen werden.

Bei der Verwendung von Splines ist deren Typ primär durch die Implementierung in der Robotersteuerung vorgegeben. Typischerweise unterscheiden sich die Spline-Implementierungen der verschiedenen Hersteller. Der wesentliche Nachteil der Lösung GG-A ist also die starke Herstellerabhängigkeit.

Alternativ kann aus dem Spline, der eine einfache Nutzerinteraktion ermöglicht, eine Bahn aus Geradenstücken approximiert werden, wie in Lösung GG-B skizziert.

Das wesentliche Argument für die Wahl der Lösung GG-B besteht in der Herstellerunabhängigkeit. Die entwickelten Programme können auf jedes Robotersystem portiert werden. Die Nachteile sind überschaubar: ein Roboterprogramm auf Basis von Linienzügen muss bezüglich der Abbildungstoleranzen untersucht werden, ebenfalls müssen der Interpolationstakt und die maximal gültige Punktezahl ausreichen.

#### 4.1.2 Konzeption von Art und Umfang der Nutzerinteraktion

Die Nutzerinteraktion mit dem zu entwickelnden Programmiersystem ist der wesentliche Faktor, der es von einem Programmierhandgerät oder einem Offline-Programmiersystem unterscheidet. Die in Tabelle 3.12 aufgeführte Anforderung MMI-4 „intuitives Programmierinterface“ verlangt, dass der Nutzer durch ein teilweise automatisches, möglichst intelligentes Programmiersystem unterstützt wird.

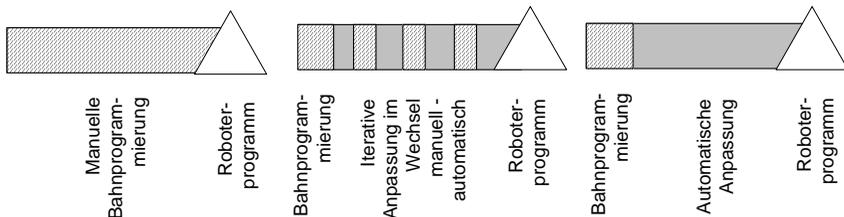


Bild 4.1: Varianten des Wechsels zwischen manueller und automatischer Programmierung: Rein manuelle Bahnprogrammierung und Anpassung (P-1, links), iterative Bahnprogrammierung und Anpassung (P-2, Mitte), automatische Bahnprogrammierung und Anpassung (P-3, rechts).

Eine wie in P-1 (siehe Bild 4.1) dargestellte Programmierung verlangt ein tiefes Wissen des Nutzers über das Robotersystem, er muss viele Arbeitsgänge unterschiedlicher Komplexität ausführen. Eine vollautomatische Anpassung wie in P-3 verlangt die Implementierung eines Expertensystems, welches nicht nur prozessabhängig, sondern meist auch sehr spezifisch anwendungsabhängig entwickelt werden muss. Alternative P-2 stellt einen Kompromiss der Extreme dar. Unter Berücksichtigung der Anforderungen (P-1 nicht intuitiv genug für MMI-4, P-3 sehr aufwändig und kostenintensiv, also im Widerspruch zu GS-1) wird die Variante P-2 gewählt.

Auf der Basis der Varianten GG-B und P-2 kann der Verfahrensablauf als Grundlage der folgenden konkreteren Teilkonzepte erstellt werden. Bild 4.2 zeigt die Programmgenerierung als Wechsel zwischen manuellen Programmierschritten (M1 bis M3) und automatischen Verarbeitungsschritten (A1 bis A3).

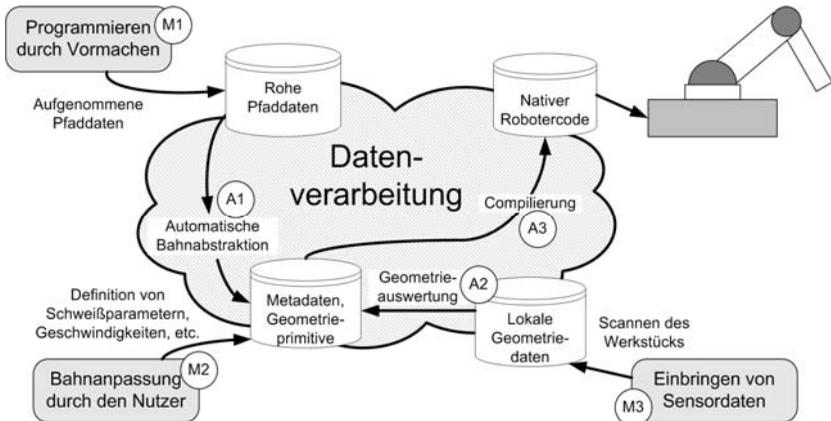


Bild 4.2: Verfahrensablauf der Programmgenerierung. Die manuellen Arbeitspunkte M1, M2 und M3 (optional) wechseln sich ab mit den automatischen Verarbeitungsschritten A1, A2 (optional) und A3.

Dabei stellt der auf der Basis der Anforderungen (siehe Tabelle 3.12) ausgewählte Verfahrensablauf nur eine von mehreren möglichen Alternativen dar. Es ist nicht ausgeschlossen, dass diese Alternativen ebenfalls zu technisch und wirtschaftlich tragfähigen Lösungen entwickelt werden können.

## 4.2 Konzeption der Mensch-Maschine-Interaktion

In diesem Abschnitt werden die für eine Realisierung notwendigen weiteren Komponenten diskutiert. Wieder wird das resultierende Gesamtsystem auf der Basis der in Tabelle 3.12 vorgestellten Anforderungen ermittelt.

### 4.2.1 Konzeption der Interaktionsschnittstellen

Der Nutzer muss auf verschiedenste Weise mit dem Robotersystem interagieren, von der Bahnprogrammierung über Parametrierung etwa des Override bis hin zur Auswahl von Prozessparametern oder Verwaltungstätigkeiten wie dem Speichern von Programmen.

Abgeleitet aus den notwendigen Tätigkeiten in Abschnitt 3.1.2 und dem Verfahrensablauf in Bild 4.2 ergeben sich 6 Interaktionsszenarien während der Programmierung des Robotersystems: das Vormachen der Bahn, die Kommandierung während des Vormachens der Bahn, die Nachbearbeitung der Geometrie und der Parameter, die Verwaltung und die Kommandierung in der Fertigung.

		<b>Mod1 Haptik – Roboter</b>	<b>Mod2 Haptik – Zeigestift</b>	<b>Mod3 Maus / Key- board / PHG</b>	<b>Mod4 Touchscreen</b>	<b>Mod5 Sprache</b>
	Bezeichnung / Beschreibung	Führen mit KMS, etc.	Getrackter Stift	Standard-PC Interface	Grafische Interaktion durch TS	Verbale Komman- dierung
<b>Vormachen der Bahn</b>	Bewegung des Roboters ent- lang der Bahn	+ Keine Ki- nematik- probleme - langsam	+ schnell und leicht - Kinematik- probleme	./.	./.	./.
<b>Komman- dierung Program- mierung</b>	Setzen des Override, Start und Stopp der Aufnahme, ...	./.	./.	- nicht am Platz - wenig ge- eignet für Werkstatt- umgebung	+ intuitiv - nicht am Platz	+ immer verfügbar + zusätzliche Modalität - Lärmresis- tenz
<b>Nachbear- beitung – Geometrie</b>	Auswählen und Verschie- ben oder Lös- chen von Punkten, ...	./.	./.	+ bekannt + genau - Arbeitsplatz notwendig	+ intuitiv + möglich am Roboter - präzise Eingabe schwieriger	./.
<b>Nachbear- beitung – Parameter</b>	Setzen von Geschwindig- keiten, Pro- zesspara- metern, ...	./.	./.	+ bekannt + genau - Arbeitsplatz notwendig	+ intuitiv + möglich am Roboter	./.
<b>Verwaltung</b>	Laden und Speichern von Programmen	./.	./.	+ bekannt	- komplex	./.
<b>Komman- dierung Fertigung</b>	Start und Stopp der Programme	./.	./.	+ mit PHG	- indirekt	./.

Tabelle 4.2: Morphologischer Kasten der Interaktionssituationen und der Modalitäten.

Die wesentlichen Interaktionsmodalitäten sind in Tabelle 4.2 mit den Interaktionsszenarien zu einem morphologischen Kasten entwickelt worden. Neue Eingabemethoden wie etwa in (Wörn 2008) sind in späteren Versionen möglich. Der Schrägstrich „/“ symbolisiert, dass keine als sinnvoll erachtete Lösung für diese Kombination aus Interaktionsszenario und Modalität existiert. So sind beispielsweise viele weitere Szenarien mit einem Sprachverarbeitungssystem möglich, etwa das Laden und Speichern von Dateien. Auf Grund der Erfahrung mit den entsprechenden Sprachsystemen und insbesondere ihrer Akzeptanzprobleme wegen der Lautstärke in der Werkstatt und den oft schwierigen Dialogen wurden sie nicht weiter herangezogen. Ebenso wäre es mit einer Handführung möglich, Dateien zu laden und zu speichern, dies ist aber nicht sinnvoll.

Auf der Basis der Anforderungen sind die gerasterten Zellen ausgewählt worden. Wie oben heißt dies nicht, dass andere Lösungskombinationen nicht auch technisch und wirtschaftlich überzeugende Interaktionssysteme hervorbringen könnten.

Für die Führung des Roboters wurde die Lösungsalternative Mod1 ausgewählt, also das Führen des Roboters. Dies hat im Gegensatz zu Mod2 den Vorteil, dass Kinematikfragen wie die Erreichbarkeit des Punktes oder die Möglichkeit der Kollision mit dem Werkstück während der Programmierung implizit beantwortet werden. Ein Programmiersystem auf Zeigestiftbasis wie Mod2 muss diese Berechnungen virtuell nachvollziehen und ist damit komplexer und weniger verlässlich, insbesondere wenn nur ein Zellenmodell, aber kein Werkstückmodell vorhanden ist. Der Nachteil der langsameren Interaktion wird als weniger wichtig bewertet.

Für die Kommandierung während der Programmierung, also etwa das Setzen des Override, wird Mod5 ausgewählt. Ein Sprachinterface hat hier den großen Vorteil, dass es im Prozess bedient werden kann, ohne die Hände vom Roboter zu entfernen. Der Nachteil der Anfälligkeit für Lärm wird in Kauf genommen; hier können technische Lösungen helfen oder mit wenig Aufwand ein redundantes Interaktionssystem implementiert werden, das beim Versagen der Sprachinteraktion genutzt wird.

Die Nachbearbeitung bezüglich Geometrie und Parametern wird mit einem Touchscreen vorgenommen. Im Gegensatz zum normalen Arbeitsplatz mit Maus und Keyboard kann er direkt in die Zelle integriert werden und entsprechend den Umgebungsanforderungen ausgelegt werden. Die mit dem Touchscreen erreichbare Genauigkeit ist ausreichend.

Die Verwaltung von Dateien wird - wenn irgend möglich - auch mit dem Touchscreen ausgeführt, also mittels vorgeschlagener Dateinamen und durchlaufender Nummerierung. Falls dies nicht geht, ist eine Hilfstatatur zu verwenden.

Start und Stopp des Fertigungsbetriebes ist über das Programmierhandgerät vorzunehmen.

### 4.2.2 Konzeption der Führungseinheit

Die Detailkonzeption der Interaktionsmodalität Mod1 umfasst die technische Umsetzung unter der Berücksichtigung wirtschaftlicher Randbedingungen. Die Variante Mod1 legt fest, dass der Roboter zur Bahnprogrammierung geführt werden soll

Für eine nähere Diskussion in Frage kommen die Lösungsvarianten Fü-1 „Ein KMS“, Fü-2 „Zwei KMS“, Fü-3 „Joystick“. Weitere, im Stand der Technik angesprochene Varianten kommen auf Grund der Anforderungsliste nicht in Frage: Eine Kraftsteuerung mittels Achs-Momenten-Sensoren kommt ebenso wie die Kraftführung auf der Basis der Motorströme nur für einzelne Robotertypen in Frage (Anforderung ST-2), das passive Führen einer ausbalancierten Kinematik ist für Schweißroboter nicht verfügbar (Anforderung GS-3). Tabelle 4.3 detailliert Lösungsvarianten.

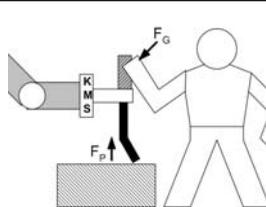
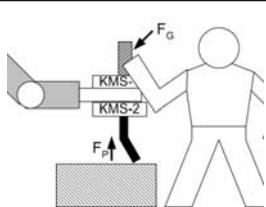
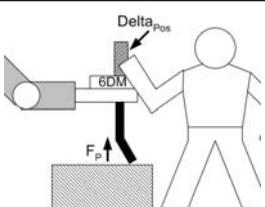
	Fü-1: Ein KMS	Fü-2: Zwei KMS	Fü-3: Joystick
Skizze			
Beschreibung	Ein Kraft-Momenten-Sensor ist zwischen TCP und Führungseinheit montiert. Er misst sowohl Führungs-, als auch Prozesskräfte	Ein KMS misst die Führungskräfte, ein zweiter die Prozess- und Kontaktkräfte. Aus beiden Signalen wird eine Bewegungsvorgabe für den Roboter erzeugt.	Ein Joystick auf der Basis einer 6-D-Maus ist am Führungselement angebracht, der Roboter bewegt sich entsprechend der Joystickbewegung.
Kosten	ca. 2.500,- €	ca. 4.500,- €	ca. 1.000,- €
Vorteile	Einfache Implementierung, Detektion von Kollisionen, Begrenzung der Kraft bei Kontakt	Feineres Führen durch die messtechnische Unterscheidung von Führungs- und Prozesskräften. Kein Zurückspringen.	Günstige Lösung
Nachteile	Keine Unterscheidung zwischen Führungs- und Kontaktkräften, dadurch ist Zurückspringen bei zu geringer Bandbreite möglich	Hohe Kosten, komplexe Realisierung.	Weniger feinfühlig, kein Schutz vor Kollisionen und zu starkem Kontakt.

Tabelle 4.3: Lösungsalternativen zur Führung des Roboters

Prinzipielle Unterschiede zwischen den Lösungsvarianten Fü-1 bis Fü-3 liegen im Sensorprinzip, in der Abdeckung der Detektion aller auftretenden Kräfte und in der Feinfühligkeit. Sowohl das Messprinzip des Kraft-Momenten-Sensors als auch das des Joysticks auf der Basis

einer 6-D-Maus beruhen auf Verformung durch Krafteinwirkung. Beim KMS werden kleinste Verwindungen in einer Stahlstruktur durch Dehn-Mess-Streifen erfasst, bei der Maus detektieren optische Sensoren die Verschiebung von Strahlschlitzen, fixiert durch Federn. Die unterschiedlichen Prinzipien haben zur Folge, dass die 6-D-Maus gefühlt wegbasiert arbeitet, während der Kraft-Momenten-Sensor gefühlt kraftbasiert funktioniert.

Ein weiterer, wesentlicher Unterschied in den Lösungen ist die Position der Sensoren in den Kraft-Einleitungsstrecken Griff-Roboter und Schweißspitze-Roboter. Optimal ist hier die Variante FÜ-2, da sie durch ihren Aufbau Prozess- und Kontaktkräfte differenziert. Die KMS können entsprechend der einwirkenden Kräfte ausgewählt werden und erlauben so eine feinfühligere Führung als mit einem einzelnen KMS. Die Kosten, die Komplexität und die Wartbarkeit allerdings verändern sich in die negative Richtung.

Problematisch bei FÜ-1 ist die Addition von Prozess-, Kontakt- und Führungskräften vor dem KMS. Die Steuerung kann nicht unterscheiden, ob eingeleitete Kräfte durch Führung oder Kontakt entstehen.

Kritischer ist die Situation bei FÜ-3, wo Prozess- und Kollisionskräfte nicht detektiert werden. Hier kann der Nutzer den Roboter in Richtung einer Kollision führen, bis ein zusätzliches Schutzelement, etwa ein Kollisionsschutz oder die Schleppfehlerüberwachung, anschlagen.

Die Variante FÜ-3 scheidet auf Grund der Kollisionsgefahr aus, bei den Varianten FÜ-1 und FÜ-2 wird der Kosten- und Komplexitätsvorteil von FÜ-1 stärker gewichtet als der Sensitivitätsvorteil von FÜ-2.

### **4.2.3 Sicherheitskonzept zur Mensch-Roboter-Interaktion**

Die Mensch-Roboter-Interaktion während der Programmierung muss nicht nur zuverlässig funktionieren, sondern auch sicher sein. Dazu wird sie so umgesetzt, dass das Gesamtsystem nach der Maschinenrichtlinie (MRL 2006) zertifiziert werden kann, sie muss insbesondere den Vorgaben der Norm für die Sicherheit von Industrierobotern (DIN EN ISO 10218-1 2006) folgen. Direktere Formen der physikalischen Mensch-Roboter-Kooperation, wie etwa in (DeSantis 2008) dargestellt, sind industriell heute noch nicht umsetzbar.

Bisher gibt es wenige Umsetzungen einer nach dieser Norm sicheren Mensch-Roboter-Interaktion, Beispiele sind im Abschnitt 2.3.5 aufgeführt. Vier Möglichkeiten der erlaubten Zusammenarbeit sind in (DIN EN ISO 10218-1 2006) beschrieben, und zwar

1. der sichere Betriebshalt,
2. die Verwendung eines Zustimmuschalters bei reduzierter Geschwindigkeit,
3. die Verwendung eines sicheren Abstandssensors, oder
4. die Leistungs- bzw. Kraftbegrenzung auf 80 W bzw. 150 N.

Unter Beachtung der Anforderungen nach Tabelle 3.12, insbesondere SI-3 (markterhältliche Komponenten) und GS-3 (Auslegung der Zelle dem Prozess entsprechend) muss das Sicher-

heitskonzept auf den Möglichkeiten 1 und 2 beruhen. Für Möglichkeit 3 notwendige sichere und hochauflösende Sensoren sind nicht verfügbar, eine Leistungs- bzw. Kraftbegrenzung nach Möglichkeit 4 erfüllt nicht die Prozessanforderungen in Bezug auf Handhabungsgewicht, Arbeitsraum und Geschwindigkeit.

Alle Funktionalitäten müssen entsprechend Performance Level 3 der (DIN EN ISO 13849-1) umgesetzt sein. Zum Führen eines Standard-Industrieroboters ist eine Kombination aus sicherem Betriebshalt zum Eintritt des Nutzers und Zustimmschalter zur Bewegung des Roboters notwendig. Das Layout sowie die Verkettung der Sicherheitskomponenten sind in Bild 4.3 dargestellt.

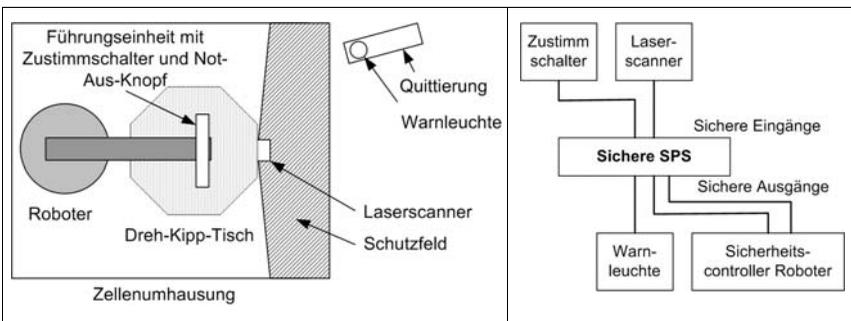


Bild 4.3: Layout (links) und Verkettung (rechts) der sicherheitsrelevanten Komponenten der Roboterzelle

Für die Detektion einer Person im Arbeitsraum ist ein Laserscanner zuständig, das Führen des Roboters wird über einen 3-stufigen Zustimmschalter quittiert. Diese Signale werden von einer sicheren SPS kombiniert und entsprechend einer Wahrheitstabelle an die sicheren Eingänge der Sicherheitssteuerung des Roboters übermittelt. Eine Warnleuchte ist angeschaltet, sobald der Roboter im kollaborierenden Betrieb ist, diverse Not-Aus-Schalter sind an den Hauptkomponenten des Aufbaus angebracht.

Die Verknüpfung der Eingangssignale mit den Ausgangssignalen ist in Tabelle 4.4 dargestellt, sie wird in der sicheren SPS vorgenommen. Zur Umsetzung dieses Sicherheitskonzeptes ist eine sichere Hardware wie etwa in (Heiligensetzer 2005) oder (Som 2007) beschrieben notwendig.

Die sichere SPS verknüpft zwei Eingänge miteinander, es gibt also theoretisch vier mögliche Fälle. Zwei Kombinationen von Eingangssignalen sind der Programmierung zuzuordnen, als der Anwesenheit von Personen im Arbeitsraum des Roboters. Ein weiterer Fall ist für den vollautomatischen Fertigungsbetrieb vorgesehen, hier ist die Anwesenheit eines Menschen nicht erlaubt. Der vierte Fall ist nicht durch eine reale Situation abgebildet. Wenn der Zustimmschalter gedrückt ist, obwohl das Schutzfeld des Laserscanners nicht durchbrochen wurde, ist ein Fehlerzustand erreicht.

Wahrheitswerte/ Situation	Eingang 1: Laserscanner	Eingang 2: Zustimm- schalter	Ausgang 1: Betriebshalt	Ausgang 2: Geschwindig- keitsüber- wachung	Ausgang 3: Warnleuchte
Programmierung: Person in der Nähe	Schutzfeld verletzt	Nicht gedrückt	Angefordert	Angefordert	Angefordert
Programmierung: Person führt Roboter	Schutzfeld verletzt	Gedrückt	Nicht angefordert	Angefordert	Angefordert
Fertigungsbetrieb	Schutzfeld frei	Nicht gedrückt	Nicht angefordert	Nicht angefordert	Nicht angefordert
Fiktiv	Schutzfeld frei	Gedrückt	Fehlerzustand (angefordert)	Fehlerzustand (angefordert)	Nicht angefordert

Tabelle 4.4: Wahrheitstabelle der Sicherheitszustände

## 5 Entwicklung eines Verfahrens zur Adaption von Bahndaten für die intuitive Programmierung

### 5.1 Erstellung eines Modells der Aufgabe

Der Nutzer nimmt während der Programmierung die Bahn des Roboters auf. Diese lässt sich für eine spätere Nutzung formalisieren, dabei basieren die Positionsangaben auf der Vereinbarung der Koordinatensysteme  $KS_W$  (Weltkoordinatensystem),  $KS_T$  (Tool-Koordinatensystem) und  $KS_{DKP}$  (Koordinatensystem auf dem Dreh-Kipp-Tisch) siehe Bild 5.1.

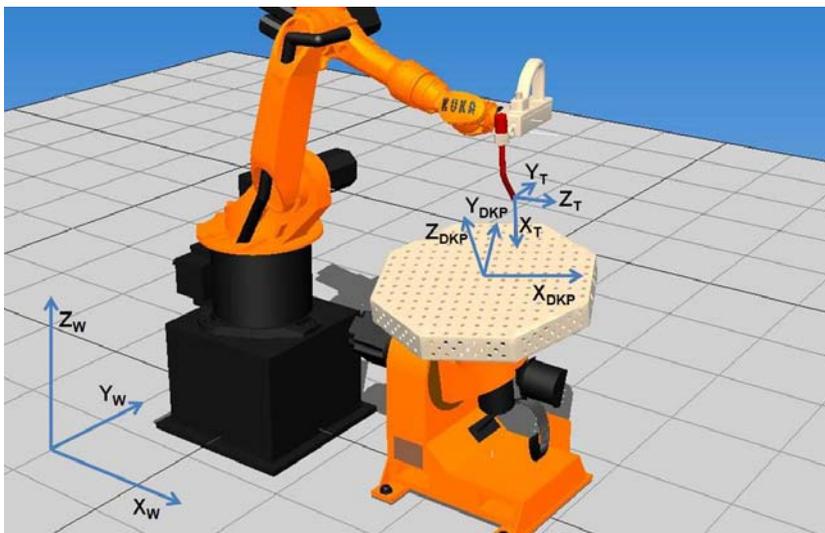


Bild 5.1: Vereinbarung der Koordinatensysteme  $KS_W$ ,  $KS_T$  und  $KS_{DKP}$

### 5.2 Formale Lösung

Um aus einer Menge von aufgenommenen 3-D-Punkten eine für den Roboter verwendbare Trajektorie zu erzeugen, werden drei wesentliche Fragestellungen untersucht:

- Kompression der aufgenommenen Bahndaten  
Während des Programmierens durch Vormachen werden in kurzem Zeittakt die aktuellen Positionen des Roboters aufgenommen. Diese Daten müssen für eine weitere Verarbeitung möglichst verlustfrei komprimiert werden.
- Segmentierung der Bahn  
Die in Kapitel 4 definierten verschiedenen Bahnabschnitte, etwa Geraden, Kreisbögen oder Splines, müssen erkannt und vereinzelt werden.

- Mapping der einzelnen Segmente  
Den einzelnen Segmenten müssen parametrisierte Geometrieprimitive zugewiesen werden, so dass diese vom Benutzer interaktiv angepasst werden können.

Das Vorgehen der Bahnabstraktion sieht also wie in Bild 5.2 dargestellt aus. Neben den oben vorgestellten Vorgängen muss eine Möglichkeit für den Benutzer vorgesehen werden, die Ergebnisse der automatischen Stufen anpassen zu können.

