

Franz-Josef Zeller

*Sensorplanung und schnelle Sensorregelung
für Industrieroboter*

Franz-Josef Zeller

*Sensorplanung und schnelle
Sensorregelung für Industrieroboter*

Herausgegeben von
Professor Dr.-Ing. Klaus Feldmann,
Lehrstuhl für
Fertigungsautomatisierung und Produktionssystematik

FAPS



Carl Hanser Verlag München Wien

Als Dissertation genehmigt von der Technischen Fakultät
der Friedrich-Alexander-Universität Erlangen-Nürnberg

Tag der Einreichung:	12. Juni 1995
Tag der Promotion:	18. Sept. 1995
Dekan:	Prof. Dr. Dr. h.c. F. Durst
Berichterstatter:	Prof. Dr.-Ing. K. Feldmann apl. Prof. Dr.-Ing. habil. W. Bär

Die Deutsche Bibliothek - CIP - Einheitsaufnahme

Zeller, Franz-Josef:

Sensorplanung und schnelle Sensorregelung für Industrieroboter
/ Franz-Josef Zeller. Hrsg. von Klaus Feldmann. - München ;
Wien : Hanser, 1996

(Fertigungstechnik - Erlangen ; 51)

Zugl.: Erlangen, Nürnberg, Univ., Diss., 1995

ISBN 3-446-18601-8

NE: GT

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdrucks
und der Vervielfältigung des Buches oder Teilen daraus,
vorbehalten.

Kein Teil des Werkes darf ohne schriftliche Genehmigung des
Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein
anderes Verfahren), auch nicht für Zwecke der Unterrichts-
gestaltung - mit Ausnahme der in den §§ 53, 54 URG ausdrücklich
genannten Sonderfälle -, reproduziert oder unter Verwendung
elektronischer Systeme verarbeitet, vervielfältigt oder
verbreitet werden.

© Carl Hanser Verlag München, Wien 1995

Herstellung: Gruner Druck GmbH, Erlangen-Eltersdorf

Printed in Germany

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Assistent am Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik der Friedrich-Alexander-Universität Erlangen-Nürnberg.

Herrn Professor Dr.-Ing. K. Feldmann, dem Leiter dieses Lehrstuhls am Institut für Fertigungstechnik, danke ich herzlich für die wohlwollende und großzügige Förderung dieser Arbeit. Mein besonderer Dank gilt ebenfalls Herrn apl. Professor Dr.-Ing. habil. W. Bär, Lehrstuhl für Regelungstechnik der Universität Erlangen-Nürnberg, für die Übernahme des Korreferates und für die Betreuung von Studien- und Diplomarbeiten am Institut für Elektrotechnik. Herrn Professor Dr. F. Hofmann, Inhaber des Lehrstuhls für Informatik IV (Betriebssysteme) der Universität Erlangen-Nürnberg, danke ich für die Unterstützung bei der Durchführung des Promotionsverfahrens.

Viele komplexe Aufgaben im technischen Bereich lassen sich heute nur mehr im Team erfolgreich bewältigen. Daher möchte ich an erster Stelle meinem Kollegen, Herrn Dipl.-Ing. U. Schönherr, danken, der in perfekter Teamarbeit zur gemeinsamen Bewältigung zahlreicher Projekte beitrug und durch seine unermüdliche Diskussionsbereitschaft großen Anteil am Gelingen der Arbeit hat. Ferner gilt mein Dank allen Kolleginnen und Kollegen, speziell jenen der Forschungsgruppe für Steuerungs- und Sensortechnik, für die stets offenen fachlichen Diskussionen und die effektive Zusammenarbeit. Weiterhin danke ich den Studenten und wissenschaftlichen Hilfskräften, die mich beim Erstellen dieser Arbeit unterstützt haben. Hier sind besonders Herr Dipl.-Ing. R. Wolf, Herr Dipl.-Ing. P. Feulner, Herr Dipl.-Ing. D. Kögel, Herr Dipl.-Ing. J. Albert und Herr Dipl.-Inf. H. Pölsterl zu nennen.

Herrn Dr. H.-J. Thon und Herrn Dipl.-Ing. M. Zäh, Mitarbeiter der Siemens AG, sowie der Bayerischen Forschungsstiftung danke ich für die großzügige Förderung des Vorhabens.

Mein ganz besonderer Dank gilt jedoch meiner Frau Birgit, die mich stets bestärkte und durch ihr fortwährendes, großes Verständnis wesentlich zum Gelingen dieser Arbeit beigetragen hat, sowie meinen beiden Kindern Florian und Caroline. Nicht zuletzt und besonders herzlich danke ich mich bei meinen Eltern, die mich immer nach besten Kräften unterstützt haben.

Sensorplanung und schnelle Sensorregelung für Industrieroboter

- Inhaltsverzeichnis -

Abkürzungen und Formelzeichen	IV
1. Einführung.....	1
2. Situationsanalyse der Sensorintegration in heutigen Robotersteuerungen	4
2.1 Nutzenanalyse der Sensorintegration.....	4
2.1.1 Technologisch - ökonomischer Nutzen	4
2.1.2 Kompensation von Roboterungenauigkeiten	7
2.1.3 Kompensation von Umgebungsungenauigkeiten.....	11
2.1.4 Programmierung.....	14
2.1.5 Direkte Regelung von Prozeßgrößen	19
2.2 Formen der Sensorintegration	20
2.2.1 Klassifikation der Formen der Sensorintegration	20
2.2.2 Korrektur von Positionen in Roboterprogrammen.....	25
2.2.3 Sensorgestützte Off-line-Bahnprogrammierung.....	26
2.2.4 On-line-Bahnadaption durch Sensorregelung	26
2.2.5 Gesteuerte Bahnadaption mit vorlaufendem Sensor.....	27
2.2.6 Bahnadaption durch Zusatzachsen	29
2.2.7 Bewertung der Verfahren	30
2.3 Defizite bestehender Systeme	32
3. Definition des Anforderungsprofils eines Sensor-Roboter-Systems	34
3.1 Technologische Zieldefinition.....	34
3.2 Globale Systemziele	40
3.3 Anforderungen an die Sensorik	40
3.4 Aufgabenprofil einer Robotersteuerung.....	41
3.5 Forderungen an einen Sensorrechner	43
3.6 Anforderungsprofil eines integrierten Gesamtsystems	44

4. Entwurf eines konfigurierbaren Gesamtsystems zur Sensorintegration	46
4.1 Struktur des integrierten Gesamtsystems ISRS	46
4.1.1 Struktureller Aufbau des Gesamtsystems	46
4.1.2 Hardwareaufbau	50
4.1.3 Funktionsmerkmale der Robotersteuerung	52
4.1.4 Einsatz eines Echtzeit-Multitasking-Kerns	53
4.2 Entwurf des Sensorsteuerungssystems SC	54
4.2.1 Aufbau des Sensorsteuerungssystems SC	54
4.2.2 Parametrierung des Sensorsteuerungssystems	61
4.3 Transputersystem zur schnellen Sensorverarbeitung	63
4.3.1 Entwurf des Softwaresystems	63
4.3.2 Abbildung des Softwaresystems auf ein Transputercluster	66
4.3.3 Kopplung des Transputersystems mit der Sensorsteuerung SC	68
4.4 Programmentwicklung und Simulation des Programmablaufs	69
4.4.1 Strukturierung des Simulationssystems	70
4.4.2 Merkmale der Funktionseinheiten	72
4.4.3 Einsatzverhalten des Simulationssystems	78
4.4.4 Erstellung von Roboterprogrammen	79
4.5 Entwurf und Simulation von Sensorregelkreisen	80
5. Adaption der Bewegungsbahn durch Sensorregelung im Interpolationstakt	82
5.1 Modellierung des kartesischen Sensorregelkreises	82
5.1.1 Struktur des kartesischen Sensorregelkreises	82
5.1.2 Modellierung eines lagegeregelten Industrieroboters	84
5.2 Identifikation der kartesischen Regelstrecke	91
5.2.1 Verfahren zur Bestimmung der Modellparameter	91
5.2.2 Ergebnisse der Systemidentifikation	93
5.3 Entwurf und Untersuchung von Regelungskonzepten	95
5.3.1 Zielsetzung und Randbedingungen	95
5.3.2 Untersuchung von parameteroptimierten Reglern	96
5.3.3 Untersuchung von strukturoptimalen Reglern	99
5.3.4 Vergleich der untersuchten Regelungen	104
5.4 Bewertung der kartesischen Sensorregelung	105
5.4.1 Ableitung und Validierung des Schleppfehlers	105
5.4.2 Einsatzbeispiele der kartesischen Sensorregelung	111

6. Schnelle Bahnkorrektur durch Sensorregelung im Lageregeltakt	112
6.1 Strukturierung des Transputersystems	112
6.2 Modell des axialen Sensorregelkreises	117
6.3 Untersuchung des schnellen Sensorregelkreises	121
6.4 Vergleich und Bewertung der Sensorregelung im Interpolations- und Lageregeltakt	124
7. Flexible Integration multisensorieller Daten	129
7.1 Motivation und Defizite der Integration multisensorieller Daten	129
7.2 Allgemeine Strukturierung von Multisensorsystemen	131
7.3 Entwurf eines Systems zur Integration multisensorieller Daten	133
7.3.1 Anforderungsprofil eines Multisensorsystems	133
7.3.2 Hierarchische Strukturierung	134
7.3.3 Schnittstellen und Datenflüsse	136
7.3.4 Modellierung mit logischen Einheiten	138
7.3.5 Einbindung des Systems in ein Echtzeitsystem	140
7.4 Flexible Konfiguration des Multisensorsystems	141
8. Rechnergestützte Planung und Auswahl von Sensoren	143
8.1 Problematik der Sensorauswahl und -planung	143
8.2 Systematik zur Auswahl von Sensoren	144
8.3 Meßprinziphierarchie zur Strukturierung des Auswahlvorgangs	145
8.4 Rechnergestütztes Auswahlsystem	146
8.4.1 Struktur des Auswahlsystems	147
8.4.2 Entscheidungsfindung	148
8.4.3 Wissensbasis	150
8.4.4 Anwenderschnittstelle	151
9. Zusammenfassung	153
Literaturverzeichnis	155

Abkürzungen und Formelzeichen

Abkürzungen:

ADW	Analog-Digital-Wandler
CAD	Computer Aided Design
CP	Continous Path
CPU	Central Processing Unit
DPR	Dual-Ported-RAM
EDV	Elektronische Datenverarbeitung
FIPO	Feininterpolator
GI	Gleichung
IPO	Interpolation
IRL	Industrial Robot Language
ISRS	Integriertes Sensor-Roboter-System
LR	Lageregler
MKS	Mehrkörpersystem
NC	Numeric Control
PC	Personal Computer
PTP	Point To Point
RAM	Random Access Memory
RC	Robotersteuerung (Robot Control)
RC-SIM	Robotersimulator
SBC	Einplatinen-Rechner (Single-Board-Computer)
SC	Sensorsteuerung (Sensor Control)
TCP	Werkzeugbezugspunkt (Tool Center Point)
ZKS	Zentrale Kommunikation und Steuerung

Formelzeichen:

$A(z)$	Nennerpolynom der z-Übertragungsfunktion
\underline{A}	Matrix A
$\hat{A}(z)$	Nennerpolynom der geschätzten z-Übertragungsfunktion
a_i	Koeffizient des Nennerpolynoms $A(z)$
a_{ij}	Splinekoeffizienten
α	Rampenwinkel
α_i	Winkel des i. Gelenks

$B(z)$	Zählerpolynom der z-Übertragungsfunktion
$\hat{B}(z)$	Zählerpolynom der geschätzten z-Übertragungsfunktion
b_i	Koeffizient des Zählerpolynoms $B(z)$
d	Ordnung der Systemtotzeit
d_i	Zählerkoeffizient der diskreten Reglerübertragungsfunktion
e	Fehler, Schleppfehler, Regelabweichung
F	Kraft
$F(s)$	Laplace-Übertragungsfunktion
F_A	Fläche des Anwenderprofils
F_S	Schnittfläche
f	Funktion f
f^{-1}	Inverse Funktion der Funktion f
f_{45°	Frequenz im Bodediagramm bei einer Phase von -45°
f_{ij}	Splinefläche im Segment (i,j)
$G(z)$	z-Übertragungsfunktion
G_j	Gewichtungsfaktor des j . Übereinstimmungsmaßes
$G_{PI}(z)$	z-Übertragungsfunktion des PI-Reglers
$G_{PID}(z)$	z-Übertragungsfunktion des PID-Reglers
$G_R(z)$	z-Übertragungsfunktion des Reglers
$G_S(z)$	z-Übertragungsfunktion der Strecke
i	Laufindex
\underline{J}	Jacobi-Matrix
j	Laufindex
K_i	Koordinatensystem im Körper i
K_{Off}	Kosten der Off-line-Programmierung
K_{On}	Kosten der On-line-Programmierung
K_P	Lagereglerversärkung
K_{Sens}	Kosten der sensorgestützten Programmierung
K_V	K_V -Wert
K_{VS}	K_V -Wert der Sensorregelung
K_{VSDB}	K_V -Wert der Sensorregelung mit Deadbeat-Regler
K_{VSPI}	K_V -Wert der Sensorregelung mit PI-Regler
k	diskrete Zeitpunkte
L	Lagrange-Funktion
λ	Lagrange-Multiplikator
M	Moment
$m+d$	Ordnung des Systems
$P(z)$	z-Übertragungsfunktion des deadbeatgeregelten Regelkreises

p_i	Koeffizient von $P(z)$
q	verallgemeinerter Gelenkwinkel
q_{ist}	verallgemeinerter Gelenkwinkelwert
q_{kor}	verallgemeinerter Gelenkwinkelkorrekturwert
q_{RC}	verallgemeinerter Gelenkwinkel der Bahnplanung
q_s	verallgemeinerter Gelenkwinkelsollwert
S_{Prog}	Stundensatz des Programmierarbeitsplatzes
S_{Rob}	Stundensatz der Roboteranlage
S_{Sens}	Stundensatz der sensorgestützten Roboteranlage
s	Laplace-Operator
jT_i	Homogene Transformation vom Koordinatensystem i ins Koordinatensystem j
T_A	Abtastzeit
T_{An}	Anregelzeit
T_{Aus}	Ausregelzeit
T_D	Ersatzzeitkonstante des Drehzahlregelkreises
T_G	Ausgleichszeit
T_{IPO}	Interpolationszykluszeit
T_{kor}	Zeit für die Programmkorrektur
T_L	Ersatzzeitkonstante des Lageregelkreises
T_{Mod}	Zeit für die Modellerstellung
$T_{ProgOff}$	Zeit für die Off-line-Programmerstellung
T_{ProgOn}	Zeit für die On-line-Programmerstellung
$T_{ProgSens}$	Zeit für die Sensorprogrammerstellung
T_{rD}	resultierende Ersatzzeitkonstante des symmetrierten Drehzahlregelkreises
T_T	Totzeit
U	Übereinstimmungsmaß
U_{ij}	Übereinstimmungsmaß des i . Anwenderprofils mit dem j . Meßprinzipprofil
u	Stellgröße
v	Geschwindigkeit
v_B	programmierte Bahngeschwindigkeit
v_Z	Korrekturgeschwindigkeit
ω	Kreisfrequenz
x	kartesische Koordinate
x_{ist}	kartesische Istkoordinate
x_{kor}	kartesische Korrekturkoordinate
x_s	kartesische Sollkoordinate
\underline{x}	Vektor x
z	diskrete Bildvariable

1. Einführung

Das Handeln heutiger Unternehmen ist stark geprägt durch eine zunehmende Ausrichtung auf den Kunden und dessen Wünsche [41, 140]. Dies führt im wesentlichen dazu, daß die Produktentwicklungszeiten drastisch reduziert werden müssen und die Produktqualität zu steigern ist. In Verbindung mit einer erhöhten Produktivität, der Einsparung von Kosten und einer wirkungsvollen Qualitätssicherung hängt der Erfolg eines Unternehmens zunehmend von verkürzten Durchlaufzeiten und einer höheren Flexibilität ab [116]. Weitreichende Änderungen des Fertigungsprinzips und eine Umgestaltungen der Fertigungsanlagen sind die Folge, um so schnell und flexibel reagieren und so am Markt bestehen zu können. Bei den hohen Lohnkosten in der Bundesrepublik Deutschland sind diese Bestrebungen nur durch eine angepaßte Automatisierung und damit eine Erhöhung der Produktivität lösbar [131]. Die Automatisierung fordert aber flexible, leistungsfähige und an die jeweilige Aufgabe kostengünstig adaptierbare und integrierbare Komponenten und Lösungen, um so die Produktivität zu steigern.

Industrieroboter besitzen eine hohe Flexibilität und können so bei geringen Stückzahlen und hoher Variantenvielfalt die verschiedenen Bearbeitungstechnologien und die Montage kostengünstig automatisieren. Die anzustrebende Flexibilität und die Adaptierbarkeit von Industrieroboteranlagen kann in vielen Fällen durch den Einsatz von Sensoren und den Aufbau von Regelkreisen verbessert oder erst sinnvoll erreicht und sichergestellt werden [45].

Bei der Integration von Sensoren treten einige Probleme und Defizite auf. Insbesondere sind im Augenblick geeignete, kostengünstige Sensoren, die die während eines Fertigungs- oder Montageprozesses bestehenden hohen Anforderungen an das Sensorsystem hinsichtlich Signalverarbeitungsgeschwindigkeit und Störsicherheit erfüllen, nur beschränkt verfügbar [110].

Die Planung und Auswahl geeigneter Sensoren ist bis heute durch die Suche in einer Vielzahl von Katalogen der meist klein- oder mittelständisch strukturierten Hersteller charakterisiert. Dies bedeutet, daß hier ein über viele Jahre erworbenes Erfahrungswissen des Projektierers notwendig ist. Insbesondere mangelt es an Systematiken zur Planung und Auswahl von Sensoren und an geeigneten rechnergestützten Systemen zur Durchführung dieser Arbeitsschritte [36, 40].

Heutige Industrierobotersteuerungen weisen keine bzw. nur eingeschränkte Schnittstellen zur Integration von Sensordaten und mangelnde Verarbeitungsmöglichkeiten auf Seiten der Robotersteuerung auf [136, 90].

Bestehende Systeme sind nicht an die jeweiligen Anforderungen der Bearbeitungstechnologie kostengünstig adaptierbar. Die mangelnde Flexibilität rührt daher, daß augenblickliche Lösungen meist speziell für eine Aufgabe projektiert und so nur sehr eingeschränkte Sensorschnittstellen geschaffen wurden. Insbesondere die bei Bearbeitungstechnologien mit hohen Bearbeitungsgeschwindigkeiten erforderlichen, schnellen Sensorschnittstellen fehlen gänzlich.

Deshalb ist es Ziel dieser Arbeit, ausgehend von der Analyse und der Charakterisierung bestehender Lösungen und den daraus abgeleiteten Anforderungen an eine flexible Sensorintegration, ein konfigurierbares und adaptierbares Sensorrechnersystem für Industrieroboter zu entwerfen und zu realisieren. Dieses System muß sowohl für eine einfache, zeitunkritische Sensordatenintegration adaptierbar sein, als auch die bei der schnellen Sensorregelung notwendigen Verarbeitungsgeschwindigkeiten ermöglichen. Das integrierte Gesamtsystem soll die Ausführenden einer Sensorregelung optimal bei der Programmierung, beim Reglerentwurf, beim Test und bei der Dokumentation unterstützen.

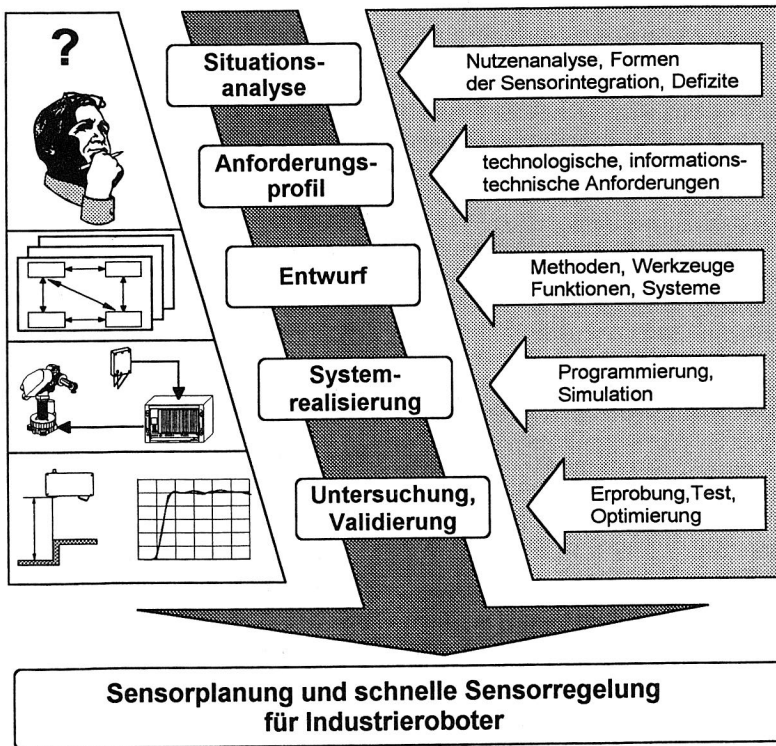


Bild 1: Vorgehensweise in dieser Arbeit

Insbesondere sind simulative und praktische Untersuchungen von Sensorregelkreisen mit unterschiedlichen Regelungs- und Integrationskonzepten vorzunehmen, um so präzise Aussagen zur erzielbaren Dynamik und der damit z.B. bei der sensorgestützten Konturbearbeitung erreichbaren Genauigkeit machen zu können. Die dabei durchgeführten, praktischen Arbeiten an

einer Modellanlage dienen gleichzeitig zur Validierung und Optimierung des integrierten Gesamtsystems.

Das Defizit der mangelnden Verfügbarkeit von geeigneten intelligenten Sensorsystemen für die Sensorregelung soll mit dem Entwurf eines flexiblen, konfigurierbaren Systems zur Fusion von multisensoriellen Daten überwunden werden. Damit ist es möglich, die von einfachen und kostengünstigen Sensoren gelieferten Einzelinformationen so zu fusionieren, daß verschiedenste, komplexe Bearbeitungsaufgaben sensorgeführt gelöst werden können.

Die bisher sehr mühsame Suche nach geeigneten Sensoren zur Lösung einer Meßaufgabe soll durch eine Systematik zur Auswahl und Planung von Sensoren vereinfacht werden. Dieses Schema läßt sich in ein rechnergestütztes Sensorauswahlsystem umsetzen, das die Projektierung eines Sensorsystems systematisiert und damit wesentlich vereinfacht.

2. Situationsanalyse der Sensorintegration in heutigen Robotersteuerungen

Gegenstand der Situationsanalyse ist die Untersuchung des aktuellen Zustands der Sensorintegration in heutigen, robotergestützten Fertigungsumgebungen. Dazu wird zunächst die Notwendigkeit des Sensoreinsatzes für Fertigungsanlagen allgemein und der daraus erwachsende technologisch-ökonomische Nutzen analysiert. Auf Basis der Nutzenanalyse werden die bisher bekannten Möglichkeiten der Sensorintegration und Sensorregelung strukturiert und charakterisiert, um anschließend Defizite der bestehenden Lösungen zu erarbeiten und zu bewerten.

2.1 Nutzenanalyse der Sensorintegration

Die Nutzenanalyse fokussiert zunächst globale, technisch-ökonomische Kenngrößen für die Wertigkeit der Sensorintegration in Fertigungsanlagen. In einem zweiten Schritt wird der Nutzen der Sensorik an den verschiedenen Einsatzbereichen innerhalb einer Fertigungsanlage detailliert untersucht.

2.1.1 Technologisch - ökonomischer Nutzen

In rechnergestützten Fertigungs- und Montageanlagen spielt die Integration von Sensoren eine immer entscheidendere Rolle [47]. Wesentliches Ziel der Integration von Sensoren ist der Aufbau von Regelkreisen, die Fehler und Toleranzen des Prozesses ausgleichen, die Haupt- und Nebenzeiten des Arbeitsgangs reduzieren und so eine kostengünstige, flexible, sichere und qualitativ hochwertige Herstellung des Produkts ermöglichen. Im Detail lassen sich die folgenden, wesentlichen Nutzenpotentiale der Sensorintegration aufzeigen.

Kosten:

Beim Einsatz von Sensoren in einer Fertigungsanlage entstehen zunächst zusätzliche Kosten, die meist aus Investitionskosten für die Sensorik, dem zusätzlichen Rechnersystem und der notwendigen Software bestehen. Oftmals kann aber dieser Kostensprung durch mögliche Einsparungen bei den Zuführsystemen oder Spannmitteln kompensiert werden [35, 57, 153]. Die Verminderung von Nebenzeiten wie z.B. Rüstzeiten, Programmierzeiten trägt zusätzlich zu einer Reduktion der Kosten bei.

Anlagen- und Prozeßsicherheit:

Mit Hilfe von Sensoren können nicht vorhersehbare Unbestimmtheiten an einer Fertigungsanlage erfaßt werden. Unter Verwendung dieser Daten kann der Prozeß an die neuen Gegebenheiten adaptiert werden oder, falls eine Adaption nicht mehr möglich ist, kann die Anlage sicher angehalten werden („fail safe“). So lassen sich z.B. unzulässige Abweichungen der realen

Umwelt gegenüber der modellierten, programmierten Welt erfassen und zur automatischen Programmanpassung heranziehen [65, 156].

Qualität:

Die gesteigerte Qualität der Produkte wird im wesentlichen durch die direkte Erfassung der qualitätsbestimmenden Größen mit Sensoren und deren Regelung erzielt. Bisher konnte oftmals nur dann ein hochqualitatives Produkt gefertigt werden, wenn mehrere indirekte Prozeßgrößen steuernd auf einem optimalen Sollwert gehalten werden und einwirkende Störgrößen definierte Maximalwerte nicht überschritten. Die direkte Regelung der Prozeßgrößen hat außerdem den Vorteil, daß die Programmierung und die Systemnutzung für den Technologen vereinfacht wird, da dieser die Sollwerte der Prozeßgrößen meist kennt [42, 65]. Die Zahl der Programmier- und Bedienerfehler reduziert sich folglich.