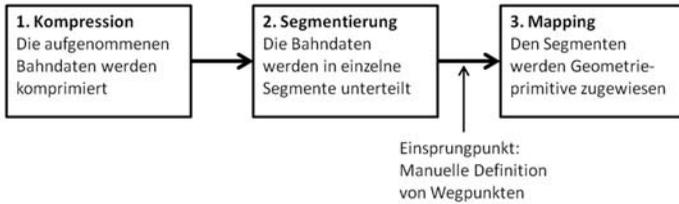


Bild 5.2: Vorgehen zur Bahnabstraktion

Ziel dieses Abschnittes ist es, aus den verschiedenen Möglichkeiten der Kompression, der Segmentierung und des Mappings die optimale Kombination zu ermitteln. Dies wird durch die Implementierung und den anschließenden numerischen Vergleich verschiedener Lösungskombinationen erreicht.

### 5.2.1 Testimplementierung der Algorithmen

Die Kombination von Algorithmen zu Kompression, Segmentierung und Mapping der Bahndaten ergeben verschiedene Gesamtverfahren, diese werden in einer 2-D Testumgebung an Hand einer Fehlerfunktion verglichen. Die Fehlerfunktion wird in einem ersten Schritt als Fläche zwischen der aufgenommenen Originalbahn und der erzeugten komprimierten Bahn definiert:

$$f(t) = \int_0^t f_{res} - f_{orig} dt \quad (5.1)$$

Die Umsetzung der Flächenberechnung erfolgt über die Triangulation der Zwischenflächen. In späteren Schritten stellt diese Fehlerfunktion eine Komponente einer Gütefunktion dar.

Die zu untersuchenden Algorithmen wurden unter VisualC++ implementiert und in eine Anwendung mit 2-D-Nutzeroberfläche eingebettet. Diese erlaubt sowohl das Laden von realen Bahndaten als auch die Definition von Bahnen mit Hilfe der Maus. Erkannte Eckpunkte werden als gelbe / helle Kreise dargestellt, Spline-Stützpunkte als blaue / dunkle Kreise, siehe Bild 5.3. Sowohl die Originalbahn als auch die automatisch erzeugte Bahn werden visualisiert. Erkannte Punkte können in ihrem Modus (Eckpunkt / Stützpunkt) interaktiv geändert

werden, ebenso können überzählige Punkte gelöscht werden. Die Triangulation zur Berechnung der Fehlerfunktion kann eingeblendet werden.

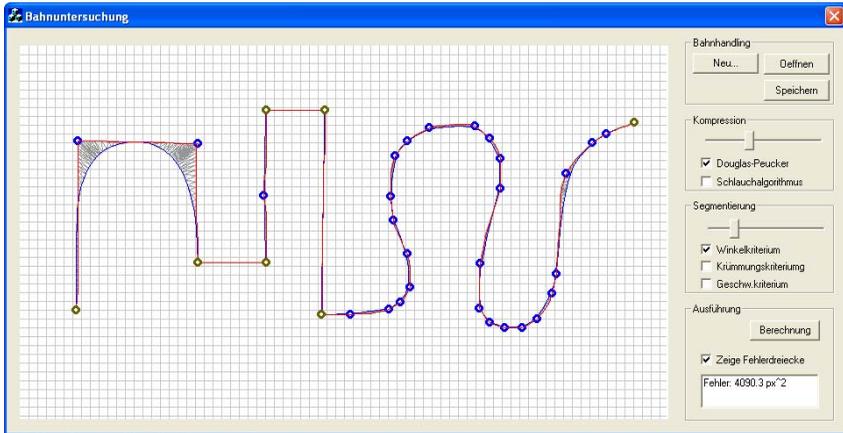


Bild 5.3: Grafische Oberfläche der Testumgebung zum Vergleich der Algorithmen, im linken Bildbereich die Darstellung der Triangulation zur Berechnung der Fehlerfunktion.

Die Verfahrenskombinationen werden unter drei Bahnkurven mit jeweils besonderen Charakteristika getestet. Die Besonderheiten der Bahnkurven bestehen in der Aufteilung der Bahn in Geraden und Kurven, die Bahnen 1 und 3 sind eine Kombination, während Bahn 2 nur auf Kurvendarstellung angewiesen ist, siehe Bild 5.4.

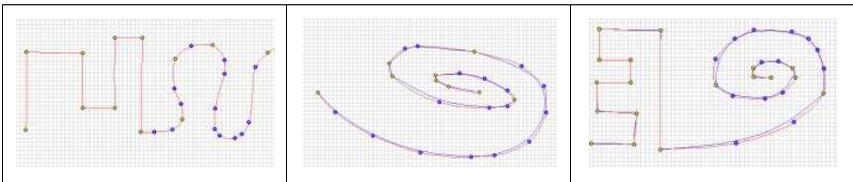


Bild 5.4: Optimale Ergebnisse von Kompression und Mapping. Diese Bahnkurven mit ihren Metadaten werden als Soll-Daten verwendet.

Die Ergebnisse der Untersuchungen sind in den folgenden Abschnitten dargestellt.

### 5.2.2 Kompression der Bahndaten

Bei der kontinuierlichen Aufnahme von Positionsdaten in kurzem Zeittakt fallen große Datenmengen an. Ziel der Kompression ist es, diese Datenmengen mit einem akzeptablen, wenn möglich adaptierbaren Verlust an Genauigkeit zu komprimieren. Dazu wird versucht, die Punkte der aufgenommenen Bahn auszudünnen. Ein Überblick ist in (Heckbert 1997) zu

finden, bekannte Ansätze basieren auf dem Krümmungsverhalten (Teh 1989) und dem Douglas-Peucker-Algorithmus (Hershberger 1992). In der Robotik ist ein Schlauchalgorithmus in Verwendung (Seyfarth 2004).

Der Douglas-Peucker-Algorithmus ermittelt rekursiv einen Linienzug, der die Eingabekontur mit entsprechender Genauigkeit approximiert, erstmals im Bereich der Computergrafik eingesetzt. Für eine Ausgangskurve, die eine Menge von Punkten

$$K: \{P_1, \dots, P_i, \dots, P_n\},$$

darstellt sowie ein Toleranzmaß  $\varepsilon > 0$  wird die Strecke zwischen  $P_1$  und  $P_n$  gebildet. Es wird nun der Punkt  $P_m$  mit  $2 \leq m \leq n - 1$  gesucht, der den größten orthogonalen Abstand von der Strecke  $\overline{P_1 P_n}$  besitzt:

$$d_{max} = \max_{i=2..n-1} d_{orthogonal} (P_i, \overline{P_1 P_n}). \quad (5.2)$$

Ist  $d_{max} \leq \varepsilon$  oder die Menge nur zwei Elemente groß, so wird der Algorithmus abgebrochen. Andernfalls startet die Rekursion mit den zwei Teilmengen  $K_1: \{P_1, \dots, P_m\}$  und  $K_2: \{P_m, \dots, P_n\}$ . Abbildung Bild 5.5 zeigt den Algorithmus.

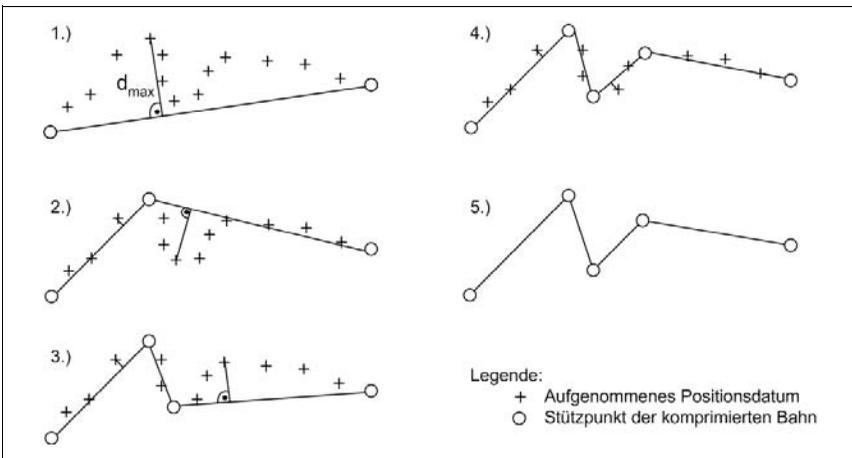


Bild 5.5: Rekursive Kompression mit dem Douglas-Peucker Algorithmus

Wesentlich für die Funktionalität ist die Festlegung der entsprechenden Toleranz  $\varepsilon$ . Sie bestimmt die Anzahl der erzeugten Stützpunkte, also der nach der Kompression verbleibenden Bahnpunkte. Diese Toleranz kann fix gesetzt werden oder variabel gestaltet sein. Das Toleranzmaß muss entsprechend der später notwendigen Gesamtgenauigkeit gesetzt werden.

Um für die Robotik anwendbar zu sein, muss der Algorithmus aus dem 2-D Anwendungsfall auf den 6-D Anwendungsfall erweitert werden, also auch bei Winkeländerungen ohne Positionsänderungen muss der Algorithmus einen Zwischenpunkt setzen.

Ein bereits in der Robotik zur Datenkompression verwendeter Algorithmus basiert auf einer Schlauchbetrachtung, beschrieben in (Seyfarth 2004). Es werden verschiedene Kriterien verwendet, die relevanten Punkte zu detektieren: die Abweichung um einen bestimmten Betrag von einer Geraden aus den letzten beiden Stützpunkten, die Umkehr auf einer Geraden oder die Verstellung der Orientierung um einen bestimmten Winkel.

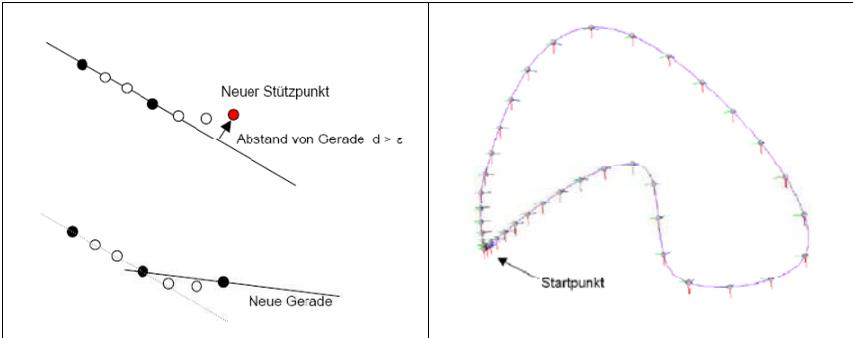


Bild 5.6: Schlauchalgorithmus, Quelle: (Seyfarth 2004)

Eine wesentliche Unterscheidung kann im Zeitpunkt der Aufnahme gemacht werden:

- Aufnahme zeitdiskreter Punkte, spätere Kompression  
Diese Alternative ist speicherintensiv, erlaubt aber in der späteren Bearbeitung die Anwendung verschiedener Filter.
- Aufnahme der bereits komprimierten Punkte  
Bei der online-Kompression werden die Algorithmen während des Führens durchgeführt. Dies führt zu wesentlich kleineren Datendateien, aber auch zu eingeschränkten Möglichkeiten der Nachbearbeitung.

Die Implementierung der Kompressionsalgorithmen führte zu den in Bild 5.7 dargestellten Ergebnissen.

Die Auswertung ergibt, dass der Douglas-Peucker-Algorithmus deutlich zielgerichteter komprimiert, die von ihm erzeugten Fehlerwerte liegen zu einem großen Teil unter denen des Schlauchalgorithmus, bei einer geringeren Anzahl von Stützpunkten.

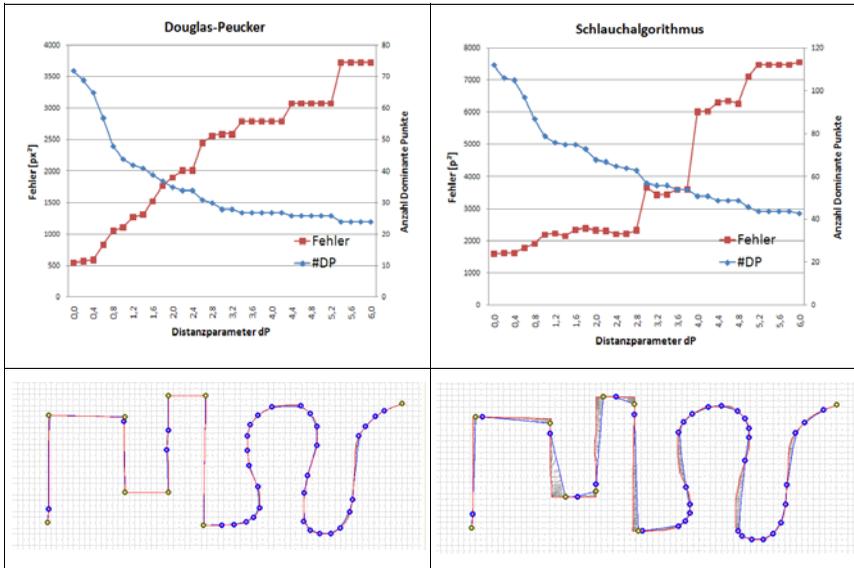


Bild 5.7: Fehlerwerte der Datenkompression mittels Douglas-Peucker (links oben) und Schlauchalgorithmus (rechts oben) für die drei Referenzkurven und eine Parametervariation von 0,01 bis 6. Unten die Abbildung der Ergebnisse für Douglas-Peucker (links) und Schlauchkriterium (rechts) bei jeweils 42 bzw. 43 erzeugten Dominanten Punkten

Das für den Bediener relevante Kriterium ist die Bahntreue bei guter Nutzbarkeit. Das Vergleichskriterium der Algorithmen ist also der Fehler bei einer gleichen Anzahl von Stützpunkten. Hier erreicht der Douglas-Peucker-Algorithmus bessere Werte:

- Schlauchalgorithmus: 44 Stützpunkte bei einem Fehler von 7489 px<sup>2</sup>
- Douglas-Peucker: 44 Stützpunkte bei einem Fehler von 1111 px<sup>2</sup>

Die Darstellung der ausgewerteten Kurven in Bild 5.7, unten, zeigt eine Ursache der deutlich unterschiedlichen Performance auf: Der Schlauchalgorithmus setzt die Stützpunkte so, dass sich insbesondere bei der Abbildung von Linienzügen mit großen Winkeln hohe Fehler ergeben. Die Performance des Algorithmus ist besser im zweiten Teil der Kurve, wo Freiformflächen nachgebildet werden. Im Anwendungsfall des Schweißens sind aber Linienzüge vorherrschend, daher ist die Untersuchung auf der Basis der Referenzkurve 1 gerechtfertigt.

Einen großen Vorteil hat der Schlauchalgorithmus dadurch, dass er online verwendet werden kann, er kann also während der Datenaufnahme die Datenmenge reduzieren. Dies ist mit dem rekursiven Ansatz des Douglas-Peucker-Algorithmus nicht möglich. In der hier beschriebenen Anwendung ist dieser Vorteil allerdings nicht relevant.

Für die weiteren Untersuchungen wird der Douglas-Peucker-Algorithmus verwendet.

### 5.2.3 Segmentierung der Bahn und Mapping von Geometrieprimitiven

Die in der Kompressionsphase gefundenen Stützpunkte müssen durch Geometrieprimitive verbunden werden, um eine Trajektorie zu erzeugen. Dies geschieht im einfachsten Fall durch Geradenabschnitte, wodurch eine Polylinie erzeugt wird. Alternativen dazu sind insbesondere Kreisbögen und Splines. Diese Geometrieprimitive werden auch von verschiedenen Robotersteuerungen unterstützt.

Die umfassendsten Möglichkeiten bietet eine Unterteilung der Bahn in Geraden und Splines. Die meisten Prozessabschnitte können durch Geraden abgebildet werden, für gekrümmte Prozessbahnen oder Verbindungsbewegungen können Splines verwendet werden.

Für die Anwendung im CAD-Bereich wurden die Bézier-Splines entwickelt, die sich aus einer Linearkombination der Bernstein-Polynome ergeben (Schwarz 1997):

$$x_k(t) = \sum_{i=0}^n b_{ik} B_i^n(t; a, b), (k = 1, 2, \dots, d) \quad (5.3)$$

wobei  $d$  für die Raumdimension zwei oder drei steht. Für Position und Orientierung sind zwei Splines in jeweils 3 Dimensionen zu verwenden. Eine Alternative sind die Non Uniform Rational B-Splines (NURBS), eine Verallgemeinerung der Bézier-Splines. Die in Bild 5.8 rechts dargestellte Bahn wurde durch NURBS approximiert.

Eine Eigenschaft der NURBS ist für die hier untersuchte Anwendung allerdings nicht optimal, Stützpunkte werden nicht als Kurvenpunkte verwendet. Dies ist in dieser Anwendung aber notwendig. Daher wurden die einfacheren kubischen Splines für die Approximation verwendet. Diese verbinden ihre Stützpunkte durch Kurven dritter Ordnung, an den Übergangspunkten ist die Gesamtkurve mindestens einmal stetig differenzierbar. Splines sind gegeben durch (Walter 1997):

$$E = \frac{1}{2} \int_{x_0}^{x_n} s''(x)^2 dx \quad (5.4)$$

Für die Lösung des sich aus dieser Minimierungsaufgabe samt der notwendigen Nebenbedingungen ergebenden linearen tridiagonalen Gleichungssystems wird ein Algorithmus aus (Press 1992) mit linearer Laufzeit verwendet. Die Ergebnisse der Approximation von Kurven mit kubischen Splines und NURBS ist in Bild 5.8 dargestellt.

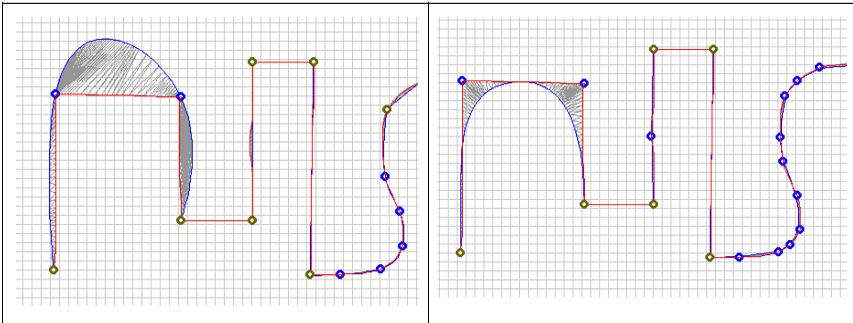


Bild 5.8: Vergleich der Implementierung von kubischen Splines (links) mit NURBS (rechts), plakativ mit einer großen Fehlerfläche.

Die Entscheidung, ob die in der Kompressionsstufe gefundenen Stützpunkte durch Geraden oder Splines verbunden werden, wird mit Hilfe eines Eckenkriteriums getroffen. Der zu Grunde liegende Gedanke ist hier, dass Eckpunkte schlecht durch Kurven verbunden werden können. Formalisiert wurde die Entscheidung an Hand der folgenden Regeln getroffen:

- Sind Punkte  $P_i$  und  $P_{i-1}$  Ecken, dann werden sie durch eine Gerade verbunden
- Wenn  $P_i$  zweitletzter Punkt ist, werden  $P_i$  und  $P_{i+1}$  durch eine Gerade verbunden
- In allen anderen Fällen wird ein Spline aus den Punkten  $P_i$  bis  $P_{i+n}$  erzeugt, wobei  $P_{i+n}$  der erste Eckpunkt nach  $P_i$  ist.

Die vorher notwendige Zuordnung der Punkte als Ecke wird mit Hilfe eines Winkelkriteriums oder auf der Basis des Krümmungsverhaltens vorgenommen. Beide Methoden arbeiten lokal.

Unter Verwendung eines Winkelkriteriums wird ein Punkt  $P_i$  als Eckpunkt definiert, wenn der Vektor  $v_1 = P_{i-1} - P_i$  um einen Winkel  $> \alpha$  von Vektor  $v_2 = P_{i+1} - P_i$  abweicht. Je nach Anwendung gilt  $5^\circ < \alpha < 60^\circ$ . Der Winkel  $\alpha$  berechnet sich aus

$$\alpha = \cos^{-1} \frac{v_1 \cdot v_2}{|v_1| \cdot |v_2|} \quad (5.5)$$

Alternativ dazu kann ein Krümmungskriterium (Teh 1989) verwendet werden, also die Richtungsänderung je Längeneinheit. Der Kehrwert der Krümmung ist der Krümmungsradius, der Radius desjenigen Kreises, der in einer Umgebung des Berührungspunkts die optimale Näherung darstellt. Im Fall einer ebenen Bahnkurve ergibt sich die Krümmung im Punkt  $(x, f(x))$  aus

$$\kappa = \left| \frac{f''(x)}{(1 + f'^2(x))^{3/2}} \right| \quad (5.6)$$

Für den späteren Fall im 3-D stellt sich die Krümmung im Punkt  $\vec{r} = \vec{r}(t)$  dar als

$$\kappa = \frac{\|\vec{r}'(t) \times \vec{r}''(t)\|}{\|\vec{r}'(t)\|^3} \quad (5.7)$$

Die Berechnung erfolgt im oben dargestellten Testprogramm numerisch durch den Fit eines Kreises mit Hilfe von 3 Punkten. Um fehlerhafte und nicht vergleichbare Krümmungswerte zu vermeiden, muss dabei auf die Äquidistanz der Punkte geachtet werden. Dies wird durch Interpolation erreicht. Bild 5.9 ist ein Beispiel für die Berechnung der Krümmungen. Die zwei dargestellten Kreise mit den Durchmessern  $d_{01}$  und  $d_{02}$  verdeutlichen das Kriterium. Liegen die Punkte im Wesentlichen auf einer Geraden, so ist ein großer Durchmesser notwendig. In Ecken sind sehr kleine Kreise das Ergebnis. Liegen die Punkte exakt auf einer Linie, wird eine Ausnahmebehandlung durchgeführt.

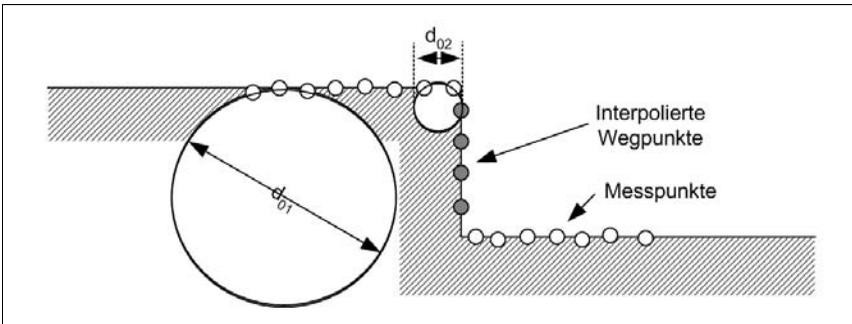


Bild 5.9: Schematische Darstellung der Berechnung der Krümmungswerte, hier mit integrierter Interpolation nicht äquidistanter Punkte.

Die Untersuchungen ergeben, dass die beiden Algorithmen, obwohl generell nicht ineinander überführbar, für diese Aufgabenstellung austauschbar sind. Der nichtlineare Teil des Krümmungskriteriums steigt bei Annäherung an  $0^\circ$ , während die hier relevanten Winkel im Bereich größer  $10^\circ$  bis maximal  $100^\circ$  liegen. Die untere Grenze wird durch die vorhergehende Kompressionsstufe definiert, Winkel sehr nahe an  $0^\circ$  entstehen auf Grund der Kompression nicht. In den genutzten Bereichen ist die Kurve linear gut approximierbar, siehe Bild 5.10.

Alternative Verfahren beruhen etwa auf der Wavelet-Transformation (Lee 1995) oder über das Hinzuziehen von prozessorientierten Parametern, etwa dem zeitlichen Verlauf der Bahn. Es ist zu vermuten, dass in den Ecken eine langsamere, abrupte Bewegung stattfindet, während der Rest der Bahn flüssig programmiert wird. Dies ließe sich über Zeitstempel differenzieren.

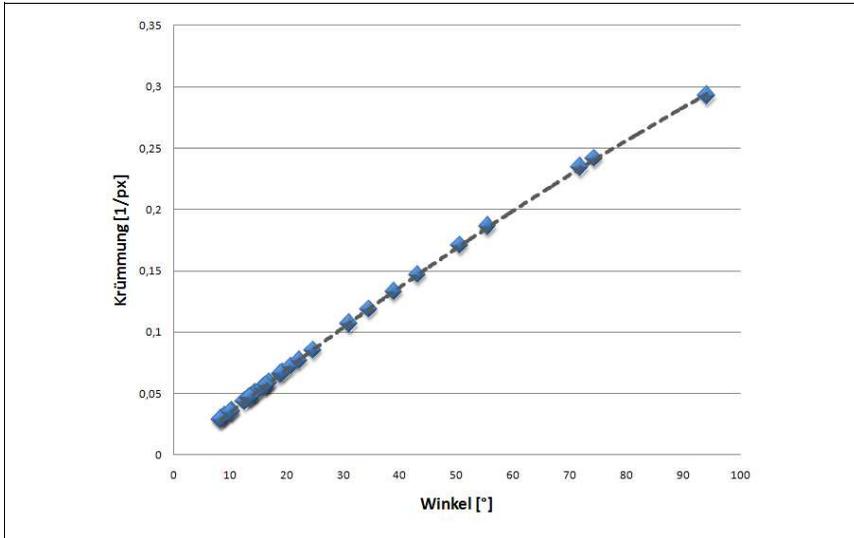


Bild 5.10: Graph der Ergebnisse des Winkel- und Krümmungskriteriums für eine komprimierte Bahn. Eine Trendlinie 2. Ordnung wurde gefittet.

### 5.3 Manuelle Wahl der optimalen Kompressionsparameter

Die weiteren Versuche werden auf der Basis der dargestellten Auswertungen mit der Verfahrenskombination Douglas-Peucker zur Datenkompression und dem Winkelkriterium zur Segmentierung durchgeführt. Ziel der Versuche ist die Einstellung der Verfahrensparameter. Dazu werden Versuche über eine Parametervariation des Kompressionsparameters  $\Delta_{DP}$  und des Segmentierungsparameters  $\alpha_{dist}$  durchgeführt, siehe Tabelle 5.1.

Parameter	Variation	Schrittweite
$\Delta_{DP}$	0,001 px bis 8,0 px	0,3 px
$\alpha_{dist}$	0° bis 60°	2°

Tabelle 5.1: Variationsbreiten der Parameter der Kompressions- und Segmentierungsverfahren

Die Bewertungsfunktion der resultierenden Bahnen berechnet die Fläche zwischen den beiden Kurven mittels eines Triangulationsalgorithmus, sie ist in Bild 5.8, links, für die Referenzkurve 1 dargestellt.

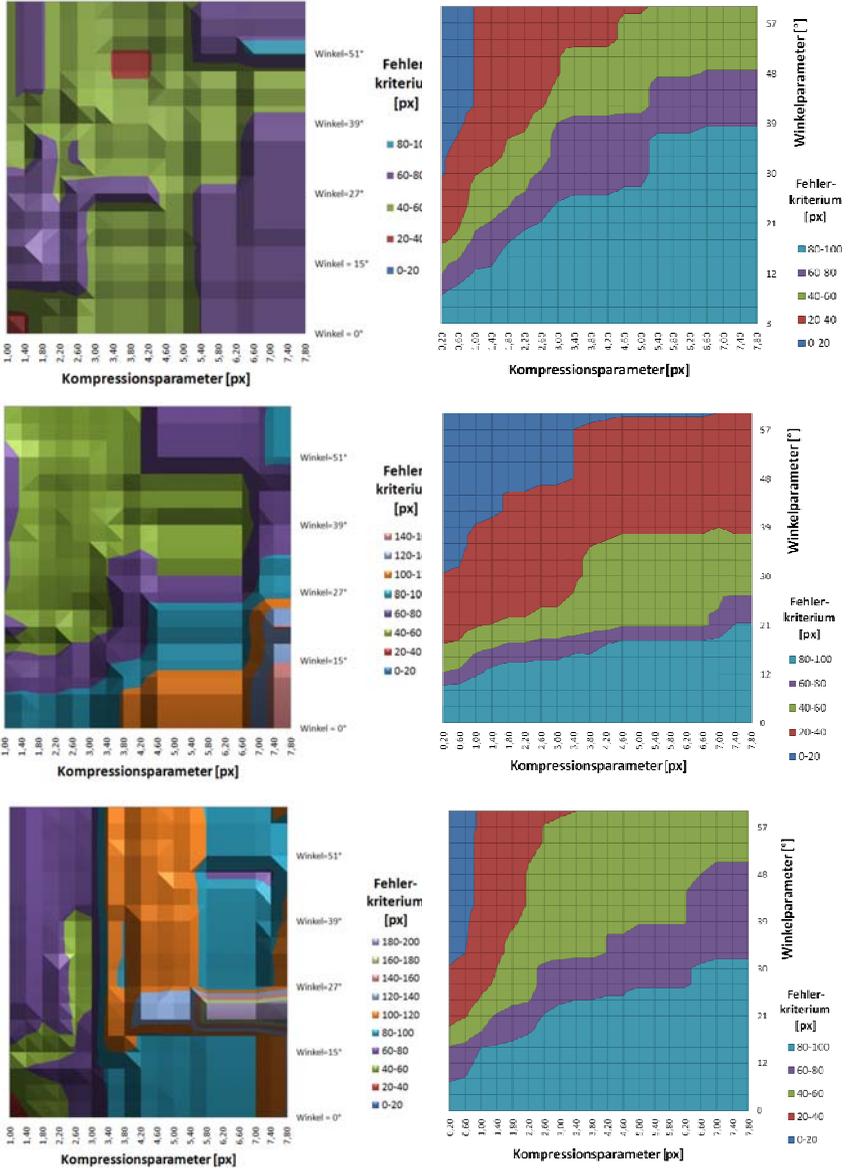


Bild 5.11: Fehlerkriterium (links) als Produkt aus Fehler und Anzahl der Dominanten Punkte und Anteil der Eckpunkte (rechts) bei verschiedenen Parameterkombinationen für die Verfahrens-kombination Douglas-Peucker-Algorithmus mit Winkelkriterium für die Referenzkurven 1 bis 3 (von oben nach unten).

Die Auswertung der oben dargestellten Testläufe führt zu den folgenden Erkenntnissen:

- Die Anzahl der erzeugten Stützpunkte sowie der resultierende Fehler zeigen teilweise einen sprunghaften Verlauf.
- Es werden bei zunehmend größerer Kompression mehr Eckpunkte erzeugt.
- Bei den Referenzbahnen 1 und 3, den Geraden – Kurven – Kombinationen, werden bis auf die extrem geringen Kompressionsparameter immer auch Eckpunkte erzeugt. Für Referenzbahn 2 tritt der Fall der kompletten Abbildung über Kurvenpunkte deutlich schneller ein.
- Bei den Referenzkurven 1 und 3 befinden sich Minima in der linken unteren Ecke, also bei einer extrem feinen Approximation durch Geradenstücke. Bei der Referenzkurve 1 findet sich zusätzlich noch ein Minimum im ersten Quadranten, also für mittlere Kompressionsgrade und eine starke Nutzung von Splines.

Die Auswahl der optimalen Parameter für die Referenzkurven erfolgt unter drei absteigend gewichteten Zielkriterien:

- Minimierung der Anzahl Dominanter Punkte
- Minimierung der Eckpunkte
- Minimierung des Fehlers

Ein mögliches Ergebnis ist in Bild 5.12 dargestellt.

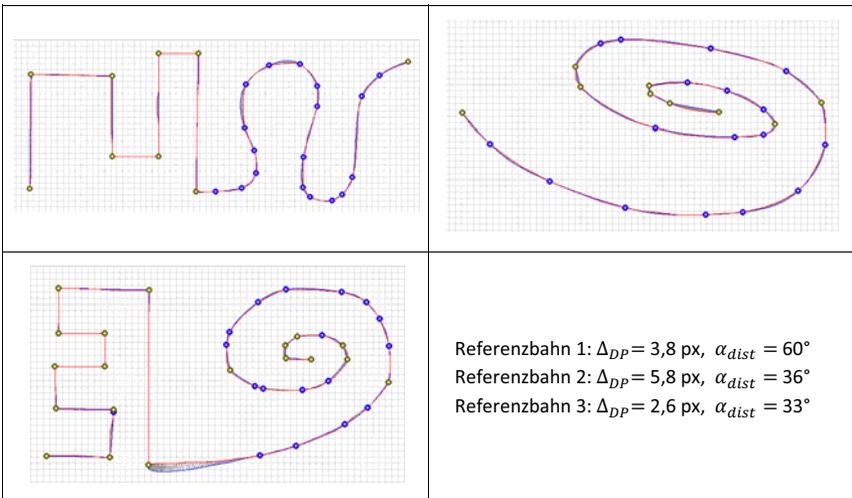


Bild 5.12: Ergebnis der Kompression unter den manuell gewählten Parametern für die Referenzkurven 1 (links oben), 2 (rechts oben) und 3 (links unten).

Verwendet wurden die in der Abbildung aufgeführten Parameter. Basierend auf dieser Ausgangsbasis werden die Bahnen manuell weiter angepasst, etwa um das Überschwingen bei Referenzkurve 3 zu eliminieren.

## 5.4 Automatische Wahl der optimalen Kompressionsparameter

Eine feste Wahl von Kompressionsparametern ist nicht möglich, da sich die optimalen Parametersätze je nach Anwendung unterscheiden. Daher ist eine automatische Wahl der Parameter wünschenswert.

Vor der Umsetzung stellen sich zwei wesentliche Fragen:

- Wie sieht die Gewichtung der oben angeführten Zielkriterien aus, um daraus ein Gütekriterium abzuleiten?
- Welcher Algorithmus wird verwendet, um das globale Minimum des Gütekriteriums zu finden?

Eine der möglichen Fehlerfunktionen wird ausgewählt:

$$\min g() \text{ mit } g(\Delta_{DP}, \alpha_{dist}) = 2,0 \text{ fehler} \cdot 5,0 n_{DP} \cdot (1 + p_c) \quad (5.8)$$

Dabei bezeichnet  $n_{DP}$  die Anzahl der gefundenen Dominanten Punkte,  $p_c$  den Anteil der Eckpunkte an den Dominanten Punkten. Die Wichtung der Zielkriterien ist mit einem nicht zu überschreitenden geometrischen Fehler überlagert.

Für Minimierungsalgorithmen stellt sich durch die fehlende Stetigkeit der Fehlerfunktion das Problem, dass Standard-Abstiegsverfahren nicht verwendet werden können, daher wird ein Monte-Carlo-Verfahren verwendet. Hier ist keine Optimalitätsgarantie gegeben, dafür ist die Laufzeit durch die Anzahl der Testläufe kalkulierbar. Die Ergebnisse der automatischen Parameterwahl sind in Tabelle 5.2 dargestellt.

Referenzbahn	Automatisch gefundene Parameter	Ergebnisse der Kompression
1	$\Delta_{DP} = 4,08 \text{ px}$ $\alpha_{dist} = 44,06^\circ$	Gütefunktion: 39,37
2	$\Delta_{DP} = 0,15 \text{ px}$ $\alpha_{dist} = 52,27^\circ$	Gütefunktion: 41,83
3	$\Delta_{DP} = 0,7 \text{ px}$ $\alpha_{dist} = 36,0^\circ$	Gütefunktion: 22,42

Tabelle 5.2: Ergebnisse der automatischen Parameterwahl. Das Monte-Carlo Verfahren wurde mit 100 Zyklen verwendet.

Weitere Ansätze, die Erkennungsrate zu verbessern, gehen in die Richtung der Künstlichen Intelligenz. Das System lernt, welche Korrekturen durch den Nutzer gemacht werden und kann diese Korrekturen nach und nach eigenständig anwenden. Diese Ansätze werden nicht in dieser Arbeit verfolgt.

## **5.5 Zusammenfassung der Erkenntnisse und Schlussfolgerungen**

Für die formal beschriebene Aufgabe der automatischen Bahnabstraktion wurde eine Verfahrenskombination ermittelt. Diese besteht aus der Kompression der Daten mittels Douglas-Peucker-Algorithmus und dem Mapping von Geraden und Splines basierend auf einem Winkelkriterium. Alternativen dieser Verfahrenskombination, etwa bezüglich der Komprimierung, der Geometriezuordnung oder auch der zugeordneten Geometrieprimitiven, wurden nach numerischen Vergleichen als weniger geeignet zurückgewiesen.

In einem zweiten Schritt wurden die einzustellenden Parameter des Verfahrens untersucht, insbesondere der Kompressionsparameter des Douglas-Peucker-Algorithmus und der Grenzwinkel für die Unterscheidung zwischen Gerade und Spline. Hier wurden für drei Referenzbahnen manuell optimale Werte bestimmt, dann wurde ein Optimierungsverfahren auf der Basis der Monte-Carlo-Methode mit anschließender lokaler Optimierung entwickelt.

Diese automatisierte Kompressionsstufe erreicht eine akzeptable Bahngüte, mit nur wenigen manuellen Korrekturen ist ein optimaler Zustand erreicht.

Weitere Verbesserungen wären durch den Einsatz von lernenden Verfahren möglich, die beispielsweise diese verbleibenden manuellen Korrekturen übernehmen.

## 6 Entwicklung eines Verfahrens zur Bahnoptimierung auf der Basis der sensorisch erfassten lokalen Werkstückgeometrie

Wesentlich für viele Anwendungen ist die Einhaltung von Toleranzen bezüglich der Position des Werkzeugs. Schweißanwendungen bewegen sich hier in einem Rahmen von 0,1 bis 0,5 mm, siehe Abschnitt 3.2.2. Die handgeführte Bahn kann diese Toleranzen nicht immer einhalten, ebenso können bei einer rein handgeführt programmierten Bahn keine Prozessmodule eingebunden werden, die auf der Werkstückgeometrie beruhen.

Eine Lösung, die wie in Anforderung GS-7 aus Tabelle 3.12 keine CAD-Daten voraussetzt, aber trotzdem die Programmierung basierend auf der Werkstückgeometrie erlaubt, ist die Aufnahme der Geometrie mittels eines am Roboter angebrachten Sensors, auch unter dem Begriff *Geometric Reverse Engineering GER* bekannt (Rahayem 2008). Meist auf der Basis von 2,5-D Scannerdaten gibt es verschiedene Ansätze, höher aggregierte Informationen aus den Daten zu gewinnen: (Rabbani 2006) unterteilt in kantenbasierte, oberflächenbasierte und scanlinienbasierte Segmentierungsverfahren. Alternativen sind in Bild 6.1 zusammengestellt.

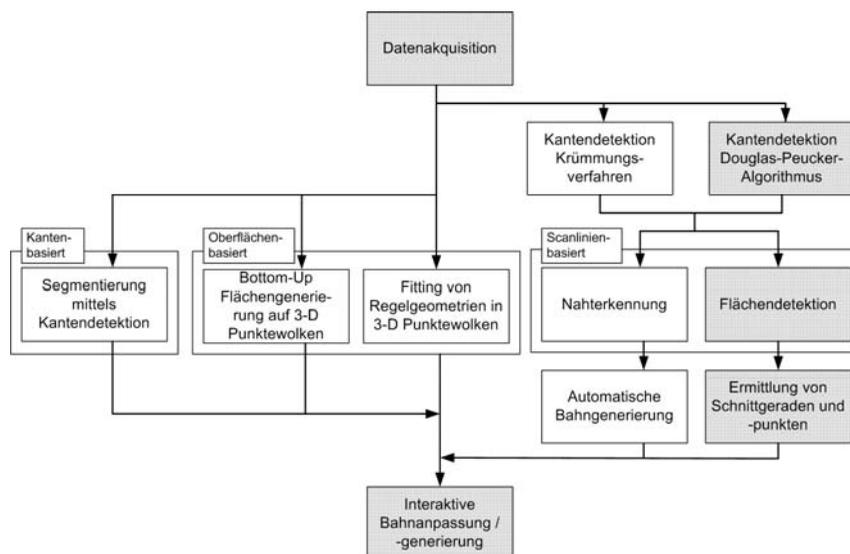


Bild 6.1: Alternative Verfahren der Datenauswertung und -nutzung. Unterlegt: das in Abschnitt 6.4 realisierte Verfahren der scanlinienbasierten Flächenrückführung.

Kantenbasierte Verfahren arbeiten häufig zweistufig, in (Sappa 2001) wird in einem ersten Schritt eine binäre Kantenkarte aufgestellt, im zweiten Schritt werden hieraus Konturen extrahiert. Einfachere Verfahren basieren auf einer Auswertung der Gradienten, einem Line-Fitting oder den lokalen Flächennormalen (Bhanu 1985).

Die Ansätze zur oberflächenbasierten Flächengenerierung lassen sich unterteilen in Bottom-Up-Ansätze, die auf der Basis lokaler Informationen wie der Flächennormale Segmente erzeugen (Rabbaní 2006). Top-Down-Ansätze passen Segmente in die Punktwolke ein und verfeinern so lange, bis ein Threshold erreicht wird (Westkämper 2006-3, Ahn 2004, Xiang 2004).

In dieser Arbeit werden zwei Verfahren auf der Basis der Scanline verfolgt, die automatische Nahterkennung und die Flächendetektion auf Basis eines Region-Growth-Verfahrens. Die automatische Nahtauswertung besitzt Ähnlichkeiten mit den Algorithmen der Online-Nahtverfolgungssensoren (Liangyu 2007, ServoRobot 2010), wenn sie auch anders angewendet wird. Das Region-Growth-Verfahren (Jiang 1994, Khalifa 2003) erzeugt aus den Scanlinien Ebenen. Beide Verfahren werden an die speziellen Anforderungen der Aufgabenstellung angepasst. Insbesondere stellt sich die Frage, wie die auf 2,5-D-Daten ausgelegten Verfahren mit mehreren Scans, also 3-D-Daten operieren können.

## 6.1 Datenakquisition

Die Datenakquisition verläuft mit Hilfe eines Laserlinien-Triangulations-Sensors. Besonders für den Anwendungsfall des Schweißens geeignet sind Nahtverfolgungssensoren, diese sind durch Kühleinrichtungen und Cross-Jet Schutzsysteme auf Schweißumgebungen angepasst. Allgemein verwendbare Sensoren hingegen lassen einen breiteren Datenzugriff zu.

Zwei Modelle von Nahtverfolgungssensoren, Falldorf und MEL-Sensor, sind mit den wesentlichen Daten in der folgenden Tabelle 6.1 aufgeführt, weitere Hersteller sind unter anderem Meta-Scout GmbH, MSC Mess-, Sensor- und Computertechnik GmbH, Bober-Optosensorik und Meßtechnik GbR oder Servo Robot Inc. Die Systeme beruhen auf dem Verfahren der Laser-Triangulation, aber auch auf 2-D-Bildverarbeitung.

In dieser Arbeit werden Laser-Triangulations-Sensoren weiter untersucht. Sie bieten einerseits die für Schweißumgebungen notwendige Robustheit, andererseits sind sie während der Fertigung für ihre Standard-Aufgabe, die Nahtverfolgung und damit den Ausgleich von Toleranzen und Verzug verwendbar.

	Falldorff S7 65/48	Nahtverfolgungssensor Arbeitsabstand 65 mm Sichtbereich / Auflösung X: 36 – 77mm / 56 µm Sichtbereich / Auflösung Z: 70 mm / 130 µm Profile / s: > 50 Cross-Jet, wechselbares Schutzglas, Kühlung möglich
	MEL-Sensor MZDW 160/40	Nahtverfolgungssensor Arbeitsabstand 130 mm Sichtbereich / Auflösung X: 30 – 40mm / 50 µm Sichtbereich / Auflösung Z: 75 mm / 150 µm Profile / s: 50 Wechselbares Schutzglas, Kühlung möglich

Tabelle 6.1: Markterhältliche Laser-Triangulations-Sensoren zur Nahtverfolgung im Schweißprozess

Die im Sensorkoordinatensystem aufgenommenen Höhenlinien werden mit der aktuellen Roboterposition verrechnet und im DKP-Koordinatensystem  $KS_{DKP}$  gespeichert.

## 6.2 Filterung der Sensordaten

Die vom Sensor übermittelten Messdaten sind nicht gefiltert, das heißt es sind Messfehler durch Staubpartikel, stark lichtabsorbierende oder reflektierende Oberflächen enthalten. Um diese Messfehler zu eliminieren, werden Filter eingesetzt:

- Intensität: Ist die Intensität des reflektierten Laserlichtes zu gering, so wird keine verlässliche Entfernung des Oberflächenpunktes übermittelt. Punkte, deren Intensität unter einer vorzugebenden Schwelle liegen, werden nicht weiter betrachtet.
- Ausreißer: Einzelne Messpunkte, die einen großen z-Abstand zu den beiden benachbarten Punkten haben, werden als Ausreißer, verursacht etwa durch Staub, gelöscht.
- Abstand der Punkte zum Rand: Punkte am Rand der Scanlinie lassen in ihrer Qualität häufig nach. Über einen Parameter wird definiert, ob Randpunkte verwendet werden.
- Abstand der Punkte zueinander: Um die Datenmenge zu reduzieren, kann ein Mindestabstand definiert werden.

## 6.3 Scanlinienbasierte Nahterkennung

Die auf einer einzelnen Scanlinie basierende Nahterkennung ist der einfachste Fall der Betrachtung. Aus dieser Scanlinie können unter entsprechenden Vorbedingungen sowohl das Zentrum der aufzubringenden Schweißnaht, als auch die Werkstückebenen extrahiert werden. Unter Einbeziehung der Auswertungen der vorherigen Scanlinien lässt sich daraus die

Position und Orientierung des Roboters berechnen. Dieses Vorgehen ist die Basis der Auswertelgorithmen bei der Nahtverfolgung.

Im Gegensatz zu den verfügbaren Nahtverfolgungssensoren, die eine konventionell programmierte Bahn anpassen, werden die Sensordaten hier zur Verbesserung der Qualität des Programmierens durch Vormachen genutzt.

Der Ablauf der scanlinienbasierten Nahterkennung ist in Bild 6.2 dargestellt. Der Roboter wird mit aktiviertem Sensor grob über die Bahn geführt, dabei wird die Werkstückgeometrie als Folge von Höhenprofilen aufgenommen. Jedes Höhenprofil wird dabei zusammen mit der entsprechenden Roboterpose gespeichert. Die Auswertung erfolgt im Anschluss, nach der Filterung werden Kantenpunkte in den Scanlinien ermittelt, hier wie in Abschnitt 5.2.2 über das Krümmungskriterium. Der entsprechende Parameter wird adaptiv über eine Histogramm- auswertung ermittelt. Die verbleibenden Kantenpunkte werden an Hand heuristischer Annahmen weiter ausgewertet. Als letzter Punkt wird aus den ermittelten Kantenpunkten eine Sollposition des Roboters erzeugt.

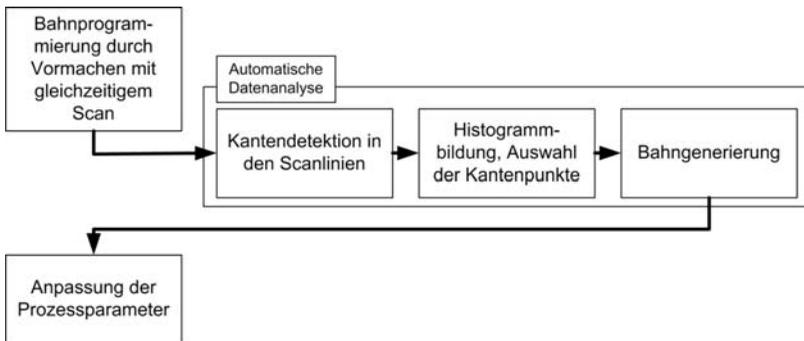


Bild 6.2: Ablauf der scanlinienbasierten Bahnadaption mittels Sensordaten.

Bei der Ermittlung der Sollposition des Roboters ist auf die Stetigkeit der Roboterbahn, also das Zusammenpassen der einzelnen aus den Höhenprofilen ermittelten Positionen zu achten. Ziel des Algorithmus ist es, die Schweißspitze sowohl für I-Nähte als auch für Kehlnähte korrekt zu positionieren.

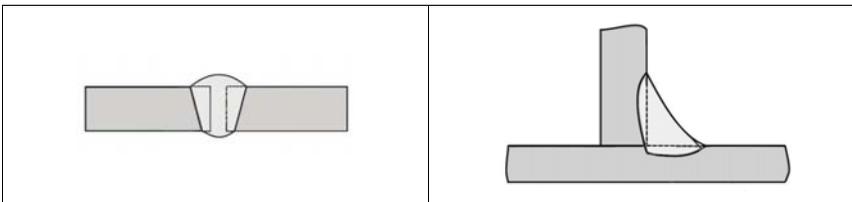


Bild 6.3: I-Naht zur Verbindung aneinanderstoßender Bleche (links), Kehlnaht zur Verbindung senkrecht aufeinander stehender Bleche (rechts). Quelle: (Fischer 1992).

### 6.3.1 Kantendetektion an Hand des Krümmungskriteriums

Ziel der Kantendetektion ist es, die für die Erkennung der I- oder Kehlnähte notwendigen Kanten zu detektieren. Im Falle der I-Naht sind das zwei bis vier Kanten, im Falle der Kehlnaht ist eine Kante zu detektieren.

Die gefilterten Scannerdaten werden dazu nach ihrer Krümmung ausgewertet. Es wird, wie in Abschnitt 5.2.2 dargestellt, ein Kreis in jeweils drei benachbarte Punkte gelegt, die Krümmung im mittleren Punkt wird als das Inverse des Radius angenommen. Es werden linear interpolierte Punkte für die Berechnung verwendet, um Äquidistanz zu gewährleisten.

Eine resultierende Krümmungskurve für eine I-Naht ist in Bild 6.4 dargestellt. Die Krümmung verläuft um die Nulllinie herum, nur für vier Punkte finden sich größere Ausschläge. Diese vier Punkte sind die Kanten der Naht, jeweils zwei an Ober- und Unterseite der Bleche.

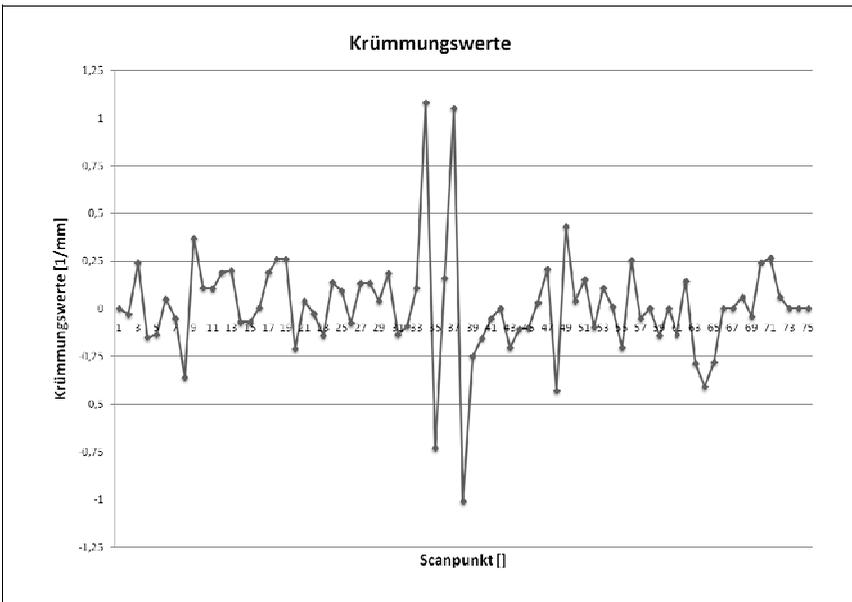


Bild 6.4: Verteilung der Krümmungswerte für eine I-Naht mit interpolierten Scanpunkten.