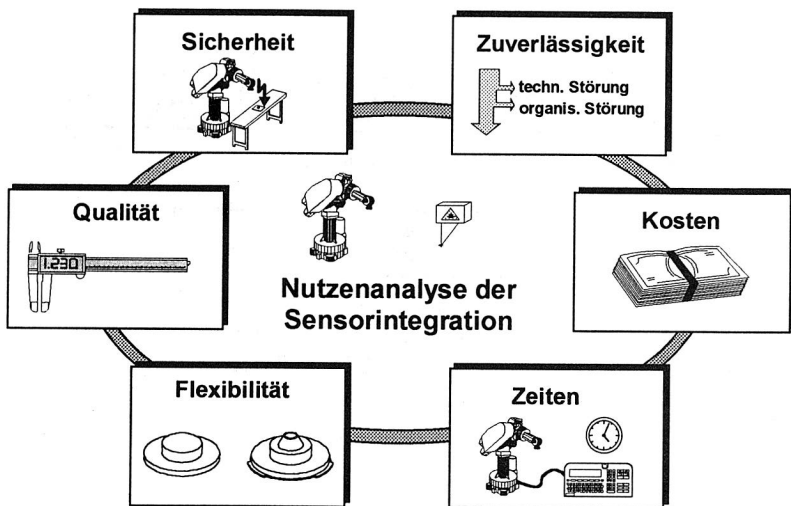


Bild 2: Technologisch - ökonomischer Nutzen der Sensorintegration

Flexibilität:

Flexibilität läßt sich nach [74] in Umrüst- und in Umbau-Flexibilität untergliedern. Durch den Einsatz von Sensoren können notwendige Umrüstungen gänzlich vermieden oder wesentlich vereinfacht werden. So lassen sich durch eine automatische Sensorkorrektur bestehende Programme an die neuen Gegebenheiten schnell anpassen, ohne daß ein aufwendiger Teach-in- oder Off-line-Programmiervorgang notwendig wird [65, 93]. Die Anforderungen an die Spannmittel innerhalb einer Fertigungsanlage und deren Anzahl verringern sich beim Einsatz von Sensoren, da die Werkstückpositionierung durch automatische Sensorkorrekturen we-

sentlich ungenauer erfolgen kann. Die damit verbundene Vereinfachungen beim Umbau der Anlage erhöhen die Umbau-Flexibilität beim Einsatz von Sensoren [51].

Zeiten:

Im wesentlichen lassen sich die Programmier- und die Umrüstzeiten beim Einsatz von sensor-gestützten Anlagen reduzieren. So muß bei der Off-line-Programmierung von Roboteranlagen meist eine sehr präzise Vermessung und Kalibrierung des Roboters und seiner Umgebung wie z.B. der Zuführeinheiten und der Spannmittel erfolgen, um Differenzen zwischen der im Off-line-Programmiersystem hinterlegten Modellwelt und der realen Welt zu bestimmen. Wird dies nicht vorgenommen, so wird eine zeit- und kostenaufwendige Teach-in-Programmierung der Anlage notwendig. Diese Problematik kann in vielen Fällen durch eine automatische Sensor-korrektur der Bewegungsbahnen effizient gelöst werden [124].

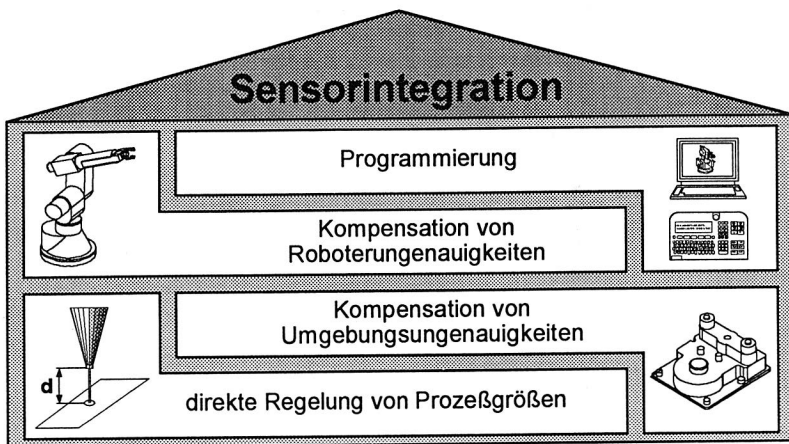


Bild 3: Nutzenpotentiale der Sensorintegration

Zuverlässigkeit:

Sensorgestützte Fertigungsanlagen können Schwierigkeiten, die von einer ungenauen Modellierung und von sich nicht deterministisch verändernden Umgebungsbedingungen herrühren, oftmals bewältigen. Die bessere Anpaßbarkeit und die situationsgerechte Reaktion auf Veränderungen an der realen Anlage und deren Umwelt bewirkt eine höhere Robustheit und Fehlertoleranz [1]. Durch die Integration von Sensoren erlangen sie ein „adaptives Verhalten“, das zu einer Steigerung der Zuverlässigkeit der Fertigungsanlage führt [57].

2.1.2 Kompensation von Roboterungenauigkeiten

Industrieroboter kommen bei der Durchführung von Handhabungsaufgaben in der Montage und bei verschiedenen Bearbeitungsaufgaben wie Kleben, Laserbearbeitung und Entgraten zum Einsatz. Im Unterschied zu NC-Maschinen besitzen Industrieroboter aber weitaus geringere Bahn- und Positioniergenauigkeiten.

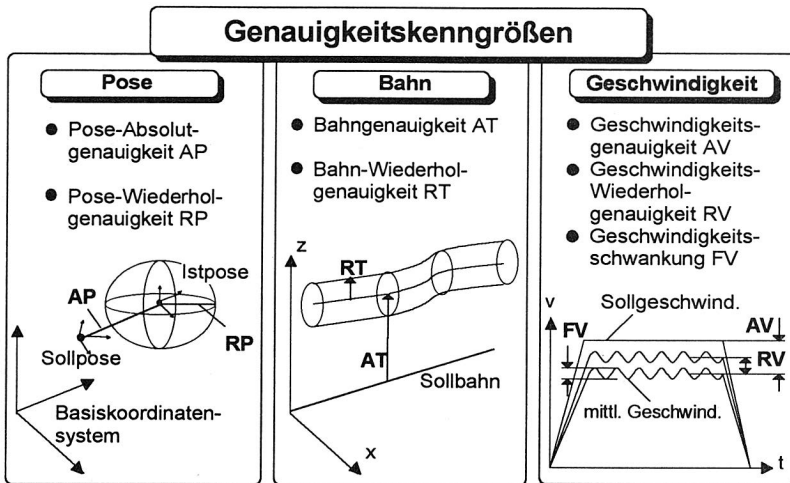


Bild 4: Genauigkeitskenngrößen von Industrierobotern

Als Gründe sind der bei weitem größere Arbeitsraum, die erheblich höheren Bahngeschwindigkeiten und die größere Beweglichkeit des Industrieroboters gegenüber der NC-Maschine anzuführen. Außerdem werden die Werkstücke und die Werkzeuge bei einer NC-Maschine in einem gemeinsamen, sehr steifen Maschinenbett gehalten, und die Antriebsstränge von NC-Maschinen weisen eine erheblich höhere Steifigkeit auf als die von Knickarmrobotern. Somit sind die Bahn- und Positioniergenauigkeiten von NC-Maschinen zwangsläufig größer als die von Industrierobotern [119]. Die Genauigkeitskenngrößen eines Roboters werden an Hand von standardisierten Kenngrößen, die in den nationalen und internationalen Standards wie der VDI-Richtlinie 2861 [150], der ISO 9283 [68] oder der RIA 15.05 [3] festgelegt sind, beschrieben und sind in Bild 4 skizziert [37].

Ursachen von Posefehlern:

Die Genauigkeit von Posen, die von einem Industrierobotern angefahren werden, sind hauptsächlich vom mechanischen Aufbau, von der kinematischen Kette des Roboters und von auftretenden Veränderungen dieser Größen über die Zeit bestimmt. Abweichungen zwischen der realen Roboterkinematik und dem Kinematikmodell, das zur Berechnung von Koordinaten-

transformationen in der Robotersteuerung verwendet wird, führen zu Posefehlern des Roboters. Auf Grund der begrenzten fertigungstechnischen Genauigkeiten am realen Roboter treten insbesondere Fehler bei den Roboterarmlängen, bei der Lage und Orientierung der Achsdrehmittelpunkte und der Nullage der Achsen auf. Untersuchungen zum Einfluß von kinematischen Fehlern auf die Posegenauigkeit zeigen, daß relativ große Fehler in den kinematischen Parametern des Roboters zu großen Fehlern in der Pose des Roboters führen [113]. Der Schluß vom resultierenden Posefehler auf die ursächlichen Einzelfehler in der kinematischen Kette ist wegen den nichtlinearen und verkoppelten kinematischen Beziehungen nur beschränkt durchführbar. Diese Sachverhalte wurden u.a. durch Arbeiten von [151] bestätigt.

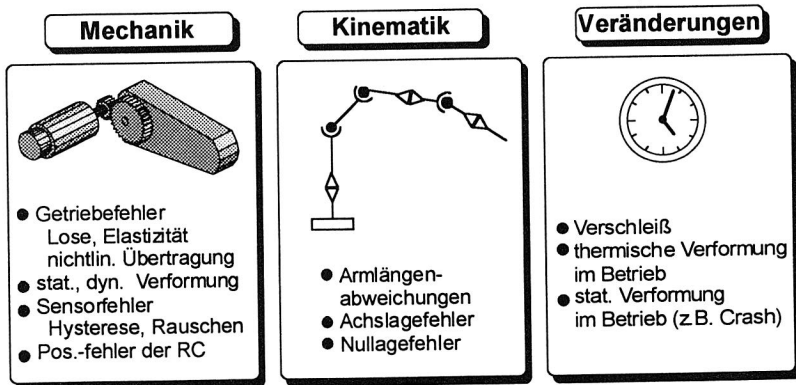


Bild 5: Ursachen von Posefehlern bei Industrierobotern

Der Posefehler eines Industrieroboters wird weiter durch mechanische Fehler im Antriebsstrang des Roboters beeinflusst [10]. Insbesondere das nichtlineare Winkelübertragungsverhalten des Getriebes und auftretende Hysteresen des Getriebes, die durch Lose verursacht werden, bewirken einen zusätzlichen Beitrag zum Posefehler. Von [113] wurden dabei Achswinkelfehler bis zur Größe von $1,3^\circ$ beobachtet. Zusätzliche Achswinkelfehler können durch die Getriebeelastizität, die bei den meisten Industrierobotern zwar geringe Werte annimmt, auftreten. Statische und dynamische Verformungen der Achskörper eines Roboters tragen auf Grund der hohen Steifigkeit zunächst kaum zum Posefehler bei [113]. In der Literatur wird außerdem über Posefehler, die vom Rauschen, von der Hysterese und von einer mangelnden Auflösung des Winkelmeßsystems herrühren, berichtet [160]. Schließlich kann der Posefehler von Positionierfehlern der Lageregelkreise der Robotersteuerung beeinflusst werden [10]. Von [151] durchgeführte Untersuchungen an einem Industrieroboter zeigen außerdem eine starke Abhängigkeit des Posefehlers von den Achsgeschwindigkeiten und der verwendeten Handlast auf. Höhere Geschwindigkeiten bewirken signifikant größere Posefehler, wogegen größere Handlasten deutlich geringere Posefehler zur Folge haben.

Als Folge des Verschleißes von mechanischen Bauteilen können nach längerem Betrieb des Roboters Posefehler, die auf Ursachen wie Getriebelose oder Elastizität zurückzuführen sind, zunehmen. Bei hochpräzisen Anwendungen stellt die Verformung der Achskörper als Folge von Temperaturänderung während der Anlaufphase eines Roboters eine weitere Fehlerquelle dar. Schließlich läßt es sich im industriellen Dauerbetrieb eines Industrieroboters nicht gänzlich vermeiden, daß beim Einrichten neuer Programme Kollisionen mit der Umgebung auftreten, die zu statischen Verformungen der Achskörper oder zu Schäden am Antriebsstrang eines Roboters führen [54]. Statische Verformungen verändern die tatsächliche Kinematik der Roboter und bewirken so einen größeren Posefehler.

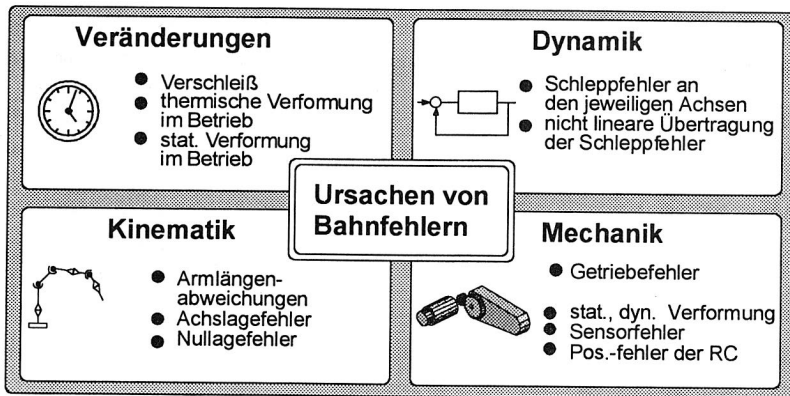


Bild 6: Ursachen von Bahnfehlern bei Industrierobotern

Ursachen von Bahnfehlern:

Die Ursachen von Bahnfehlern bei Industrierobotern sind analog zu den Posefehlern in der Kinematik, der Robotermechanik und Veränderungen verschiedener Größen über die Zeit zu lokalisieren und zeigen die selben Charakteristika [61]. Zusätzlich stellen sich dynamische Bahnfehler ein. In den Lageregelkreisen der einzelnen Roboterachsen kommt es bei stetiger Änderung der Führungsgrößen zu Schleppfehlern, die von der verwendeten Struktur und den Parametern der Lageregelung und der Veränderung des Lagesollwert über die Zeit abhängig sind. So kann bei steigender Bahngeschwindigkeit ein Ansteigen des Bahnfehlers beobachtet werden. Die Schleppfehler der einzelnen Achsen werden über die nichtlineare Roboterkinematik in die kartesischen Bahnfehler des Roboters übertragen [113].

Da bei den verschiedenen Bearbeitungstechnologien definierte Bahngeschwindigkeiten gefordert werden, um die gewünschte Bearbeitungsqualität am Werkstück zu erzielen, verbleibt zur Reduzierung des Schleppfehlers nur eine Verbesserung des dynamischen Verhaltens der Lageregelkreise. Durch den Einsatz einer Zustandsregelung bzw. von Geschwindigkeits- und Be-

schleunigungsvorsteuerungsstrategien und von lernenden Vorsteuerkonzepten konnte eine Reduzierung des Schleppfehlers beobachtet werden [82, 113].

Einfluß des Werkzeugs auf die Pose-/Bahngenauigkeit:

Da bei einer Bearbeitungsaufgabe letztendlich das Werkzeug an eine definierte Pose bzw. entlang einer definierten Bahn verfahren werden muß, trägt außerdem das Werkzeug zum resultierenden Fehler bei. Hauptfehlerursachen sind dabei Differenzen zwischen der in der Robotersteuerung hinterlegten und der tatsächlichen Koordinatentransformation, die die Verschiebung und Verdrehung des Handflanschkoordinatensystems des Roboters gegenüber dem Werkzeugkoordinatensystem (TCP) beschreibt. Nicht zu unterschätzen sind daneben die im Laufe der Zeit auftretenden Veränderungen am Werkzeug als Folge des Verschleißes eines Werkzeugs [50]. Insbesondere bei Technologien wie dem Lichtbogenschweißen, den Schleif- und Entgrattechnologien unterliegt das Werkzeug einer nicht unwesentlichen Abnutzung. Im Gegensatz dazu tritt beim Laserstrahlschneiden und beim Laserstrahlschweißen auf Grund des Werkzeugs Laser zunächst keinerlei Verschleiß auf [90].

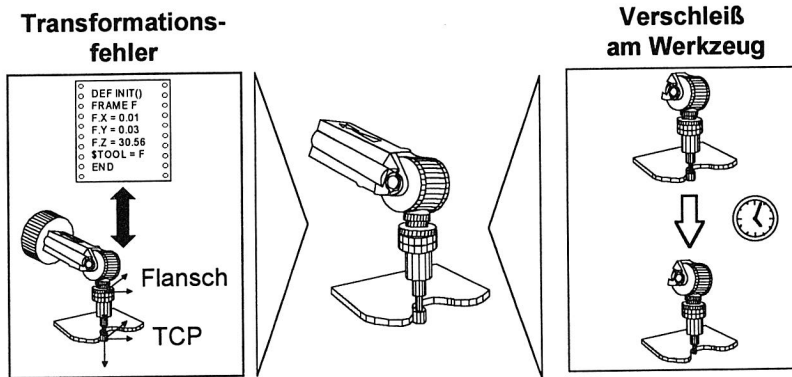


Bild 7: Ursachen von Werkzeugfehlern

Kompensation durch Sensorik:

Eine Vielzahl von Bahn- bzw. Posefehlern, die von Roboter- oder Werkzeugungenauigkeiten hervorgerufen werden, lassen sich durch den Einsatz von Sensorik kompensieren [110, 119, 129]. Dabei ist es unerheblich, an welcher Stelle der Fehler entstanden oder auf welche Ursache er zurückzuführen ist, da der resultierende Gesamtfehler von Sensoren erfaßt und geeignet kompensiert wird. Somit können auch Aspekte wie z.B. Verschleißerscheinungen ausgeglichen werden, die im Laufe der Zeit die Bearbeitungsaufgabe negativ beeinflussen. Wesentliche Voraussetzung ist aber, daß das Sensorsystem die entscheidenden Größen störungsfrei erfassen kann und über entsprechende Eingriffe in der Robotersteuerung verfügt.

2.1.3 Kompensation von Umgebungsungenauigkeiten

Die jeweilige Bearbeitungsaufgabe, die mit einem Industrieroboter durchgeführt werden soll, kann von Ungenauigkeiten in der Umgebung des Industrieroboters negativ beeinflusst werden. Als Hauptursachen für Ungenauigkeiten in einer Roboterumgebung sind Form- und Lagetoleranzen der Werkstücke, Abweichungen der Positionier- und Spannmittel und Unbestimmtheiten von geometrischen Merkmalen wie z.B. die Gratausprägung zu nennen [81, 122, 127].

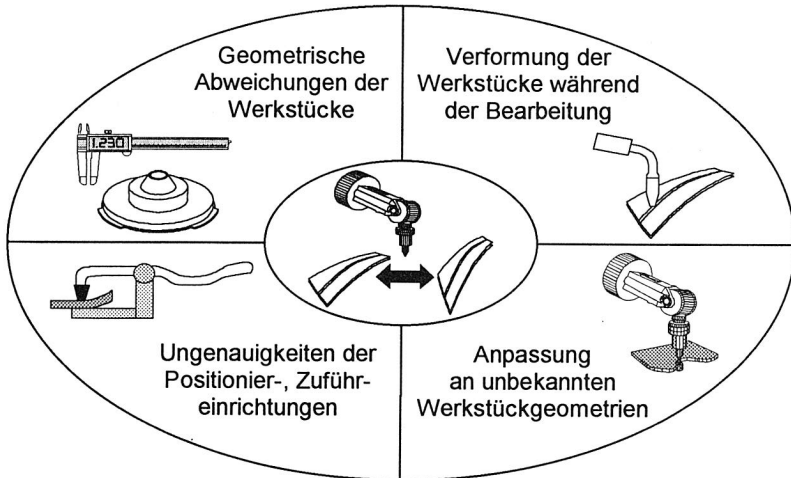


Bild 8: Ursachen von Ungenauigkeiten in der Roboterumgebung

Abweichungen in der Aufspannung:

Abweichungen in der Aufspannung werden sowohl von Spannfehlern als auch von Abweichungen der Spannmittel hervorgerufen. Spannfehler treten dann auf, wenn das zu bearbeitende Werkstück vor Betätigung des Spannmittels mangelhaft positioniert wurde. Durch den Einsatz entsprechend genauer aber kostenintensiver Positionier- und Zuführsysteme kann dies weitgehend vermieden werden [31]. Außerdem führen Späne oder Materialreste, die am Werkstück oder am Spannmittel haften und dann zwischen Werkstück und Spannmittel geraten, zu weiteren Spannfehlern. Zwischenliegende Grate können während des Spannvorgangs außerdem Formtoleranzen der Werkstücke bewirken [136]. Die Abweichungen der Spannmittel lassen sich in Form- und Lageabweichungen untergliedern. Abweichungen in der Form des Spannmittels werden durch zu große Nachgiebigkeiten oder durch thermische Einflüsse und dem damit resultierenden Verzug des Spannmittels verursacht. Toleranzen in der Lage des Spannmittels werden meist durch Kollisionen hervorgerufen. Die hohen Kräfte, die in einem

solchen Fall wirken, führen zu einer Verschiebung oder Beschädigung des Spannmittels [18, 136].

Toleranzen der Werkstücke aus der Vorfertigung:

Durch unvermeidbare Toleranzen in der Bauteilvorfertigung bestehen oftmals nicht unerhebliche Differenzen zwischen der Werkstücksollgeometrie und der tatsächlichen Geometrie des Werkstücks [32, 157]. Dies tritt v. a. bei Bauteilen, die in verschiedenen Losen gefertigt wurden, verstärkt zu Tage. Die geometrischen Toleranzen werden in Lage-, Maß- und Formtoleranzen unterteilt [63].

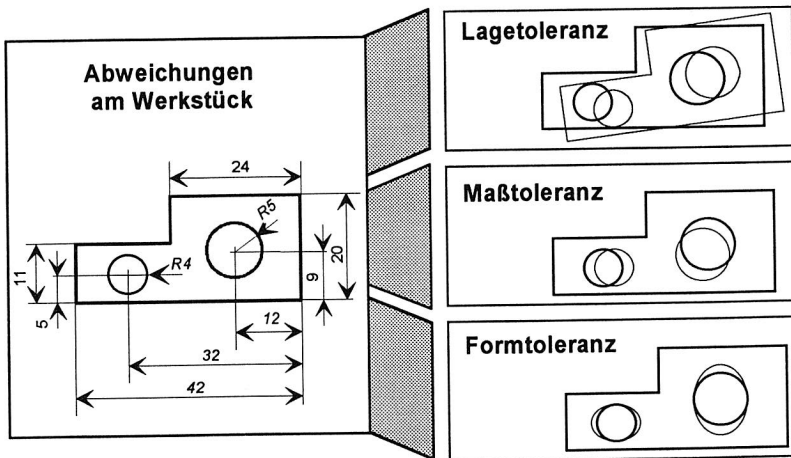


Bild 9: Klassifizierung von Werkstücktoleranzen

Liegt eine Lageabweichung vor, so ist die Werkstückkontur gegenüber der vorgegebenen Sollkontur um eine Translation und eine Rotation im Raum verschoben. Durch eine Verschiebung und Drehung der programmierten Bewegungsbahnen lassen sich die Bewegungsbahnen an die Werkstückkontur anpassen. Lageabweichungen können sowohl am gesamten Werkstück wie auch an Teilsegmenten des Bauteils auftreten [114].

Bei Maßabweichungen sind einzelne Maße am Werkstück fehlerhaft [7]. Durch Rotation und Translation des Bauteils kann keine Übereinstimmung mit der Sollgeometrie erzielt werden. Ein für das gesamte Bauteil erstelltes Bewegungsprogramm muß in einem solchen Fall in wesentlichen Teilen neu geschaffen werden [63].

Unter Formabweichungen werden alle Toleranzen am Werkstück subsummiert, bei denen die ursprüngliche Gestalt oder einzelne Konturkomponenten des Bauteils wie z.B. eine Gerade nicht eingehalten werden [112]. Formabweichungen treten v.a. bei großen elastischen oder wenig formbeständigen Werkstücken wie bei meterlangen 2-3 mm dicken Blechplatten oder

großvolumigen Tiefziehteilen auf und sind nur mit hohem Aufwand an Spannmitteln zu vermeiden [114].

Verformung der Werkstücke während der Bearbeitung:

Bei verschiedenen Bearbeitungstechnologien kann eine Verformung des Werkstücks während des Bearbeitungsprozesses beobachtet werden. Die Deformation des Bauteils gefährdet oftmals den weiteren Verlauf des Prozesses und führt zu einer geringen Produktqualität [53, 114]. Freiwerdende Werkstoffspannungen, der Wärmeverzug und die Wärmeausdehnung verursachen eine Verformung der Werkstücke während der Bearbeitung [63]. Typischerweise treten diese Effekte meist beim Laserstrahlschneiden, -schweißen und beim Bahnschweißen auf, wobei die Wärmeausdehnung und der Wärmeverzug bei den Laserstrahlbearbeitungsverfahren durch die geringere Wärmeeinbringung geringer ausfällt [111]. Bei kraftschlüssigen Bearbeitungsverfahren können die einwirkenden Kräfte bei einer entsprechenden Elastizität des Bauteils eine Deformation des Werkstück bewirken.

Unbestimmtheit von Werkstückgeometrien und -merkmalen:

In einigen Anwendungsfällen sind die Geometrien von Werkstücken und deren Merkmale wie z.B. Grate, an denen sich die Bewegungsbahn des Roboters ausrichten soll, apriori nicht oder unzureichend bekannt [39, 156]. Dies tritt meist bei der Produktion von Prototypen auf, so daß nur die Möglichkeit besteht, die Geometriedaten der Bauteile mit Abtastsystemen (Scannern) off line zu gewinnen [111] oder den Prozeß durch den Einsatz entsprechender Sensorik on line an die Bauteilgeometrie zu adaptieren. Mit Hilfe von Nahtfolgesensoren kann so z.B. die Bewegungsbahn des Schweißroboters an die tatsächlich vorliegende Nahtgeometrie der Bauteile angepaßt werden [53].

Beim Entgraten hängt die Bewegungsbahn und die Bahngeschwindigkeit des Werkzeugs sehr stark von der Lage des Grats am Bauteil und dessen Ausprägung ab [28]. Die Gratgeometrie, die meist vereinfacht durch die Gratstärke und die Grathöhe beschrieben wird, ist in Verbindung mit der Anpreßkraft und der Fräserdrehzahl ein wesentliches Kriterium für die Festlegung der Bahngeschwindigkeit [127]. Aus diesem Grund sollte die Bahngeschwindigkeit des Werkzeugs laufend an die aktuelle Gratausprägung und die Anpreßkraft angepaßt werden, um ein qualitativ hochwertiges Bearbeitungsergebnis zu erhalten.