Im allgemeinen Fall ist die Krümmungsverteilung weniger eindeutig. Daher ist es notwendig, den Schwellenwert zur Erkennung eines Kantenpunktes adaptiv festzulegen. Dies wird über eine Histogrammauswertung vorgenommen.

Die Krümmungswerte werden dazu in diskrete Klassen eingeteilt. Die besten Ergebnisse wurden dabei mit einer Diskretisierung in 0,05-er Schritten erzielt, dann ergibt sich die Anzahl der Klassen aus:

$$n_b = \frac{\kappa_{\max}}{0,05} \quad (6.1)$$

Ein solches Histogramm ist in Bild 6.5 dargestellt. Es zeigt viele Punkte mit geringer Krümmung, dann eine Serie leerer Klassen und im Anschluss 7 Punkte mit hoher Krümmung.

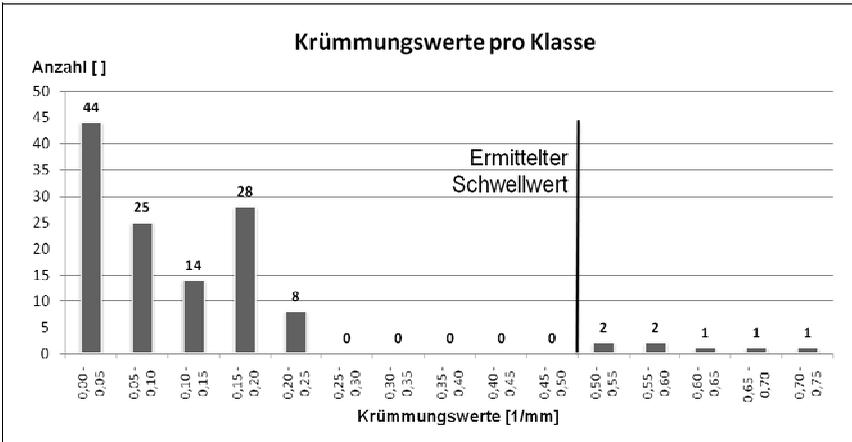


Bild 6.5: Histogramm der Krümmungswerte zur adaptiven Definition des Schwellwertes der Klassifikation

Dieses Histogramm zeigt den typischen Verlauf einer Krümmung. Der Schwellwert ist in der (obersten) Nullserie festzulegen. Nur dann ergibt sich eine klare Trennung zwischen Kanten- und Flächenpunkten. Dieser entwickelte Algorithmus adaptiert das Prinzip der Krümmungsuntersuchung durch die Interpolation und adaptive Schwellwertbildung die Anforderung der vollständigen Automatisierbarkeit.

### 6.3.2 Auswahl der Kantenpunkte an Hand heuristischer Kriterien

Das in Bild 6.5 dargestellte Histogramm liefert 7 Kantenpunkte in der untersuchten Scanlinie einer Kehlnaht. Bild 6.6 zeigt die Darstellung einer solchen Kehlnaht mit weiteren Kantenpunkten. Für die weitere Bearbeitung muss die relevante Kante sowie die Orientierung der anschließenden Ebenen gefunden werden. Dazu werden heuristische Verfahren verwendet.

In einem ersten Schritt werden Ausreißer entfernt. Einzelne als Kante erkannte Punkte werden durch den Vergleich mit den davor und dahinter liegenden Profilen als Ausreißer erkannt und nicht weiter ausgewertet.

Im Anschluss werden die Kantenpunkte eliminiert, deren Abstand zur Schweißspitze zu groß ist, es wird also die Annahme ausgewertet, dass der Bediener den Roboter mit der Schweißspitze nahe über die zu schweißende Naht geführt hat.

Im Anschluss an diese Auswertung bleiben nur noch die gewünschten ein bis vier Kantenpunkte übrig, die eine Kehl- bzw. I-Naht definieren.

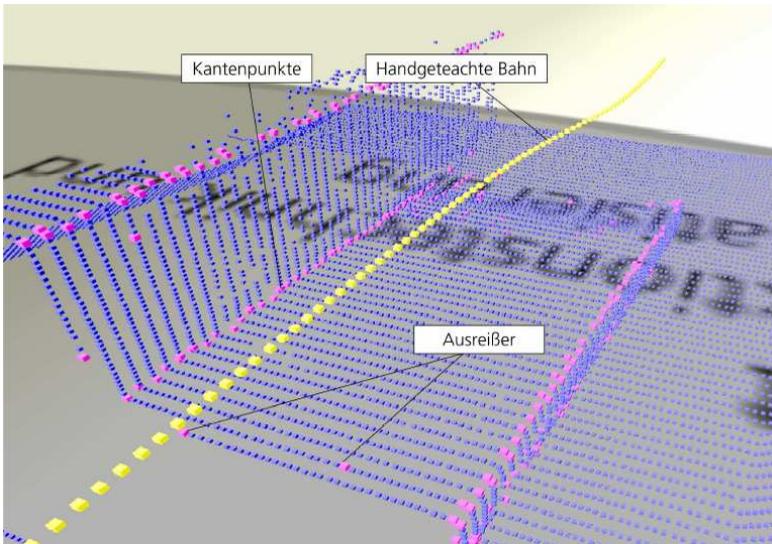


Bild 6.6: Auswahl der Kantenpunkte einer Kehlnaht.

### 6.3.3 Erzeugung einer Bahn

Aus der Bestimmung der Kantenpunkte muss eine Schweißrohr- und damit Roboterposition erzeugt werden, die in der gewünschten Orientierung und etwas oberhalb der Naht entlang führt. Ein Algorithmus für I-Nähte wird dargestellt, ebenso die Nutzung dieses Algorithmus auch für Kehlnähte.

Bild 6.7 stellt die notwendigen Informationen dar. Die zu erzeugende Roboterposition liegt senkrecht über der Mitte der Naht.

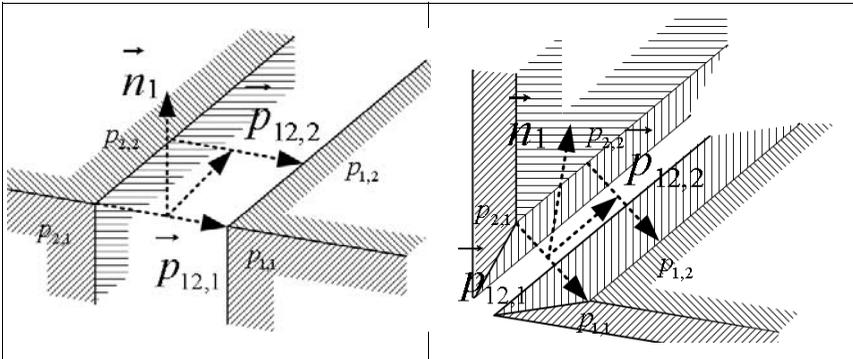


Bild 6.7: Erzeugung einer Roboterbahn aus den Kanteninformationen einer I-Naht (links), Übertragung der Situation einer I-Naht auf eine Kehlnaht (rechts).

Formal ergibt sich die Bahn aus den folgenden Überlegungen:

$$\vec{p}_{12,1} = p_{1,1} - p_{2,1} \quad \text{und} \quad \vec{p}_{12,2} = p_{1,2} - p_{2,2} \quad (6.2)$$

bezeichnen zwei Verbindungsvektoren der ermittelten Kanten. Mittels Vektoraddition und Kreuzprodukt ergibt sich die Normale  $\vec{n}_1$ . Die Sollposition der Schweißspitze ergibt sich damit aus

$$\vec{p}_{TCP\_neu} = p_{2,1} + \frac{1}{2}\vec{p}_{12,1} + \vec{n}_1 \cdot \frac{d}{|\vec{n}_1|} \quad (6.3)$$

Der Parameter  $d$  definiert den Abstand der Schweißspitze zur Oberfläche.

Die Ermittlung der Bahnposition einer Kehlnaht erfolgt analog, aus den Informationen der Kehlnaht wird eine quasi-I-Naht erzeugt.

### 6.3.4 Test und Verifikation

Der entwickelte Algorithmus wurde an verschiedenen Testbauteilen verifiziert. Bild 6.8 zeigt die Auswertung einer Kehlnaht, Bild 6.9 die Auswertung einer I-Naht. Aus beiden Auswertungen wurden Roboterprogramme erzeugt.

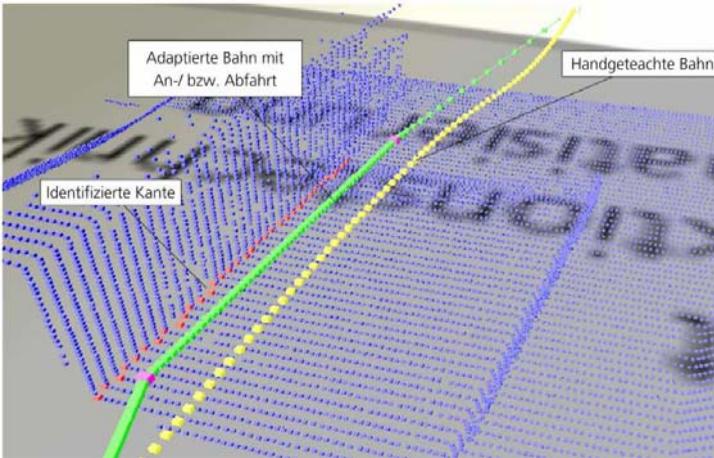


Bild 6.8: Auswertung einer Kehlnaht

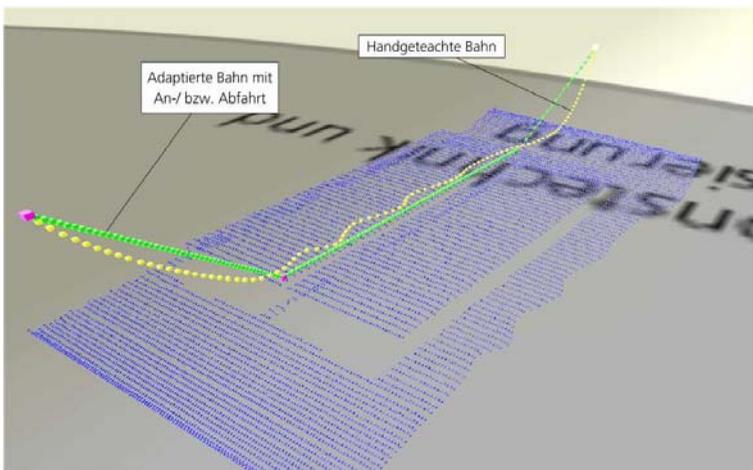


Bild 6.9: Auswertung einer I-Naht

Die Vorteile des Verfahrens der scanlinienbasierten Nahterkennung sind:

- Vollautomatische Sensordatenauswertung und Generierung einer präzisen Roboterbahn aus der grob vorgemachten Bahn.
- Geringe Komplexität durch Einschränkung auf lokale Auswertung

Die Nachteile des Verfahrens sind mit den Vorteilen eng gekoppelt:

- Die Datenaufnahme ist nur in Kombination mit der Programmierung möglich, ein späteres Abscannen ist nicht zulässig.
- Der Scanner muss immer annähernd senkrecht zur Naht orientiert sein. Das Abfahren von komplexeren Bauteilen ist schwierig, das sensorunterstützte Schweißen einer geschlossenen Kehlnaht mit Orientierungswechsel beispielsweise ist nicht möglich.

Zusammenfassend ist das Verfahren sehr gut geeignet für Bauteile mit einfachen, langen und überwiegend geraden Nähten.

Für komplexere Bauteile ist die Verwendung des im nächsten Kapitel dargestellten Verfahrens der scanlinienbasierten Flächenrückführung sinnvoll.

## 6.4 Scanlinienbasierte Flächenrückführung

Einen allgemeingültigeren Ansatz als im vorherigen Abschnitt beschrieben verfolgt die scanlinienbasierte Flächenrückführung. Der in diesem Abschnitt entwickelte Algorithmus erzeugt Flächen aus den mit dem Scanner aufgenommenen lokalen Werkstückgeometrien. Auf der Basis dieser Flächen können Schnittgeraden ermittelt und die manuell programmierte Bahn angepasst werden.

Bild 6.10 stellt den Ablauf der scanlinienbasierten Flächenrückführung dar. Zuerst wird die Schweißbahn manuell programmiert, im Anschluss, also auch zeitversetzt oder in anderer Orientierung, wird das Bauteil gescannt. Als erster Auswertungsschritt werden wie in 6.3.1 die Kanten auf den einzelnen Scanlinien ermittelt. Dazu kann der Douglas-Peucker-Algorithmus oder die krümmungsbasierte Kantendetektion verwendet werden.

Im Anschluss findet auf den zwischen den Kanten definierten Geraden ein Region-Growth-Verfahren statt, das zu Flächen verschiedener Größe führt.

Diese Flächen können interaktiv ausgewählt werden, die dann berechnete Schnittgerade wird zur Optimierung der manuell programmierten Bahn verwendet.

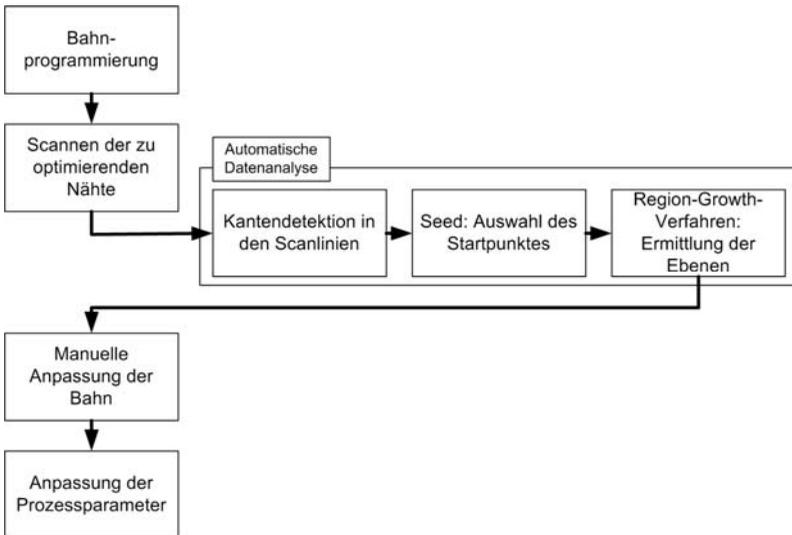


Bild 6.10: Ablauf der Bahnadaption mittels bei der scanlinienbasierten Flächenrückführung.

Bild 6.11 zeigt die für die Entwicklung verwendeten Testdaten: Einfache, generierte Daten mit einem weißen Rauschen auf der linken Seite, den Scan eines Gallerieträgers auf der rechten Seite. Der Gallerieträger besitzt eine für das Anwendungsspektrum mittlere Komplexität.

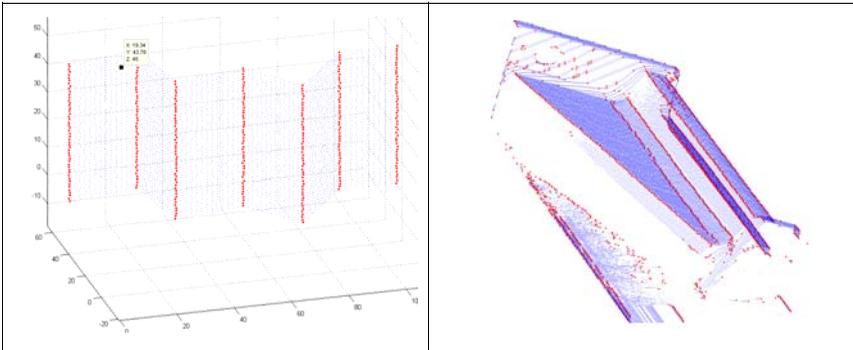


Bild 6.11: Generierte Testdaten mit leichtem Rauschen (links), Scandaten des Gallerieträgers (rechts), jeweils mit dunkler markierten Kantenpunkten.

### 6.4.1 Segmentierung mittels Region-Growth-Verfahren

Der Region-Growth-Ansatz beruht darauf, eine Menge von Geraden daraufhin zu untersuchen, inwieweit sie Flächen bilden. Es wird ein spezialisierter Ansatz nach (Jiang 1994) als Grundlage genommen. Dieser Ansatz geht davon aus, dass der Scandatensatz aus ebenen Teilregionen besteht. Insbesondere für den Anwendungsfall des MAG-Schweißens bei KMU ist diese Annahme häufig gültig. Wenn notwendig, könnten aus den Ebenen komplexere geometrische Objekte ermittelt werden.

Der erste Schritt ist die Vorverarbeitung der Scandaten, um Ausreißer zu eliminieren und Rauschen zu erfassen. Im zweiten Schritt werden die Scanlinien in Geradenzüge gewandelt. Ein Startpunkt (Seed-Element) für das Region-Growth wird ausgewählt, der günstige Ausgangsbedingungen besitzt. In der Folge wird untersucht, ob die benachbarten Linienelemente mit diesem Seed-Element eine Ebene bilden.

Durch das höhere Abstraktionsniveau mit Linienzügen im Gegensatz zu Einzelpunkten, ist der Ansatz von Jiang sehr effizient. Bild 6.12 zeigt das Flussdiagramm der Basisversion des Algorithmus.

In dieser Arbeit steht die Integration des Algorithmus in die funktionale Schweißroboterzelle im Vordergrund. Daher ist eine Frage, was an dem Algorithmus bei der Transformation von 2,5-D-Scandaten auf 3-D-Scandaten geändert werden muss. In der Anwendung werden mehrere Scans mit unterschiedlicher Orientierung aufgenommen, wodurch sich echte 3-D-Daten ergeben können.

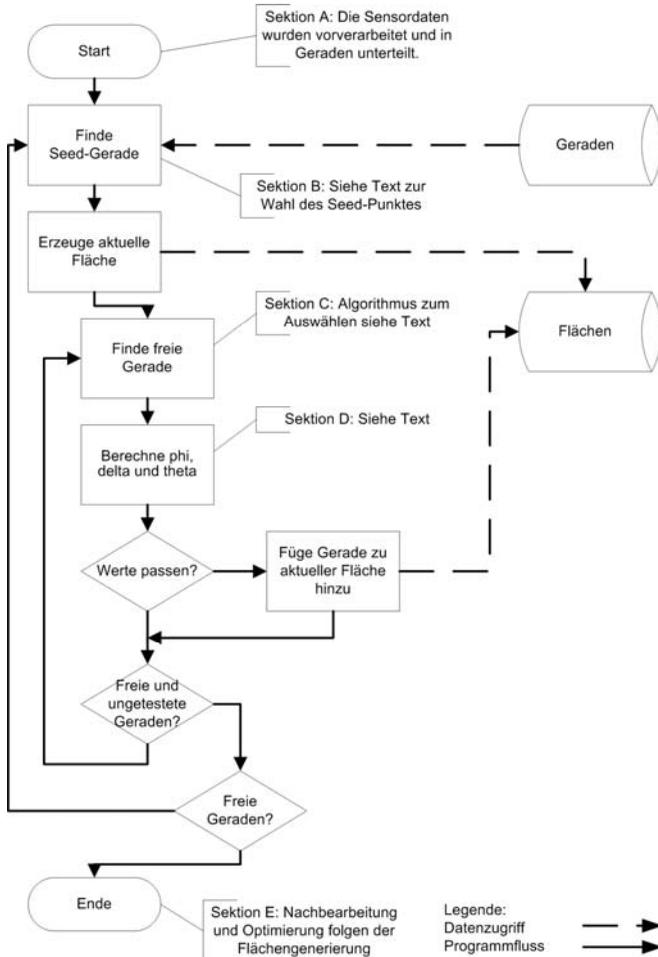


Bild 6.12: Flussdiagramm der Auswertungsschritte des Region-Growth-Algorithmus.

Der in Bild 6.12 in Form eines Flussdiagramms dargestellte Algorithmus arbeitet auf der Basis von Geraden. Die mit dem Laser-Triangulations-Sensors aufgenommenen Scanlinien müssen also vor Beginn des Algorithmus gefiltert und in Linienzüge umgewandelt werden. Dies geschieht mit dem in 5.2.2 vorgestellten Douglas-Peucker-Algorithmus, alternativ über das ebenfalls vorgestellte Krümmungskriterium. Der zweite, nicht-triviale Programmpunkt des Algorithmus (Sektion B) besteht in der Auswahl eines Seed-Punktes. Hier muss eine noch nicht in eine Fläche integrierte Gerade gefunden werden, die als Startpunkt für die nächste Fläche verwendet wird. Die Auswahl dieser Seed-Geraden hat sowohl Auswirkungen auf die Laufzeit des Verfahrens als auch auf die Qualität der erzeugten Flächenmengen. Der ein-

fachste Ansatz ist, eine zufällig ausgewählte Gerade zu verwenden, weitere Optimierungen sind möglich. Im Anschluss an die Wahl der Seed-Gerade  $g_s$  gilt es, die weiteren zur Fläche gehörigen Geraden zu ermitteln, dies wird durch das Prüfen verschiedener Parameter durchgeführt. Die zu prüfende Gerade  $g_p$  wird in Sektion C ausgewählt. Hier ist der rein zufällige Ansatz nicht sinnvoll, da zwei Geraden nur Teil einer Fläche sein können, wenn alle Geraden zwischen ihnen ebenfalls zu dieser Fläche gehören. Die Auswahl kann also, je nach Implementierung der Datenstruktur, über die Indizierung erfolgen (plus 1 und minus 1) oder aber bei deutlich höherem Rechenaufwand über das Merkmal des minimalen Abstandes.

Wenn die beiden zu testenden Geraden gefunden wurden, werden in Sektion D die zwei wesentlichen für die Zugehörigkeit relevanten Parameter Winkelabweichung  $\varphi$  und Abstand  $\delta$  berechnet. Auf der Basis der beiden Geradengleichungen

$$g_s(t) = \vec{r}_1 t + \vec{a}_1 \quad \text{und} \quad g_p(t) = \vec{r}_2 t + \vec{a}_2 \quad . \quad (6.4)$$

berechnen sie sich als

$$\varphi = \cos^{-1} \frac{\vec{r}_1 \cdot \vec{r}_2}{|\vec{r}_1| |\vec{r}_2|} \quad \text{und} \quad \delta = \frac{|(\vec{a}_1 - \vec{a}_2) \cdot (\vec{r}_1 - \vec{r}_2)|}{|\vec{r}_1 \times \vec{r}_2|} . \quad (6.5)$$

Dabei wird der kleinste Abstand der beiden Geraden verwendet.

Dem Sonderfall von zwei Geraden, die nach den Kriterien Richtungsvektor und Abstand einer Ebene zugehörig sind, aber so zueinander verschoben sind, dass sie zwei Ebenen bilden, wird über eine Untersuchung der Verschiebung Rechnung getragen.

Aus den von der Kantendetektion gelieferten Anfangs- und Endpunkte  $\vec{p}_\alpha$  und  $\vec{p}_\omega$  berechnen sich die Mittelpunkte der Geraden  $\vec{p}_m$  und die Längen der Geraden  $l$ :

$$l_s = |\vec{p}_{\omega s} - \vec{p}_{\alpha s}|, \quad l_p = |\vec{p}_{\omega p} - \vec{p}_{\alpha p}| \quad (6.6)$$

$$\vec{p}_{ms} = \vec{p}_{\alpha s} + \frac{\vec{p}_{\omega s} - \vec{p}_{\alpha s}}{2}, \quad \vec{p}_{mp} = \vec{p}_{\alpha p} + \frac{\vec{p}_{\omega p} - \vec{p}_{\alpha p}}{2}$$

Wenn für den Abstand der Mittelpunkte  $\Delta_M = |\vec{p}_{mp} - \vec{p}_{ms}|$  gilt

$$\Delta_M < \frac{l_s + l_p}{2}, \quad (6.7)$$

so befinden sich die beiden Geraden nicht in Überdeckung. Sie werden dann nicht als einer gemeinsamen Fläche zugehörig gekennzeichnet.

Wenn sich die berechneten Parameter innerhalb der vorgegebenen Toleranzen befinden, wird die untersuchte Gerade der aktuellen Fläche zugeordnet, andernfalls wird sie für diese

Fläche verworfen. Im Anschluss iteriert der Algorithmus, bis alle Geraden einer Fläche zugeordnet sind.

Im Anschluss findet in Sektion D des Flussdiagrammes in Bild 6.12 eine entsprechende Aufbereitung der Daten statt. In einem ersten Schritt werden alle Flächen, die nur eine zu kleine Anzahl von Geraden auf sich abbilden konnten, entfernt. Dies sind oft Flächen mit nur einer Geraden, die zur Terminierung des Algorithmus notwendig sind. Um aus allen Geraden eine einheitliche Fläche mit exakt definiertem Aufsetz- und Normalenvektor zu bekommen, wird ein Least-Squares-Verfahren verwendet, das auf den Anfangs- und Endpunkten der zugeordneten Geraden arbeitet, siehe diverse Programm-Bibliotheken oder (Press 1992). Um ein realistisches und trotzdem leicht darstellbares Polygon der Fläche zu bekommen, werden alle zur späteren Darstellung genutzten Randpunkte entlang des Normalenvektors auf die Fläche gemappt. Bild 6.13 zeigt die Ergebnisse für die beiden Testdatensätze.