Kompensation von Umgebungsungenauigkeiten:

Die auftretenden Toleranzen am Werkstück können in vielen Fällen durch den Einsatz von Sensoren kompensiert werden [110]. Dazu müssen Sensoren zur Verfügung stehen, die die geometrischen Änderungen am Werkstück oder eines Werkstückmerkmals erfassen [156]. Die Sensorsignale bewirken entsprechende Korrekturbewegungen des Roboters, so daß der Prozeß an die Werkstücktoleranzen adaptiert wird.

Andere Möglichkeiten zur Lösung der Problematik bestehen darin, die Toleranzen der Werkstücke zu verringern oder die Bewegungsbahnen an der Anlage neu zu programmieren. Sollen die Toleranzen der Werkstücke verringert werden, so können zunächst höhere Genauigkeitsanforderungen an die Vorfertigung der Bauteile gestellt werden. Dies hat aber oftmals zur Folge, daß die Kosten der vorgefertigten Bauteile auf Grund des höheren Fertigungsaufwands steigen. Die Abweichungen der Werkstücke während der Bearbeitung und in der Aufspannung lassen sich nur durch technisch aufwendige Positionier- und Spannvorrichtungen vermeiden [31]. Sollen vorab nicht bestimmbare Werkstückgeometrien oder -merkmale bearbeitet werden, so ist eine Automatisierung dieser Bearbeitungsaufgabe ohne den Einsatz von Sensoren nur schwer durchführbar und hat meist eine höhere Zahl von Fehlfunktionen der Anlage oder eine reduzierte Produktqualität zur Folge [156]. Eine Neuprogrammierung der Bewegungsbahnen der Applikationen kommt aus wirtschaftlichen Gründen meist nicht in Frage, da die zusätzlichen Programmier- und Stillstandzeiten der Anlage die Kosten zusätzlich erhöhen.

2.1.4 Programmierung

Bei der Programmierung von Industrierobotern kann grundsätzlich zwischen direkten Verfahren (on line) und indirekten Verfahren (off line) unterschieden werden. On-line-Verfahren nutzen den Roboter direkt als Programmierwerkzeug, wogegen bei Off-line-Verfahren die Programmierung an einem Rechner fern ab vom Zielsystem durchgeführt wird [158].

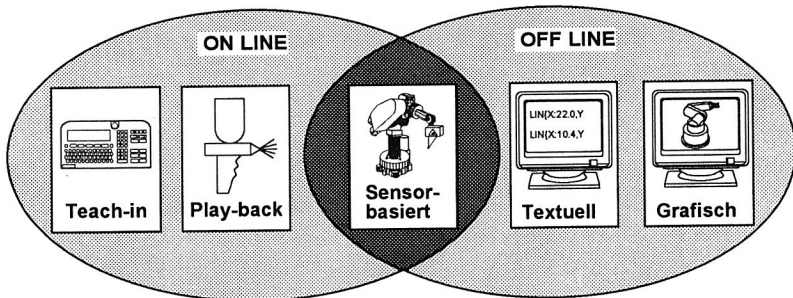


Bild 10: Gebräuchliche Programmierverfahren für Industrieroboter

Die beiden wichtigsten Methoden der On-line-Programmierung sind das Teach-in- und das Play-back-Verfahren. Die Play-back-Methode wird v.a. bei der Programmierung von Beschichtungs- oder Lackieraufgaben verwendet, wobei sie aber zusehends von expliziten Off-line-Verfahren verdrängt wird. Das Teach-in-Verfahren stellt im Augenblick das meist verwendete Verfahren zur Programmierung von Industrierobotern dar [59, 62]. Die sensorgestützte Programmierung kann zur Unterstützung der On-line- und der Off-line-Programmierung eingesetzt werden. Ein Programm wird dabei durch die Integration von Bahnkorrektur-

werten, die von Sensoren erfaßt wurden, an die reale Umwelt schnell angepaßt [119, 120, 161].

Bewertung von On-line-Programmierverfahren:

Die Programmierung von Industrieroboter erfolgt heutzutage überwiegend mit Hilfe von On-line-Programmierverfahren. Oftmals sind aber die Bahnwiederholgenauigkeiten der Roboter bei unterschiedlichen gewählten Geschwindigkeiten im Programmier- und Automatikbetrieb entsprechend groß, so daß erst nach wiederholtem Feinprogrammieren eine für die Bearbeitungstechnologie befriedigende Roboterarbeitsbewegung erzielt werden kann [119]. On-line-Programmierverfahren besitzen außerdem den Nachteil, daß die Roboteranlage während des Programmiervorgangs nicht betrieben werden kann. Um eine hohe zeitliche Auslastung der Anlage sicherzustellen, werden die Programmierarbeiten weitgehend in Nachtschichten oder an Wochenenden vorgenommen [158]. Bei Neuanlagen kann die Programmierung erst in einer sehr späten Aufbauphase, kurz vor Produktionsbeginn, durchgeführt werden. Dies verdeutlicht den Vorteil der Off-line-Programmierung, denn ca. 85 % der Programmierarbeiten können off line erfolgen [76].

Anwendungsgrenzen zeigen On-line-Programmierverfahren dann, wenn parallel arbeitende Roboter innerhalb einer Roboterzelle in ihrer Bewegung synchronisiert werden müssen oder Einzelaufgaben optimal an mehrere Roboter zu verteilen sind. Die off line erstellten Programme lassen sich im Rechner interaktiv simulieren und bezüglich Taktzeit und Programmablauf optimieren. Kollisionen zwischen Roboter und Umgebung können erkannt werden, und Entwurfsfehler in der Roboterzelle bzw. Konstruktionsfehler am Werkstück können frühzeitig entdeckt und behoben werden [76].

Bewertung von Off-line-Programmierverfahren:

Off-line-Programmiersysteme, wie sie in [49] und [163] vorgestellt werden, zeigen Probleme bei der Modellierung der Roboteranlage. Zunächst besteht die Möglichkeit, die notwendigen Geometriedaten Konstruktionsunterlagen zu entnehmen und mit Hilfe von 3D-CAD-Funktionen ins Programmiersystem zu übertragen. Dieses Verfahren ist sehr zeitaufwendig und fehlerträchtig. Der Aufwand der Modellierung läßt sich reduzieren, indem Geometriedaten, die meist in einem 2D-CAD-System vorliegen, verwendet werden. Die 2D-Daten müssen manuell nachbearbeitet werden und enthalten oftmals für die eigentliche Bahnprogrammierung unwesentliche Details, die die Antwortzeiten des Programmiersystems unnötig erhöhen. Zusätzlich bestehen Probleme biegeschlaffe Komponenten wie Schlauchpakete zu modellieren [158].

Große Schwierigkeiten bestehen bisher bei der Übertragung der off line erstellten Programme an die reale Roboteranlage. Auf eine Feinabstimmung der Programme mit Hilfe eines On-line-Verfahrens kann bisher nicht verzichtet werden [60]. Eine mangelnde Übereinstimmung des internen Rechnermodells mit der realen Roboteranlage sind die Ursache für diese Vorge-

hensweise. Die Gründe von Abweichungen zwischen Simulation und Realität können an verschiedenen Orten der in Bild 11 dargestellten Wirkungskette lokalisiert werden [11].

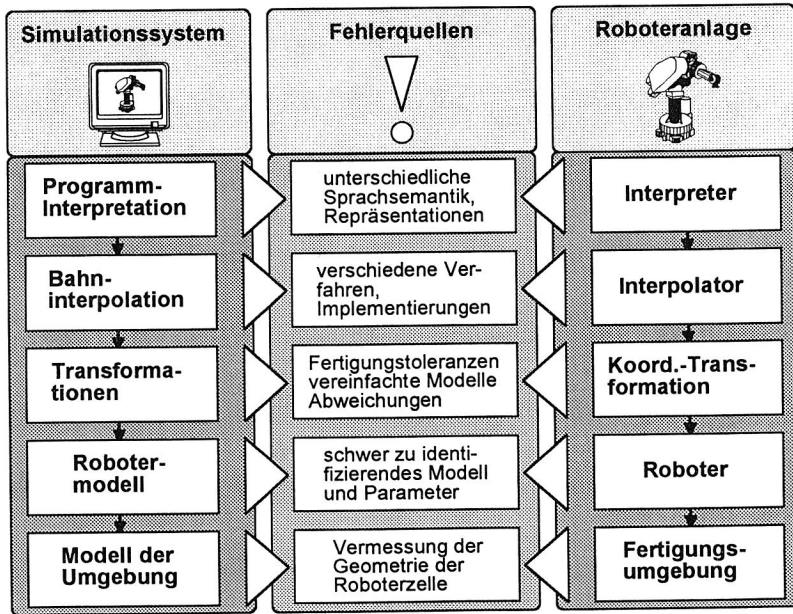


Bild 11: Fehlerquellen bei Off-line-Programmiersystemen

Um die zudem auftretenden Ungenauigkeiten in der Roboterumgebung gering zu halten, werden bei der Anlagenprojektierung hochgenaue Zuführ- und Spannmittel vorgesehen, und man fordert eine hohe Konturtreue des Werkstücks. Trotzdem kommt es häufig v.a. bei Tiefzieh-teile zu Verspannungen, die zu einem Auffedern nach dem Herausnehmen aus der Werk-zeugform führen. Um das CAD-Modell an die Realität anzupassen, berichtet [137] über die Vermessung des Werkstücks und des Spannmittels auf einer Koordinatenmeßmaschine. Dieses Vorgehen ist für eine Serienfertigung bei weitem zu aufwendig.

Bei der Projektierung einer Roboterzelle, die mit Hilfe von Off-line-Verfahren programmiert werden soll, bestehen oftmals Probleme, da die von den Roboterherstellern angegebenen Kenngrößen der Roboter ausschließlich an Hand von on line programmierten Bahn-punkten gemessen werden. Dadurch werden aber Fehler, die aus Fertigungstoleranzen des Roboters, der kinematischen Kette des Roboters und der Robotersteuerung resultieren, kompensiert. Dies hat zur Folge, daß die mit Hilfe von On-line-Programmierverfahren bestimmten Kenn-größen keinerlei Rückschlüsse auf die Genauigkeit von off line programmierten Industrierob-otern zulassen [9, 10, 142]. Industrieroboter, die off line programmiert werden, sollten

demnach durchwegs höhere Genauigkeitskenngrößen besitzen. Es bieten sich zwei Möglichkeiten an, um dieses Ziel zu verwirklichen. Zunächst kann durch die Wahl engerer Fertigungs- und Montagetoleranzen die Genauigkeit des Industrieroboters erhöht werden. Dies hat aber zur Folge, daß sich die Fertigungskosten des Roboters stark erhöhen [70]. Aus diesem Grund wurde bisher der Weg beschritten, den Roboter exakt zu vermessen. Dabei konnte festgestellt werden, daß der wesentliche Anteil des Gesamtfehlers von systematischen, kinematischen Fehlern verursacht wird [70]. Eine softwaretechnische Kompensation dieser Fehler innerhalb der Steuerung oder des Off-line-Programmiersystems ist möglich. Die Fehlerkompensation sollte geeigenerweise auf dem Off-line-Programmiersystem durchgeführt oder vom Roboterhersteller bereits in der Systemsoftware integriert sein, wie in [70] berichtet wird. Diese Vorgehensweise hat aber bis heute kaum Verbreitung gefunden. Zusätzlich muß der Roboter in festgelegten Zeitintervallen nachkalibriert werden, um Drifterscheinungen zu kompensieren [144].

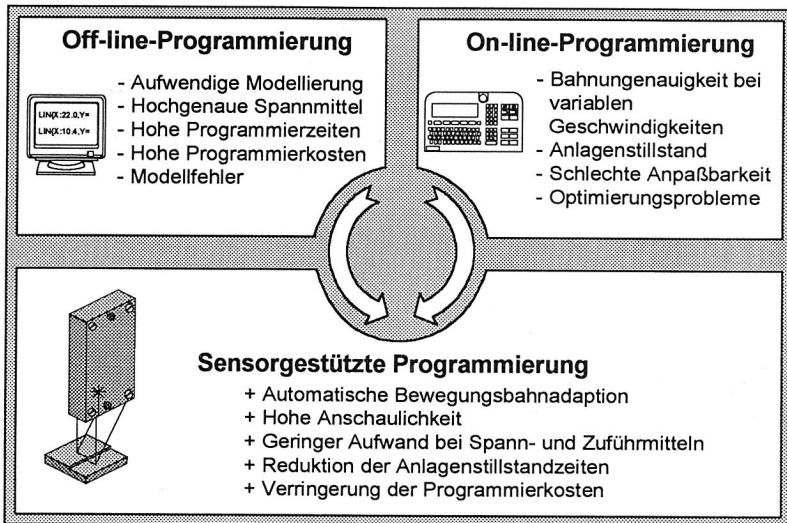


Bild 12: Programmierverfahren im Vergleich

Bewertung der sensorgestützten Programmierung

Die sensorgestützte Programmierung und Bahnadaption besitzt den bedeutenden Vorteil, daß alle Systemfehler, unabhängig wie diese entstanden sind, kompensierbar sind [60]. Bei weniger komplexen Anwendungen kann auf die Bahnprogrammierung des Roboters verzichtet bzw. eine wenig präzise Bahnprogrammierung durchgeführt werden [124]. Der Roboter bewegt sich dann alleinig auf Grund des Sensorsignals. Soll der Roboter apriori nicht bekannten Merkmalen des Werkstücks wie z.B. Graten folgen, so ist diese Aufgabenstellung einzig mit der

sensorgestützten Programmierung lösbar. Die Zeiten, die man zur Erstellung eines Roboterprogrammes unter Anwendung der sensorgestützten Programmierung benötigt, beschränken sich in der Regel auf wenige Minuten [7]. Ein weiteres Aufgabenfeld der sensorgestützten Programmierung ist die selbsttätige Feinprogrammierung off line erstellter Programme [120], um so wirkungsvoll die Off-line-Programmierung von Anlagen zu unterstützen.

Wirtschaftlichkeit von Programmierverfahren:

Zur Frage der Wirtschaftlichkeit des Einsatzes der verschiedenen Programmierverfahren werden die nachfolgenden, aufgeführten Kriterien zur Bewertung herangezogen. Der Einsatz eines grafischen-interaktiven Programmiersystems setzt eine sehr leistungsfähige Rechnerhardware voraus, die zu nicht unerheblichen Kostensätzen für den Programmierplatz führen. Die Stundensätze eines Rechnerarbeitsplatzes sind dabei im Mittel erheblich höher als die einer Roboteranlage. Der mit der Programmierung beauftragte Mitarbeiter muß über entsprechende Erfahrung mit den komplexen Konstruktions- und Programmiersystemen verfügen, so daß höhere Personalkosten im Vergleich zur On-line-Programmierung zu erwarten sind. Zusätzliche Kosten entstehen während der Korrekturphase am Roboter sowie durch die erhöhten Anforderungen an die Spanneinrichtungen und des Roboters [158].

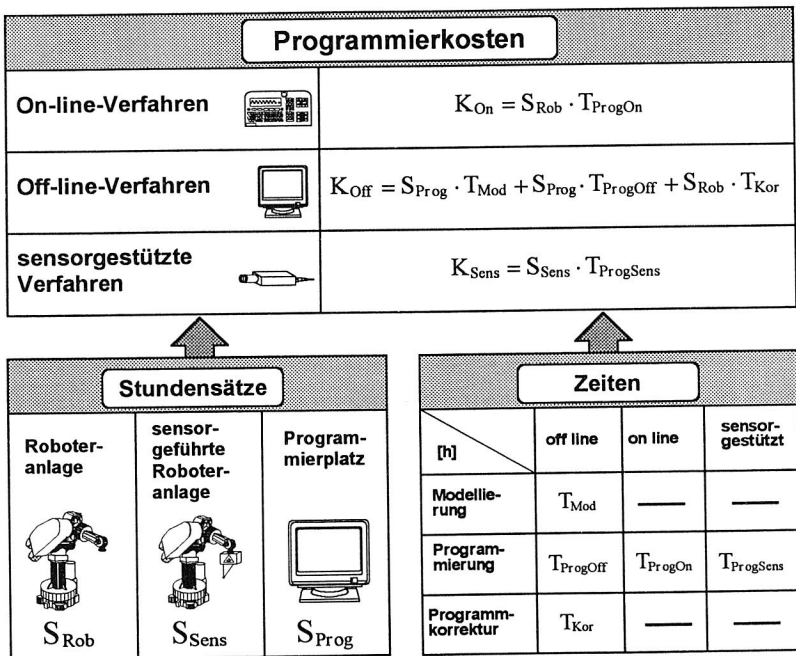


Bild 13: Berechnung von Programmierkosten

Auf Grund von zusätzlichen Investitionskosten für die Sensorik aber gleichzeitigen Kosteneinsparungen bei den Zuführ- und Spannmittleinheiten fallen die Stundensätze eines sensorgestützten Roboters geringfügig höher aus als die Stundensätze einer vergleichbaren nicht sensorgestützten Anlage.

Grundsätzlich läßt sich feststellen, daß die Programmierzeiten stark von der Komplexität der Anwendung v.a. der Bahngeometrie abhängen. Mit steigender Komplexität steigen die Programmierzeiten beim On-line-Verfahren wesentlich stärker an, als dies beim Off-line-Verfahren der Fall ist. Ab einer bestimmten Komplexität werden die Programmierzeiten des On-line-Verfahrens sogar größer als die des Off-line-Verfahrens [158].

Die sensorgestützte Programmierung kann als unterstützendes Verfahren sowohl bei der On-line- wie der Off-line-Programmerstellung die Programmierzeiten senken. Insbesondere bei der Programmierung von komplexen Bewegungsbahnen können durch die automatische Bahnadaption die besonders zu Buche schlagenden Zeiten für das „Feinprogrammieren“ und „Einfahren“ der Anlage drastisch gesenkt werden.

2.1.5 Direkte Regelung von Prozeßgrößen

Die Sensorunterstützung bei der Programmerstellung versucht letztendlich, die bekannten Probleme der anderen Verfahren zur Roboterprogrammierung zu vermeiden. Anstatt die Toleranzen des Roboters und der Umgebung durch eine aufwendige Roboterkalibrierung oder dem Einsatz aufwendiger Spannmittel zu kompensieren, beeinflusst die sensorgestützte Programmierung einzig die Relativlage zwischen dem Werkzeug und dem Werkstück. Die Einhaltung der geforderten geometrischen Beziehungen zwischen dem Werkzeug und dem Werkstück besitzt außerdem den wesentlichen Vorteil, automatisch alle Systemfehler - unabhängig davon, wie sie entstanden sind - zu kompensieren [12, 42, 60].

Außer den geometrischen Größen beeinflussen zusätzlich Technologiegrößen den zu führenden Prozeß. So wirken sich beim Laserstrahlschneiden die Bahngeschwindigkeit, der Abstand zwischen Düse und dem Werkstück und deren Orientierung sowie die Laserleistung auf die Qualität des Schnittergebnisses aus [17, 153]. Ähnliche Zusammenhänge zwischen geometrischen und technologischen Größen weisen auch die Bearbeitungstechnologien Entgraten und Bahnschweißen auf. Es liegt nun nahe, die prozeßbeeinflussenden Größen des jeweiligen Bearbeitungsverfahrens an Hand eines gesamtheitlichen Steuerungs- und Regelungskonzepts geeignet zu beeinflussen (Bild 14). Die Relativlage zwischen Werkzeug und Werkstück ist durch den Einsatz von Sensorik bekannt. Sie kann in Grenzfällen auf Grund des verzögerten Verhaltens der Roboterantriebe nicht rechtzeitig korrigiert werden. Trotzdem besteht die Möglichkeit, z.B. durch eine Veränderung der Laserleistung, den Bearbeitungsprozeß sicher zu führen. Erst die direkte Erfassung der prozeßbeeinflussenden Größen durch Sensoren sowie deren Steuerung bzw. Regelung ermöglicht eine effiziente Lösung der Probleme.

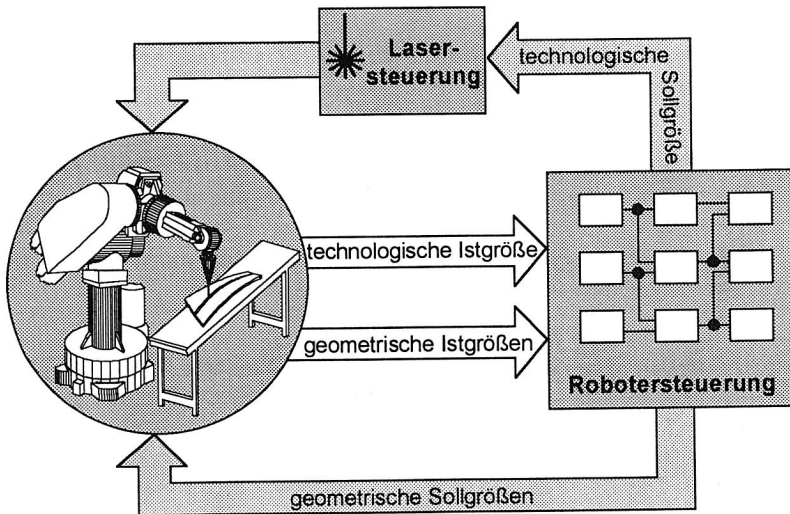


Bild 14: Beispiel eines gesamtheitlichen Steuerungs- und Regelungskonzepts

Einen weiteren Vorteil der sensorgestützte Programmierung besteht darin, daß die Anschaulichkeit bei der Programmierung der Anlage für den Anwender zunimmt. Der Anwender, meist ein Technologie, der genaue Kenntnisse über die optimalen Prozeßgrößen besitzt, wird von der aufwendigen, geometrischen Programmierung der Anlage entlastet. Die wenigen für die Programmierung des Roboters notwendigen Vorgaben von geometrischen Werte beschränken sich dabei auf anschauliche Größen wie den relativen Abstand zwischen Werkstück und Werkzeug oder die Bahngeschwindigkeit.

2.2 Formen der Sensorintegration

Die bekannten Formen der Sensorintegration werden zunächst einer Strukturierung unterzogen. Dazu kommen bewährte, je nach Zielsetzung verschiedene Klassifikationsschemen zum Einsatz. Im weiteren werden die Formen der Sensorintegration detailliert an Hand qualifizierter Einsatzkriterien diskutiert und bewertet.

2.2.1 Klassifikation der Formen der Sensorintegration

Die verschiedenen Arten des Sensoreinsatzes und deren Integration in Robotersteuerungen lassen sich zunächst nach der informationstechnischen Integrationsstelle in der Robotersteuerung einordnen. Robotersteuerungen sind meist hierarchisch strukturiert und bestehen aus den Teilsystemen Ablauf-, Bewegungs-, Aktions-, Dialog- und Datenaustauschsteuerung sowie dem Programmiersystem [21].

Sensorinformationen greifen meist auf den verschiedenen Abarbeitungsebenen innerhalb der Bewegungs- und der Ablaufsteuerung der Robotersteuerung ein (Bild 15). Die Form der Problemstellung und die darauf basierende Lösungsstrategie erfordert einen bestimmten Typ von Integrationsstelle auf der entsprechenden Hierarchieebene der Robotersteuerung. Folglich werden die verschiedenen Formen der Sensorintegration nach der jeweiligen Eingriffsstelle in der Robotersteuerung und deren steuerungstechnischen Charakteristika klassifiziert.

Programmablauf:

Der Programmablauf wird durch das Roboterprogramm festgelegt. Der Interpreter separiert das Roboterprogramm in Komponenten, die jeweils an die Aktions-, Bewegungs-, und Ablaufsteuerung zur weiteren Bearbeitung weitergegeben werden. Über die Eingriffsstelle 1 am Interpreter (Bild 15) besteht die Möglichkeit, in den Programmablauf einzugreifen und Reaktionen wie Programmverzweigungen, Unterprogrammaufrufe und Veränderungen der Parameter von Bewegungsbefehlen zu bewirken. Der Interpreter übergibt die Bewegungsbefehle des Roboterprogramms an die Präparation, die vorbereitende Parameterberechnungen für die Bahninterpolation vornimmt. Da diese Module azyklisch ausgeführt werden, können Sensordaten nur einmal je Programmlauf übertragen werden und eine entsprechende Reaktion bewirken. Diese Form der Sensorintegration wird in Kapitel 2.2.2 näher betrachtet.

Bahnplanung:

Die Bahnplanung führt die Berechnung von Führungsgrößen für die Lageregelung abhängig von dem im Roboterprogramm vorgegebenen Bewegungsbefehl durch. Dazu berechnet der kartesische Bahninterpolator abhängig vom vorgegebenen Interpolationstyp in einem festen Zeitraster, dem sogenannten Interpolationstakt, kartesische Bahnstützpunkte zwischen dem Bahnstart- und dem Bahnendpunkt. Die Zykluszeit des Interpolationstakts beträgt typischerweise 10-40 ms. Der Interpolator bestimmt die Stützpunkte in der Weise, daß sich eine vorgegebene Bewegung mit den spezifizierten Bahngeschwindigkeits- und Bahnbeschleunigungsprofilen ergibt. Der Einsatz von stetigen, besonders „glatten“ Geschwindigkeitsprofilen hat sich als günstig erwiesen, da andernfalls die Robotermechanik oftmals zu Schwingungen angeregt wird [102, 167]. Die berechneten Bahnstützpunkte werden nun zyklisch an die inverse Koordinatentransformation übertragen.

Über die Eingriffsstelle 2 (Bild 15) können die kartesischen Bahnstützpunkte zyklisch verändert oder ersetzt werden. Damit besteht die Möglichkeit, die Bewegungsbahn des Roboters zyklisch unter der Verwendung von aktuell gemessenen Sensorwerten regelnd oder steuernd zu beeinflussen. Diese Formen der Sensorintegration werden in den Kapitel 2.2.3, 2.2.4 und 2.2.5 weiter ausgeführt.