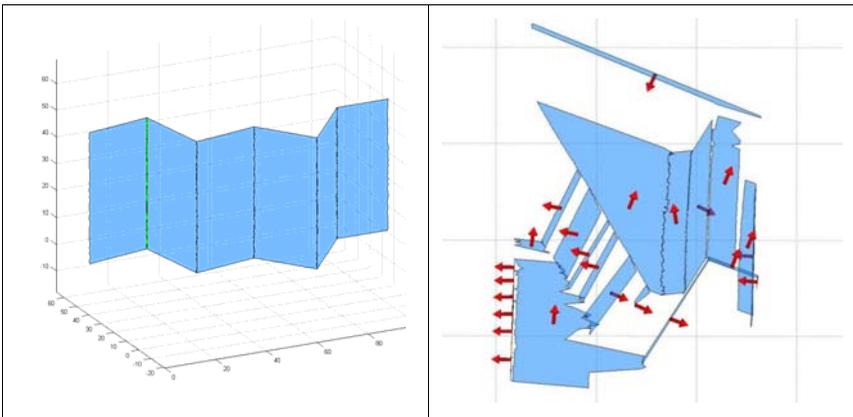


Bild 6.13: Segmentierte Flächen der Testdatensätze

### 6.4.2 Optimierung des Region-Growth-Verfahren

In dem hier beschriebenen Szenario ist es möglich, Punkte des Werkstücks mehrmals mit unterschiedlichen Orientierungen zu scannen. Mit diesen Scandaten und dem hier verfolgten Verfahren werden die entsprechenden Flächen erzeugt, allerdings auch mehrfach, sofern keine weiteren Vorkehrungen getroffen werden. Da in einer Fläche nur Geraden mit einem ähnlichen Richtungsvektor subsumiert werden, kann eine reale Fläche durch zwei oder mehr Geradenmengen mit unterschiedlichen Richtungsvektoren, also aus unterschiedlichen Scans, repräsentiert werden. Dies kann zu Schwierigkeiten bei der Interaktion führen, ebenso zu Ungenauigkeiten in der Bahnplanung.

Abhilfe kann hier die Untersuchung der Abbild-Flächen auf identische reale Flächen schaffen. Wenn dieselbe reale Fläche als Vorbild nachgewiesen werden kann, kann eine gemeinsame Fläche auf der Basis der Anfangs- und Endpunkte aller Geraden erstellt werden.

Die Bedingungen hierfür sind die Identität der Ebenen und die Überschneidung der durch die Randpunkte der Geraden definierten Flächen, siehe Bild 6.14.

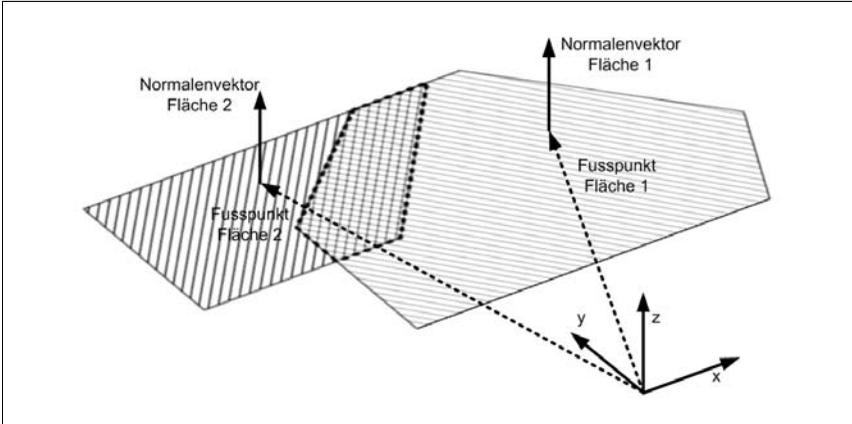


Bild 6.14: Flächenrepräsentationen derselben realen Fläche mit unterschiedlichen Richtungsvektoren der Basisgeraden.

Es gelten drei Bedingungen, die Flächen sind in Koordinatenform

$E: n_1x_1 + n_2x_2 + n_3x_3 = y$  gegeben:

$$\cos^{-1} \frac{\vec{n}_1 \cdot \vec{n}_2}{|\vec{n}_1| \cdot |\vec{n}_2|} < \varepsilon_\varphi \quad (6.8)$$

für die Identität der Normalenvektoren,

$$\frac{n_1x_1 + n_2x_2 + n_3x_3 - y}{\sqrt{n_1^2 + n_2^2 + n_3^2}} < \varepsilon_\Delta \quad (6.9)$$

für die identische Lage der Ebenen über die Hessesche Normalform, und

$$\exists \vec{p}_i \in \mathfrak{F}_2, \vec{p}_i \in \mathfrak{F}_1 \quad (6.10)$$

für die Überschneidung der Flächen. Die letzte Bedingung kann mit Methoden der Computergrafik nachgewiesen werden, indem die Randpunkte  $\vec{p}_i$  der Fläche  $\mathfrak{F}_1$  über die Kriterien der Umlaufzahl oder der Kreuzungszahl auf die Zugehörigkeit zu Fläche  $\mathfrak{F}_2$  getestet werden. Beide Algorithmen sind rechenintensiv.

### 6.4.3 Ermittlung der Schnittgeraden, Interaktion

Auf Grund der möglicherweise großen Anzahl von ermittelten Flächen findet eine manuelle Adaption der programmierten Bahnen an die Werkstückgeometrie statt. Entsprechend den Anforderungen an die Interaktion mit dem Robotersystem wird nur eine begrenzte Anzahl von Vorgehensweisen umgesetzt. Bild 6.15 stellt beispielhaft die geometrischen Zusammenhänge der Anpassung eines Eckpunktes dar.

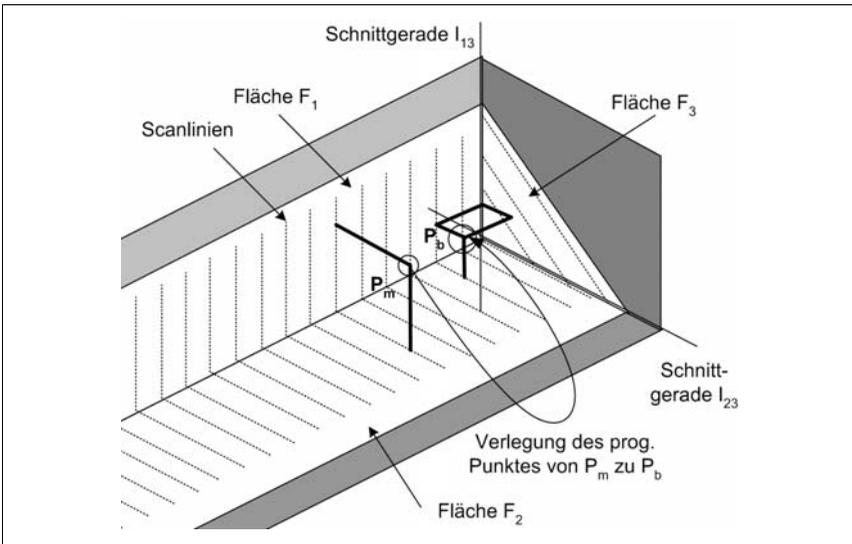


Bild 6.15: Anpassung eines per Hand programmierten Punktes an die ermittelten Geometriedaten des Werkstücks

Mit den gegebenen Daten, den ermittelten Flächen der Werkstückgeometrie einerseits und den manuell programmierten Bahnpunkten andererseits lassen sich im Wesentlichen vier Vorgehensweisen definieren:

- Zentrierung an Gerade  
Aus zwei Flächen wird ihre Schnittgerade berechnet, der programmierte Punkt wird entlang des zur Schnittgerade senkrechten Verbindungsvektors angenähert.
- Verschiebung entlang Gerade  
Aus zwei Flächen wird ihre Schnittgerade berechnet, der programmierte Punkt wird entlang dieser Geraden verschoben.
- Zentrierung an Eckpunkt  
Aus drei Ecken wird ein Schnittpunkt berechnet, der programmierte Punkt wird zur Winkelhalbierenden hin verschoben.

- Verschiebung zum Eckpunkt  
Aus drei Ecken wird ein Schnittpunkt berechnet, der programmierte Punkt wird zum Schnittpunkt hin verschoben.

Um hier die intuitive Interaktion weiterhin zu gewährleisten, wird nur ein Werkzeug weiter entwickelt, die Anpassung der programmierten Punkte mit definiertem Abstand. Hierzu werden zwei oder drei Ebenen in einer 3-D-Oberfläche ausgewählt, der Bahnpunkt wird so in Richtung Eckpunkt verschoben, dass die in der Oberfläche definierten Distanzen eingehalten werden.

#### 6.4.4 Reduzierung des geometrischen Fehlers

Die Algorithmen zur Flächenrückführung ermitteln zuverlässig relevante Lösungen. Weiter zu untersuchen sind Laufzeit (siehe 6.4.5) und Fehlerverhalten des Algorithmus. Die Fehlerbetrachtung und -optimierung ist notwendig, um die Toleranzen des Prozesses einhalten zu können.

Für das Fehlerverhalten relevante Einflüsse sind:

- die Genauigkeit des Roboters,
- die Referenzierung des Sensorsystems,
- Auflösung und Linearität des Sensors sowie die
- Toleranzen der Algorithmen.

Der durch den Roboter induzierte Fehler wird als zwischen Absolut- und Wiederholgenauigkeit liegend angenommen. Da eine Bewegung des Roboters basierend auf kartesischen Koordinaten errechnet wird, kann die Wiederholgenauigkeit des Roboters nicht erreicht werden. Andererseits werden die Berechnungen im kartesischen Raum nur lokal ausgeführt, nicht über den gesamten Arbeitsraum. Daher wird ein Fehler von 0,1 mm bis 0,3 mm angenommen.

Die Referenzierung des Sensorsystems wird an Hand der CAD-Daten des Griffes vorgenommen und durch Messungen eines mit dem Roboter eingemessenen Referenzkörpers optimiert. Im Anschluss stimmen die Sensordaten bis auf 0,5 mm mit den Referenzdaten des Roboters überein.

Die weiteren Toleranzen sind:

Auflösung Sensordaten:	0,15 mm
Formfehler Geraden:	$< 2 \text{ Tol}_{DP}$ , durch Versuche als 0,4 mm abgeschätzt
Formfehler Ebenen:	$< 2$ Formfehler Geraden, also $< 0,8$ mm

Die bei der Ermittlung der Kantenpunkte und beim Ebenenfitting auftretenden Fehler sind in Bild 6.16 veranschaulicht.

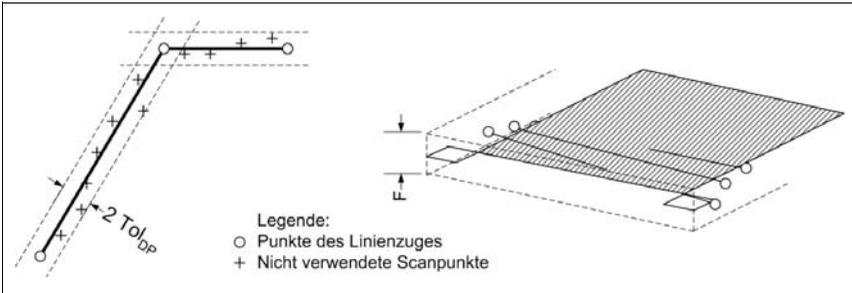


Bild 6.16: Worst-Case-Szenarien zur Fehlerbetrachtung, Douglas-Peucker-Algorithmus (links) und Ebenenfitting (rechts)

Der Fehler, der durch die Algorithmen der Kantendetektion auftritt, ist durch die entsprechenden Parameter beeinflussbar. Werden die Parameter allerdings zu gering gewählt, ist keine Kompression spürbar, was sich durch eine überhöhte Anzahl von resultierenden Flächen auswirkt. Der oben angegebene Wert von 0,4 mm ist der durch Versuche optimierte Toleranzparameter.

In der aktuellen Implementierung ist also ein Worst-Case-Fehler von bis zu 2,3 mm denkbar. Dieser Fehler trat in den durchgeführten Versuchen nicht auf.

Eine Optimierung des möglichen Fehlers ist durch die folgenden Ansätze möglich:

- Verringerung des Roboterfehlers durch den Einsatz eines absolutvermessenen Roboters.
- Einsatz eines Sensors mit höherer Auflösung
- Nutzung des Douglas-Peucker-Algorithmus nur zur Segmentierung des Scanline, Erstellung der Teillinien in den Segmenten durch einen Least-Squares-Algorithmus. Der Fehler ist dann in der Größenordnung der Sensorauflösung zu erwarten.
- Erstellung der Ebenen auf der Basis aller Scanpunkte und mittels eines Least-Squares-Algorithmus. Der Fehler ist dann in der Größenordnung der Sensorauflösung zu erwarten.

Durch diese Schritte wäre ein Worst-Case-Gesamtfehler  $< 1$  mm erreichbar. Dieser Wert wäre ausreichend, insbesondere in den hier relevanten Anwendungen.

### 6.4.5 Reduzierung der Laufzeit

Der Auswertalgorithmus zeigt bei der in Bild 6.12 entworfenen Basisversion das Verhalten  $O(n^2)$  als Best-Case und Worst-Case (Sperschneider 2002). Diese quadratische Laufzeit ist nicht optimal, da insbesondere bei größeren Scans einige Hunderttausend Punkte ausgewertet werden müssen. Auf Standardrechnern liegt die Laufzeit aber im nicht störenden Bereich bis zu einigen Sekunden. Eine Reduktion um einen Faktor wäre also ausreichend, es ist kein vollkommen anderer Ansatz notwendig.

Ansätze zur Optimierung der Laufzeit sind:

- Intelligentes Setzen der Seed-Punkte
- Abschätzung der maximal möglichen Anzahl der Punkte je Ebene, Abbruch bei zu kleiner Anzahl.

### 6.4.6 Use-Case- und Klassen-Entwurf

Die Algorithmen zur scanlinienbasierten Flächenrückführung wurden unter MatLab entworfen und getestet, müssen aber in die grafische Oberfläche der Schweißroboterzelle integriert werden. Der relevante Teil der Algorithmik wird mit UML modelliert und mittels eines CASE-Tools in Code umgesetzt.

Basis für die Umsetzung ist das Klassendiagramm in Bild 6.17. Für die mathematischen Funktionen wurde die MatLab C++ Bibliothek verwendet. Diese ist Teil der Math Library Toolbox und implementiert die bisher verwendeten Algorithmen.

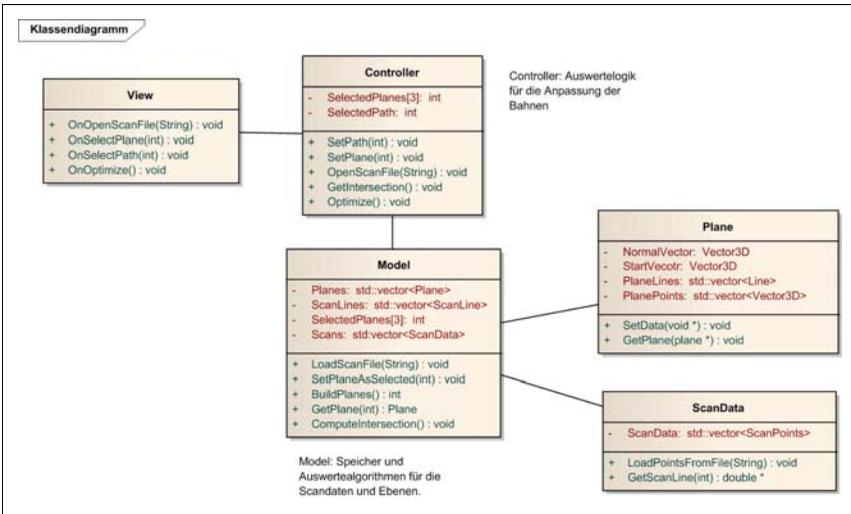


Bild 6.17: Klassendiagramm zur Bahnadaption basierend auf Flächendetektion.

Die beiden wesentlichen Use-Cases sind das Laden einer neuen Datei mit Scandaten, woraufhin die Auswertung getriggert wird, und die Interaktion, also das Auswählen von Ebenen und die folgende Anpassung von Bahnpunkten. Sie sind in Bild 6.18 dargestellt.

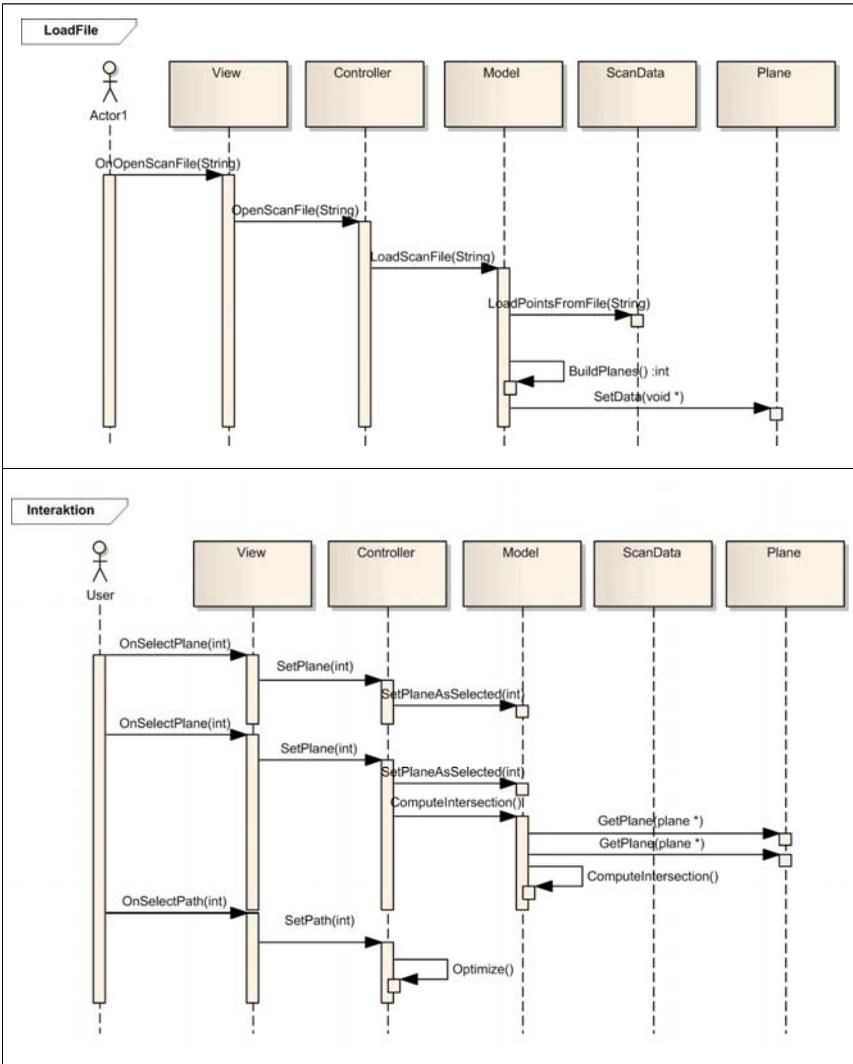


Bild 6.18: Use-Case-Diagramme „Laden Datei“ und „Interaktion“

# 7 Realisierung und Erprobung des Programmierverfahrens

## 7.1 Gesamtaufbau der Versuchszelle

Bild 7.1 zeigt den Aufbau der zur Erprobung verwendeten Schweißroboterzelle am Fraunhofer IPA. Die Hauptkomponenten sind in Tabelle 7.1 aufgeführt.

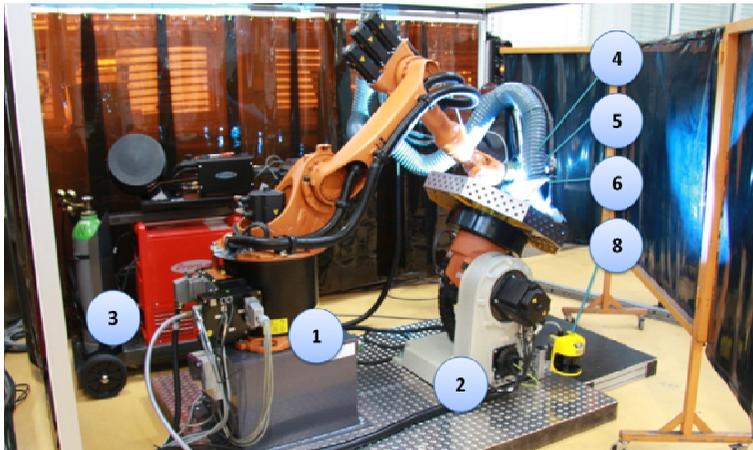


Bild 7.1: Übersichtsbild der realisierten Schweißroboterzelle

Nr.	Komponente	Kurzbeschreibung
1	Industrieroboter	KUKA KR16, Steuerung KRC2
2	Dreh-Kipp-Tisch	KUKA DKP400, eingebunden als externe Achsen, gekoppelter Betrieb möglich
3	Schweißquelle	Fronius TP3200, verbunden mit der Robotersteuerung über RobInterface5000
4	Führungsriff	Handgriff zum Führen des Roboters, Zustimmschalter und Aufnahme Knopf integriert (siehe Bild 7.2)
5	Kraft-Momenten-Sensor	Aufnahme der Bewegungsvorgaben (siehe Bild 7.2)
6	Laser-Triangulations-Sensor	MEL M2DW 75/30, Möglichkeit zur Kühlung (siehe Bild 7.2)
7	Touchscreen	Beckhoff Touchscreen mit Industrie-PC zur Steuerung und Nachbearbeitung (siehe Bild 7.3)
8	Laserscanner	SICK S3000, Teil des Sicherheitssystems

Tabelle 7.1: Hauptkomponenten der Schweißroboterzelle

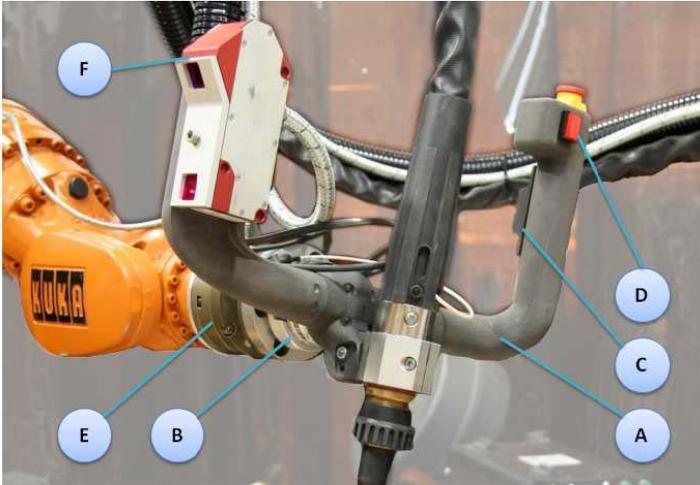


Bild 7.2: Werkzeug des Schweißroboters mit Führungsgriff (A), Kraft-Momenten-Sensor (B), Zustimmungsschalter (C), Aufnahme-Taster (D) und Kollisionsschutz (E), links ist der Laser-Triangulations-Sensor (F) an Stelle des zweiten Handgriffes angebracht.

## 7.2 Eingesetzte Teilsysteme und Komponenten

### 7.2.1 Industrieroboter und Dreh-Kipp-Tisch

Als Handlungssystem wird ein KUKA Industrieroboter vom Typ KR16 verwendet. Dieser Roboter besitzt ein Handhabungsgewicht von 16 kg bei einer maximalen Reichweite von 1611 mm. Zwei zusätzliche Achsen eines Dreh-Kipp-Tisches vom Typ KUKA DKP400 erlauben es, das Werkstück in Wannenlage vorzupositionieren.

Die Steuerung KRC2 kann KRL-Programme (KUKA Robot Language) ausführen und verfügt über eine Ethernet-Schnittstelle zum Datenaustausch im Interpolationstakt von 12 ms. Diese RSI-Ethernet-Schnittstelle ermöglicht die Handführung. KRL-Programme können über eine Client-Server-Architektur von einem externen Rechner hochgeladen und angewählt werden.

Die integrierte SafeRobot-Option erlaubt die Überwachung von achsweisen Sicherheitsbereichen und Maximalgeschwindigkeiten.

### 7.2.2 Schweißanlage

Der MAG-Schweißprozess wird von einer Fronius-Schweißquelle TransPuls Synergic 3200 mit Kühlgerät FK 4000-R FC, Drahtvorschub VR 1500 4R/F++ Roboter und Rohrbogen Robacta300 36° übernommen. Das Schweißsystem ist über die Schnittstelle RobInterface5000 an das Handhabungssystem angeschlossen. Das Roboterinterface bietet die Möglichkeit des Job-Betriebes, lässt aber auch die direkte Adaption der Prozessparameter zu. Ebenso liefert es Informationen über den Status des Schweißsystems.

### **7.2.3 Sensorsystem**

Zur Aufnahme der lokalen Werkstückgeometrie wird der Laserscanner M2DW 75/30 der Fa. MEL Mikroelektronik GmbH verwendet. Der Scanner projiziert eine Laserlinie, deren Höhenverlauf nach dem Triangulationsprinzip gemessen wird. Der Scanner ist für den Einsatz in Schweißrobotern konzipiert, er ist für Luft- oder Wasserkühlung vorbereitet und besitzt einfach austauschbare Schutzgläser.

Der Scanner besitzt einen Messbereich in Z-Richtung von 75 mm bei 0,15 mm Auflösung, in X-Richtung von 40 mm bei einer Auflösung von 0,05 mm. Die Scanfolge beträgt 50 Hz. Der Scanner wird per Ethernet in die Schweißroboterzelle integriert.

### **7.2.4 Mensch-Maschine-Schnittstelle**

Der Bediener muss in zwei Phasen der Programmierung mit dem Robotersystem interagieren, während des Führens und während der Nachbearbeitung. In beiden Fällen wurde darauf geachtet, die Interaktion möglichst intuitiv zu gestalten.

Das Führen des Roboters wird über ein haptisches Interface durch Kraftregelung vorgenommen. Der Bediener greift den Roboter am Führungsriff und zieht oder schiebt ihn in die entsprechende Richtung. Die von ihm aufgebrauchten Kräfte werden von dem Kraft-Momenten-Sensor gemessen und in Bewegungen umgesetzt.

Die während des Führens zusätzlich notwendige Interaktion kann über eine Sprachschnittstelle oder den Touchscreen vorgenommen werden. Die vorzunehmenden Einstellungen umfassen:

- Auswahl, Führung von Roboter oder Dreh-Kipp-Tisch
- Auswahl, 3-D oder 6-D Führung
- Auswahl des Override
- Speichern eines Punktes

Per Touchscreen sind die Einstellungen auf einer grafischen Bedienoberfläche vorzunehmen, siehe Bild 7.3, links. Die Nutzung der Sprachschnittstelle hat den Vorteil, dass der Bediener keine Hand vom Roboter lösen muss, der Arbeitsfluss wird nicht gestört. Das Sprachinterface basiert auf dem Sphinx-Erkennen der Carnegie-Mellon-University und wird über eine Ethernet-Schnittstelle eingebunden. Detaillierte Informationen über das System und die weiteren Arbeiten, insbesondere zur Nutzung in industriellen Umgebungen, finden sich unter (Hollmann 2008).



Bild 7.3: Interaktion mittels Touchscreen (links) und Sprachinterface mittels Kehlkopfmikrofon (rechts)

Das Speichern eines Punktes kann per Sprachkommando oder über einen Taster auf dem Führungsriff erfolgen.

Für die Nachbearbeitung der aufgenommenen Bahnen kann der Touchscreen oder eine Standard-Büroumgebung, also Maus und Tastatur, verwendet werden. Die Software, näher beschrieben in Abschnitt 7.4, ist auf eine Bedienung mit minimaler Texteingabe hin optimiert.

Wesentlich für die Umsetzung der hier beschriebenen Interaktionsmöglichkeiten ist ein entsprechendes Sicherheitskonzept. Das in Abschnitt 4.2.3 beschriebene Sicherheitskonzept wurde dazu umgesetzt. Für den Transfer in die jeweilige Anwendungsumgebung sind eine Überarbeitung der Risikoabschätzung und die Anpassung des Sicherheitskonzeptes notwendig. Für einen langfristigen Einsatz, etwa in einem Stahlbauunternehmen, ist die sehr offene Konzeption wie hier beschrieben nicht sinnvoll; eine Umhausung und die Nutzung staubresistenter Sensoren sind dann vorzusehen.

Für zukünftige Anwendungen ist auch in Betracht zu ziehen, inwieweit Arbeiten zur Kollisionsfolgenabschätzung für das Sicherheitskonzept relevant werden, dargestellt etwa in (Oberer-Treitz 2008).

## 7.3 Steuerungskomponenten

### 7.3.1 Software-Architektur

Um die in den vorherigen Abschnitten dargestellten Funktionalitäten abbilden zu können, wurde eine entsprechende Hard- und Software-Architektur aufgestellt. Wesentliche Aspekte bei der Konzeption war das Einhalten der Echtzeit-Anforderungen in der Kommunikation zwischen Roboter und PC: der für das Führen des Roboters verantwortliche PC muss innerhalb eines festen Zeitfensters auf die Anfrage des Roboters antworten. Daher wurden alle

nicht zeitkritischen Prozesse auf einem zweiten PC abgebildet. Ebenso wurde die Kommunikation auf unterschiedlichen Netzen realisiert. Bild 7.4 stellt die Gesamtarchitektur dar.

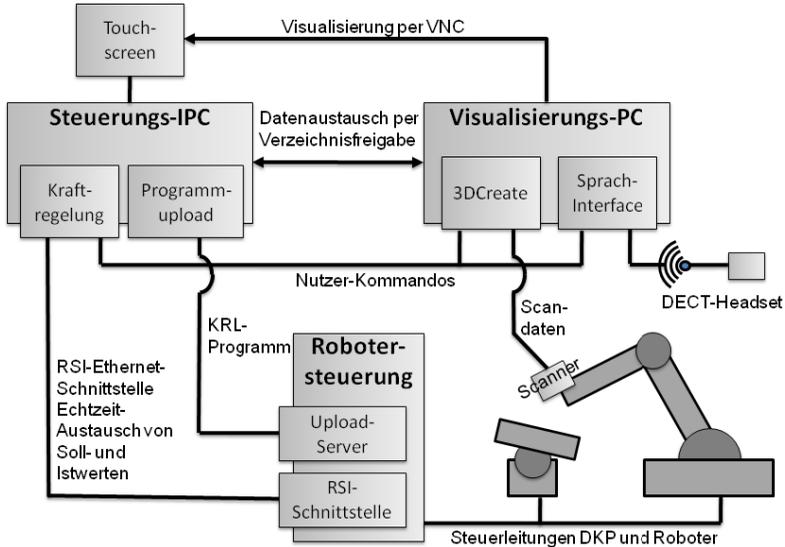


Bild 7.4: Gesamtarchitektur der Softwarekomponenten

### 7.3.2 Steuerungskomponenten zur manuellen Führung des Roboters

Die Führung des Roboters erfolgt mittels einer Admittanz-Regelung, vergleiche (Albuschäffer 2002). Feinfühligere Methoden sind etwa in (Koepe 2001) zu finden. Die gemessenen Kraft- und Momentenwerte des Sensors werden in eine 6-D-Zielposition transformiert, diese wird dann der Positionsregelung zum Stellen der Achsen übergeben. Die Auswertung des Kraft-Momenten-Sensors erfolgt in einem externen Industrie-PC, der zur Übergabe der Werte an die KUKA-Steuerung die RSI-Ethernet-Schnittstelle verwendet. Diese Schnittstelle übermittelt im Interpolationstakt von 12 ms Soll- und Ist-Positionen des Roboters und lässt Änderungen der Sollwerte zu. Weitere Informationen finden sich in (Meyer 2006).

Neben den vom Bediener aufgebrachtten Kräften und Momenten werden die Gewichtskräfte und -momente am Führungsgriff vom Sensor gemessen. Diese werden in der Startposition und bei Bedarf genullt, da sie sich aber bei einer wechselnden Orientierung verändern, können sie mathematisch extrahiert werden. Hierzu ist die Voraussetzung, dass alle biegeschlafenen Komponenten hinter dem Sensor an der Roboterstruktur fixiert sind.

### 7.3.3 Steuerungskomponenten zur Nachbearbeitung des aufgenommenen Programmes

Die Nachbearbeitung der aufgenommenen Bahn erfolgt in dem Offline-Planungstool 3DCreate der finnischen Firma Visual Components (VC 2008). Die in dieser Arbeit entwickelten Funktionalitäten werden als COM-Elemente eingebunden. Die Elemente sind in der Lage, Bahnen mit den entsprechenden Sensordaten zu laden, eine automatische Vorverarbeitungsstufe und die manuellen Anpassungen durchzuführen und KRL-Robotercode zu erzeugen. Weiterhin übernimmt die Software die Visualisierung der Zelle.

Das COM-Modul InTeach-Program erlaubt dabei die Bedienung der Schweißroboterzelle während der Programmierung der Bahn. Hier können über Touchscreen oder Sprachinterface die Bedienparameter ausgewählt werden, siehe Abschnitt 7.2.4. Die Visualisierung der Zelle erfolgt über geladene CAD-Modelle, deren Achswinkel entsprechend den realen Komponenten nachgestellt werden.

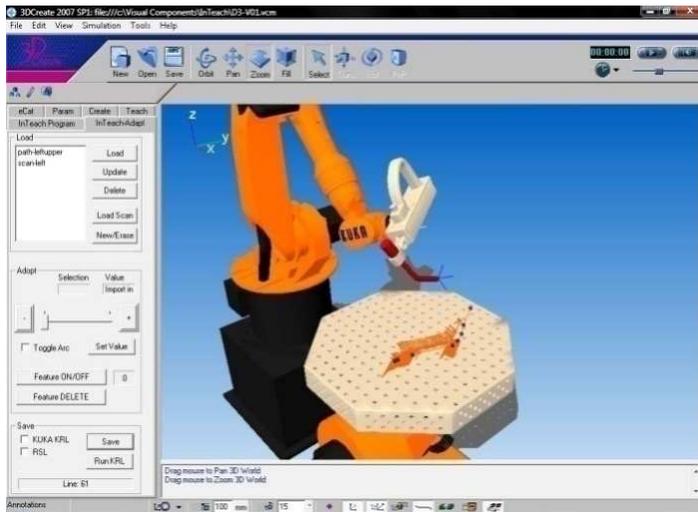


Bild 7.5: Software-Schnittstelle zur Programmierung der Bahn und ihrer Nachbearbeitung

Das COM-Modul InTeach-Adapt erlaubt die Nachbearbeitung der Bahn und die Erzeugung von Robotercode. Es können mehrere Bahnsegmente geladen werden, für jedes einzelne können die Prozessparameter eingestellt werden, insbesondere Start und Stopp des Schweißprozesses und die Verfahrgeschwindigkeiten. Ebenso können Punkte manipuliert werden, also gelöscht oder in ihren Koordinaten verschoben.

Wenn zusätzlich zur Bahn lokale Gemetriedaten des Werkstückes mit dem Scanner aufgenommen wurden, so kann die Bahn über diese Daten optimiert werden. Mehrere Scans können über einen Dialog hochgeladen und entsprechend der Algorithmen in Abschnitt 6.4 aus-

gewertet werden. Dargestellt werden dem Bediener nicht die rohen Punktwolken, sondern die erkannten Flächen.

Die Nachbearbeitung erfolgt in diesem Fall in vier Schritten:

- Selektion der zwei oder drei Flächen, mit deren Hilfe die Bahn optimiert werden soll, mit dem Mauszeiger. Zwei Flächen sind für die Ausrichtung an Ihrer Schnittgeraden notwendig, drei Flächen für die Ausrichtung an einem Eckpunkt.
- Selektion der zu optimierenden Bahnpunkte mit dem Mauszeiger.
- Auswahl der Optimierungsfunktionen über das Dialogfeld. Die Optimierung erfolgt schrittweise, Abstandswerte werden als Feedback in dem Dialogfeld dargestellt. Die Möglichkeiten der Optimierung sind in Abschnitt 6.4.3 dargestellt.
- Da die Bahn des Roboters verändert wurde, ist eine Simulation des neuen Programmes notwendig. In dieser Simulation wird sowohl die Erreichbarkeit als auch die Kollisionsfreiheit mit den gescannten Flächen geprüft.

Im Anschluss an die Nachbearbeitung der Bahn wird Robotercode aus den Bahndaten erzeugt, einerseits KRL-Code zum Ausführen auf dem KR16, andererseits RSL-Code, ein XML-basiertes Format für die 3DCreate-interne Maschinenprogrammierung. Der RSL-Code erlaubt es, das erzeugte Programm vor der Ausführung in einer Simulation ablaufen zu lassen, um die Bahn auf Sinnfälligkeit zu prüfen. Die Erzeugung des KRL-Codes ist in Abschnitt 7.3.4 näher beschrieben.

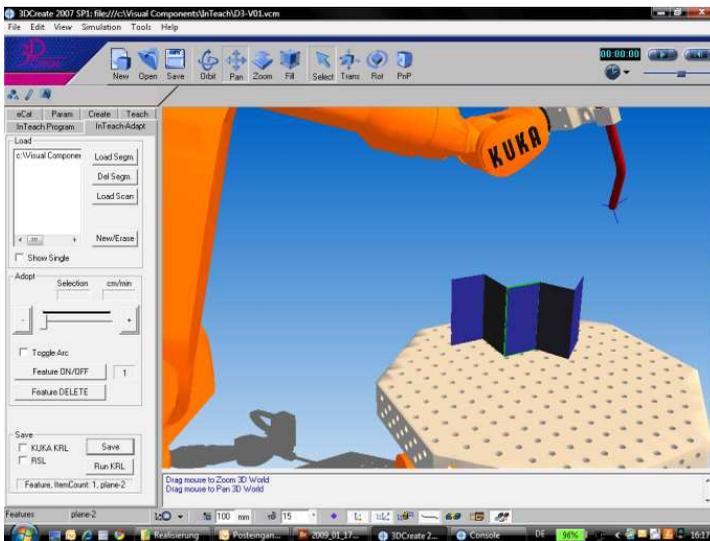


Bild 7.6: Screenshot der Sensordatenauswertung

### 7.3.4 Format des erzeugten RoboterCodes

Um einen späteren Fertigungsbetrieb mit hoher Verfügbarkeit garantieren zu können, wird ein natives Roboterprogramm in KRL (KUKA Robot Language) erzeugt und auf die Steuerung geladen. Der Roboter wird im Anschluss daran wie eine Standard-Fertigungszelle über das Programmierhandgerät bedient.

Bei der Erzeugung des KRL-Programmes müssen drei Anforderungen beachtet werden:

- *Spätere Adaptierbarkeit*  
Bediener und Programmierer müssen für spätere Anpassungen des Programmes und seiner Punkte die Standard-Methoden der KUKA-Steuerung verwenden können. Dazu ist notwendig, dass Bewegungsbefehle im so genannten Inline-Format erzeugt werden, nicht in einem einfacheren direkten Format.
- *Ansteuerung der Stromquelle*  
Schweißspezifische Fragestellungen wie das Füllen von Nahtendkratern, überlagerte Pendelbewegungen und das Error-Handling müssen industrietauglich implementiert werden. Dazu wird das Applikationspaket KUKA.ArcTech verwendet, ein Softwarepaket zur Ansteuerung Job-basierter Stromquellen (KUKA 2008).
- *Koordinierte Bewegung von Roboter und Dreh-Kipp-Tisch*  
Um auch Nähte aus mehreren Teilbahnen immer in Wannenlage schweißen zu können, müssen Roboter und Dreh-Kipp-Tisch eine koordinierte, gleichzeitige Bewegung ausführen. Dies muss in der Steuerung und im Roboterprogramm hinterlegt sein.

Die Anforderungen werden mittels eines KRL-Programmes erfüllt, das aus einer Quelldatei (\*.src) und einer Datendatei (\*.dat) besteht. Stark vereinfacht dargestellt enthält die erste die Bewegungsbefehle auf der Basis von Punktvariablen, die zweite Datei löst die Punktvariablen in XYZABCE1E2-Koordinaten auf.

Der Transfer von der grafischen Oberfläche auf das KRL-Programm ist einfach für Bewegungsbefehle ohne Prozess, mit einsetzendem Prozess muss die Definition des Start- und Stopp-Punktes auf diverse Prozessbefehle heruntergebrochen werden. Im Fall von KUKA.ArcTech sind das:

- ArcOn: Hinbewegung zum Schweißpunkt mit Start des Schweißprozesses
- ArcSwitch: Bewegung zu einem Schweißzwischenpunkt mit aktivem Prozess
- ArcOff: Bewegung zum Endpunkt der Naht mit aktivem Prozess, Stopp des Prozesses

Die Zuordnung erfolgt entsprechend dem Übergangsgraphen in Bild 7.7.

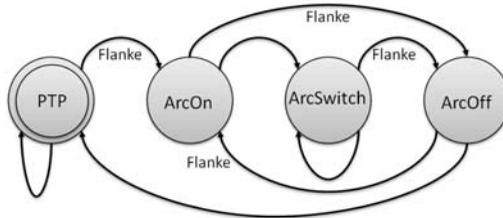


Bild 7.7: Übergangsgraph zur Zuordnung von Nahtpunkten. Eine Flanke entsteht beim Wechsel des Schweißprozesses von aktiv zu passiv bzw. umgekehrt.

Die erzeugten Roboterprogramme entsprechen den Anforderungen und sind ablauffähig.

## 7.4 Ablauf der Programmierung

Der Ablauf der Programmierung der entwickelten Schweißroboterzelle ist in Bild 7.8 dargestellt, er unterteilt sich in die drei Blöcke Programmierung, Optimierung und Fertigung. Der erste Block, die Programmierung, wird mit den in dieser Arbeit entwickelten Verfahren und Komponenten durchgeführt. Dabei kann gewählt werden, ob die Roboterbahn durch lokale Werkstückdaten, also Scandaten, optimiert werden soll.

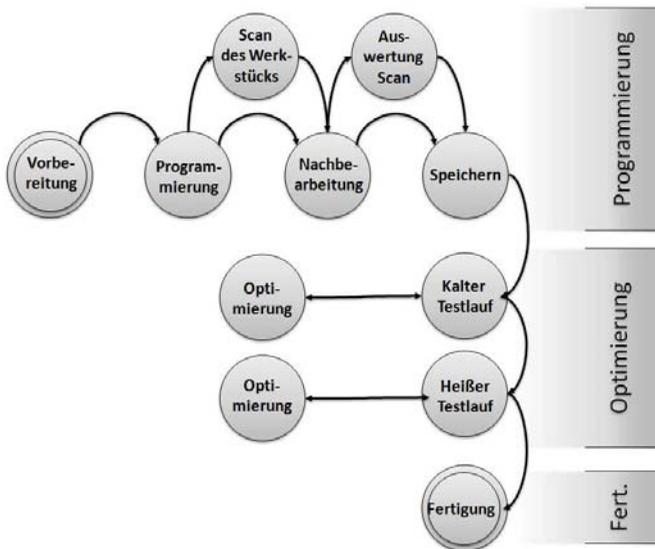


Bild 7.8: Ablauf der Programmierung, Optimierung und Fertigung

Die beiden sich anschließenden Blöcke, Optimierung und Fertigung, laufen vollständig auf der Robotersteuerung mit deren Bordwerkzeugen ab. So werden Punkte etwa über die Ak-

tualisierung der Inline-Befehle mittels TouchUp (KUKA 2008) vorgenommen. Ebenso wird für die Fertigung ein Standard-KRL Programm ausgeführt.

Diese Aufteilung erhöht die Verfügbarkeit des Robotersystems im Industrieinsatz.

## 7.5 Erprobung am Beispiel eines Galerieträgers

Die Schweißroboterzelle wurde bei dem Metallbauunternehmen Maschinenbau Treffler (Treffler 2008, Tabelle 7.2) installiert, um Tests und Optimierungen in einer realen Umgebung vornehmen zu können. Die Fa. Treffler stellt unter anderem Schweißkonstruktionen her, typische Produkte sind in Bild 7.9 dargestellt. Die Losgrößen dieser Produkte liegen typischerweise zwischen 10 und 100 Stück je Auftrag.



Bild 7.9: Typische Schweißkonstruktionen gefertigt bei Fa. Treffler: Lagersitze, landwirtschaftliche Maschinen, Galerieträger (v.l.n.r.)

Kennzeichen	Ausprägung
Name und Adresse	Maschinenbau Treffler, Pöttmes-Echsheim
Produkte	Metall-, Maschinenbau, Landtechnik
Anzahl der Mitarbeiter	55 Angestellte, 17 Auszubildende
Schichtmodell	Zweischichtbetrieb
Erfahrung mit Robotik	Schweißroboterzelle KUKA KR16

Tabelle 7.2: : Kenndaten der Fa. Maschinenbau Treffler

Der Test der Schweißroboterzelle in der Fertigungshalle der Fa. Treffler verfolgte mehrere Ziele:

- Verifikation der Funktionalität in einer realen Umgebung
- Test und Kommentierung der Anlage durch Mitarbeiter der Fa. Treffler
- Zeitnahme bei der Programmierung von Bauteilen unter Nutzung verschiedener Vorgehensweisen bei der Programmierung.

Die Schweißroboterzelle wurde direkt in der Fertigungshalle aufgebaut, siehe Bild 7.10.



Bild 7.10: Test der in der Fertigung aufgebauten Schweißroboterzelle durch einen Mitarbeiter der Fa. Treffler

### 7.5.1 Spezifikation der Aufgabe

Während der Testphase bei der Fa. Treffler wurden zwei Schweißaufgaben näher untersucht:

- *Programmierung der Schweißnähte eines I-Trägers auf einer Grundplatte*  
Dies ist eine einfache Standardaufgabe, die als Teil komplexerer Programmieraufgaben ständig wiederkehrt. Da der Zeitrahmen der Programmierung überschaubar ist und Versuchsteile günstig herzustellen sind, wurde diese Aufgabe als Basistest definiert.  
Daten: 8 Schweißnähte, 16 schweißrelevante Punkte, insgesamt ca. 40 Punkte zu programmieren
- *Programmierung eines Galerieträgers*  
Galerieträger (Tragelemente eines Laufsteges um einen Großdiesel) werden häufig in der bei der Fa. Treffler in Betrieb befindlichen Schweißroboterzelle gefertigt. Die Prozesszeiten eines Bauteils liegen hier bei etwa 20 min, die Qualität der gefertigten Bauteile ist sehr gut. Die Programmierzeit ist mit 7 Stunden (mit dem KUKA Control Panel) hoch. Aus diesem Grund wurde hier kein direkter Vergleich gemessen, sondern nur die Machbarkeit getestet.  
Daten: 73 Schweißnähte, 177 schweißrelevante Punkte (2x73+31 Zwischenpunkte), insgesamt ca. 610 Punkte zu programmieren

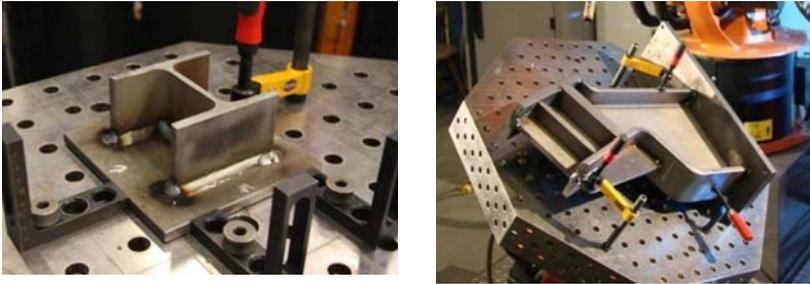


Bild 7.11: Testbauteile I-Träger (links), Galerieträger (rechts)

Beide Bauteile werden gestrahlt und gepunktet auf der Demmeler-Platte auf dem Dreh-Kipp-Tisch fixiert. Die verwendeten Schweißparameter (Jobbetrieb) sind in der folgenden Tabelle 7.3 aufgeführt.

Parameter	I-Träger	Galerieträger
Schweißspannung [V]	24,9 V / 216 A	29,2 V / 274 A
Drahttyp, Durchmesser [mm]	Rehm Unifili 3, 1 mm	Rehm Unifili 3, 1 mm
Drahtvorschub [m/s]	12,2	17,6
Verfahrensgeschwindigkeit [m/min]	40	40
Schweißgas	Corgon 18 %	Corgon 18 %
Durchfluss	12 l/min	12 l/min
Adaptionsparameter	4,9	5,9
Kraterfüllzeit [s]	0,05	0,1

Tabelle 7.3: Prozessparameter der Testbauteile

Die Parameterbreite des KUKA ArcTech-Paketes wurde nicht vollständig verwendet, es wurden etwa keine Pendelbewegungen programmiert. Die wesentlichen Parameter, beispielsweise die Endkraterfüllzeit, wurden mit Hilfe der Roboterprogrammierer von der Fa. Treffler gewählt.

Bei der Fa. Treffler haben die beiden Roboterprogrammierer die Tests durchgeführt, wobei einer eher für Fehlerbehebung in der Schicht zuständig ist, während der andere beinahe alle Programme schreibt. Die Erfassung des jeweiligen Kenntnisstandes in Bezug auf die beiden Programmierverfahren ist also unerlässlich.

Testperson	Kenntnisstand KUKA Control Panel	Kenntnisstand Programmieren durch Vormachen
1 (Fa. Treffler)	Befriedigend (Tasten)	Befriedigend (ca. 1 h)
2 (Fa. Treffler)	Sehr gut (Maus)	Ausreichend (ca. 0,5 h)
3 (Fraunhofer IPA)	Befriedigend (Tasten)	Sehr gut (Entwickler)
4 (Fraunhofer IPA)	Befriedigend (Tasten)	Sehr gut (Entwickler)
5 (Fa. Treffler)	Keine Kenntnisse bzgl. Roboterprog.	Ausreichend (ca. 0,5 h)

Tabelle 7.4: Kenntnisstand der Testpersonen bzgl. Roboterprogrammierung mittels KCP und Programmieren durch Vormachen. Bei der KCP-Steuerung ist die Art des Verfahrens angegeben, über die kartesischen Achstasten oder die 3-D-Maus.

### 7.5.2 Test und Ergebnisse des Programmierens durch Vormachen

Der erste Schritt der Programmierung, die Festlegung der Schweißpunkte, wurde von den Mitarbeitern der Fa. Treffler nach kurzer Einführung (<15 min) an den Testwerkstücken durchgeführt, siehe Bild 7.12. Auf Grund der Erfahrung bei der Programmierung des Standard-Roboters waren sowohl das Vorgehen als auch das Umgehen mit kinematischen Problemen wie Singularitäten klar.



Bild 7.12: Programmierung der beiden Testwerkstücke durch Mitarbeiter der Fa. Treffler

Das qualitative Feedback der Tester war positiv, so wurde die Intuitivität und die Schnelligkeit insbesondere beim Verfahren im Raum als positiv bewertet. Optimierungsbedarf wurde im Bereich der Störkonturen des Handgriffs gesehen, ebenso an weiteren, inzwischen behobenen Punkten (Behandlung von Singularitäten, Durchgängigkeit der Bedienung).

Bei den systematischen Aussagen ist zwischen der Erfahrung der Tester zu unterscheiden. Tester 2, sehr erfahren in der Programmierung und geübt im Verfahren des Roboters mit der KUKA 3-D-Maus, findet weniger Vorteile als Tester 1, der nicht so häufig den Fertigungsroboter programmiert.

Tester 5 fand sich schnell in die Programmierumgebung ein und war nach sehr kurzer Zeit in der Lage, prozessfähige Roboterprogramme durch Vormachen zu programmieren.