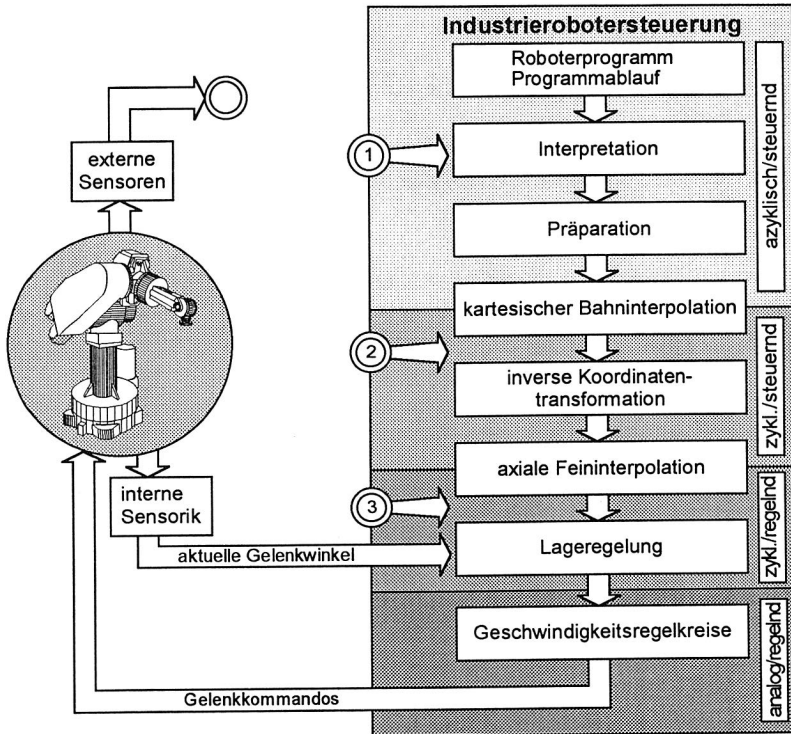


Bild 15: Hierarchische Struktur einer Robotersteuerung und mögliche Integrationsstellen für Sensorinformationen

Inverse Koordinatentransformation und Lageregelung:

Die Lageregelung hat die Aufgabe, die von der Bahnplanung (Interpolation) ermittelten Führungsgrößen genau und verzögerungsfrei in die Bewegung des Roboters umzusetzen. Bei der Positionsregelung von Industrierobotern hat sich das Prinzip der dezentralen Kaskadenregelung durchgesetzt. Dabei wird jede Bewegungsachse, unter Vernachlässigung der nichtlinearen Kopplungen, als vom Mehrkörpersystem isoliert betrachtet und jeweils separat in einer Kaskade geregelt. Folglich müssen die kartesischen Führungsgrößen in axiale Führungsgrößen transformiert werden. Dazu muß die inverse Koordinatentransformation ausgeführt werden [38]. Die inverse Koordinatentransformation, die im Interpolationstakt berechnet wird, ist eine sehr rechenintensive Operation, so daß die Zykluszeit des Interpolationstakts nicht weiter verringert werden kann. Um eine möglichst gleichmäßige Bewegung des Roboters zu erzielen, wird deshalb oft eine Interpolation der axialen Führungsgrößen im Zeitraster des Lageregeltakts durchgeführt. Die Zykluszeit des Lageregeltakts besitzt typischerweise Werte von

1-10 ms. Die interpolierten, axialen Führungsgrößen werden an die Lageregelung übergeben. Über die Eingriffsstelle 3 (Bild 15) können die axialen Führungsgrößen der Lageregelung zyklisch an Hand von aktuell gemessenen Sensorwerten verändert werden. Diese Eingriffsstelle weist sehr geringe, systeminterne Verzögerungszeiten auf, da sie direkt die Führungsgrößen der Lageregelung beeinflusst. Diese Form der Sensorintegration wird in Kapitel 2.2.4 weiter detailliert.

Klassifikation nach dem Anbringungsort der Sensorik:

Die Formen der Sensorintegration kann man außerdem in die folgenden Fälle, die sich hinsichtlich des Anbringungsortes der Sensoren unterscheiden, einteilen [125]:

1. Der Sensor erfaßt, da er örtlich und damit zeitlich vorlaufend bezüglich dem TCP montiert ist, die zukünftigen, relativen, geometrischen Beziehungen zwischen Werkzeug und Werkstück. Die sich bei der Sensorintegration ergebenden Totzeiten und das verzögerte Verhalten des Roboters lassen sich kompensieren, da die Korrektursignale mit einem entsprechenden, zeitlichen Vorhalt der Robotersteuerung zugeführt werden können.
2. Der Sensor ist direkt am Werkzeugbezugspunkt (TCP) angebracht und erfaßt die aktuell vorliegenden, geometrischen Beziehungen zwischen Werkzeug und Werkstück. Die Sensorsignale können dabei meist unmittelbar als Korrektursignale an die Robotersteuerung übertragen werden.
3. Der Aufnehmer des Sensorsystems ist ortsfest an der Roboteranlage montiert und überblickt zumindest Teilbereiche der Roboterszene. Die mangelnde Auflösung des Sensorsystems, die durch den großen Meßbereich des Sensors bedingt ist, und Abschattungen der Szene durch das Werkstück, das Werkzeug und den Roboter bereiten große Probleme.
4. Der Sensor besitzt eine eigene Kinematik, die es ihm erlaubt, Aktionen des Werkzeugs oder Merkmale des Werkstücks zu beobachten. Der Einsatz dieser Lösung wird aber stark eingeschränkt, da zusätzliche Kosten für den Geräteaufwand entstehen und überdies weiterer Platz für den Anbau benötigt wird.

Verknüpft man die möglichen Eingriffstellen für Sensorsignale an einer Robotersteuerung mit den Sensoren, die entsprechend den vorhergehenden Fällen 1-4 angebracht wurden, so ergeben sich sechs grundsätzlich sinnvolle Klassen von Sensorintegrationsformen. Diese Klassen, die in den Kapiteln 2.2.2 - 2.2.6 weiter betrachtet werden, können zunächst jeweils einem Wert der drei folgenden Klassifikationspaare zugeordnet werden (Bild 16):

Azyklisch/zyklisch:

Liegt eine zyklische Form der Sensorintegration vor, so übernimmt die Robotersteuerung in einem festen Zeittakt laufend Daten von den Sensoren. Im azyklischen Fall übertragen die Sensoren meist einmalig Daten an die Robotersteuerung [132].

Steuernd/regelnd:

Die Begriffe sind im Sinn der DIN-Norm 19226 [22] zu verstehen. Es besteht zum einen die Möglichkeit, Sensorwerte so zurückzuführen, daß über den Sensor, die Robotersteuerung und den Roboter ein Regelkreis aufgebaut wird. Im anderen Fall werden die Sensorwerte steuernd an die Robotersteuerung übertragen [155].

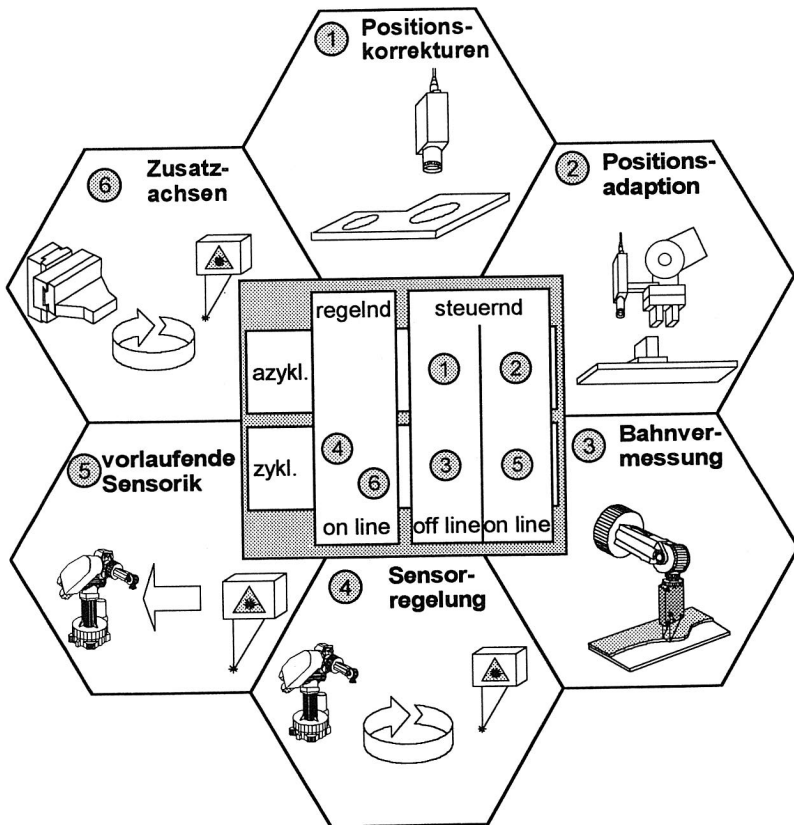


Bild 16: Formen der Sensorintegration und deren Charakteristika

On line/off line:

Werden die aktuellen Roboterbewegungen von nahezu gleichzeitig erfaßten Sensorwerten, die in die Robotersteuerung integriert werden, beeinflusst, so handelt es sich um eine Off-line-Integration. Andernfalls basiert die Bewegung des Roboters auf zuvor gemessenen und gespeicherten Sensorwerten.

2.2.2 Korrektur von Positionen in Roboterprogrammen

Unter Verwendung eines ortsfesten Sensorsystems, meist einer Kamera mit nachfolgendem Bildverarbeitungssystem, kann eine unbekannte Lage- oder Orientierungsänderung des Werkstücks oder eines Werkstückmerkmals erfaßt werden. Die Position des Werkstücks, die im Sensorkoordinatensystem vorliegt, wird in ein der Robotersteuerung bekanntes Koordinatensystem, wie z.B. das Fußkoordinatensystem des Roboters, transformiert. Legt das Roboterprogramm fest, daß die Position des Werkstücks bei der Abarbeitung des Programms angefahren werden soll, so kann die bisherige im Programm hinterlegte Position durch die aktuell ermittelte Position ersetzt werden. Die Sensorinformation ist dazu über die Eingriffsstelle 1 (Bild 15) rechtzeitig vor der Bearbeitung des Bewegungsbefehls an die Robotersteuerung zu übertragen. Die Bestimmung der Korrekturposition kann sowohl off line wie on line zur Roboteraktion erfolgen. Diese Form der Sensorintegration hat bereits industriellen Einsatz gefunden [96].

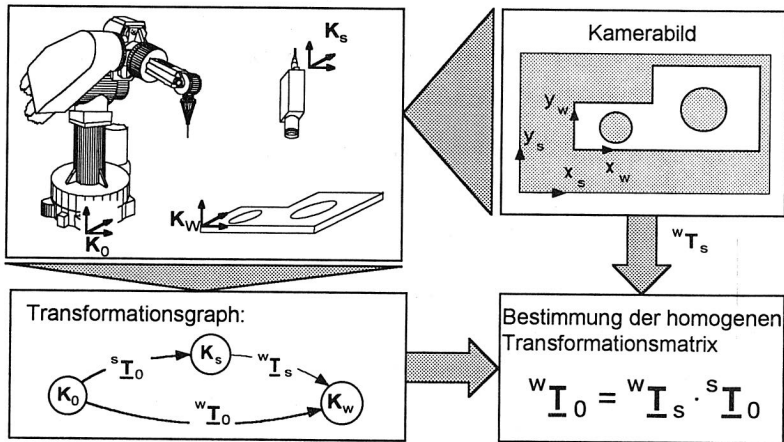


Bild 17: Azyklische Korrektur von Positionen

Die mangelnde Auflösung des Sensorsystems, die durch den großen Meßbereich des Sensors bedingt ist, und Abschattungen können dabei Schwierigkeiten bereiten. Durch die ortsfeste Anbringung des Sensorsystems wird außerdem die Genauigkeit dieser Form der Sensorintegration um die Wiederholgenauigkeit des verwendeten Roboters verschlechtert. Eventuell vorhandene Fehler in der Koordinatentransformation, die den Zusammenhang zwischen dem Sensor- und dem Roboterfußkoordinatensystem beschreibt, tragen zum Gesamtfehler der Anordnung bei.

2.2.3 Sensorgestützte Off-line-Bahnprogrammierung

Genügt die azyklische Positionskorrektur nicht mehr den gestellten Anforderungen, so kann die Werkstückkontur vor der Bearbeitung vermessen und eine anschließende Korrektur aller Bahnstützpunkte vorgenommen werden. Die Bahnkorrekturdaten werden dazu steuernd über die Eingriffstelle 2 (Bild 15) in die Robotersteuerung eingebracht. Damit lassen sich alle Toleranzen eines Werkstücks kompensieren, soweit diese sich nicht erst bei der Bearbeitung einstellen. Dieses Verfahren erfordert aber einen sehr hohen Aufwand, da die Werkstückkontur zunächst vermessen werden muß, bevor aus den gemessenen Abweichungen die erforderliche Verschiebung der Bahnstützwerte errechnet werden kann [125]. Aus diesem Grund ist ein zweimaliges Abfahren der Werkstückkontur notwendig. Für die Berechnung der Verschiebung der Bahnstützwerte ist in der Robotersteuerung oder in einem zusätzlichen Rechner entsprechend Speicherkapazität und Rechenleistung zur Verfügung zu stellen. Durch die Definition von Bahntoleranzschränken läßt sich die Zahl der abzuspeichernden Bahnstützpunkte wesentlich verringern, und die Bahnplanung für die im 2. Schritt folgende Bewegungsgenerierung vereinfacht sich [48]. Bei verschiedenen, kraftschlüssigen Bearbeitungsverfahren wie dem Entgraten oder Schleifen ist dieses Verfahren wegen der sich einstellenden Elastizität des Roboters nur beschränkt einsetzbar [136].

2.2.4 On-line-Bahnadaption durch Sensorregelung

Bei diesem Verfahren bewegt sich der Industrieroboter auf einer vorher programmierten Bahn. Der Sensor ist am Werkzeugbezugspunkt (TCP) angebracht und erfaßt zyklisch die Abweichung der Roboterbahn von der Sollbahn, die durch Merkmale am Werkstück, wie z.B. einer Kante, festgelegt ist. Aus der detektierten Abweichung berechnet ein Sensorrechner Korrektursignale, die zyklisch an die Robotersteuerung übertragen werden.

Durch die Rückkopplung der sensorgestützt erfaßten Bahn kann eine zusätzliche Regelkaskade um die Lageregelkreise der Robotersteuerung aufgebaut werden. Jegliche Toleranzen des Werkstücks oder der Spannmittel sowie Ungenauigkeiten des Industrieroboters, unabhängig ob sie vor oder während der Bearbeitungsaufgabe entstehen, können kompensiert werden. Entscheidende Voraussetzung ist dabei, daß die Abweichungen vom eingesetzten Sensorsystem erfaßt werden und die Dynamik der Sensorregelung ausreichend ist.

Diese Form der On-line-Bahnadaption erfordert eine zyklische Integration der Sensorsignale nach dem kartesischen Bahninterpolator im Interpolationstakt der Robotersteuerung (Bild 18). Dazu müssen kartesische Korrekturwerte in derzeit typischen Taktzeiten von 10-40 ms berechnet werden. Totzeiten im Sensorregelkreis, die sich bei der Erfassung der Sensorwerte und deren Integration in die Robotersteuerung ergeben, verringern in diesem Fall die Gesamtdynamik des Sensorregelkreises [110, 120, 121].

Ein zweiter Sensorregelkreis kann durch die zyklische Integration der Sensorsignale nach dem axialen Interpolator im Lageregeltakt der Robotersteuerung (Bild 18) aufgebaut werden. Hierfür sind axiale Korrekturwerte im Lageregeltakt, der bei heutigen Steuerungen typischerweise Werte von 1-10 ms aufweist, an die Robotersteuerung zu übertragen. Die Verringerung der Taktzeit um den Faktor 4-10 läßt eine Reduktion der Totzeiten im Sensorregelkreis erwarten, die wiederum eine verbesserte Gesamtdynamik des Sensorregelkreises zur Folge hat [110].

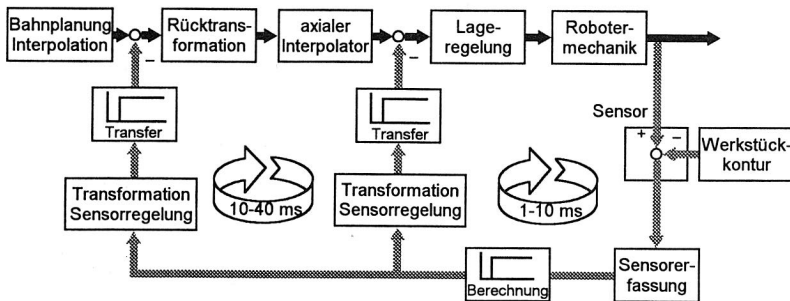


Bild 18: Verschiedene Formen von Sensorregelkreisen

Oftmals stellt die Anwendung zusätzliche Anforderungen an die Form der Sensorintegration. Zusatzachsenlösungen oder vorlaufend montierte Sensoren können aus Gründen der Zugänglichkeit am Werkstück in einigen Fällen nur beschränkt eingesetzt werden. Bei vielen Anwendungen muß außerdem die Meßwerterfassung direkt am Werkzeugbezugspunkt (TCP) erfolgen [77]. Hierzu gehören Anwendungen wie

- ☐ die Abstandsregelung, wenn aus Zugänglichkeitsgründen keine vorlaufende Sensorik oder Zusatzachsen am Werkzeug verwendet werden können,
- ☐ die Kraftregelung beim Entgraten und Schleifen von Gußwerkstücken,
- ☐ die Kraft/Momentenüberwachung und -korrektur bei der Montage.

2.2.5 Gesteuerte Bahnadaption mit vorlaufendem Sensor

Bei dieser Form der Sensorintegration ist der Sensor nicht am Werkzeugbezugspunkt (TCP) des Roboters angebracht, sondern in einem festen Abstand vor dem TCP in Richtung der aktuellen Bewegung montiert. Dadurch daß der Sensor vorlaufend angebaut ist, liefern die Sensordaten eine Prädiktion des Bahnverlaufs der Werkstückkontur. Aus den erfaßten Sensordaten können zukünftige Bahnstützstellen berechnet und in die Robotersteuerung nach dem kartesischen Bahninterpolator im Interpolationstakt integriert werden. Durch die rechtzeitige Übertragung der zukünftigen Sollwerte an die Robotersteuerung lassen sich Totzeiten bei der Datenübertragung und Totzeiten innerhalb der Robotersteuerung weitgehend kompensieren, und ermöglichen dem Roboter dem Bahnverlauf der Werkstückkontur zu folgen [63, 112].

Diese Steuerungsstrategie wird als lokale Bahnplanung bezeichnet, da abhängig von den prädiktiv erfaßten Sensorwerten eine lokal begrenzte Bahngenerierung unter der Verwendung von verschiedenen Interpolationsverfahren durchgeführt wird [167]. Da die tatsächliche TCP-Position (Regelgröße) nicht rückgekoppelt wird, stellt diese Bahnführung eine Steuerung dar [63].

Dem Einsatz von vorlaufenden Sensoren stehen einige Hinderungsgründe gegenüber. Durch das steuernde Verhalten der Bahnplanung mit vorlaufendem Sensor können nicht alle Bahnfehler kompensiert werden. Systematische Diskrepanzen zwischen den Transformationen in der Robotersteuerung und der tatsächlichen Kinematik des Roboters können nicht ausgeglichen werden [63].

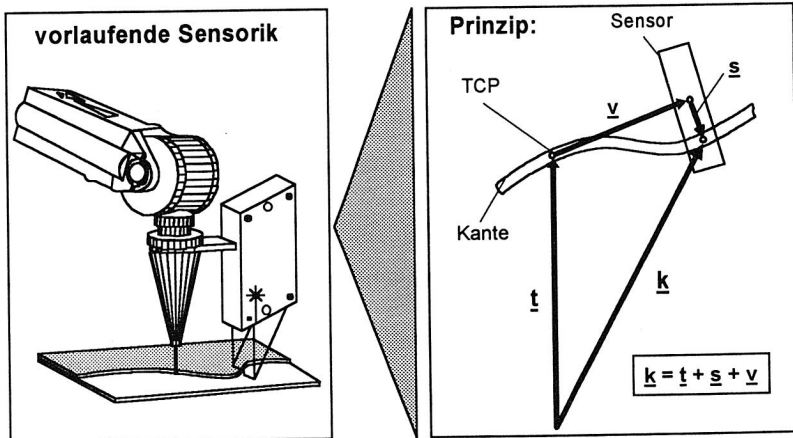


Bild 19: Bahnadaption mit vorlaufender Sensorik

Die starre, vorlaufende Ankopplung des Sensors an den Werkzeugbezugspunkt hat den Nachteil, daß jeweils nur der Sensor oder das Werkzeug in Nominalorientierung über einer gekrümmten Werkstückkontur stehen kann. Ist eine Nominalorientierung des Werkzeugs durch den Bearbeitungsprozeß zwingend vorgegeben, so können Meßbedingungen, wie z.B. der maximale Meßwinkel des Sensorsystems, verletzt werden, die zu unbrauchbaren Sensorwerten führen [125]. Insbesondere bei stark gekrümmten Werkstückkonturen stellen die eingeschränkten Meßbereiche der Sensoren ein Problem dar (Bild 20). Diese Problematik kann meist nur durch den Einsatz zusätzlicher Aktorik, die das Sensorsystem in die erforderliche Position bringt, aufgelöst werden. Der Einbau zusätzlicher Aktorik zur Positionierung der Sensorik ist aber unter fertigungstechnischen Randbedingungen zu aufwendig, zu kostspielig und zu störanfällig, um verbreitet eingesetzt zu werden. Außerdem verschärft sich hierdurch das Problem der Zugänglichkeit am Werkstück [123].

Eine Verringerung des Vorlaufs der Sensoren könnte diese Problematik entschärfen, ist aber oftmals aus mechanischen Gründen (Baugröße) oder wegen der Verletzung der zeitlichen Vorlaufbedingung nur beschränkt möglich [32].

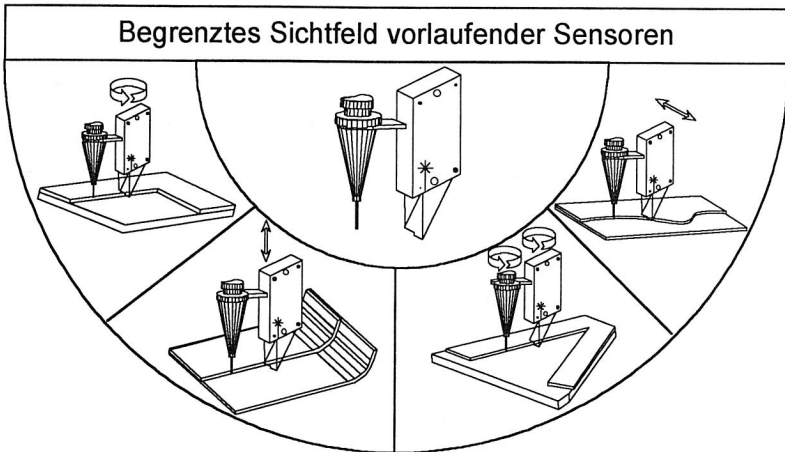


Bild 20: Problemfelder von vorlaufender Sensorik

Muß eine Bahn von einem Startpunkt bis zu einem Endpunkt adaptiert werden, so ergeben sich Probleme durch den Vorlauf der Sensorik. Am Startpunkt liegt keine Sensorinformation zur Sollbahn vor, da der Sensor, um die Vorlaufstrecke versetzt, zu messen beginnt. Am Endpunkt können sich ähnliche Probleme ergeben.

2.2.6 Bahnadaption durch Zusatzachsen

Bei dieser Form der Sensorintegration wird zur Bahnadaption ein Zusatzachsensystem zwischen die Roboterhandwurzel und das Werkzeug montiert. Ein Positionsregelkreis, der aus den Komponenten Sensor, Sensorwertverarbeitung, Regelung und Zusatzachse besteht, sorgt für die Einhaltung der geometrischen Beziehungen zwischen dem Werkzeug und dem Werkstück (Bild 21). Die sensorgeführte Positionsregelung der Zusatzachsen erfolgt dabei vollständig unabhängig von der Bewegung des Roboters [95, 122].

Die Zusatzachsen wirken als kartesische Achsen im Werkzeugkoordinatensystem, und der Sensor nimmt seine Messungen meist im Werkzeugkoordinatensystem vor. Folglich kann die Berechnung der Positionsvorgaben für die Zusatzachse ohne zeitaufwendige Koordinatentransformationen durchgeführt werden. Da für Korrekturbewegungen nicht mehr der ganze Roboterrumpf sondern nur noch die Zusatzachse und das Werkzeug bewegt werden muß, läßt sich eine höhere Gesamtdynamik des Zusatzachsensensorregelkreises im Vergleich zu anderen Lösungen erzielen [95, 122].

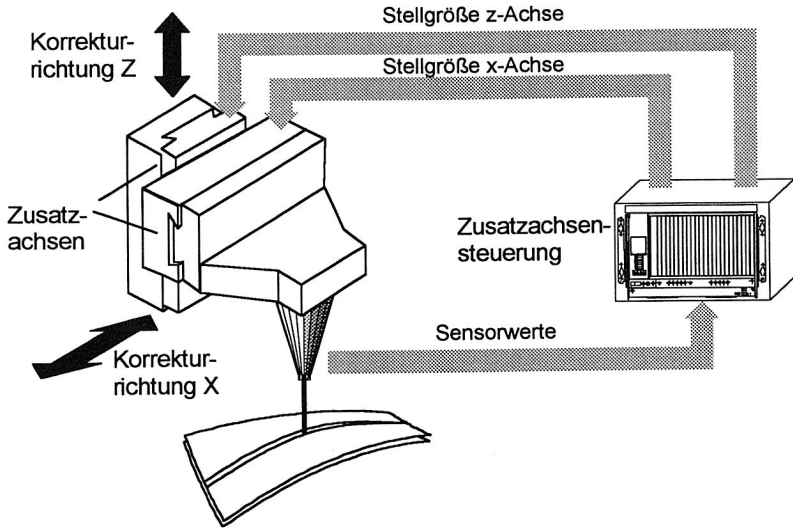


Bild 21: Struktur der Bahnadaptation mit Zusatzachsen

Die Vorteile sind aber mit einen Nachteilen verbunden. Die Korrekturmöglichkeiten sind durch die Zahl der Freiheitsgrade des Zusatzachsensystems vorgegeben (1 bis 2 Freiheitsgrade) und außerdem beschränkt durch den maximalen Hub- oder Schwenkbereich der Zusatzachsen. Dies hat zur Folge, daß die Positionsabweichungen nur in Richtung der Zusatzachsen erfolgen sollten und nie den maximalen Verstellbereich der Zusatzachsen überschreiten dürfen. Die Forderung nach einer geringen Bauhöhe und einem geringen Gewicht der Zusatzachsen, um möglichst die Nutzlast des Roboters nicht zu verringern, verschärft das Problem des eingeschränkten Verstellwegs der Zusatzachse [123]. Außerdem erfordert diese Lösung einen nicht unerheblichen Geräteaufwand und Platzbedarf [125].

2.2.7 Bewertung der Verfahren

Die unterschiedlichen Verfahren zur Integration von Sensordaten in Robotersteuerungen weisen abhängig von den Anforderungen der jeweiligen Anwendung Vor- und Nachteile auf. Eine Bewertung der Verfahren an Hand der nachfolgenden Kriterien wurde in Tabelle 1 vorgenommen:

- ☐ **Kompensation von Abweichungen:** Die Fähigkeit des Verfahrens die verschiedenen Formen von Abweichungen zu kompensieren, wurde unter Verwendung der folgenden Fragestellungen vorgenommen: Können Lage-, Formtoleranzen, die vor der Bearbeitung am Werkstück vorliegen, von dem Verfahren kompensiert werden? Können Lage-, Formtoleranzen, die während der Bearbeitung am Werkstück entstehen, von dem Ver-

fahren kompensiert werden? Kann das Verfahren Bahnabweichungen, die der Roboter verursacht, ausgleichen? In wievielen Dimensionen kann die Bahnadaption durch das Verfahren erfolgen?

- Zuverlässigkeit und Aufwand: Schließlich werden die Zuverlässigkeit und der hard- und softwaretechnische Aufwand des jeweiligen Verfahrens abgeschätzt. Ein hoher Aufwand verbunden mit einer größeren Komplexität korreliert in den meisten Fällen auch mit einer geringeren Zuverlässigkeit des Verfahrens.

Formen der Sensorintegration	Kompensation von				Dimension bis zu	hohe Zuverlässigkeit	geringer Aufwand
	Lageabweichungen vor der Bearbeitung	Formabweichungen vor der Bearbeitung	Abweichungen während der Bearbeitung	Roboterabweichungen			
Korrektur von Positionen in Roboterprogrammen (Kap. 2.2.2)	↑	↓	↓	↓	6	↑	↑
Sensorgestützte Off-line-Bahnprogrammierung (Kap. 2.2.3)	↑	↑	↓	↓	6	↗	↗
On-line-Bahnadaption durch Sensorregelung (Kap. 2.2.4)	↑	↑	↗	→	6	↗	↗
Gesteuerte Bahnadaption mit vorlaufendem Sensor (Kap. 2.2.5)	↑	↑	↗	↘	6	→	→
Bahnadaption durch Zusatzachsen (Kap. 2.2.6)	↑	↑	↗	↗	2	→	↘

↑: Kriterium sehr gut erfüllt

→: Kriterium meist erfüllt

↓: Kriterium nicht erfüllt

↗: Kriterium gut erfüllt

↘: Kriterium schlecht erfüllt

Tabelle 1: Bewertung der Formen der Sensorintegration

2.3 Defizite bestehender Systeme

Heutige Industrieroboter werden nur in geringen Stückzahlen mit zusätzlicher, externer Sensorik ausgerüstet. Dies ist v.a. darin begründet, daß Industrierobotersteuerungen oft über keine oder nur sehr eingeschränkte Möglichkeiten verfügen, Sensorsignale zu integrieren [90, 155]. Bestehende Lösungen, die oftmals von Forschungseinrichtungen oder Universitätsinstituten vorgeschlagen wurden, beschäftigten sich oft mit sehr spezifischen Problemen. Die unterschiedlichen Problemstellungen, die durch eine Vielzahl von Einsatzfällen mit verschiedenen Randbedingungen charakterisiert sind, konnten deshalb nicht umfassend und flexibel gelöst werden [63, 155, 156]. Die folgenden Sachverhalte stellen die wesentlichen Hemmnisse für die Integration von Sensorik an Roboteranlagen dar:

- Die Sensorsysteme konnten die hohen Anforderungen der Fertigungs- und Montageprozesse bisher nur unzureichend erfüllen. Wesentliche Forderungen sind dabei eine schnelle Sensordatenerfassung und -verarbeitung [130]. Die Erfassung der Sensorwerte muß kostengünstig mit hoher Zuverlässigkeit und Störsicherheit erfolgen, um Störgrößen, die durch das Werkstück oder durch den Bearbeitungsprozeß hervorgerufen werden, zu eliminieren [12]. Eine schnelle Integration der Sensordaten macht zusätzlich eine enge Synchronisation zwischen dem Sensorsystem und der Robotersteuerung erforderlich [111]. Insbesondere sollen Ansätze, die diese Probleme unter Verwendung von mehreren Sensoren (Multisensorik) zu lösen versuchen, weiterverfolgt werden [91].
- Die bisherigen Industrierobotersteuerungen sind wenig geeignet für die Integration von Sensoren, da sie meist über nur sehr eingeschränkte Möglichkeiten zu einer flexiblen Sensordatenintegration und -verarbeitung verfügen [42, 43, 90, 95, 155]. Bestehende Lösungen zeichnen sich dadurch aus, daß durch vermeidbare Totzeiten in der Sensorwerterfassung und der Integration in die Robotersteuerung die Reaktionsschnelligkeit des Systems herabgesetzt wird [95, 121, 156]. Die Ursachen dafür sind [167]:
 - Totzeiten der Sensorsysteme, die durch geringe Erfassungsraten und hohe Zeiten bei der Aufbereitung und Auswertung der Information bedingt sind,
 - Zeitverzögerungen während der Interaktion von Sensorsystem und Robotersteuerung (Übertragung, Transformationen),
 - Mangel an Untersuchungen zu geeigneten Regelungskonzepten, die ein Führungsverhalten mit möglichst geringen Verzögerungszeiten des Sensorregelkreises sicherstellen.

Eine notwendige Standardisierung von Sensorschnittstellen an Robotersteuerungen konnte bis heute nicht durchgeführt werden [95]. Es wurden zwar verschiedene Konzepte [30, 80] und der Entwurf der Vornorm DIN V 66311 [25] für eine standardisierte Sensorschnittstelle ausgearbeitet, aber nicht verwirklicht.

- Die bisherigen zur rechner- und informationstechnischen Integration von Sensoren bestehenden Systeme sind in den meisten Fällen pragmatisch am Einzelfall orientiert [152]. Diese System können die höheren Anforderungen an die Flexibilität nicht erfüllen, da sie keine „offenen“ Systemstrukturen aufweisen, die z.B. eine nachträgliche Integration von Sensor- oder Steuerungsmodulen oder eine Parametrierung des Systems erlauben [93, 164]. Dazu ist es notwendig, geeignete Strategien zu entwickeln, die den Anwender unterstützen, komplexe Sensorverarbeitungsroutinen und Regelungsstrategien zu entwickeln und im System zu realisieren [32]. Insbesondere muß die Anwenderprogrammierung, deren Ziel es ist, eine sensorgestützt ablaufende Anwendung zu implementieren, durch entsprechende Hilfsmittel unterstützt und erleichtert werden [101]. Ein solches noch zu entwerfendes System sollte es z.B. ermöglichen, daß die Bedienung der Sensor-Roboter-Anlage nur von der Robotersteuerung aus erfolgt.

3. Definition des Anforderungsprofils eines Sensor-Roboter-Systems

Zunächst lassen sich die technologischen Ziele, die verschiedene Bearbeitungstechnologien an eine integrierte Sensor-Roboter-Lösung stellen, definieren. Die technologische Zieldefinition dient als Ausgangsbasis einer nachfolgend durchgeführten Spezifikation der Anforderungsprofile von Teilkomponenten des Sensor-Roboter-Systems.

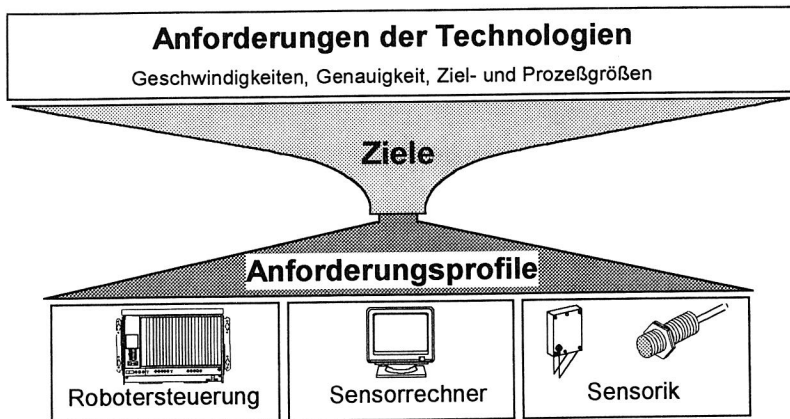


Bild 22: Ableitung der Technologieanforderungen

Insbesondere lassen sich die Aufgaben der Robotersteuerung und die Anforderungen an die Sensorik und an den Sensorrechner separieren und für den späteren Entwurf eines Systems nutzen. Abschließend können die Anforderungen an ein integriertes Gesamtsystem benannt werden, das den Anwender bei Programmierung, Simulation und Test am Sensorrechnersystem optimal unterstützen soll.

3.1 Technologische Zieldefinition

Die unterschiedlichen Materialbearbeitungsverfahren und die Montage stellen vielfältige Anforderungen an die Sensorintegration eines Robotersystems (Tabelle 2). Die Laserbearbeitungsverfahren sind dadurch charakterisiert, daß der Abstand zwischen dem Werkstück und dem Werkzeug sehr genau eingehalten werden muß, um eine exakte Fokussierung des Laserstrahls auf der Werkstückoberfläche und damit eine optimale Wärmeeinbringung am Werkstück zu erzielen. Eine konstante Wärmeeinbringung erfordert außerdem, daß der Roboter den Laserstrahl mit konstanter Geschwindigkeit entlang der gewünschten Bahn führt. Beim Laserstrahlschweißen sollte außerdem der Laserstrahl der Naht am Werkstück mit hoher Genauig-

keit, die von der jeweiligen Art des Stoßes abhängt, folgen [153]. Die typischen Kenngrößen bei der Laserstrahlbearbeitung sind der Tabelle 2 zu entnehmen [18, 17, 31, 53].

Beim Bahnschweißen bestehen ähnliche Anforderungen an die Bewegungsführung des Werkzeugs wie bei der Laserstrahlbearbeitung. Verfahrensbedingt sind aber die Forderungen hinsichtlich der Bahngeschwindigkeit und der erforderlichen Genauigkeiten geringer [12].

Der Auftrag von Kleb- oder Dichtstoffen nimmt bei den bisher betrachteten, berührungslosen Bearbeitungsverfahren eine Sonderstellung ein, da neben der Nutfolge die Einhaltung einer konstanten Bahngeschwindigkeit die Hauptforderung an die Bewegungsführung sind. Die Bahngeschwindigkeit sollte ggf. an den aktuellen Klebstoffdurchfluß angepaßt werden. Das Bearbeitungsverfahren ist durch hohe Bahngeschwindigkeiten bei mittleren Bahngeauigkeiten charakterisiert [79].

Anwendungen	Bahngeschwindigkeiten, Kräfte	Genauigkeiten
Laserstrahlschneiden	1 - 15 m/min	0,2 mm
Laserstrahlschweißen	1 - 6 m/min	0,2 mm
Laserstrahlbiegen	1 - 10 m/min	0,2 mm
Bahnschweißen	0,5 - 3 m/min	0,8 mm
Kleberauftrag	10 - 30 m/min	0,7 mm
Entgraten von Kunststoff	5 - 15 m/min	0,5 mm
Entgraten von Keramik	0,1 - 1 m/min	0,3 mm
Kleingerätemontage	5 - 500 N	0,3 mm

Tabelle 2: Bearbeitungstechnologien und ihre Kenngrößen [12, 18, 31, 79]

Die Integration von Sensoren wird bei kraftschlüssigen Bearbeitungsverfahren wie dem Entgraten und dem Schleifen von Rohteilen und in der Montage angestrebt. Beim Entgraten muß hauptsächlich die Position des Grats und seine Geometrie erfaßt und das Werkzeug entlang des Gratverlaufs am Werkstück geführt werden. Die Bahngeschwindigkeit des Werkzeugs sollte abhängig von der Gratgeometrie variiert werden, um eine optimale Abtragleistung zu erzielen und eine Überlastung des Werkzeugs zu verhindern [127]. Die Abtragleistung wird beim Schleifen im wesentlichen von den wirkenden Andruckkräften und der Wirkzeit, die sich direkt von der Bahngeschwindigkeit ableitet, bestimmt. Die möglichen Bahngeschwindigkeiten hängen dabei stark von den Werkstoffeigenschaften des Werkstücks und dem verwendeten

Werkzeug ab. In Tabelle 2 sind typische Werte für Kunststoff- und Keramikwerkstücke aufgeführt [97, 127].

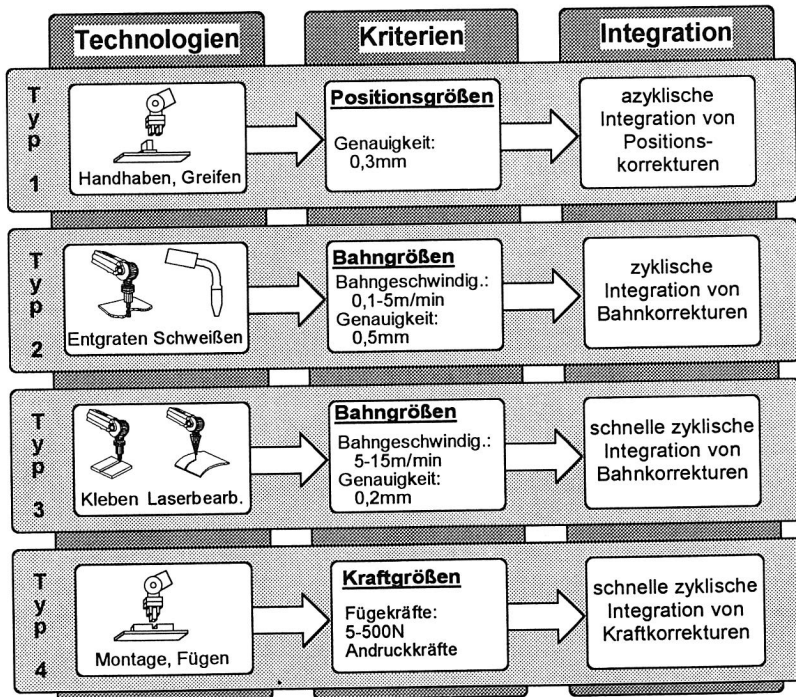


Bild 23: Ableitung der Anforderungen an die Sensordatenintegration

Bei der Gerätemontage stellen Greif- und Fügevorgänge den Hauptanteil der Montagefunktionen dar, die sensorisch sinnvoll unterstützt werden können. Der Greifvorgang wird hauptsächlich vom exakten Positionieren des Greifers über dem Werkstück beeinflusst. Dies kann durch eine vorab erfolgte Bestimmung der Werkstückposition geeignet sensorisch ausgeführt werden. Die für den Fügevorgang kennzeichnenden Prozeßgrößen sind hauptsächlich die Fügekräfte und -momente sowie die Fügetoleranzen. Die statisch wirkenden Fügekräfte liegen oftmals im Lastbereich des Roboters, so daß sie von diesem direkt aufgebracht werden können. Die Anforderungen an die Einhaltung der aufzubringenden Kräfte sind nicht sehr hoch, Abweichungen von 10% und mehr können problemlos hingenommen werden. Bei der Montage von Kunststoffteilen kommen häufig größere Fügetoleranzen vor.

Analysiert man die in Tabelle 2 aufgeführten Verfahren an Hand ihrer technologischen Kenngrößen, so lassen sie sich in vier Typklassen einteilen (Bild 23). Dabei kann zunächst zwischen Verfahren unterschieden werden, bei denen das Werkzeug entlang einer definierten Bahn oder

auf einer beliebiger Bahn zu einer vorgegebenen Position bewegt werden muß. Weiter lassen sich Verfahren zusammenfassen, die definierte Kräfte am Werkzeug aufbringen sollen. Schließlich können Bearbeitungstechnologien, die eine exakte Bahnführung des Werkzeugs voraussetzen, in die Typklassen 2 und 3 eingeordnet werden. Die Einteilung erfolgt dabei nach den bei der Bearbeitung notwendigen Bahngrößen, die hauptsächlich durch die Bahngeschwindigkeit und die Bahngenauigkeit beschreiben werden. Die verschiedenen Typen stellen aber auch unterschiedliche Anforderungen an die Integration von Sensordaten, die eine Bahn- bzw. Positionsadaption des Roboters bewirken sollen:

1. Der Typklasse 1 sind Montagevorgänge, bei denen Positionen mit einer mittleren Genauigkeit von 0,3 mm einzunehmen sind, zugeordnet. Typische Beispiele hierfür sind das Greifen eines Werkstücks. Sollen diese Vorgänge sensorunterstützt durchgeführt werden, so muß das Sensor-Roboter-System eine azyklische Integration von Positionskorrekturen ermöglichen.
2. Beim Bahnschweißen bzw. Entgraten muß die Bahn des Werkzeug laufend an den Grat bzw. an die Naht des Werkstücks angepaßt werden. Das Werkzeug bewegt sich dabei typischerweise mit einer Bahngeschwindigkeit von 0,1 - 5m/min und einer erforderlichen Bahngenauigkeit von ca. 0,5 mm über das Werkstück. Eine zyklische Integration von Bahnkorrekturwerten ist erforderlich, um eine sensorgestützte Adaption der Bewegungsbahn zu erreichen.
3. Die Bearbeitungstechnologien der Laserstrahlbearbeitung, des Klebenauftrags und des Entgratens von Kunststoffen unterscheiden sich von den Verfahren des Typs 2 in den höheren Anforderungen hinsichtlich der Bahngeschwindigkeit und der Bahngenauigkeit. Folglich wird bei einer sensorgestützten Adaption der Bewegungsbahn eine schnelle zyklische Integration von Bahnkorrekturwerten erforderlich.
4. Die Typklasse 4 vereint Fertigungsverfahren, wie z.B. das Schleifen oder das Fügen, bei denen definierte Andruck- oder Fügekräfte auf das Werkzeug aufzubringen sind. Sollen diese Vorgänge sensorgestützt durchgeführt werden, so ist eine schnelle, zyklische Integration von Kraftkorrekturwerten notwendig.

Um die technologischen Anforderungen der vier Typklassen von Fertigungsverfahren zu erfüllen, sollte ein Sensor-Roboter-System über die erforderlichen Formen von Integrationsstellen verfügen. Bisher steht aber noch nicht fest, welche geometrischen Bahn-, Positions-, und Kraftgrößen abhängig von der Anwendung zu integrieren sind. Dazu lassen sich die verschiedenen Anwendungen in die nachfolgenden fünf Typklassen einordnen. Die Sensordaten werden dabei jeweils in der gleichen Form in die Robotersteuerung integriert:

Positionskorrektur:

Der Endpunkt einer Bahn soll hierbei an die aktuellen Erfordernisse, wie z.B. die Änderung einer Greifposition bei der Montage, angepaßt werden. Um alle Möglichkeiten einer Positionskorrektur verwirklichen zu können, sollten alle 6 Positionsgrößen (Translation und Rotation) unabhängig voneinander azyklisch zur aktuellen Bewegung korrigierbar sein.

Abstandsführung:

Viele Anwendungen wie das Bahnschweißen benötigen eine präzise Regelung des Abstands zwischen Werkzeug und Werkstück. Das Werkzeug sollte oftmals eine definierte Orientierung zur Werkstückoberfläche einnehmen. Um diese Anforderung zu erfüllen, müssen Abstandskorrekturen zyklisch an die Robotersteuerung übertragen werden. Solange die Werkstückkontur nicht stark gekrümmt ist, kann auf eine zusätzliche Orientierungsführung verzichtet werden.

Nahtfolge:

Eine Vielzahl von Anwendungen wie der Kleberauftrag erfordern, daß das Werkzeug einer Naht oder Nut am Werkstück folgt. Der Abstand zur Naht muß einen vorgegebenen Wert einnehmen und die Orientierung des Werkzeug relativ zur Naht darf innerhalb von vorgegebenen Grenzen schwanken. Außerdem muß die Bahngeschwindigkeit des Werkzeugs eine definierte Größe einhalten. Solange das Werkstück keine ausgeprägte, dreidimensionale Kontur aufweist, kann die Orientierungsführung des Werkzeugs oft vernachlässigt werden. In diesem Fall sind Korrekturen der Bahn in alle drei Koordinatenrichtungen des Werkzeugkoordinatensystems notwendig, um sowohl den Abstand zum Werkstück und die Nahtfolge als auch die Bahngeschwindigkeitsvorgabe erfüllen zu können. Bei einer ausgeprägten, dreidimensionalen Werkstückkontur müssen außerdem 2-3 Orientierungswinkel korrigiert werden. Die Korrekturgrößen müssen zyklisch bzgl. des Werkzeugkoordinatensystems übertragen werden.

Geschwindigkeitsführung:

Anwendungen wie das Entgraten erfordern eine laufende Anpassung der Bahngeschwindigkeit an die aktuelle Gratgeometrie des Werkstücks. Falls zusätzlich keine weiteren, geometrischen Größen beeinflußt werden müssen, kann diese Forderung durch eine zyklische Änderung des Overrides, der ein Maß für die prozentuelle Reduktion der programmierten Bahngeschwindigkeit darstellt, erfüllt werden. Andernfalls muß die Änderung der Bahngeschwindigkeit über eine Korrektur der kartesischen Bahngrößen erreicht werden.

Kraftregelung:

Bei Anwendungen wie dem Fügen von Bauteilen oder dem Schleifen von Werkstücken müssen oftmals definierte Kräfte auf das Werkstück aufgebracht werden. Dazu lassen sich zyklisch

Korrekturkräfte bestimmen, die dann unter Verwendung der Steifigkeitsmatrix des Roboters in Positionskorrekturen umgerechnet und an die Robotersteuerung übertragen werden.

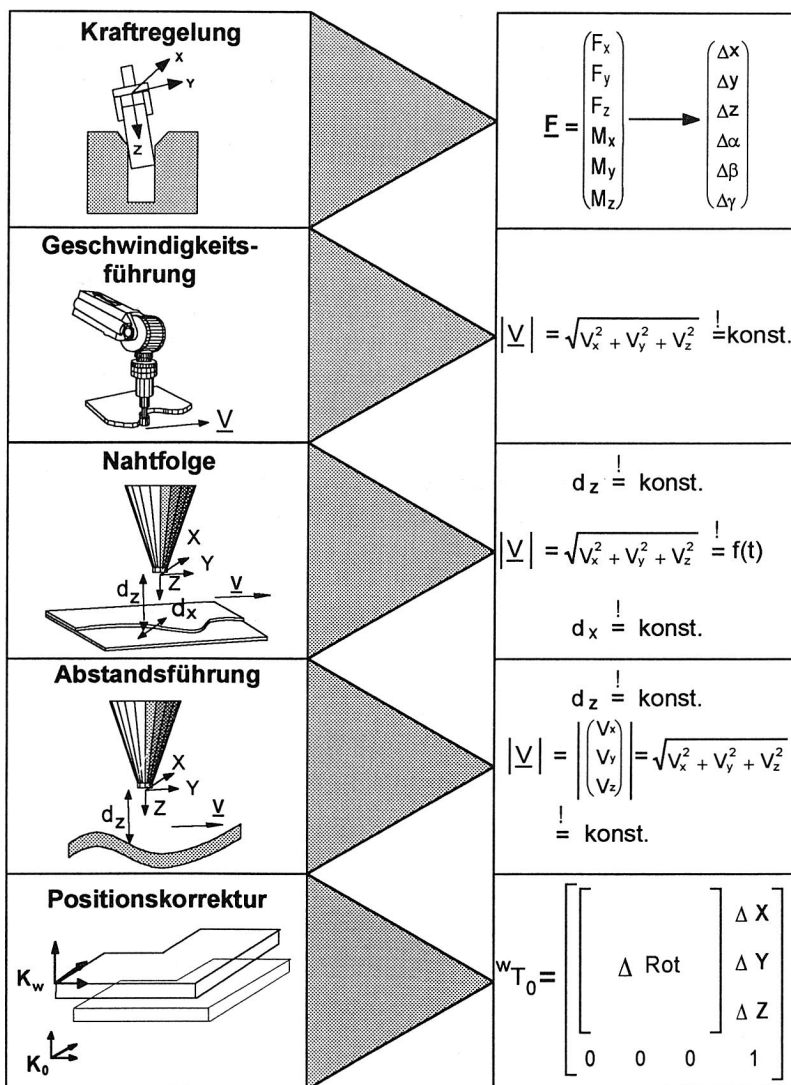


Bild 24: Technologische Typklassen und ihre Anforderungen

3.2 Globale Systemziele

Ein Sensor-Roboter-System, das eine sensorgestützte Bewegungsführung ermöglicht, muß eine hohe Flexibilität aufweisen. So müssen unterschiedliche Sensoren angeschlossen werden können, um eine Vielzahl von unterschiedlichen Anwendungen mit ein und dem selben System implementieren zu können. Dies erfordert eine entsprechende Konfigurierbarkeit und Adaptierbarkeit des Hardware- und Softwaresystems, um so eine angepaßte, effiziente und zuverlässige Lösung des jeweiligen Anwendungsproblems zu verwirklichen. Es soll damit vermieden werden, daß je Problemstellung ein entsprechendes System neu entwickelt wird.

Im Mittelpunkt der Anforderung an ein Sensor-Roboter-System steht die schnelle und effiziente Erfassung und Verarbeitung von Sensordaten. Insbesondere sollen die Daten mit minimalen Verzögerungszeiten zwischen den Teilkomponenten des Systems übertragen werden. Beim Aufbau von Sensorregelkreisen ist eine hohe Gesamtdynamik der Regelung erforderlich, um bei hohen Bahngeschwindigkeiten Fehler in der Werkstückkontur mit der geforderten Genauigkeit auszugleichen. Dazu sind im wesentlichen kurze An- und Ausregelzeiten der Sensorregelung gefordert.

Wesentliche Anforderung an ein Sensor-Roboter-System sind niedrige Kosten sowohl in der Anschaffung, im Unterhalt sowie bei der Erstellung von Applikationen. Dies bedingt eine an die jeweilige Applikation angepaßte Hardwarelösung als auch effiziente Mittel bei der Erstellung und dem Test von Anwenderprogrammen.