### 7.5.3 Test und Ergebnisse der grafischen Nachbearbeitung

Die programmierten Bahnen müssen, wie im Ablaufdiagramm in Bild 7.8 dargestellt, nachbearbeitet werden, um aus den bisher rein geometrischen Daten ein komplettes Roboterprogramm mit den entsprechenden Prozessbefehlen (ArcOn, ArcSwitch, ArcOff) zu machen. Dazu wird die 3-D-Oberfläche verwendet. Ebenso werden während der Bahnprogrammierung Parameter wie der Override über die grafische Oberfläche eingestellt.

Sowohl die Bedienung als auch die Nachbearbeitung konnte von den Mitarbeitern der Fa. Treffler nach einer kurzen Einführung von ca. 20 Minuten eigenständig vorgenommen werden, siehe Bild 7.13.

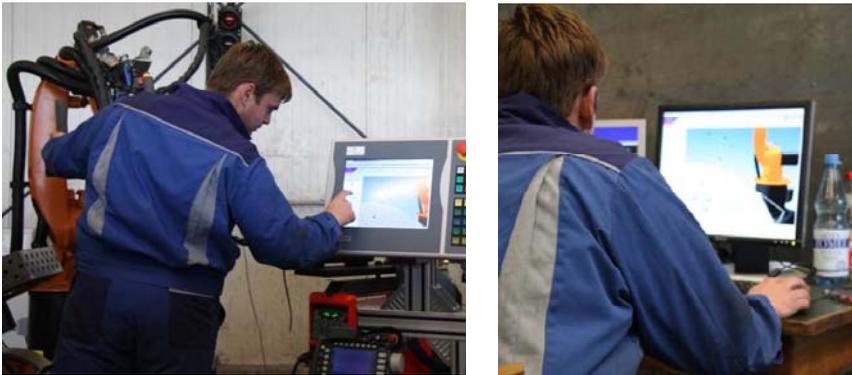


Bild 7.13: Verwendung des Touchscreens zur Bedienung des Robotersystems (links), Nachbearbeitung der Bahn mittels Maus und 3-D-Oberfläche (rechts)

Der Kern des qualitativen Feedbacks war der Wunsch nach Minimierung der Nachbearbeitung. Diesem Feedback folgend wurden Algorithmen integriert, die etwa die Bahngeschwindigkeiten anhand des Prozesszustandes als Defaultwert setzen: schnell bei Luftfahrt, langsamer bei Annäherung, Prozessgeschwindigkeit während der Schweißaktivität.

### 7.5.4 Test und Ergebnisse der sensorbasierten Bahnoptimierung

Die Versuche zur sensorbasierten Bahnoptimierung wurden in den Versuchsfeldern des Fraunhofer IPA vorgenommen.

Zum Scannen der relevanten Bauteilbereiche wird der Führungsriff um 90° gedreht, so dass der in Bild 7.2 gezeigte Laser-Triangulations-Sensor lotrecht zum Werkstück steht. Der Scan-

ner wird langsam und gleichmäßig über das Werkstück geführt. In der Auswertung können mehrere Scans geladen werden.

Die Optimierung an Hand von Scannerdaten ist funktionsfähig und verbessert die Genauigkeit der programmierten Bahnpunkte, insbesondere bei einer schlechten Ausgangsgenauigkeit. Versuche wurden mit dem I-Träger und auch mit dem Gallerieträger durchgeführt.

Mehrere Ergebnisse lassen sich festhalten:

- Die Genauigkeit von Bahnpunkten kann optimiert werden, die Punkte können auf weniger als 1 mm genau in gescannte Eckpunkte positioniert werden. Dies ist für den Prozess des MAG-Schweißens bei vielen Bauteilen ausreichend.
- Das mehrmalige Scannen eines Ausschnittes der Bauteilgeometrie führt zu Problemen in der bisherigen Implementierung.
- Die Orientierung während des Scans muss in der bisherigen Implementierung konstant gehalten werden.

Auch unter dem Aspekt des zusätzlichen Zeitaufwandes ist das Scannen der Bauteilgeometrie nicht für alle Werkstücke geeignet. Die Genauigkeit und damit die Qualität der programmierten Bahn lassen sich aber durch das Vorgehen verbessern.

### 7.5.5 Prozesszeiten der Programmierung

Eine wesentliche Aussage der Tests bei der Fa. Treffler sind die notwendigen Programmierzeiten. Es wurden Zeitnahmen der Mitarbeiter der Fa. Treffler und des Fraunhofer IPA gemacht.

#### Versuch 1: I-Träger, Vergleich von KCP und Programmieren durch Vormachen

Bauteildaten: 8 Schweißnähte, 16 schweißrelevante Punkte, insgesamt ca. 40 Punkte zu programmieren

	Testp. 1 (Treffler)	Testp. 2 (Treffler)	Testp. 3 (FhG IPA)	Testp. 4 (FhG IPA)	Testp. 5 (Treffler)
Programmierung mit KCP	26	16	35	80	
Kalter Testlauf, Korrektur	16	7	16	6	
<i>Summe KCP</i>	42	23	51	86	
Programmierung durch Vormachen	10	13,5	9	15	25
Nachbearbeitung	2	1,5	1	2	2
Kalter Testlauf, Optimierung	13	6	8	12	13
<i>Summe Programmieren durch Vormachen</i>	25	21	18	29	40
<b>Reduktion der Programmierzeit</b>	<b>40 %</b>	<b>9 %</b>	<b>65 %</b>	<b>66 %</b>	

Tabelle 7.5: Zeitaufnahme in [min] der Prozessschritte bei der Programmierung des I-Trägers

Die Auswertung der Versuche ergibt als wesentliche Punkte:

- Die Programmierung durch Vormachen ist über alle Testpersonen betrachtet um 46% schneller als mit dem KCP
- Für geübte Programmierer ist die Zeitersparnis gering; die Bewegung des Roboters zu den Punkten hin ist schneller, dafür ist die Nachbearbeitung der Bahn als zusätzlicher Schritt zu bewältigen.
- Insbesondere für KMU ist es vorteilhaft, dass auch Personal ohne Robotererfahrung schon nach kurzer Lernphase produktiv Bahnen programmieren kann.

#### Versuch 2: Galerieträger, Vergleich von KCP und Programmieren durch Vormachen

Bauteildaten: 73 Schweißnähte, 177 schweißrelevante Punkte (2 x 73 + 31 Zwischenpunkte), insgesamt ca. 610 Punkte zu programmieren

Auf Grund der Komplexität des Bauteils ist kein realistischer Vergleich möglich. Probleme bezüglich der Vergleichbarkeit sind: Prototypenstatus der Programmierumgebung, unterschiedliche Robotertypen (KR16 Kurzarm zu Langarm) mit daraus resultierenden Erreichbarkeitsproblemen.

## 7.6 Schlussfolgerungen aus Realisierung und Erprobung

Die entwickelte Programmierumgebung wurde, wie in den vorhergehenden Abschnitten beschrieben, bei der Fa. Maschinenbau Treffler unter realen Umgebungsbedingungen getestet. Es wurden Versuche mit mehreren Personen unterschiedlicher Qualifikationen durchgeführt, mehrere Werkstücke wurden gefertigt. Die gemachten Erfahrungen führen zu wichtigen Schlussfolgerungen:

- Die Programmierung des Roboters mit der Programmierumgebung war allen Testern nach einer kurzen Einführung von etwa 30 Minuten möglich. Hier spielte es keine Rolle, ob die Tester in der Bedienung des Roboters mit dem Standard-Programmierhandgerät erfahren waren. Wesentliche Voraussetzung für dieses Ergebnis ist die Intuitivität durch die haptische und grafische Schnittstelle.
- Die Bewegungsführung ist besonders für weniger erfahrene Bediener deutlich schneller. Dies ist unter anderem auf den nicht notwendigen Transfer der PHG-Tasten in die entsprechenden Koordinatensysteme.
- Wichtig ist eine durchgehende Bedienung ohne Nutzung des Keyboards. Die Interpretation der dreidimensional dargestellten Bahndaten stellt keine Herausforderung für die Prozessexperten dar. Die Eingabe von Dateinamen oder Daten aber ist zu ungewohnt.
- Insgesamt ist die Programmierumgebung für Prozessexperten geeignet, die nicht täglich Industrieroboter programmieren. Hier ist ein hohes Potenzial gegeben. Erfahrene Programmierer haben geringere Vorteile.

Die Erprobung konnte die Funktionalität der Programmierumgebung in realen Umgebungen nachweisen. Ebenso konnten Einsparungen in der Programmierzeit nachgewiesen werden.

## 8 Zusammenfassung und Ausblick

Industrieroboter werden vornehmlich in der Großindustrie eingesetzt, insbesondere in den Branchen Automobilbau, chemische Industrie, Elektrotechnik/Elektronik und Maschinenbau. Dort steht der Industrieroboter seit Jahrzehnten für flexible Automatisierung auf höchstem Niveau. Die dort entwickelten Lösungen und Vorgehensweisen können aber nicht auf andere Branchen und insbesondere andere Betriebsgrößen übertragen werden. Kleine und mittlere Unternehmen etwa haben Probleme, Industrieroboter einzusetzen, da sie in anderen Losgrößen arbeiten und andere Infrastrukturmöglichkeiten haben. Sie stecken häufig in einer Automatisierungszwickmühle, sie müssen rationalisieren, um dem Wettbewerbsdruck standzuhalten, können die in Großbetrieben etablierten Methoden aber nicht anwenden.

Dabei ist die Anzahl der kleinen und mittleren Betriebe groß, 99,8 % der Betriebe gehören in diese Kategorie, sie beschäftigen 67 % aller Werkstätigen. Der Bedarf an Lösungen für diese Unternehmen ist also enorm.

Um dieses Defizit zu mindern, war das Ziel dieser Arbeit die Entwicklung eines Verfahrens zu schnellen und intuitiven Programmierung von Industrierobotern. Das Verfahren soll an der Beispielanwendung MAG-Schweißen verifiziert werden.

Die Interaktion mit der Schweißroboterzelle soll dabei multimodal vom Prozessexperten, dem Schweißer, vorgenommen werden. Mittels Programmierung durch Vormachen kann der Schweißer den Roboter entlang der zu schweißenden Nähte verfahren, eine Kraftregelung führt den Roboter dahin, wo der Bediener ihn hindrückt. Ein besonderer Fokus der Arbeit liegt auf der dann folgenden Nachbearbeitung der Bahn und der Schnittstelle zum Bediener.

So wurde ein Verfahren entwickelt, das es erlaubt, die aufgenommenen Bahndaten zu komprimieren und Geometrieprimitive zuzuweisen. Die Kompression nutzt den Douglas-Peucker-Algorithmus, für das Mapping der Daten werden entsprechende Kriterien erstellt. Verschiedene Kombinationen von Algorithmen wurden implementiert und verglichen.

Der zweite Entwicklungsschwerpunkt lag in der Verbesserung der manuell aufgenommenen Bahn auf der Basis lokaler Werkstückgeometrie. Diese Geometriedaten bestehen aus einer Punktwolke, die mit einem Laserscanner aufgenommen werden kann. Zwei Verfahren zur Auswertung wurden entwickelt, ein auf den einzelnen Scanlinien basierendes Verfahren zur Optimierung der Bahn, sowie ein Verfahren zur Flächenrückführung, basierend auf einem Region Growth Ansatz. Mit Hilfe der Flächeninformationen kann die Bahn interaktiv optimiert werden.

Die entwickelten Verfahren und Algorithmen wurden in ein Testsystem integriert, das auf einem KUKA KR16 mit Dreh-Kipp-Tisch und einer Fronius-Schweißstromquelle beruht. Der Schweißroboter wurde unter anderem bei dem KMU Maschinenbau Treffler, einem Unternehmen mit ca. 70 Mitarbeitern, getestet. Dort wurde einerseits sichergestellt, dass das Robotersystem reale Bauteile unter den entsprechenden Umgebungsbedingungen fertigen

kann, andererseits wurden Programmier-tests mit Mitarbeitern der Firma durchgeführt. Programmiert wurden Galerieträger für Großmotoren, etwa 1 m lange Stahlkonstruktionen mit 70 Einzelschweißnähten und einer Prozesszeit von 20 Minuten.

Die Testläufe zeigten, dass die Referenzbauteile mit der Schweißroboterzelle und dem entwickelten Programmierverfahren gefertigt werden können. Die Auswertung der Tests zeigte, dass die Ziele der Arbeit, die Vereinfachung der Programmierung und die Verkürzung der Programmierzeit, erreicht wurden. Prozessexperten ohne Kontakte mit der Roboterprogrammierung konnten nach weniger als 2 Stunden einfache Werkstücke schweißen. Mit der konventionellen Roboterprogrammierung erfahrene Werker konnten ihre Arbeit um 9 % bis 40 % beschleunigen.

Der aufgebaute Schweißroboter ist funktional und in sich geschlossen. Trotzdem bieten sich, aufbauend auf dieser Arbeit, weitere Entwicklungen an.

Die Verfahren zur Optimierung der Bahn an Hand von Sensordaten können durch einen hybriden Ansatz ergänzt werden: falls CAD-Informationen über das Werkstück vorhanden sind, könnten diese mit den Sensordaten fusioniert und zur Optimierung der Bahn genutzt werden (Meyer 2010). Weiterhin kann es sinnvoll sein, komplexere Informationen aus den Sensordaten zu generieren, etwa durch die Extraktion von Regelgeometrien.

Die Interaktion mit dem Bediener kann durch eine Verbesserung der Schnittstellen optimiert werden. Im Bereich der Sprachinteraktion ist dies insbesondere die Verwendung eines Kehlkopfmikrofons, welches jegliche Umgebungsgeräusche ausblendet. Zu leisten wäre die Transformation der aufgenommenen Schwingungen dahingehend, dass sie von den üblichen Verarbeitungseingängen verstanden werden: die im Rachen- und Mundraum stattfindende Stimmbildung muss den Signalen hinzugefügt werden. Ebenso ließe sich die Interaktion durch Komponenten der Künstlichen Intelligenz verbessern. Diese lernen aus den Aktionen des Nutzers und schlagen im weiteren Verlauf bei bekannten Situationen die wahrscheinlichen Werte und Aktionen vor.

In Bezug auf die Applikation bietet sich die Integration eines Expertensystems an, das auf der Basis der Geometrie- und Prozessparameter weitere Einstellungen optimiert. Die Erweiterung des Verfahrens auf neue Anwendungen bietet ebenfalls Entwicklungsmöglichkeiten.

## 9 Abstract

### **Trajectory Recording and Post Processing while Programming Industrial Robots using Programming by Demonstration**

To enable especially small and medium companies to program industrial robots in an intuitive and fast way a programming environment has been developed based on the programming by demonstration paradigm. The programming environment enables the recording and sensor assisted post processing of trajectories in welding scenarios. The programming environment has been validated at a welding shop floor.

**Introduction, Deficit and Goals** – Since nearly 60 years industrial robots are a building block to increase productivity through automation. Especially in the automotive industry, engineering and chemical industry industrial robots are commonly used. Solutions for set up, programming and maintenance are well developed. But these procedures cannot be transferred without problems to small and medium companies (SMEs). Here the boundary conditions, especially regarding company size, prevent e.g. specialized robot departments. Regarding the economics SMEs often do have to cope with small lot sizes. Industrial robots are only viable in these use cases, if the programming can be conducted in an intuitive and fast way, preferably by the process expert.

**State of the Art** – The most spread programming methods for industrial robots are online programming using the teach pendant, and offline programming. Further methods are being developed and tested in many variations, lots of them in the area of programming by demonstration. Here the trajectory to be recorded is shown the robot system directly on the work piece, e.g. by guiding the robot, by using a teach wand or by using a teach kinematic. Regarding trajectory post processing the text based adaption on the teach pendant is commonly used. Other possibilities comprise graphical teach pendants or graphical user interfaces on the computer.

**Definition of Subsystems** – Based on the analysis of the welding robot six sub systems with the according requirements have been defined: the handling system, the safety system, means to record the trajectory, means to post process the trajectory, the sensor system and the process equipment.

**Concept of the Programming Environment** – The guidance of the robot will be realized using a force torque sensor and generating new position values in the interpolation cycle. Haptic, graphic and verbal interactions allow the user to combine different modalities, e.g. change the velocity using a spoken command while guiding the robot. The safety concept follows the collaboration regulations of the ISO 10218-1 with safe reduced speed and an enabling switch as main components. The recorded trajectory will be stored and processed as line and spline segments. A laser triangulation sensor allows to use local geometry information of the work piece to optimize the recorded path. KRL robot language will be generated after the post processing.

**Development of the Trajectory Post Processing** – Several algorithms to enable the graphical trajectory post processing have been developed and compared. Necessary steps are the data compression using the Douglas-Peucker-Algorithm, the segmentation and mapping into line and spline elements using a corner criterion.

**Development of the Trajectory Optimization using Sensor Information** – A higher precision can be reached when incorporating local work piece data. The scan data from a laser triangulation sensor is filtered and processed using a scanline and a region growth based approach. The scanline based approach can be used to generate programs for simple, mostly straight welding seams. The region growth based approach enables the worker to scan the work piece, to select generated planes, and to adapt the manual programmed points to the computed edges and corners.

**Realization Set Up** – The developed programming environment in a welding scenario has been realized and tested using a KUKA KR16 with turn table DKP400 and a Fronius TP3200 welding source. A guiding handle with a JR3 force torque sensor and the necessary safety equipment (enabling button and emergency stop) has been attached to the robot. A Beckhoff IPC with touchscreen enabled the interaction during trajectory recording and post processing. A laser triangulation sensor from MEL-Sensor has been attached to allow scanning the work piece.

**Realization Results** – The welding robot cell has been tested at the facilities of Fraunhofer IPA and in the shop floor of Maschinenbau Treffler, a welding contract manufacturer with approx. 70 welders in southern Germany. The tests at Treffler have been conducted with the goals of optimizing reliability and ease of use under realistic production conditions, and to assess programming times.

The programming environment and the realized robot cell allowed producing common parts out of the Treffler production. Experienced welders without robot programming knowledge have been able to program simple parts with an introduction time of only few hours. Robot programmers could decrease the necessary programming time between 9 % and 40 %, depending on their routine with the teach pendant.

**Summary** – The goal of setting up an intuitive and fast programming environment has been successfully realized in the use case of a welding robot.

**Further work** – From the current point of view development directions towards the integration of manual programming and CAD based programming seem to be promising. Currently SMEs often do not work on 3D CAD drawings, but if they are available they could be used to interactively define the process trajectory. Other possible developments can be found in the realization of a throat microphone command interpreter, and the generalization of the programming environment to further applications.

## 10 Referenzen

- ABB 2004            ABB: FlexPendant case study: **Klößner Desma** (2004)  
[http://library.abb.com/global/scot/scot241.nsf/veritydisplay/8b76462f397dd271c12570d00038979f/\\$File/rs\\_case\\_desma\\_en\\_print.pdf](http://library.abb.com/global/scot/scot241.nsf/veritydisplay/8b76462f397dd271c12570d00038979f/$File/rs_case_desma_en_print.pdf) (20.06.2010)
- ABB 2008            ABB: Product Description **Force Control for Machining** (2004)  
<http://www.abb.com/product/us/9AAC133110.aspx> (20.06.2010)
- Ahn 2004            Ahn, Sung J. **Least Squares Orthogonal Distance Fitting of Curves and Surfaces in Space**. Berlin: Springer, 2004
- Albu-Schäffer 2002    Albu-Schäffer, Alin; Hirzinger, Gerd: **Cartesian Impedance Control Techniques for Torque Controlled Light-Weight Robots**. In: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2002. Washington, DC, Mai 2002, S. 657-663
- Bhanu 1985            Bhanu, B.; Lee, S.K.; Ho, C.C.; Henderson, T.C.: **Range Data Processing: Representation of Surfaces by Edges**. Salt Lake City: University of Utah, Department of Computer Science, 1985
- Biggs 2003            Biggs, Geoffrey; MacDonald, Bruce.: **A Survey of Robot Programming Systems**. In: Proceedings of the Australasian Conference on Robotics and Automation, Brisbane, Australia, 2003, paper 33
- Bischoff 2002            Bischoff, Rainer; Kazi, Aarif; Seyfarth, Markus: **The morpha style guide for icon-based programming**. In: VDI-Berichte Nr. 1679. Düsseldorf: Deutscher Ingenieur-Verlag, 2002, S. 41-46
- Blackbirdrobotics 2010    Blackbird Robotersysteme GmbH: **Firmenhomepage**  
<http://www.blackbird-robotics.de> (15.06.2010)

- Breckweg 2006 Breckweg, Achim; Meyer, Christian ; Drechsler, Klaus ; Ruger, Olaf: **Fast and Intuitive Robot Programming for Mandrel Guiding of Braiding Machines for Textile Preforming**. In: Erath, Marc A. (Ed.); SAMPE Europe: The Use of Composites in the 21st Century to Save Energy and Weight / CD-ROM : SEICO 07, 28th SAMPE EUROPE International Conference April 2-4th, 2007, Paris. Proceedings. Riehen / Basel, CH, 2007
- Colombo 2006 Colombo, D.; Dallefrate, D.; Tosatti, L.M.: **PC based control systems for compliance control and intuitive programming of industrial robots**. In: VDI-Wissensforum: ISR 2006 - ROBOTIK 2006 : Proceedings of the Joint Conference on Robotics, May 15-17, 2006, Munchen: Visions are Reality. Dusseldorf, 2006. (VDI-Berichte 1956).
- Denkena 2004 Denkena, B.; Worn, H.; Hein, B.; Apitz, R.; Kowalski, P.; Mages, D.: **Vereinfachte Programmierung von Industrierobotern**. In: wt Werkstattstechnik online, Jahrgang 94/2004, S. 442-446
- DeSantis 2008 De Santis, Agostino; Siciliano, Bruno; De Luca, Alessandro; Bicchi, Antonio: **An atlas of physical human-robot interaction**. In: Mechanism and Machine Theory 43.3 (2008), 2008, S. 253-270
- DeStatis 2008 Statistisches Bundesamt (Hrsg.): **Statistisches Jahrbuch 2008**: Fur die Bundesrepublik Deutschland. Wiesbaden, 2008
- Dilthey 1995 Dilthey, Ulrich: **Schweitechnische Fertigungsverfahren Band 2, Verhalten der Werkstoffe beim Schweien**. Dusseldorf: VDI-Verlag, 1995
- Dilthey 2005 Dilthey, Ulrich: **Schweitechnische Fertigungsverfahren 1**. Berlin u. a.: Springer, 2005
- DIN EN 999 Norm: DIN EN 999 2008-10: **Sicherheit von Maschinen - Anordnung von Schutzeinrichtungen im Hinblick auf Annaherungsgeschwindigkeiten von Korperteilen**
- DIN EN ISO 6947 Norm: DIN EN ISO 6947 1997-05: **Schweinahte - Arbeitspositionen - Definitionen der Winkel von Neigung und Drehung**
- DIN EN ISO 8373 Norm: DIN EN ISO 8373 1996-08: **Industrieroboter - Worterbuch**

- DIN EN ISO 10218-1 Norm: DIN EN ISO 10218 2006-06: **Roboter für Industrieumgebung - Sicherheit - Teil 1: Roboter**
- DIN EN ISO 13849-1 Norm: DIN EN ISO13849-1 2008-12 **Sicherheit von Maschinen - Sicherheitsbezogene Teile von Steuerungen - Teil 1: Allgemeine Gestaltungsleitsätze**
- Drechsler 2009 Drechsler, K.: **Kooperation zwischen GE und TUM im Bereich Kohlefaser-Verbundwerkstoffe**. Aktuelle Meldungen der Technischen Universität München, 03. Juli 2009. [http://portal.mytum.de/pressestelle/meldungen/news\\_article.2009-07-03.7218603417](http://portal.mytum.de/pressestelle/meldungen/news_article.2009-07-03.7218603417) (20.06.2010)
- Drews 1997 Drews, P.; Schmid, D.; Volkholz, V. (Hrsg.): **Roboter in der Werkstatt**. Frankfurt: Maschinenbau-Verlag, 1997
- EU-KMU 2008 Europäische Kommission: **Vorrang für den Mittelstand**. Ausgabe 2008. Luxemburg: Amt für Veröffentlichungen der Europäischen Union, 2008
- Eurostat 2008 Europäische Gemeinschaften / Statistisches Amt (Eurostat): **EUROSTAT Jahrbuch 2003 : Der statistische Wegweiser durch Europa**. Luxemburg : Amt für amtliche Veröffentlichungen der Europäischen Gemeinschaften, 2008
- Fischer 1992 Fischer, Ulrich (Leitung): **Fachkunde Metall**. Haan-Grünten: Verlag Europa-Lehrmittel, 1992
- Foitzik 2001 Foitzik, Bernhard: **Wer beschichten kann, kann auch programmieren**. In JOT Journal für Oberflächentechnik, Jahrgang 2001, Ausgabe 7. Wiesbaden: Vieweg+Teubner, 2001, S. 12-14
- Garcia 2007 Patentschrift DE102007008238A1: **Fertigungseinrichtung** (20.12.2007). DaimlerChrysler AG, Alexander Alonso Garcia
- Hans 2005 Hans, Matthias: **Eine modulare Kontrollarchitektur für den Hol- und Bringdienst von Roboterassistenten**. Heimsheim : Jost-Jetter, 2005. (IPA-IAO Forschung und Praxis; 412). (Stuttgart, Univ., Fak. Maschinenbau, Inst. für Industrielle Fertigung und Fabrikbetrieb, Diss. 2005)