Schließlich muß der Anwender bei der Erstellung einer sensorgestützten Applikation durchgehend unterstützt werden. Insbesondere mit Hilfe von geeigneten Entwicklungs- und Simulationswerkzeugen als auch Hilfsmitteln zur Analyse, Visualisierung und Archivierung von Testdaten und erprobten Programmen lassen sich die Programme effizient erstellen, testen und verwalten. Außerdem sollte ein Sensor-Roboter-System von einer einzigen Anwenderschnittstelle aus bedienbar sein. Die abgeleiteten Anforderungen lassen sich nun auf die verschiedenen Teilkomponenten eines Sensor-Roboter-Systems abbilden.

3.3 Anforderungen an die Sensorik

Die innerhalb eines Sensor-Roboter-Systems verwendeten Sensorsysteme müssen vielfältige Anforderungen, die oftmals abhängig von den jeweiligen Erfordernissen der Anwendung sind, erfüllen (Bild 25).

Die Sensorsignalerfassung und -verarbeitung muß hohe Auswerte- und Verarbeitungsgeschwindigkeiten aufweisen, um die Verzögerungszeiten des Sensorsystem gering zu halten [47, 155]. Dies ist v.a. bei der On-line-Bahnadaption eine unverzichtbare Voraussetzung. Das Sensormeißprinzip sollte auch über eine hohe Signalauflösung sowie eine entsprechende Meßgenauigkeit verfügen [31].

Ferner muß der Meßaufnehmer robust gegenüber dem Einsatz in unmittelbarer Prozeßumgebung und den damit verbundenen Störeinflüssen sein, so daß immer eine sichere Bereitstellung der notwendigen Information gewährleistet ist [31]. Das Sensorsystem sollte bei einer Vielzahl von Anwendungen, die ähnliche Rahmenbedingungen aufweisen, flexibel einsetzbar sein [53].

Um die Zugänglichkeit am Werkstück und die Nutzlast des Roboters nicht wesentlich zu beeinträchtigen, müssen Sensorsysteme über einen möglichst kleinen, robusten mechanischen Aufbau und ein geringes Gewicht verfügen. Die Sensorsysteme sollten außerdem in verschiedenen Anschlußvarianten zur Verfügung stehen, um den Installations- und Aufbauaufwand in Grenzen zu halten [47].

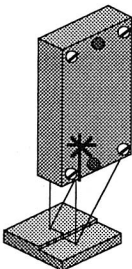
Anforderungen an die Sensorik 	Signalverarbeitung: <ul style="list-style-type: none"> - hohe Verarbeitungsgeschwindigkeit - hohe Auflösung
	Robustheit: <ul style="list-style-type: none"> - Anwendungsflexibilität - Störsicherheit
	Anbau: <ul style="list-style-type: none"> - geringe Baugröße, Gewicht - Anschlußflexibilität
	Programmierung: <ul style="list-style-type: none"> - Parametrierbarkeit - Unterstützung bei der Programmerstellung

Bild 25: Anforderungsprofil an die Sensorik

Um die Integration der Sensoren ins System zu erleichtern, müssen ferner Programmodule zur Sensorwerterfassung und -verarbeitung zur Verfügung stehen. Damit kann die Programmerstellung im Sensor-Roboter-System oft vereinfacht und beschleunigt werden. Entsprechend sollte eine ggf. erforderliche Parametrierung des Sensorsystems automatisch durchgeführt werden [120].

3.4 Aufgabenprofil einer Robotersteuerung

Aus Gründen des Aufwands und der Machbarkeit erscheint es sinnvoll, eine industrielle Robotersteuerung einzusetzen. Damit ist man zwar an die typischen Steuerungs- und Regelungsstrukturen industrieller Robotersteuerungen gebunden, kann aber auf ein bestehendes, erprobtes Steuerungs-, Bedien- und Programmiersystem zurückgreifen. Außerdem entsteht somit keine Sonderlösung, sondern die Lösung kann in andere, industrielle Robotersteuerungen ent-

sprechend umgesetzt werden. Zudem sollte die Robotersteuerung der neuesten Steuerungs-generation angehören und über eine leistungsfähige Bewegungsführung und Bedienung verfügen.

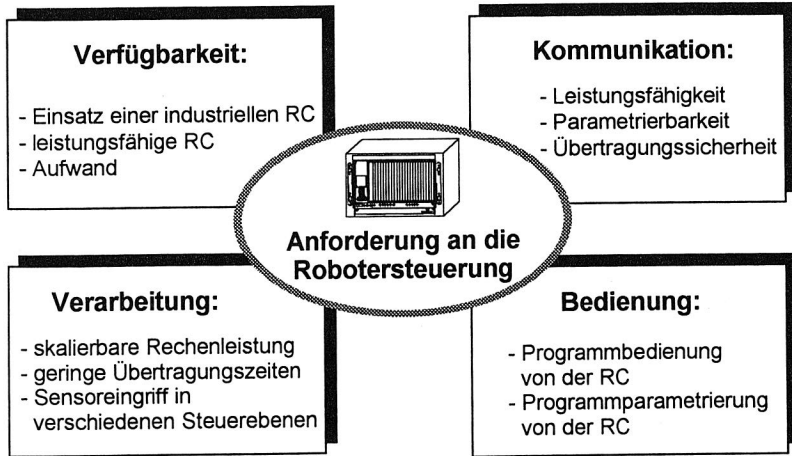


Bild 26: Anforderungen an die Robotersteuerung

Um Sensordaten geeignet in die Robotersteuerung integrieren zu können, muß die Robotersteuerung eine leistungsfähige Kommunikationsschnittstelle aufweisen [81]. Dies setzt meist eine sehr enge Kopplung zwischen der Robotersteuerung und der Sensorik voraus. Um eine an die jeweilige Anwendung angepaßte Sensorintegrationslösung zu erhalten, muß eine Parametrierbarkeit der Kommunikationsschnittstelle gefordert werden. So sollen sowohl aktuelle Zustandsdaten der Robotersteuerung wie kartesische Ist- und Sollpositionen als auch Korrekturdaten übertragen werden. Die Korrekturdaten müssen dabei sowohl in die azyklischen Hierarchieebenen als auch in die zyklischen Hierarchieebenen der Robotersteuerung eingreifen. Die Übertragung von Daten soll dabei sicher durchgeführt werden, und bei eventuell auftretenden Fehlern muß das System in einen sicheren Zustand überführt werden. Zur Berechnung der Korrekturwerte muß die Robotersteuerung um zusätzliche Rechenleistung in Form einer Baugruppe ergänzt werden. Die Baugruppe soll im weiteren als Sensorrechner bezeichnet werden. Geringe Verzögerungszeiten bei der Übertragung von Korrekturdaten und bei der Einbringung der Korrekturwerte in die Robotersteuerung sind weitere Forderungen, die an die Robotersteuerung gestellt werden, um eine schnelle Reaktion des Roboters auf Sensorsignale zu gewährleisten. An der Robotersteuerung sollte schließlich die Möglichkeit bestehen, die Sensorprogramme des Sensorrechners anwählen, starten, beenden und parametrieren zu können.

3.5 Forderungen an einen Sensorrechner

Der Sensorrechner übernimmt zunächst die Aufgabe, Sensordaten zu empfangen, die an unterschiedlichen Schnittstellen zur Verfügung gestellt werden. Entsprechend den unterschiedlichen Sensorsignalen gibt es verschiedene Sensorschnittstellen [30]. Analoge Schnittstellen sind industriell anerkannte Schnittstellen, die geringe Kosten für die Verkabelung aber eine hohe Empfindlichkeit gegenüber Störsignalen aufweisen. Ein hervorragender Schutz gegenüber elektromagnetischen Störungen sowie die Fähigkeit, Daten über große Entfernungen zu übertragen, kennzeichnet serielle, digitale Schnittstellen, wobei aber nur sehr begrenzte Übertragungsgeschwindigkeiten möglich sind. Feldbusse sind spezielle Formen von seriellen, digitalen Schnittstellen, die hinsichtlich der Übertragungsmechanismen, -sicherheit und -geschwindigkeit optimiert sind. Typische Vertreter von Feldbussen sind der Profibus [23] und der FIP [148, 149].

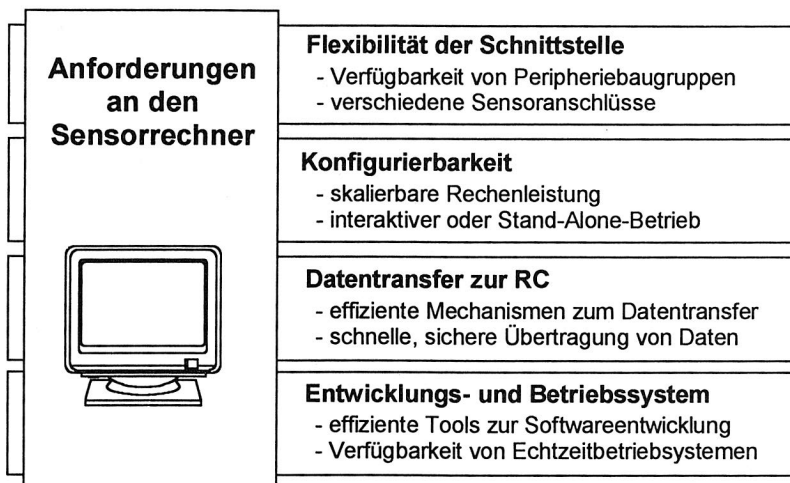


Bild 27: Anforderungen an den Sensorrechner

Ein modularer Aufbau des Sensorrechners stellt außerdem die Konfigurierbarkeit des Hardwaresystems sicher und bedingt zusätzlich eine Skalierbarkeit der Rechenleistung. Somit kann die Rechenleistung des Sensorrechners an die Erfordernisse der Sensorsignalverarbeitung angepaßt werden. Durch die Modularität des Sensorrechners können zusätzliche Funktionen der Sensorintegration, die je nach den jeweiligen Anforderungen der Aufgaben variieren, einfach integriert werden. Der Aufbau des Sensorrechners muß eine zur Robotersteuerung kompatible Hardwarestruktur aufweisen. Insbesondere sollte die Übertragung von Daten zwischen Sensorrechner und der Robotersteuerung sicher, schnell und bidirektional erfolgen und durch effiziente Hardwaremechanismen beschleunigt werden.

Schließlich müssen für die Hardwareplattform des Sensorrechners wirkungsvolle und bedienerfreundliche Betriebs- und Entwicklungssysteme zur Verfügung stehen. Nachdem die Integration von Sensordaten in der Regel on line und unter Echtzeitbedingungen erfolgt, muß ein zeitlich deterministisches Verhalten des Betriebssystems auf externe Ereignisse gefordert werden. Die Reaktion des Betriebssystems muß auf Grund der sehr kurzen Zykluszeiten in den unteren Hierarchieebenen der Robotersteuerung (1-10 ms) innerhalb kürzester Zeit erfolgen. Das Entwicklungssystem sollte ferner eine effiziente Erstellung von Anwenderprogrammen ermöglichen. Dazu müssen erprobte und wirkungsvolle Softwarewerkzeuge zur Verfügung stehen.

3.6 Anforderungsprofil eines integrierten Gesamtsystems

Das integrierte Gesamtsystem hat die Aufgabe, den Anwender während des gesamten Zeitraums, beginnend mit der Programmerstellung bis zur abschließenden Erprobung einer Sensorintegrationslösung, wirksam zu unterstützen. Dabei ist es erforderlich, entsprechende Hilfsmittel zur Verfügung zu stellen.

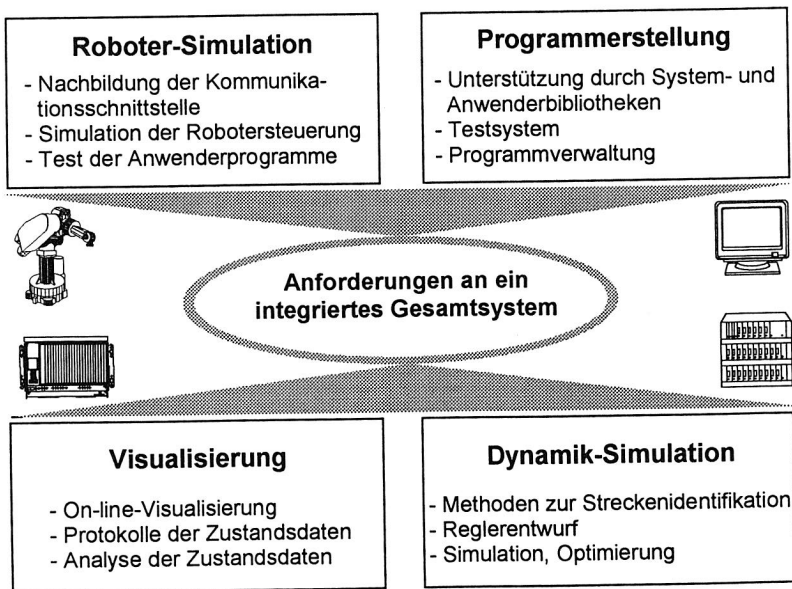


Bild 28: Anforderungen an ein integriertes Gesamtsystem

Während des Programmiervorgangs sollten mächtige Bibliotheken verfügbar sein, die Systemfunktionen, Sensorerfassungs- und -verarbeitungsmodule und anwendungsspezifische Funktionen zur Vereinfachung der Programmierung beinhalten. Schließlich kann die Wiederver-

wendbarkeit von bestehenden, erprobten Programmen durch wirkungsvolle Werkzeuge zur Programmverwaltung erhöht werden, und somit lassen sich Entwicklungskosten senken.

Zum Entwurf von wirkungsvollen Sensorsteuerungs- und Regelungskonzepten sind Werkzeuge zur Identifikation von Strecken, zum Entwurf von Reglern sowie zur Simulation des gesteuerten bzw. geregelten Gesamtsystems unabdingbar. Damit können frühzeitig Untersuchungen zur Realisierbarkeit von technologischen Forderungen wie spezifizierte Bahngeschwindigkeiten bei definierten Genauigkeiten durchgeführt werden.

Um bei der Erprobung von Sensorprogrammen Schäden an der Roboteranlage zu vermeiden, sollten die wesentlichen Teile des Roboters und der Robotersteuerung in einem Simulationssystem nachgebildet werden. Dabei ist es erforderlich, das Kommunikationsverhalten der Robotersteuerung und das Verhalten des Roboters gegenüber seiner Umwelt zu simulieren, um den Test eines neu geschaffenen Anwenderprogramms zu ermöglichen.

Außerdem sollten Möglichkeiten bestehen, die aktuellen Zustandsdaten wie Sensor- und Bahnwerte on line an der Anlage zu visualisieren. Damit wird dem Anwender ein Hilfsmittel zur Verfügung gestellt, Probleme frühzeitig zu erkennen und ggf. korrigierend einzugreifen. Ferner sollten wirkungsvolle Werkzeuge es ermöglichen, Zustandsdaten protokollieren, analysieren und archivieren zu können, um Optimierungen oder qualitätssichernde Maßnahmen durchzuführen.

4. Entwurf eines konfigurierbaren Gesamtsystems zur Sensorintegration

Die in Kapitel 3 erarbeiteten Anforderungen wurden exemplarisch in das konfigurierbare Gesamtsystem ISRS (Integriertes Sensor-Roboter-System) umgesetzt, das eine flexible Sensorintegration an Robotersteuerungen vom Typ SIROTEC ACR 20 (Siemens AG) [5] ermöglicht. Der Ausführende wird in den einzelnen Entwicklungsphasen einer Sensorintegrationslösung, die Bereiche wie die Programmerstellung, den Entwurf von Sensorregelungen, den Test und die Optimierung umfassen, weitgehend vom Gesamtsystem ISRS unterstützt. Zunächst wird die Struktur des modularen Gesamtsystems ISRS entworfen und an Hand eines Architekturmodells in Teilsysteme zerlegt. Die resultierenden Komponenten werden nachfolgend vorgestellt und näher erläutert.

4.1 Struktur des integrierten Gesamtsystems ISRS

Der strukturelle Aufbau des Gesamtsystems ISRS, dessen Entwurf zunächst implementierungsunabhängig erfolgt, wird vorgestellt und im weiteren auf eine geeignete Hardwarestruktur abgebildet. Die besonderen Merkmale des Gesamtsystems ISRS wie die Funktionsmerkmale der exemplarisch eingesetzten Robotersteuerung und die Auswahl eines der Hardwareplattform angepaßten Betriebssystems werden danach präsentiert.

4.1.1 Struktureller Aufbau des Gesamtsystems

Ein integriertes Gesamtsystem zur Sensorintegration muß verschiedenste Anwendungsgebiete abdecken und den Anwender auf den unterschiedlichsten Ebenen der Programmierung und während des Tests an der Anlage optimal unterstützen. Das Gesamtsystem soll folglich flexibel im Hardwareausbau und insbesondere im Funktionsumfang der Steuerungssoftware sein. Ein modulares Architekturmodell ist die ideale Struktur, um diese Forderungen zu erfüllen und so z.B. Funktionen auf optimierten Rechnerkomponenten abzuarbeiten. Eine der wesentlichen Grundmodule des Systems stellt die Robotersteuerung dar, die als eigenständige Komponente konzipiert wird. Somit können verfügbare, industrielle Steuerungen, die die Kommunikationsanforderungen erfüllen, eingesetzt werden.

Um die Aufgabe der Sensordatenerfassung und der Sensordatenintegration durchzuführen, wird das zweite, bedeutende Grundmodul des Systems, die Sensorsteuerung SC (Sensor Control), entworfen. Die Sensorsteuerung SC ist das Gegenstück zur Robotersteuerung und gleichzeitig dessen Kommunikationspartner. Aus diesem Grund ist eine enge Kopplung zwischen der Robotersteuerung und der Sensorsteuerung anzustreben. Die beiden Grundmodule müssen deshalb über eine leistungsfähige Kommunikationsschnittstelle in Verbindung stehen, deren Hard- und Softwarearchitektur noch zu konzipieren ist.

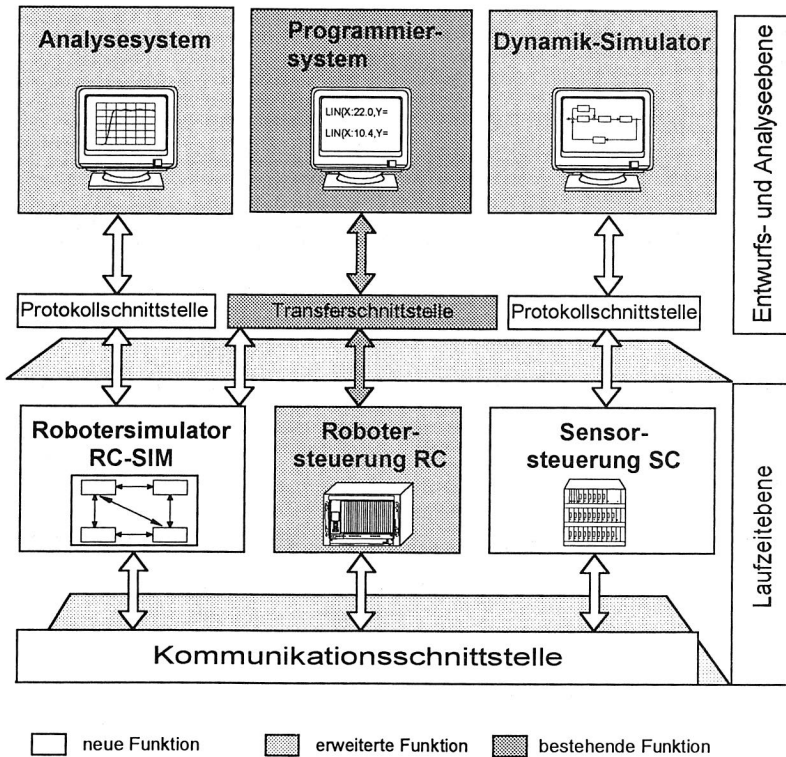


Bild 29: Architekturmodell des integrierten Gesamtsystems ISRS

Das in Kapitel 3 erarbeitete Anforderungsprofil, das eine Sensorintegration auf drei funktionalen Ebenen verlangt, kann durch den bisher vorgestellten Entwurf erfüllt werden. Die beiden Teilsysteme RC und SC ermöglichen die bidirektionale, azyklische wie zyklische Übertragung von Daten auf den Ebenen der Lageregelung, der Bahnplanung als auch der Ablaufplanung und Programminterpretation (Bild 30). Die gesamte Ablaufsteuerung der Sensorintegration wie die Initialisierung der Sensorprogramme kann von der Robotersteuerung aus vorgenommen werden, indem entsprechende Daten auf der Ebene der Ablaufplanung übertragen werden. Ferner besteht die Möglichkeit, die Übertragung von Daten entsprechend den Anforderungen zu initialisieren. Damit wird z.B. verhindert, daß Daten auf der Ebene der Lageregelung oder Bahnplanung laufend ausgetauscht werden, obwohl die Aufgabe dies nicht erfordert. Infolgedessen wird nicht unnötigerweise Rechenleistung in der Roboter- und der Sensorsteuerung beansprucht, die in solchen Fällen zu höheren Anforderungen an die Hardwarekomponenten führen. Eine hohe Flexibilität in der Anwendung wird durch die Konfigurierbarkeit und Adaptierbarkeit des Hard- und Softwaresystems nicht mit höheren Anforderungen an die Rechen-

leistung erkaufte. Die weitere Detaillierung des Aufbaus der Sensorsteuerung und der Entwurf des Kommunikationskonzepts der Datenübertragung wird in Kapitel 4.2.1 vorgestellt.

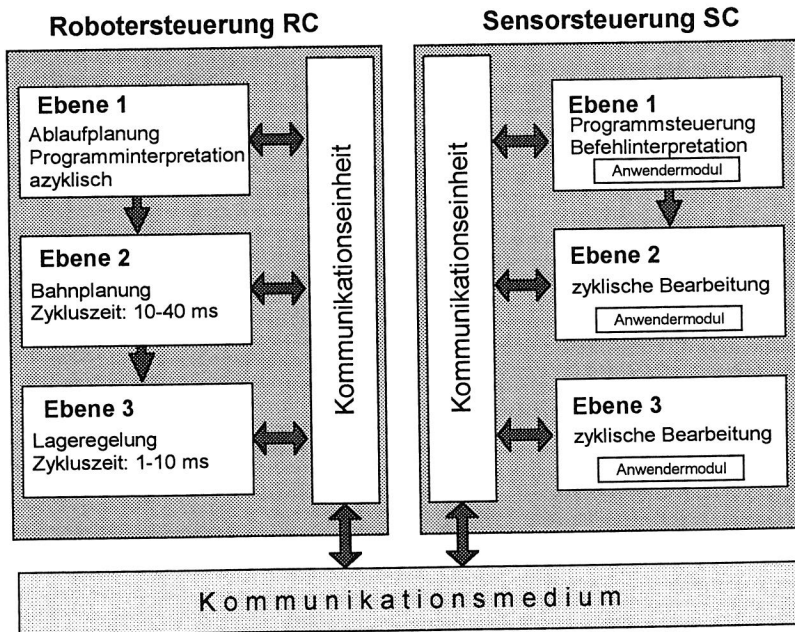


Bild 30: Kommunikationsebenen der Roboter- und Sensorsteuerung

Die Entwicklung und der Test von Anwendermodulen im SC-System lässt sich erheblich erleichtern, wenn in der Entwicklungsphase auf die Roboteranlage und die Robotersteuerung verzichtet werden kann. Durch einen Robotersimulator, der sowohl die notwendigen, informationstechnischen Strukturen als auch die erforderlichen, systemdynamischen Eigenschaften der Robotersteuerung und des Roboters nachbildet, können bereits frühzeitig die Anwendermodule der Sensorsteuerung SC als auch die zugehörigen RC-Programme erprobt werden. Eine Beschädigung der Roboteranlage, der Sensorik und des Werkstücks kann so auch in Ausnahmesituationen vermieden werden. Die Funktion der Robotersteuerung übernimmt der Simulator RC-SIM, der anstatt der Robotersteuerung über die Kommunikationsschnittstelle mit der Sensorsteuerung kommuniziert (Bild 29). Die notwendige Modellierung der Robotersteuerung und deren Simulation durch das Grundmodul RC-SIM wird in Kapitel 4.4 näher erläutert.

Die drei Grundmodule RC-SIM, RC, SC bilden die untere Ebene des Gesamtsystems ISRS, die im weiteren als Laufzeitebene bezeichnet wird. Die Übertragung von Daten zwischen den drei Modulen erfolgt über eine festgelegte Kommunikationsschnittstelle. Oberhalb der Laufzeitebene ist die Entwurfs- und Analyseebene angeordnet, die im wesentlichen Hilfsmittel zum

Entwurf und zur Validierung von Modulen der Laufzeitebene zur Verfügung stellt. Die Module der Entwurfs- und Analyseebene sind über definierte Schnittstellen mit den Systemen der Laufzeitebene verbunden (Bild 29).

Die Erstellung von Roboterprogrammen kann durch den Einsatz der Systemkomponente „Programmiersystem“ vereinfacht werden. So können mit Hilfe dieser Komponente Roboterprogramme für sensorgestützte Anwendungen off line von der Roboteranlage erstellt werden. Zusätzlich lassen sich die Roboterprogramme übersichtlich verwalten und archivieren, so daß eine bessere Wiederverwertbarkeit von bestehenden und erprobten Roboterprogrammen erreicht wird. Durch den vorgesehenen Einsatz einer kommerziellen Robotersteuerung sind meist auch geeignete Off-line-Programmiersysteme verfügbar. Über eine Transferschnittstelle werden Roboterprogramme bidirektional zwischen der Robotersteuerung und dem Programmiersystem übertragen. Falls die Robotersteuerung von RC-SIM simuliert wird, müssen die Roboterprogramme entsprechend an den Simulator transferiert werden.

Eine weitere Systemkomponente des integrierten Gesamtsystems ISRS stellt das Analysesystem dar, das eine Untersuchung der während einer Anlagensimulation mit RC-SIM ermittelten Kenngrößen erlaubt (Bild 29). Die während eines Simulationslaufs von RC-SIM erfaßten Zustandsgrößen werden zunächst über eine definierte Protokollschnittstelle an das Analysesystem übertragen. Dort besteht die Möglichkeit, die signifikanten Protokolldaten zu visualisieren und interaktiv entscheidende Zeit- und Wertebereiche der Protokolldaten zu untersuchen. Der Anwender gewinnt damit Aufschluß über Problembereiche der aktuell simulierten Anwendung und kann entsprechende Änderungen am Roboter- oder Sensorprogramm vornehmen. Dazu bedient er sich des Programmiersystems. Weitere Details des Analysesystems werden in Kapitel 4.4 besprochen.

Schließlich beinhaltet das integrierte Gesamtsystem ISRS die Komponente „Dynamik-Simulator“, der den Entwurfsprozeß von Sensorregelungen effizient unterstützen soll (Bild 29). Insbesondere lassen sich mit dem Dynamik-Simulator

- ☐ der Entwurf von Sensorregelungs- oder Steuerungsstrategien,
- ☐ die Simulation des Übertragungsverhaltens der entworfenen Konzepte,
- ☐ und die Analyse und Visualisierung von realen Testläufen an der Roboteranlage effizient durchführen.

Zur Übertragung von Zustandsgrößen, die während eines Testlaufs ermittelt werden, steht eine Protokollschnittstelle zwischen der Sensorsteuerung SC und dem Dynamik-Simulator zur Verfügung, die flexibel an die unterschiedlichen Anwendungsfälle der Sensorsteuerung angepaßt werden kann. Der Dynamik-Simulator wurde entsprechend den Anforderungen als interaktiv zu bedienendes System entworfen, das in Kapitel 4.5 näher erläutert wird.

4.1.2 Hardwareaufbau

Die Struktur des integrierten Gesamtsystems ISRS wird im folgenden auf ein qualifiziertes Hardwarearchitekturmodell abgebildet, so daß die jeweiligen Teilsysteme von ISRS auf optimalen Rechnerkomponenten bearbeiten werden. Das gewählte Architekturmodell muß insbesondere variabel im Hardwareausbau und der vorhandenen Rechenleistung sein. Ferner sollte die Möglichkeit bestehen, abhängig von der Einsatzkonstellation, die Teilsysteme unterschiedlichen Hardwarekomponenten zuzuordnen. Diese Aufgaben erfordern ein Architekturmodell, das modular aufgebaut ist und aus mehreren Rechneinheiten besteht.

Der Zusammenschluß von mehreren Rechnern wird häufig als Cluster bezeichnet. Unter einem Cluster wird üblicherweise eine Menge lose gekoppelter Prozessoren verstanden, die über einen gemeinsamen Speicher verfügen [138]. Die Minimalausführung eines Clusters besteht aus einem Prozessor, der mit einem Multitaskingbetriebssystem ausgestattet ist.

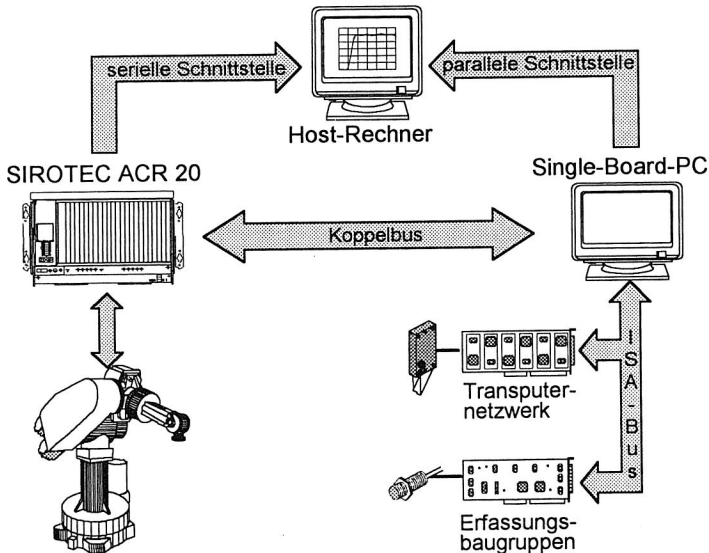


Bild 31: Konfigurierbare Hardwarestruktur des integrierten Gesamtsystems ISRS

Bedingt durch den Aufbau der eingesetzten Robotersteuerung wurde ein Cluster entworfen, das sich aus der Robotersteuerung, einem Single-Board-PC und einem Transputer-Netzwerk zusammensetzt und über 2-7 Prozessoren verfügt. Die drei Teilsysteme RC, SC und RC-SIM der Laufzeitebene des Gesamtsystems ISRS können auf dieses Rechnercluster abgebildet werden. Die RC kann über einen Koppelbus und einen gemeinsamen Speicher (DPR) mit dem Single-Board-PC kommunizieren. Der Single-Board-PC beinhaltet die Teilsysteme SC und RC-SIM

des Gesamtsystems ISRS. Für rechenintensive Operationen, wie sie bei der Integration von Sensordaten in die Lageregelungsebene abzuarbeiten sind, wird der Single-Board-PC um ein Subsystem, das aus einem Transputernetz mit bis zu 5 Prozessoren besteht, erweitert. Das über den PC-Bus und einen gemeinsamen Speicher (DPR) angekoppelte Transputernetzwerk führt außerdem Aufgaben der Sensorwerterfassung und der Koordinatentransformation durch. Aus diesem Grund verfügt das Transputernetzwerk über eigene Baugruppen zur Sensorwerterfassung, die analoge Sensorsignale mit sehr hohen Taktraten abtasten können.

Der Single-Board-PC kann außerdem mit handelsüblichen, in vielen Varianten verfügbaren, kostengünstigen PC-Baugruppen zur Aufnahme von Sensorwerten ausgestattet werden. Dabei läßt er sich sowohl als integrierter Rechner (embedded controller), der nicht als eigenständiger Rechner in Erscheinung tritt [71], als auch als Bedienstation mit Visualisierungsmöglichkeiten betreiben. Weiterhin stehen dem Entwickler und Anwender auf PC-Architekturplattformen mächtige Softwarewerkzeuge zur Verfügung. Damit konnte eine Anordnung erzielt werden, die den jeweiligen Anforderungen optimal angepaßt werden kann und in der jeweiligen Ausbaustufe die notwendigen Funktionen bereitstellt. Das Laufzeitsystem von ISRS ist somit gekennzeichnet durch eine hohe anpaßbare Rechenleistung, Erweiterbarkeit und klare Strukturierung.

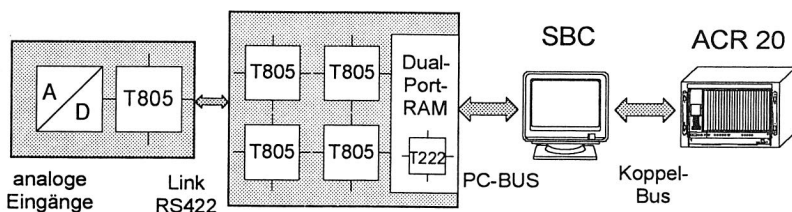


Bild 32: Hardwareaufbau des Transputernetzwerks

Die Entwurfs- und Analyseebene von ISRS wird auf einem Host-PC implementiert. Das verwendete kommerzielle Off-line-Programmiersystem, das eine PC-Architektur voraussetzt, ist über eine serielle Schnittstelle, die der Übertragung von Roboterprogrammen dient, verbunden. Der Datenaustausch zwischen dem Single-Board-PC und dem Host-PC erfolgt über eine parallele Schnittstelle. Durch den Einsatz der selben Rechnerarchitektur sowohl für die Sensorsteuerung als auch für die Entwurfs- und Analyseebene können die Module der Entwurfs- und Analyseebene auch vollständig auf dem Single-Board-PC abgearbeitet werden. Folglich kann in diesem Fall der Host-PC entfallen.

Während des Entwurfs und dem simulativen Test von Anwendungen können der Roboter-Simulator RC-SIM und die Sensorsteuerung SC unabhängig von der Roboteranlage und dem Single-Board-PC off line auf dem Host-PC betrieben werden. Insofern lassen sich die Anwender- und Roboterprogramme unabhängig von der Robotersteuerung und dem Single-Board-PC

auf dem Host-PC vorab testen. Die ausgewählte Architektur stellt somit eine größtmögliche Flexibilität sicher.

4.1.3 Funktionsmerkmale der Robotersteuerung

Eine flexibel konfigurierbare und schnelle Integration von Sensordaten erfordert moderne Robotersteuerungen, die über hohe Verarbeitungsgeschwindigkeiten, leistungsfähige Mechanismen der Bahnplanung und schnelle Kommunikationsschnittstellen verfügen. Nach einer eingehenden Untersuchung von kommerziellen Industrierobotersteuerungen hinsichtlich ihrer Einsetzbarkeit im integrierten Gesamtsystem ISRS wurde die Robotersteuerung SIROTEC ACR 20 ausgewählt. Die Robotersteuerung weist den typischen Steuerungsaufbau von industriellen Robotersteuerungen auf, womit eine weitgehende Übertragbarkeit der erzielten Ergebnisse auf andere Robotersteuerungen gleicher Struktur gegeben ist. Die Funktionsmerkmale einer modernen Robotersteuerung werden im folgenden charakterisiert [5, 6, 13, 92, 99].

Bessere Lageregelungskonzepte wie die Symmetrierung der einzelnen Achsdynamiken oder Geschwindigkeitsvorsteuerungen bewirken eine höhere Bahngenauigkeit des Roboters. Die Bewegungsbahnen sind weitgehend konturinvariant gegenüber Geschwindigkeitsreduktionen sowohl bei CP- als auch bei PTP-Bewegungen. Die Bewegungsplanung der Robotersteuerung gibt weiche und mechanikschonende Bewegungen vor. Dies wird durch die Einsatz entsprechend stetiger Geschwindigkeitsprofile erreicht. Mächtige Mechanismen, die Bewegungssätze durch Überschleifen verbinden, stehen zur Verfügung. Moderne Robotersteuerungen weisen neuartige Bedien- und Programmierkonzepte auf, die das Arbeiten mit der Robotersteuerung vereinfachen. Die Programmiersprache der Robotersteuerung ist weitgehend konform zur standardisierten Roboterprogrammiersprache IRL [26].

Die bisher verfügbaren Robotersteuerungen konnten die Forderungen nach einer flexiblen Integrationsstelle für Sensordaten nicht erfüllen [90]. Insbesondere bestanden keine Möglichkeiten, Sensordaten innerhalb der Ebene der Lageregelung zu integrieren. Die verwendete Robotersteuerung wurde deshalb prototypisch um eine flexible Kommunikationsschnittstelle zur Integration von Sensordaten entsprechend den Anforderungen erweitert. Die Kommunikationsschnittstelle weist die nachfolgenden Kennzeichen auf. Die Kopplung der Robotersteuerung mit dem Single-Board-PC erfolgt über einen gemeinsamen Speicher (DPR), der eine hohe Datensicherheit bei einer gleichzeitig verfügbaren hohen Datenübertragungsgeschwindigkeit gewährleistet [19]. Die Kommunikation wird von der Robotersteuerung durch das Auslösen von Interrupts am Single-Board-PC gesteuert. Eine rechtzeitige Reaktion der Kommunikationspartner ist damit gewährleistet, und eine hohe Systemlast, die andernfalls durch das zyklische Abfragen von Kommunikationszuständen resultiert, läßt sich verhindern. Die Sensordaten können auf allen drei Ebenen - Ablaufplanung, Bahnplanung und Lageregelung - der Robotersteuerung integriert werden (Bild 30). Zusammen mit der Konfigurierbarkeit der Kommunikationsschnittstelle ist dies eine wichtige Voraussetzung für die geforderte Flexibilität von Sensorintegrationslösungen. Schließlich stehen Kommunikations- und Sicherungsmechanismen zur

Verfügung, die z.B. den gleichzeitigen Zugriff auf Daten des gemeinsamen globalen Speichers verhindern.

4.1.4 Einsatz eines Echtzeit-Multitasking-Kerns

Das Teilsystem Sensorsteuerung SC muß auf Kommunikationsanforderungen der Robotersteuerung innerhalb eines festgelegten Zeitintervalls reagieren. Parallel dazu sollen zusätzlich Zustandsdaten der Sensorsteuerung, wie z.B. Sensorwerte, visualisiert werden. Wird während des Tests von Anwenderprogrammen der Robotersimulator RC-SIM verwendet, so müssen auf dem Single-Board-PC sowohl der Robotersimulator RC-SIM als auch die Sensorsteuerung SC des integrierten Gesamtsystems ISRS ablaufen. Teilfunktionen der beiden Systeme, wie z.B. die Kommunikationsfunktionen, müssen in einer spezifizierten Reihenfolge und zeitlich deterministisch ablaufen. Damit ist eine Echtzeitverarbeitung im Single-Board-PC unumgänglich. Die im weiteren verwendeten Begriffe zum Echtzeitbetrieb von Rechnern orientieren sich nach DIN 44300 [24].

Echtzeitbetriebssysteme unterstützen die Einhaltung von definierten Zeitbedingungen, indem sie geeignete Hilfsmittel zur Verfügung stellen. Der Kern eines Echtzeitverarbeitungssystems ist der Scheduler, der zwischen den Task umschaltet. Eine Task ist ein lauffähiges Programm mit den lokalen Daten und dem zugehörigen Programmzustand. Tasks laufen unabhängig voneinander, sie können allerdings miteinander kommunizieren und sich gegenseitig starten und anhalten. Eine Task kann abhängig von ihren Daten verschiedene Zustände annehmen. Das Echtzeitbetriebssystem verwaltet die Taskzustände und entscheidet, welche Task im Augenblick aktiv ist. Eine Task höherer Priorität kann eine niederprioritäre Task unterbrechen. Erfolgt die Unterbrechung, unmittelbar nachdem die höherprioritäre Task lauffähig wurde, so spricht man von einem preemptive Multitasking. Erfolgt dies erst nach einer definierten Zeit, so wird das Multitasking als nonpreemptive bezeichnet. Echtzeitbetriebssysteme mit einem preemptive Multitasking weisen somit kürzere Reaktionszeiten auf Ereignisse auf [58, 71].

Ein Taskwechsel kann auch durch einen Interrupt, der eine Unterbrechung des Programmablaufs durch ein äußeres Ereignis bewirkt, erfolgen. Tasks müssen synchronisiert werden, wenn sie gleichzeitig auf gemeinsame Ressourcen, wie z.B. Daten, zugreifen. Außerdem werden zwischen Tasks oftmals Daten ausgetauscht. Zur Tasksynchronisation und zur Inter-Task-Kommunikation stellt ein Echtzeitbetriebssystem Mechanismen wie Semaphoren, Mailboxen, Messages und Shared Memory zur Verfügung. Echtzeitbetriebssysteme können in die vier Klassen „Echtzeit-Kernel, Betriebssystemerweiterung, Echtzeit-Unix, Echtzeitbetriebssysteme“ eingeteilt werden [72]. An Hand der in der Literatur diskutierten Vor- und Nachteile der jeweiligen Klassen von Echtzeitbetriebssystemen läßt sich die Auswahl eines Echtzeitbetriebssystems vornehmen [46, 73, 117, 139].

Dazu werden zunächst die Anforderungen der Teilsysteme SC und RC-SIM an ein Echtzeitbetriebssystem analysiert und in einem Anforderungsprofil zusammengefaßt. Die Auswahl eines

Echtzeitbetriebssystem ist dabei oftmals eng verzahnt mit der Auswahl der Hardware-Plattform und insbesondere des Prozessors [33, 159]. Der systematisch durchgeführte Auswahlprozeß ergab, daß sich eine MS-DOS-Erweiterung sehr gut als Echtzeitbetriebssystem für den Single-Board-PC eignet. Besonders hervorzuheben sind dabei die kurzen Reaktionszeiten des Taskschedulers der eingesetzten MS-DOS-Erweiterung RTKernel [100] sowie die Kompatibilität des Echtzeitbetriebssystems zu bestehenden Teilsystemen des integrierten Gesamtsystems ISRS.

4.2 Entwurf des Sensorsteuerungssystems SC

Der Kern des integrierten Gesamtsystems ISRS stellt die Sensorsteuerung SC dar, die maßgeblich die Übertragung von Daten zur Robotersteuerung und die Kommunikationssteuerung ausführt. Entsprechend den in Kapitel 3 erarbeiteten Anforderungen wurde das zugrundeliegende Konzept und der strukturelle Aufbau der Sensorsteuerung SC entworfen. Der Aufbau der Sensorsteuerung SC wird im folgenden vorgestellt und deren Parametrierung näher erläutert.

4.2.1 Aufbau des Sensorsteuerungssystems SC

Die Sensorsteuerung SC hat hauptsächlich die Aufgabe mit der Robotersteuerung innerhalb der Ebene Ablaufplanung, Bahnplanung und Lageregelung zu kommunizieren. Dazu stehen an der Robotersteuerung zwei zyklische Eingriffstellen, die einen bidirektionalen Datentransfer im Interpolations- bzw. Lageregeltakt erlauben, und eine azyklische Eingriffstelle zur bidirektionalen Übertragung von Daten zur Verfügung [165]. Die zyklische Integration von Korrekturdaten ermöglicht eine Adaption der Bewegungsbahn, die aktuell von der Robotersteuerung verfahren wird. Andererseits kann der Single-Board-PC auf zyklische Daten der Robotersteuerung zugreifen. Der Datentransfer wird vom Single-Board-PC vorgenommen, wobei jedoch Daten, die die Lageregelschnittstelle betreffen, aus Performanzgründen direkt an das Transputernetzwerk weitergegeben werden sollen.

Die azyklische Eingriffstelle dient dazu, Daten auf der Ebene des Roboterprogramms zu transferieren. Sie kann so Sensordaten, die den weiteren Ablauf des Roboterprogramms beeinflussen, übertragen. Diese Eingriffstelle kann aber auch dazu verwendet werden, eine Programm- und Ablaufsteuerung des SC-Systems, die von Seite der Robotersteuerung bedient werden kann, aufzubauen. Eine jegliche Kommunikationsanforderung kann nur von der Robotersteuerung ausgehen. Zwischen den Kommunikationspartner RC und SC herrscht somit eine Master-Slave-Konfiguration. In der Sensorsteuerung SC sollten außerdem geeignete Mechanismen zur Erfassung und Verarbeitung von Sensordaten zur Verfügung stehen.

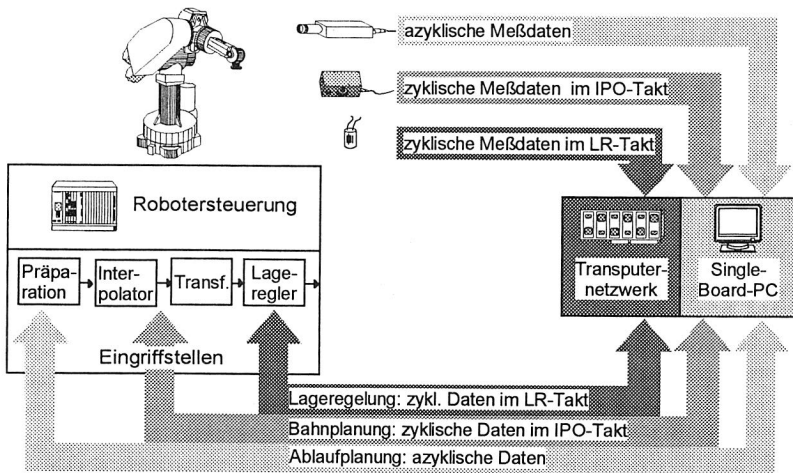


Bild 33: Kommunikationsebenen und Eingriffstellen von Roboter- und Sensorsteuerung

Auf der Basis der vielfältigen Möglichkeiten und Anforderungen soll nun ein geeigneter Entwurf der Sensorsteuerung SC vorgenommen werden. Der azyklischen Kommunikation kommt dabei eine Schlüsselfunktion zu.

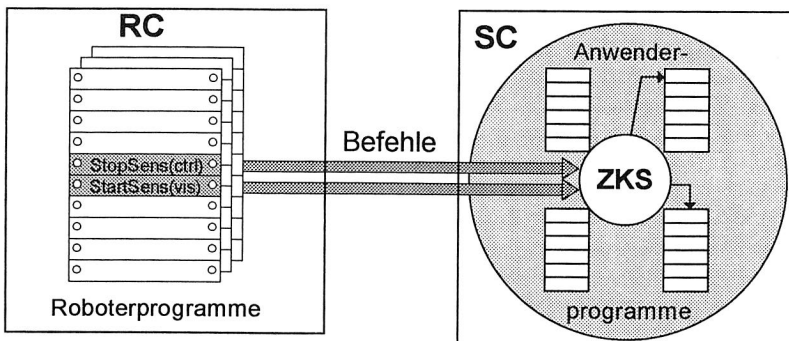


Bild 34: Steuerung der Anwenderprogramme durch die Robotersteuerung

Hauptziel der zu definierenden Sensorsteuerung SC muß es sein, Mechanismen zur Verfügung zu stellen, die es erlauben, die Sensorintegration flexibel von der Robotersteuerung aus zu steuern. Dazu wurde das Modul „Zentrale Kommunikation und Steuerung“ (ZKS) entworfen. Es ermöglicht, daß azyklische wie zyklische Anwenderprogramme der Sensorsteuerung SC von der Robotersteuerung aus angewählt, parametrisiert, gestartet und beendet werden können.

Die Modul ZKS bedient sich dabei der azyklischen Kommunikation zwischen der Roboter- und der Sensorsteuerung. Außerdem wird die zyklische Übertragung von Daten, wie sie für zyklische Anwenderprogramme notwendig ist, von ZKS unterstützt. Die Sensorsteuerung SC unterteilt sich so strukturell in das Modul ZKS, in zyklische und azyklische Anwenderprogramme und modulare Funktionen zur Sensorwerterfassung und -verarbeitung. Die Anwenderprogramme werden vom Benutzer unter Beachtung von Designvorschriften erstellt und in die Sensorsteuerung SC eingebunden. Damit steht die volle Funktionalität zur Steuerung der Anwenderprogramme von Seiten der RC zur Verfügung. Diese Aufgabe übernimmt vollständig, ohne nach außen transparent zu werden, das Modul ZKS.

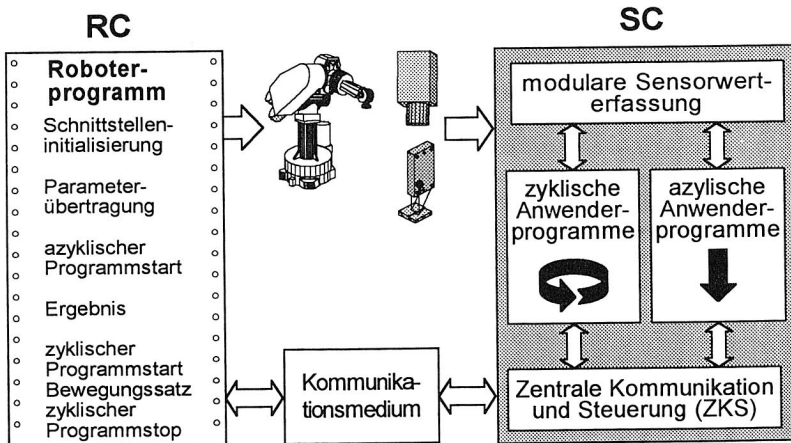


Bild 35: Aufbau der Sensorsteuerung SC und deren Steuerbarkeit durch die RC [45]

Grundmodul „Zentrale Kommunikation und Steuerung“ (ZKS):

Ziel des Entwurfs von ZKS war es, Systemdienste zur Programmsteuerung auf hohem Abstraktionsniveau zu konzipieren und diese Dienste auf die eigentlichen zyklischen und azyklischen Kommunikationsmechanismen der RC-SC-Kopplung abzubilden. Dabei wurde insbesondere darauf geachtet, daß ZKS auch einfach an andere Kommunikationsmechanismen und -medien adaptiert werden kann, um so die Sensorsteuerung SC auch an andere Kommunikationspartner (Robotersteuerungstypen) anbinden zu können. Zunächst wurde das Aufgabenprofil des Modul ZKS definiert (Tabelle 3).

Zur Programmsteuerung stehen Systemdienste zur Verfügung, die zyklische und azyklische Anwenderprogramme aktivieren/deaktivieren und starten/beenden können. Weitere Systemdienste übernehmen die Übertragung von Parametern an das SC-System oder an ein spezifisiertes Anwenderprogramm. Schließlich können Systemdienste die Kommunikation zur Robotersteuerung beenden oder eine Initialisierung des Single-Board-PCs bewirken. Diese Auf-

gaben wurden entsprechend ihrer asynchronen Art auf einen azyklischen Datenaustausch zwischen Robotersteuerung und Sensorsteuerung SC abgebildet. Zusätzlich muß ZKS geeignete Mechanismen zur Verfügung stellen, damit zyklische Anwenderprogramme auf zyklische Daten zugreifen bzw. zyklische Daten an die Robotersteuerung übertragen können. Aus Performanzgründen muß die Datenübertragung dabei sehr schnell erfolgen.

azyklische Aufgaben	zyklische Aufgaben
Programmsteuerung: <input type="checkbox"/> Starten von Anwenderprogrammen <input type="checkbox"/> Beenden von Anwenderprogrammen Parameterübergabe: <input type="checkbox"/> Parameter des Anwenderprogramms <input type="checkbox"/> Maschinendaten der RC Kontrolle des Single-Board-PCs: <input type="checkbox"/> Initialisieren des Single-Board-PCs <input type="checkbox"/> Beenden der Kommunikation	Datentransfer im Interpolationstakt: <input type="checkbox"/> aktuelle Werte der RC an die SC <input type="checkbox"/> Korrekturwerte an die RC Datentransfer im Lageregeltakt <input type="checkbox"/> aktuelle Werte der RC an die SC <input type="checkbox"/> Korrekturwerte an die RC Kontrolle der Kommunikationsbereitschaft <input type="checkbox"/> Austausch eines Lebenszeichens

Tabelle 3: Aufgaben des Grundmoduls Zentrale Kommunikation und Steuerung

Um diese Anforderungen zu erfüllen, wird für das Modul ZKS ein hierarchischer Aufbau in Form von einzelnen gekapselten Schichten gewählt, die auf die physikalischen Gegebenheiten und deren Erfordernisse abgebildet werden:

Die **physikalische Schicht** wird von allen an der Kommunikation beteiligten Hardwarekomponenten gebildet. Die Robotersteuerung und der Single-Board-PC sind dabei über einen gemeinsamen Speicher gekoppelt. Der Speicher ist unterteilt in Bereiche für azyklische und zyklische Daten sowie für Kontroll- und Statusinformationen. Ein gleichzeitiger Zugriff auf den selben Speicherbereich wird durch einen Sicherungsmechanismus (Semaphoren) verhindert. Die Robotersteuerung teilt dem Single-Board-PC eine Kommunikationsanforderung durch das Auslösen eines Interrupts mit. Je nach der Codierung des Interrupts kann dabei zwischen den unterschiedlichen zyklischen und azyklischen Kommunikationsanforderungen unterschieden werden.