- Haegele 2008 Hägele, Martin; Verl, Alexander: **Prozessinnovationen mit Industrierobotern**. In: Gesellschaft für Fertigungstechnik: Stuttgarter Impulse - Fertigungstechnik für die Zukunft : Schriftliche Fassung der Vorträge zum Fertigungstechnischen Kolloquium am 10. bis 11. September 2008 in Stuttgart. Stuttgart, 2008, S. 195-204
- Heckbert 1997 Heckbert, Paul S.; Garland, Michael: **Survey of Polygonal Surface Simplification Algorithms**. In: Proceedings of the Multiresolution Surface Modeling Course on SIGGRAPH 1997, 05.08.1997, Los Angeles
- Hein 2007 Hein, Björn; Hensel, Martin: **Modellgestützte Online-Programmierverfahren für Industrieroboter**. In: 40. Sitzung des VDI/VDE Fachausschusses 4.13 „Steuerung und Regelung von Robotern“. Frankfurt am Main: VDI/VDE, 27.07.2007
- Heiligensetzer 2005 Heiligensetzer, Peter: **Safe Operation–Safe Handling**. In: Westkämper, Engelbert (Hrsg.) ; Verl, Alexander (Hrsg.) ; Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA: 4. Workshop für OTS-Systeme in der Robotik: Sichere Mensch-Roboter-Interaktion ohne trennende Schutzsysteme. Fraunhofer IPA Workshop 2005
- Heiligensetzer 2003 Heiligensetzer, Peter: **Sichere Mensch-Roboter Kooperation**. In: 33. Sitzung des VDI/VDE-GMA Fachausschuß 4.13 „Steuerung und Regelung von Robotern“. Karlsruhe, VDI/VDE, 23.01.2003
- Helms 2006 Niehaus, Michael (Bearb.) ; Helms, Evert (Bearb.) ; Kubacki, Jens (Bearb.) ; Meyer, Christian (Bearb.) ; Parlitz, Christopher (Bearb.) ; Barth, Oliver (Bearb.) ; u.a. ; Bundesminister für Bildung und Forschung: **ASSISTOR : Abschlussbericht für das Projekt assistierende, interaktive und sicher im industriellen Umfeld agierende ortsflexible Roboter** im BMBF-Rahmenkonzept " Forschung für die Produktion von morgen" für das Themenfeld: "Schlüsselkomponente Handhabungstechnik". Stuttgart, 2006
- Helms 2007 Helms, Evert: **Roboterbasierte Bahnführungsunterstützung von industriellen Handhabungs- und Bearbeitungsprozessen**. Heimsheim : Jost-Jetter, 2007. (IPA-IAO Forschung und Praxis; 451). (Stuttgart, Univ., Fak. Maschinenbau, Inst. für Industrielle Fertigung und Fabrikbetrieb, Diss. 2006)

- Henrichs 2006 Henrichs, D.; Gecks, T.: **Multi-Camera Collision Detection allowing Object Occlusions**. In: VDI-Wissensforum: ISR 2006 - ROBOTIK 2006 : Proceedings of the Joint Conference on Robotics, May 15-17, 2006, München: Visions are Reality. Düsseldorf, 2006. (VDI-Berichte 1956).
- Hershberger 1992 Hershberger, John; Snoeyink, Jack: **Speeding up the Douglas-Peucker line-simplification algorithm**. In P. Bresnahan et al., editors, Proc. 5th Intl. Symp. On Spatial Data Handling, volume 1, Charleston, SC, 1992
- Heuser 1990 Heuser, Harro: **Lehrbuch der Analysis. Teil 1**. 8. Auflage. Stuttgart: Teubner, 1990. Definition 34.6
- Hirzinger 2002 Hirzinger, G.; Sporer, N.; Albu-Schaffer, A.; Hahnle, M.; Krenn, R.; Pascucci, A.; Schedl, M: **DLR's torque-controlled light weight robot III - are we reaching the technological limits now?** Proceedings of the ICRA 2002 IEEE International Conference on Robotics and Automation. Washington, DC, Mai 2002
- Hollmann 2008 Hollmann, Rebecca; Hägele, Martin: **The Use of Voice Control for Industrial Robots in Noisy Manufacturing Environments**. In: International Federation of Robotics: ISR 2008 : 39th International Symposium on Robotics, 15.-17. Oct. 2008, Seoul, Korea. Seoul, Korea, 2008, S. 14-18
- Huggenberger 2007 Huggenberger, Ulrich: **KUKA SmartGUI – Einfache Bedienung sowie schnelle und intuitive Programmierung**. In: Westkämper, Engelbert (Hrsg.) ; Verl, Alexander (Hrsg.) ; Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA: Industrieroboter schneller und einfacher programmieren : auch ohne Brain-to-Robot-Interface. Fraunhofer IPA Workshop F 157, 5. Dezember 2007, Stuttgart
- IFR 2009 International Federation of Robotics / Statistical Department: **World Robotics Industrial Robots 2009**. Frankfurt/Main, 2009
- Jiang 1994 Jiang, X. Y.; Bunke, H.: **Fast segmentation of range images into planar regions by scan line grouping**. Machine Vision and Applications, 1994

- Kazi 2005 Kazi, A.; Bunsendal, J.; Haag, D.; Baum, R.; Bischoff, R.: **Next Generation Teach Pendants for Industrial Robots**. In: Prassler, Erwin (Ed.) ; Lawitzky, Gisbert (Ed.) ; Stopp, Andreas (Ed.) ; Grunwald, Gerhard (Ed.) ; Hägele, Martin (Ed.) ; Dillmann, Rüdiger (Ed.) ; Iossifidis, Ioannis (Ed.): Advances in Human-Robot Interaction. Berlin u.a. : Springer, 2005, S. 176-177. (Springer Tracts in Advanced Robotics - STAR 14)
- Khalifa 2003 Khalifa, I., Moussa, M. and Kamel, M.: **Range image segmentation using local approximation of scan lines with application to cad model acquisition**. Machine Vision Applications 13(5-6). Springer-Verlag 2003.
- Kinkel 2009 Kinkel, Steffen: **Final report on the market potential of the developed new robotic technologies**. Deliverable ISE.4 des SMERobot-Projektes. 2009.
- Kleine 2007 Kleine, Oliver, Kinkel, Steffen, Jäger, Angela: **Flexibilität durch Technologieeinsatz. Nutzung und Erfolgswirkung flexibilitätssteigernder Technologien**. Mitteilungen aus der ISI-Erhebung zur Modernisierung der Produktion, Fraunhofer ISI. Karlsruhe 2007.
- Koeppe 2001 Koeppe, Ralf: **Robot Compliant Motion based on Human Skill**. Dissertation, Technische Wissenschaften ETH Zürich, Nr. 14133, 2001.
- Koeppe 2007 Koeppe, Ralf: **Robotersysteme – Anwendungen aus Industrie und Servicerobotik**. Skript zur Vorlesung am ISW der Universität Stuttgart, 2007.
- Krause 2009 Krause, Detlef: **Offline-Programmierung – Chance für kleine und mittlere Unternehmen?** In: Westkämper, Engelbert (Hrsg.) ; Verl, Alexander (Hrsg.) ; Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA: Bearbeiten mit Industrierobotern : Technologien - Anwendungen - Hardware/Software. Fraunhofer IPA Workshop F 187, 9. Nov. 2009, Stuttgart. Stuttgart : FpF - Verein zur Förderung produktionstechnischer Forschung, 2009. (Fraunhofer IPA Workshop; F 187)
- Krüger 2009 Krüger, J.; Lien, T.K.; Verl, A.: **Cooperation of human and machines in assembly lines**. In: CIRP Annals - Manufacturing Technology, Volume 58, Issue 2, 2009.
- KUKA 2008 KUKA Roboter GmbH: **Dokumentation KUKA Roboter und Steuerung, speziell „Programmierung Experte“ und „ArcTech“**. Augsburg, 2008.

- KUKA 2009 KUKA Roboter GmbH: **Produktblatt KUKA Inline-Messtechnik**.  
[http://www.kuka.com/nl\\_media/06/datenblatt\\_Messtechnik.pdf](http://www.kuka.com/nl_media/06/datenblatt_Messtechnik.pdf)  
(20. August 2009)
- Lee 1995 Lee, J-S.; Sun, Y-N.; Chen, C-H.: **Multiscale corner Detection by Using Wavelet Transform**. In: IEEE Transactions on image processing, Band 4, Nr. 1, Januar 1995.
- Leeser 1994 Leeser, K. ; Donoghue, J.P.; Townsend, W.T.: **Computer-Assisted Teach and Play: Novel User-Friendly Robot Teach Mode Using Gravity Compensation**. The Proceedings of the Robotics International/SME/Fifth World Conference on Robotics Research. 1994.
- Lego 2009 The Lego Group: Homepage Produktbereich **Mindstorms**.  
<http://mindstorms.lego.com/> (15. Januar 2009)
- Liangyu 2007 Liangyu, Li; Lingjian, Fu; Xin, Zhou; Xiang, Li: **Image Processing of Seam Tracking System Using Laser Vision**. In: T.-J. Tarn et al. (Eds.): Lecture Notes in Control and Information Sciences, 2007, Volume 362, Robotic Welding, Intelligence and Automation, Pages 319-324. Berlin: Springer, 2007
- Lozano 1983 Lozano-Perez, T.: **Robot programming**. In: Proceedings of the IEEE, Volume: 71, Issue: 7, S. 821 – 841. Juli 1983.
- MacGregor 2009 MacGregor, Campbell: **Robots to get their own operating system**. In: The New Scientist, Volume 203, Issue 2720, 5 August 2009, Pages 18-19. München: Reed Business, 2009
- Marshall 2004 Marshall, David: **ABB TeachSaver Revolutinizes Foundry Cleaning Operations**. In: Industrial Robots, Volume 31 issue 2. Bingley: Emerald, 2004
- Matthias 2008 Matthias, Björn; Kirsten, René: **Industrieroboter als Produktionsassistenten für die Automobilmontage**. Präsentation auf dem Preetag des Forschungszentrums Ladenburg der ABB Group, 17. November 2008.

- Meißner 2000      Meißner, Alexander: **Aufgabenorientierte Programmierung von Bewegungsbahnen für Großroboter mit redundanter Gelenkarmkinematik.** Heimsheim : Jost-Jetter, 2000. (IPA-IAO Forschung und Praxis; 321). (Stuttgart, Univ., Fak. Konstruktions- und Fertigungstechnik, Inst. für Industrielle Fertigung und Fabrikbetrieb, Diss. 2000)
- Meyer 2006      Meyer, Christian; Schraft, Rolf Dieter: **An Intuitive Teaching Method for Small and Medium Enterprises.** In: Pham, Duc Truong (Hrsg.) ; Eldukhri, Eldaw E. (Hrsg.) ; Soroka, Anthony. J. (Hrsg.) ; FP6 I\*PROMS Network of Excellence: Intelligent Production Machines and Systems : 2nd I\*PROMS Virtual International Conference, 3-14 July 2006. Amsterdam u.a. : Elsevier, 2006, S. 568-571
- Meyer 2007-1      Meyer, Christian ; Hollmann, Rebecca ; Parlitz, Christopher ; Hägele, Martin: **Programmieren durch Vormachen für Assistenzsysteme - Schweiß- und Klebbahnen intuitiv programmieren.** In: it - Information Technology. 49 (2007), Nr. 4, S. 238-246
- Meyer 2007-2      Meyer, Christian ; Hägele, Martin: **Roboter als Schrittmacher der Produktion : Aktuelle Entwicklungen der Industrierobotik für heute und morgen.** In: A&D Automation & Drives. (2007), Select Robotik & Automation, S. 26-29
- Meyer 2010      Meyer, Christian: **CAD, CAM und Industrieroboter - Automatische Programmengenerierung für Schweiß- und Fräsprozesse.** In: Gesellschaft für Fertigungstechnik: Stuttgarter Impulse - Fertigungstechnik für die Zukunft: Schriftliche Fassung der Vorträge zum Fertigungstechnischen Kolloquium am 29. bis 30. September 2010 in Stuttgart. Stuttgart, 2010, S. 427-438
- MRL 2006      Europäische Union: **RICHTLINIE 2006/42/EG** DES EUROPÄISCHEN PARLAMENTS UND DES RATES. Amtsblatt der Europäischen Union, Mai 2006
- Neuronics 2009      Pressevorstellung: **Im Dienste der Menschen.** In: X-Technik Automation, Ausgabe 7, November 2009, Seite 62. X-technik IT & Medien GmbH. Wilhering, 2009
- Nikon 2010-1      Nikon Metrology Produktinformationen: **ModelMaker MMDx/MMC** [http://www.nikonmetrology.com/handheld\\_scanners/mmdx\\_mmc/](http://www.nikonmetrology.com/handheld_scanners/mmdx_mmc/) (24. Juni 2010)

- Nikon 2010-2      Nikon Metrology Produktinformationen: **RCA Robot CMM Arm**  
[http://www.nikonmetrology.com/products/robot\\_cmm\\_arm/rca/](http://www.nikonmetrology.com/products/robot_cmm_arm/rca/)  
(24. Juni 2010)
- Noruk 2007      Noruk, J.; Garner, R.: **How to keep automated welding from going  
around in Circles.** WebSite TheFabricator.com, July 2007.  
[http://www.thefabricator.com/Robotics/Robotics\\_Article.cfm?ID=1659](http://www.thefabricator.com/Robotics/Robotics_Article.cfm?ID=1659)  
(10. Juni 2007)
- Oberer-Treitz  
2008-1      Oberer-Treitz, Susanne; Meyer, Christian; Verl, Alexander: **Kollisionsbe-  
wertung bei der Mensch-Roboter-Kooperation : Auswertung einer  
Crash Test Versuchsreihe mit einem Industrieroboter und einem Crash  
Test Dummy.** In: Verl, Alexander (Vorsitz) u.a.; VDI/VDE-Gesellschaft  
Meß- und Automatisierungstechnik u.a.: **ROBOTIK 2008 : Leistungsstand -  
Anwendungen - Visionen - Trends.** Tagung München, 11. u. 12. Juni 2008.  
Düsseldorf : VDI Verlag, 2008, 16 S. (CD-ROM), Abstract S. 219-222 (Band  
(VDI-Berichte 2012)
- Oberer-Treitz  
2008-2      Oberer-Treitz, Susanne; Schüssler, Matthias: **Robotergestützte Vibrati-  
onsvermessung von Karosserien läuft völlig mannos.** In: MM - Das In-  
dustrieMagazin. (2008), Nr. 22, S. 86-89
- OSHA 2008      U.S. Department of Labor der Vereinigten Staaten von Amerika, Occupa-  
tional Safety & Health Administration: **OSHA Technical Manual.**  
[http://www.osha.gov/dts/osta/otm/otm\\_iv/otm\\_iv\\_4.html](http://www.osha.gov/dts/osta/otm/otm_iv/otm_iv_4.html). (30. De-  
zember 2008)
- Paintmate 2009      Homepage der Fa. **Paint mate Robotics AB**, Schweden.  
<http://www.paintmate.se/>. (23. Januar 2009)
- Pilz 2009      Pilz: Produktinformation **SafetyEYE®. Perfect 3D Protection.**  
[http://www.pilz.com/products/sensors/camera/f/safeyeye/  
index.de.jsp;jsessionid=7C57DEC16405DD336D3824626BC57397](http://www.pilz.com/products/sensors/camera/f/safeyeye/index.de.jsp;jsessionid=7C57DEC16405DD336D3824626BC57397)  
(20. Juni 2010)
- PISA 2008      Fraunhofer IPK: Homepage des EU-Projektes **PISA: Flexible Assembly  
Systems through Workplace-Sharing and Time-Sharing Human-Machine  
Cooperation.** <http://www.pisa-ipk.org/>. (18. Dezember 2008)

- Press 1992 Press, William H.; Teukolsky, Saul A.; Vetterling, William T.; Flannery, Brian P.: **NUMERICAL RECIPES IN C: THE ART OF SCIENTIFIC COMPUTING**. S. 680. Cambridge: Cambridge University Press, 1992
- Rabbani 2006 Rabbani, T.; van den Heuvelb, F. A.; Vosselman, G.: **Segmentation of Point Clouds using Smoothness Constraints**. IAPRS Volume XXXVI, Part 5, Dresden 25.-27. September 2006
- Rahayem 2008 Rahayem, M.; Kjellander, J.A.P. & Larsson, S.: **Geometric Reverse Engineering using a Laser Profile Scanner mounted on a Industrial Robot**. Tagungsband 6th International DAAAM Baltic Conference on INDUSTRIAL ENGINEERING. Tallinn, Estonia, 2008
- Rawiel 2006 Schutzrecht EP 0 963 816 B1: **Verfahren zum Erfassen und Kompensieren von kinematischen Veränderungen eines Roboters**. Rawiel, Gerd; Wiest, Ulrich; Verl, Alexander; Pieters, Stefan. Europäisches Patentamt, 2006
- Reinhart 2008 Reinhart, G.; Munzert, U.; Vogl, W.: **A programming system for robot-based remote-laser-welding with conventional optics**. In: CIRP Annals - Manufacturing Technology, Volume 57, Issue 1, 2008, Pages 37-40. Amsterdam: Elsevier, 2008
- Rinas 2009 Rinas: Homepage der Fa. **KPS Rinas BV**. <http://www.rinas.dk/>. (3. Januar 2009)
- Sappa 2001 Sappa, A. D. and Devy, M.: **Fast range image segmentation by an edge detection strategy**. In: Third International Conference on 3-D Digital Imaging and Modeling, 2001
- Schäfer 2007 Schäfer, Reinhold: **Vereinfachte Roboter-Programmierung durch Vorführen und Sprachbefehle (12.12.2007)**. <http://www.maschinenmarkt.vogel.de> (03. Januar 2010)
- Schal 2002 Schal, Willy (Hrsg.): **Fertigungstechnik 2**. Hamburg: Verlag Handwerk und Technik GmbH, 2002
- Schnabel 2007 Schnabel, R.; Wahl, R.; Klein, R.: **Efficient RANSAC for Point-Cloud Shape Detection**. Computer Graphics Forum, Band 26-2. The Eurographics Association and Blackwell Publishing Ltd, 2007.

- Schraft 1998 Schraft, Rolf Dieter ; Schmierer, Gernot: **Serviceroboter: Produkte, Szenarien, Visionen**. Berlin; Heidelberg: Springer, 1998
- Schraft 2005 Schraft, Rolf Dieter; Meyer, Christian; Parlitz, Christopher; Helms, Evert: **PowerMate – A Safe and Intuitive Robot Assistant for Handling and Assembly Tasks**. In: IEEE / Robotics and Automation Society: 2005 IEEE International Conference on Robotics and Automation (ICRA) / DVD : Robots get closer to humans. DVD Conference and Video Proceedings. Barcelona, April 18-22, 2005. Piscataway, NJ, USA : IEEE Press, 2005, S. 4085-4090
- Schraft 2006 Schraft, Rolf Dieter, Meyer, Christian: **The need for an intuitive teaching method for small and medium enterprises**. In: VDI-Wissensforum: ISR 2006 - ROBOTIK 2006 : Proceedings of the Joint Conference on Robotics, May 15-17, 2006, München: Visions are Reality. Düsseldorf, 2006, 10 S. (CD-ROM), Abstract S. 95 (Band). (VDI-Berichte 1956)
- Schulte 2003 Schulte, Ralf, Wechselbaum, Robert, Vater, Bernd: **Manutec lebt**. Automation & Qualität, 10. Jahrgang, Sonderdruck Juni 2003.
- Schwarz 1997 Schwarz, Hans Rudolf: **Numerische Mathematik**. B. G. Teubner-Verlag, Stuttgart, 1997.
- ServoRobot 2010 ServoRobot: **ROBO-TRAC** Produktbeschreibung.  
<http://www.servorobot.com/en/manufacturing-solutions/seam-finding-tracking-adaptive-control/robo-trac.html> (20. Juni 2010)
- Seyfarth 2004 Seyfarth, Markus: **Aufzeichnung von TCP-Bewegung mit Datenreduktion**. Reis RSV Dokumentation. Obernburg, 2004.
- SMErobot 2008 Fraunhofer IPA: Homepage des EU-Projektes **SMErobot: The European Robot Initiative for Strengthening the Competitiveness of SMEs in Manufacturing**. <http://www.smerobot.org>. (12. Dezember 2008)
- Som 2007 Som, Franz: **Robotersteuerung mit neuartigen Bedien- und Programmierfunktionen**. In: Westkämper, Engelbert (Hrsg.) ; Verl, Alexander (Hrsg.) ; Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA: Industrieroboter schneller und einfacher programmieren : auch ohne Brain-to-Robot-Interface. Fraunhofer IPA Workshop F 157, 5. Dezember 2007, Stuttgart

- Sperschneider 2002 Sperschneider, Volker; Hammer, Barbara: **Theoretische Informatik. Eine problemorientierte Einführung**. Berlin: Springer, 2002
- Teh 1989 Teh, Cho-Huak; Chin, Roland T.: **On the Detection of Dominant Points on Digital Curves**. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 11, No. 8, August 1989
- Thiemermann 2005 Thiemermann, Stefan: **Direkte Mensch-Roboter-Kooperation in der Kleinteilemontage mit einem SCARA-Roboter**. Heimsheim : Jost-Jetter Verlag, 2005. (IPA-IAO Forschung und Praxis; 411). (Stuttgart, Univ., Fak. Maschinenbau, Inst. für Industrielle Fertigung und Fabrikbetrieb, Diss. 2005)
- Treffler 2008 Maschinenbau Treffler: **Homepage der Fa. Maschinenbau Treffler**, <http://www.maschinenbau-treffler.de/>. (12. Dezember 2008)
- Umbreit 2007 Umbreit, Matthias: **Normativer Rahmen für assistierende Roboter nach EN ISO 10218-1 und prEN ISO 10218-2**. Vortrag auf Workshop der Berufsgenossenschaft Metall-Süd zur Mensch-Roboter-Kooperation. 2. August 2007, Lengfurt
- VC 2008 Visual Componentes: Homepage der Fa. **VisualComponents Oy**: <http://www.visualcomponentes.com>. (27. Februar 2009)
- Verl 2007 Verl, Alexander (Herausgeber-Beirat); Schraft, Rolf Dieter (Editorial Board): **Industrial Robot: The international journal of industrial and service robots. Special Issue: Welding Industrial Robot**. Bradford, UK : Emerald Group Publishing Ltd, 2007
- Verl 2008 Verl, Alexander; Naumann, Martin: **Automatic Generation of Executable Code for a Robot Cell using UPnP and XIRP**. In: Institute for Systems and Technologies of Information, Control and Communication: Fifth International Conference on Informatics in Control, Automation and Robotics : ICINCO 2008, 11.-15. May, 2008, Funchal, Madeira. Setubal, Portugal, 2008, S. 242-248
- Verl 2009 Schutzrecht DE 102004024378B2 (20. Mai 2009). Verl, Alexander: **Verfahren zur robotergestützten Vermessung von Objekten**.

- Vogl 2009 Wolfgang Vogl: **Eine interaktive räumliche Benutzerschnittstelle für die Programmierung von Industrierobotern**. Dissertation an der TU München, iwb. Herbert Utz Verlag, 2009.
- Vosselman 2004 Vosselman, G.; Gorte, B.; Sithole G; Rabbani, T.: **Recognizing structure in laser scanner point clouds**. In: Proceedings of Conference on Laser scanners for Forest and Landscape assessment and instruments, Freiburg, Germany, 2004.
- Walter 1997 Walter, Wolfgang: **Analysis 1**. Berlin: Springer, 1997.
- Weck 2006 Weck, Manfred; Brecher, Christian: **Werkzeugmaschinen, Fertigungssysteme – Band 4: Automatisierung von Maschinen und Anlagen**. 6 Aufl. Berlin: Springer, 2006
- Westkämper 2006-1 Westkämper, Engelbert; Warnecke, Hans-Jürgen: **Einführung in die Fertigungstechnik**. 7. Auflage. Wiesbaden, Teubner, 2006
- Westkämper 2006-2 Westkämper, Engelbert: **Einführung in die Organisation der Fertigung**. Berlin, Springer, 2006
- Westkämper 2006-3 Westkämper, Engelbert; Stotz, Martin; Effenberger, Ira: **Automatische Segmentierung von Messpunktwolken in regelgeometrische Elemente**. : Technisches Messen - tm. 73 (2006), Nr. 1, S. 60-66
- Wörn 2005 Wörn, Heinz; Brinkschulte, Uwe: **Echtzeitsysteme – Grundlagen, Funktionsweisen, Anwendungen**. Berlin: Springer, 2005
- Wörn 2008 Wörn, Heinz; Schmid, Andreas: **Intuitive Human-Robot Cooperation**. Proceedings of the 11th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines CLAWAR. 2008
- Wilke 2002 Wilke, W.: **Segmentierung und Approximation großer Punktwolken**, Dissertation <http://elib.tu-darmstadt.de/diss/000255/> (25.05.2007)

- Winkler 2008      Winkler, Bernd: **Konzept zur Sicherer Mensch-Roboter-Kooperation auf Basis von Schnellen 3-D Time-of-Flight Sensoren.** In: Verl, Alexander (Vorsitz) ; Hägele, Martin (Programmkomitee) ; u.a. ; VDI/VDE-Gesellschaft Meß- und Automatisierungstechnik: ROBOTIK 2008 : Leistungsstand - Anwendungen - Visionen - Trends. Tagung München, 11. u. 12. Juni 2008. Düsseldorf : VDI Verlag, 2008
- Xiang 2004      Xiang, Rihua; Wang, Runsheng: **Range image segmentation based on split-merge clustering.** In: Proceedings of the 17th International Conference on Pattern Recognition ICPR. August 2004
- Zäh 2005      Zäh, M. F.; Vogl, W.; Patron, C.: **Interaktive Planung von Produktionssystemen mittels Augmented Reality.** Wt Werkstattstechnik online Jahrgang 95 (2005) H. 9
- Zürn 2008      Zürn, M.; Brandstetter, T; Steegmüller, D.: **Assembly 21: innovative system and assembly concept for rear axle assembly for the new C-Class with cooperating robot teams.** IEEE-IFR Joint Forum on Innovation and Entrepreneurship in Robotics and Automation, München, 11. Juni 2008