Die **Transportschicht** übernimmt den Datentransfer von und zur physikalischen Schicht des Moduls ZKS. Sie bildet dazu Datenströme auf die physikalischen Gegebenheiten ab bzw. transferiert die physikalischen Gegebenheiten in entsprechende Datenströme. Die Transportschicht entkoppelt so die höheren Schichten des Modells von der aktuellen, physikalischen Schicht. Transportaufträge erhält die Transportschicht über eine Warteschlange von der über-

geordneten Synchronisationsschicht. Ist ein Auftrag abgearbeitet, so erfolgt eine Statusmeldung an die übergeordnete Schicht.

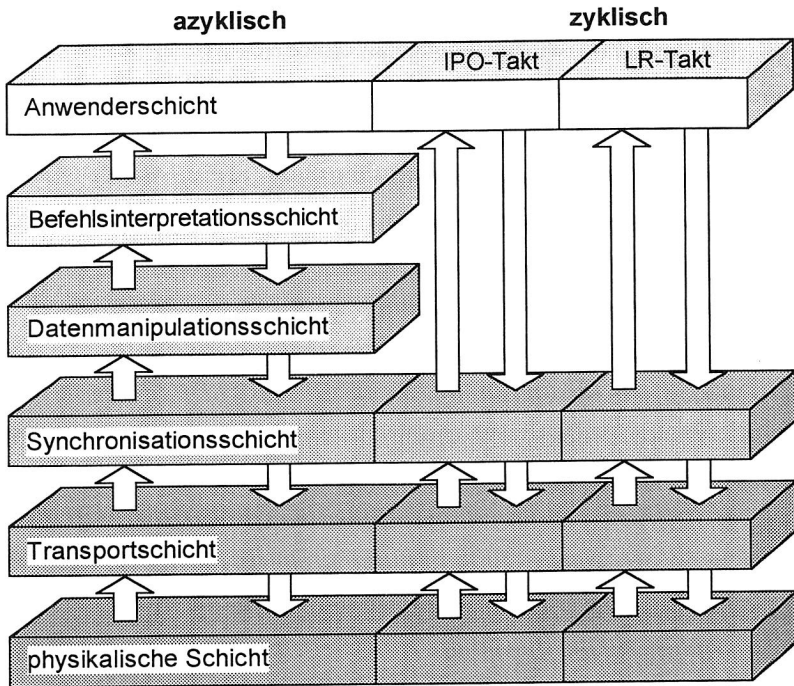


Bild 36: Schichtenmodell der Zentralen Kommunikation und Steuerung (ZKS)

Die **Synchronisationsschicht** stellt sicher, daß keine unsynchronisierten Zugriffe auf das Kommunikationsmedium erfolgen und die Kommunikationsaufträge der Anwenderprogramme entsprechend ihrer Ankunftsreihenfolge bearbeitet werden. Die Synchronisationsschicht nimmt an den ihr übergebenen Daten keinerlei Veränderungen vor, sondern verwaltet diese nur. Die eigentliche Synchronisation erfolgt über den von der Robotersteuerung ausgelösten Interrupt. Je nach dem Typus des Interrupts stößt die Synchronisationsschicht unterschiedliche Aktionen im Softwaresystem an. Bei „azyklischen“ Interrupts werden die überlagerten Schichten aktiviert. Im Falle von „zyklischen“ Interrupts werden die entsprechenden zyklischen Anwenderprogramme in Bereitschaft versetzt.

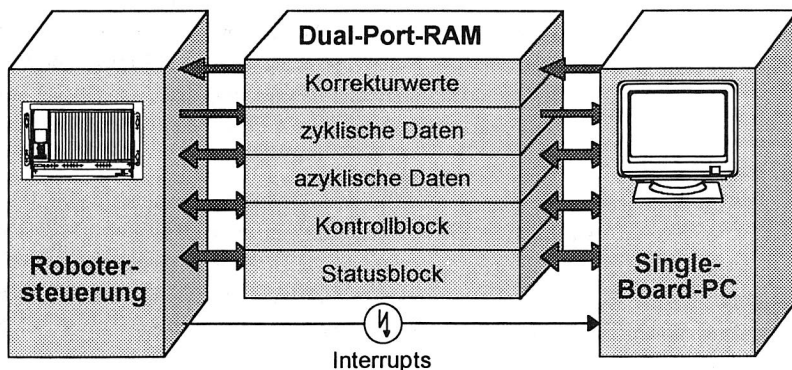


Bild 37: Struktur der physikalischen Schicht

Die Aufgabe der **Datenmanipulationsschicht** besteht in der Verwaltung der Lese- und Schreibwünsche auf das Kommunikationsmedium. Die Zugriffswünsche werden auf ihre Korrektheit geprüft und nach der erfolgreichen Kontrolle an die Synchronisationsschicht weitergegeben.

Befehl	zugehörige Systemtask
Maschinendaten übertragen	Masch_Dat()
Anwenderprogramm starten	Prog_Start()
Anwenderprogramm beenden	Prog_Abort()
Daten übergeben	More_Data()
Reglerparameter übergeben	Regler_Para()
SBC zurücksetzen	Reset()
Status abfragen	Status_Task
SBC beenden	SBC_End()

Tabelle 4: Verzeichnis der Befehle zur Steuerung der Sensorsteuerung SC

Aufgabe der **Befehlsinterpretationsschicht** ist die Interpretation der von der Robotersteuerung übertragenen Befehle und die Aktivierung der zugehörigen Systemdienste. Dazu werden von der Robotersteuerung die Befehlsnummer und die entsprechende Nummer des davon betroffenen Anwenderprogramms übertragen. Der Befehlsinterpretierer identifiziert den Befehl und aktiviert die zugehörige Systemtask. Es stehen z.Z. die in Tabelle 4 aufgelisteten Befehle zur Verfügung.

Die **Anwenderprogrammschicht** stellt die Schnittstelle zu den jeweiligen Anwenderprogrammen dar. Die Übertragung von Daten und die Steuerung der Anwenderprogramme wird von der Anwenderprogrammschicht durchgeführt. Die Initialisierung und Terminierung sowie die Ablaufsteuerung dieser Programme erfolgt an Hand der Aufträge, die von der Befehlsinterpreterschicht stammen.

Die wesentlichen Eigenschaften dieses hierarchischen Schichtmodells bestehen darin, daß die höheren Schicht von der Funktionalität der niederen Schichten voll abstrahieren. Zwischen den Schichten befinden sich klar definierte Schnittstellen. Durch diese Kapselung der Schichten kann eine höhere Flexibilität des Systems gegenüber Änderungen, wie sie sich bei Verwendung eines anderen Kommunikationsmediums ergeben, erreicht werden. In einem solchen Fall muß nur die betroffene Schicht modifiziert werden.

zyklische Anwenderprogramme

Innerhalb von zyklischen Anwenderprogrammen müssen komplexe Algorithmen, wie z.B. mehrere Koordinatentransformationen und Regelalgorithmen, abgearbeitet werden. Um eine effektive Bearbeitung dieser Funktionen zu ermöglichen, werden sie in drei Teile aufgeteilt. Eine azyklische Startfunktion wird vor der eigentlichen, zyklischen Übertragung ausgeführt. Sie initialisiert die Regler- und Filterroutinen, führt individuelle Einstellungen aus und installiert den zyklischen Teil. Der zyklische Teil beinhaltet den eigentlichen Regler, der im IPO-Takt der RC ausgeführt wird. Diese Reglerfunktion berechnet an Hand von Sensorwerten und den aktuellen Achsstellungen in Abhängigkeit der jeweiligen Reglerfunktionalität neue Korrekturwerte, die an die RC übertragen werden. Für die Terminierung der zyklischen Routinen ist eine weiteres, azyklisches Anwenderprogramm vorgesehen, das die Deinstallation der zyklischen Reglerfunktion und die Rücksetzung der individuellen Einstellungen der Startfunktion durchführt. Im System des Sensorrechners kann nur jeweils ein zyklisches Anwenderprogramm aktiv sein, jedoch können mehrere zyklische Anwenderprogramme nacheinander abgearbeitet werden.

Für die Belange der sehr schnellen Datenintegration im Lageregeltakt wird der Single-Board-PC um eine Transputerkarte erweitert. Alle zyklischen Funktionen der Reglerfunktionen und der Sensorwerterfassung werden auf dem Transputernetzwerk abgearbeitet. Die „Zentrale Kommunikation und Steuerung“ (ZKS) übernimmt dabei nur die Funktionen der Datenübertragung und der Synchronisation sowie der Steuerung der Anwenderprogramme auf dem Transputersystem. Die Kommunikation und die Strukturen des Transputersystems werden später in Kapitel 4.3 erläutert.

4.2.2 Parametrierung des Sensorsteuerungssystems

Bei der Integration von Sensoren an Roboteranlagen müssen unterschiedlichste Problemstellungen gelöst werden. Dies erfordert eine hohe Flexibilität in der Erstellung von Anwenderprogrammen. Um den Anwender trotzdem wirkungsvoll zu unterstützen, stehen zyklische und azyklische Anwenderprogrammtemplates zur Verfügung, die entsprechend den Erfordernissen parametrisiert und modifiziert werden. Der Anwender muß dazu seine Vorstellungen zur Erfassung der Sensorwerte und deren Verarbeitung sowie zur Sensorregelung und Sensorsteuerung des Roboters in azyklische und zyklische Sensorprogramme umsetzen.

Zusammen mit der Sensorerfassungsbibliothek und den ZKS-Systembibliotheken können die gewünschten, ausführbaren Programme für den Single-Board-PC geschaffen werden, die dann von der Robotersteuerung aus bedienbar sind (Bild 38). Für Testzwecke kann eine modifizierte ZKS-Systembibliothek eingebunden werden, dies es erlaubt, mit dem Robotersimulator RC-SIM anstatt mit der Robotersteuerung zu kommunizieren. Auf diese Weise können die Anwenderprogramme ohne die Gefahr einer Anlagenbeschädigung auf die gewünschte Funktion hin untersucht werden.

Auf der Robotersteuerung stehen dem Anwender Programmakros zur Verfügung, die eine einfache Steuerung und Bedienung der Anwenderprogramme auf dem Single-Board-PC ermöglichen. Dazu müssen die parametrisierten Makros in das entsprechende Roboterprogramm eingebunden werden. Nach einer Syntaxüberprüfung des Roboterprogramms durch das Offline-Programmiersystem kann das Roboterprogramm in der Robotersteuerung oder, im Falle eines Test, im Robotersimulator RC-SIM abgearbeitet werden. Damit läßt sich eine vollständige Steuerung der im Single-Board-PC hinterlegten azyklischen und zyklischen Anwenderprogramme durch die Robotersteuerung erzielen. Die in Kapitel 3 erarbeiteten Forderungen, die eine Bedienbarkeit des Sensorrechners durch die Robotersteuerung beinhalten, konnten somit erfüllt werden.

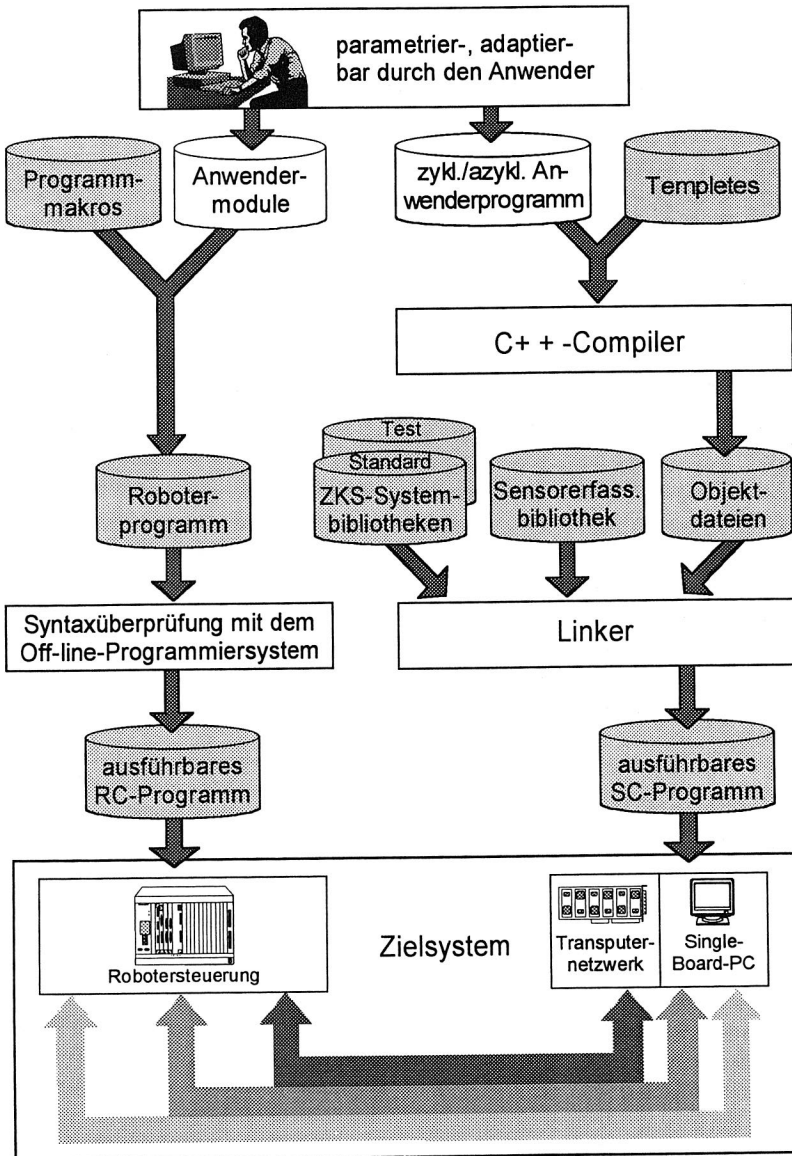


Bild 38: Datenfluß bei der Generierung und Parametrierung eines Sensorintegrationsprogramms

4.3 Transputersystem zur schnellen Sensorverarbeitung

Das Transputersystem dient als optionale Erweiterung des Single-Board-PCs, falls die umfangreichen Berechnungen sowie die Sensorwerterfassung, wie sie bei der Integration im Lage-regeltakt der Robotersteuerung anfallen, innerhalb der sehr kurzen Reaktionszeit nicht bewältigt werden können. Durch den Einsatz eines Rechnerclusters, das hier aus dem Single-Board-PC und einer variablen Anzahl von Transputern besteht, kann eine parallele Verarbeitung angestrebt werden [89]. Ein Transputercluster wurde für die Erfüllung der Aufgabe deshalb gewählt, weil sich durch Parallelanordnung der Transputer eine hohe, skalierbare Rechenleistung erzielen läßt. Die Fähigkeit der Kommunikation über serielle Schnittstellen, den sogenannten Links, ermöglicht eine schnelle Datenübertragung zwischen den einzelnen Prozessoren. Die Eignung von Transputern für die Steuerung von Robotern wurde bereits in anderen Forschungsarbeiten nachgewiesen [75, 83].

Zunächst wird im folgenden der Entwurf des Softwaresystems des Transputerclusters vorgenommen, das insbesondere die Bedienbarkeit der Anwenderprogramme auf dem Transputersystem ermöglichen soll. Durch eine parallele Strukturierung des Systems kann zusätzlich eine hohe Verarbeitungsgeschwindigkeit erzielt werden. Anschließend erfolgt eine Abbildung des Softwaresystems auf das Transputercluster mit dem Ziel, die Prozesse optimal auf die Prozessoren zu verteilen. Schließlich wird die Kopplung des Transputersystems mit dem Single-Board-PC entworfen und näher beleuchtet.

4.3.1 Entwurf des Softwaresystems

Innerhalb des Transputersoftwaresystems soll neben der schnellen Bearbeitung von zyklischen Funktionen auch die Steuerung und Bedienung von zyklischen Anwenderprogrammen ermöglicht werden. Eine Bearbeitung und Steuerung von azyklischen Anwenderprogrammen ist nicht erforderlich, da das Transputersystem nur als Subsystem des Single-Board-Computers konzipiert wurde, das rechenintensive zyklische Aufgaben löst. Die Steuerung und Bedienung der zyklischen Anwenderprogramme soll dabei vom Single-Board-PC respektive der Robotersteuerung aus durchgeführt werden. Damit sind ähnliche Funktionen zur Steuerung der Anwenderprogramme und zur Versorgung dieser mit Initialisierungsdaten erforderlich, wie sie für die Zentrale Kommunikation und Steuerung (ZKS) auf dem Single-Board-PC bereits entworfen wurden.

Die Steuerung der Anwenderprogramme erfordert zunächst einen Systemdienst, der ein angewähltes, zyklisches Anwenderprogramm startet. Der Dienst muß ferner sicherstellen, daß nicht bereits ein anderes Anwenderprogramm aktiv ist, und andernfalls eine Fehlermeldung an das Hostsystem senden. Ein weiterer Systemdienst soll ein definiertes Anhalten eines gerade aktiven Anwenderprogramms vornehmen können. Desweiteren ist es hilfreich, Initialisierungsdaten vom Hostsystem an ein zyklisches Anwenderprogramm übertragen zu können. Auf diese Weise lassen sich z.B. Regler- oder Filterparameter des Anwenderprogramms von der Robo-

tersteuerung aus modifizieren. Die Aktivität der Systemdienste läßt sich vorteilhaft durch einen azyklischen Datenverkehr zwischen dem Transputercluster und dem Single-Board-PC realisieren, da die Anforderungen hinsichtlich der Reaktionszeit gering sind. Außerdem ist die Anwenderprogrammsteuerung in dieser Weise eindeutig vom zyklischen Datenverkehr der Anwenderprogramme getrennt.

Ferner sollte ein schneller Datentransfer zwischen dem Single-Board-PC und dem Transputercluster bestehen, um zyklische Nutzdaten der Anwenderprogramme in sehr kurzer Zeit transferieren zu können. Diese Nutzdaten sind meist Größen wie aktuelle Gelenkwinkel des Roboters oder Korrekturwinkel, die zur schnellen Adaption einer Bewegungsbahn des Roboters benötigt werden. Diese Anforderung kann vorteilhaft durch einen zyklischen Datentransfer ausgeführt werden. Die Anwenderprogramme können sich in ihren Anforderungen hinsichtlich der notwendigen Rechenleistung stark unterscheiden. Folglich sollte für die Anwenderprogramme Rechenleistung, die innerhalb weiter Grenzen skalierbar ist, zur Verfügung stehen.

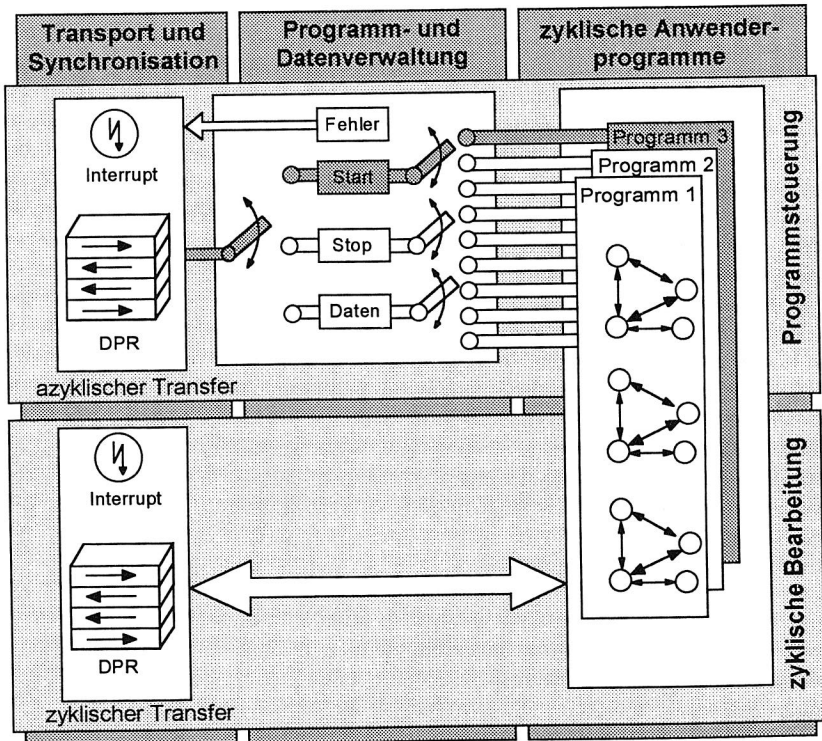


Bild 39: Datenfluß und Softwarestruktur des Transputersystems

Das Softwaresystem des Transputersystems besitzt ähnlich wie die „Zentrale Kommunikation und Steuerung“ (ZKS) eine hierarchische Schichtenstruktur, die es erlaubt, die Teilaufgaben effizient abzuarbeiten (Bild 41). Die **physikalische Schicht** besteht im wesentlichen aus einem gemeinsamen Speicher zwischen dem PC und dem Transputercluster, der getrennte Bereiche für die jeweilige Transferrichtung und für zyklische und azyklische Daten sowie für Kontroll- und Statusinformationen vorsieht. Es besteht die Möglichkeit, sowohl auf dem PC Interrupts wie auf dem Transputer Ereignisse (Events) auszulösen. Folglich beinhaltet das Kommunikationsmedium Grundfunktionalitäten, die einen schnellen Datentransfer unterstützen.

Auf die physikalische Schicht setzen die **Transport- und Synchronisationsdienste** auf. Sie synchronisieren den Zugriff auf das Kommunikationsmedium sowie den Ablauf der Anwenderprogramme. Ferner übernehmen sie den Transport von Nutzdaten vom und zum Kommunikationsmedium. Dabei werden verschiedenste Zugriffs-, Sicherungs- und Quittungsmechanismen umgesetzt. Schließlich übergibt der Transportdienst die Nutzdaten an die nachfolgenden Verarbeitungseinheiten, wobei hier zwischen azyklischen und zyklischen Datentransfer unterschieden wird. Zyklische Daten werden direkt über festgelegte Datenkanäle an die Anwenderprogramme transferiert. Azyklische Daten werden zur Auswertung an die Programm- und Datenverwaltung weitergeben.

Die **Programm- und Datenverwaltung** interpretiert die azyklischen Nutzdaten und löst die entsprechenden Systemdienste aus, die Aktionen wie das Starten oder Stoppen eines spezifizierten, zyklischen Anwenderprogramms veranlassen. Die Systemdienste wurden so konzipiert, daß sie die Steuerkommandos jeweils an alle Komponenten des Anwenderprogramms schicken, da das Anwenderprogramm meist auf mehreren Transputern verteilt ist. Andere Systemdienste ermöglichen die Übertragung von Initialisierungsdaten an das angewählte Anwenderprogramm. Außer der Parametrierbarkeit des Anwenderprogramm von Seiten der Robotersteuerung können so auch globale Daten vorab lokal an die Komponenten des Anwenderprogramms verteilt werden. Bei verteilten Rechnerarchitekturen kann so ein „Flaschenhals“, der durch eine ständige Übertragung von globalen Daten an die verschiedenen Rechnerknoten entsteht, verhindert werden [164].

In **zyklischen Anwenderprogrammen** werden Berechnungen wie Koordinatentransformationen, Regelungs- und Steuerungsalgorithmen sowie die Sensorwerterfassung und -verarbeitung durchgeführt. Der Anwender erstellt einzig das Anwenderprogramm. Nach der Einbindung in das Transputersoftwaresystem stehen die Dienste zur Steuerung und Bedienung des Anwenderprogramms vom Single-Board-PC bzw. von der Robotersteuerung aus unmittelbar zur Verfügung. Die zyklische Abarbeitung des parametrierten und aktivierten Anwenderprogramms und der resultierende, zyklische Datentransfer erfolgt einzig durch die zyklischen Mechanismen der Transport- und Synchronisationsschicht.

4.3.2 Abbildung des Softwaresystems auf ein Transputercluster

Das Transputersystem soll die zyklischen Anwenderprogramme leistungsfähig und in kurzen Zeitzyklen verarbeiten. Dies läßt sich nur durch ein optimales Zusammenwirken von Softwarekonzept und der eingesetzten Hardware erreichen. Folglich muß das bisher entworfene Softwarekonzept geeignet auf die Transputerhardware abgebildet werden. Ziel der Entwurfsarbeiten muß es sein, die einzelnen Module des Softwaresystems parallel zu strukturieren und auf die einzelnen Transputer zu verteilen, um minimale Bearbeitungszeiten zu garantieren. Der weitere Verlauf des Entwurfsvorgangs ist dadurch charakterisiert, daß eine bestmögliche Topologie eines Transputerclusters festzulegen ist. Erfahrungen zu geeigneten Prozessortopologien und deren bevorzugter Einsatz in verschiedenen Anwendungsbereichen liegen vor und sind in [109] zusammengefaßt. Außerdem wurden Werkzeuge entwickelt, die Algorithmen optimal auf eine parallele Prozessorarchitektur abbilden [94].

Diese Verfahren sind für die Abbildung des Softwarekonzepts auf ein Transputercluster wenig geeignet, da die Funktionalität und der Umfang der zyklischen Anwenderprogramme stark variieren kann, und so für jedes Anwenderprogramm ein eigener Optimierungslauf notwendig wird. Zunächst erscheint es deshalb sinnvoll, Aussagen zu erarbeiten, wie granular die Gesamtaufgabe zerlegt werden muß, um eine optimale Abbildung auf die Hardware zu erhalten. Der Begriff Granularität bezeichnet im weiteren die Feinheit der Zerlegung einer Gesamtaufgabe. [103] hat unter Verwendung eines Granularitätsschemas gezeigt, daß eine Aufgabenparallelisierung mit einer hohen Granularität wenig geeignet ist, die Rechenzeit wesentlich zu verringern. Die Ursache hierfür sind meist eine Zunahme der Kommunikationszeiten zwischen den Prozessoren, die die verringerte Rechenzeit bei weitem aufwiegt. Eigene, durchgeführte Untersuchungen haben dies wiederum bestätigt. Folglich empfiehlt es sich, eine Parallelisierung mit einer geringen Granularität auf der Ebenen von Funktionsbausteinen anzustreben. Bei der Abbildung des Softwaresystems auf die Hardware wurde entsprechend verfahren. Insbesondere können dabei die nachfolgenden Entwurfsregeln Anwendung finden [164]:

- ☐ Module, die gemeinsamen Speicher oder gemeinsame Datenkanäle nutzen, sind auf einen Transputer zu implementieren.
- ☐ Module, die nur geringe Verknüpfungen mit anderen Modulen besitzen, können auf mehrere Transputer verteilt werden. Dabei sollen nicht mehr als vier Datenkanäle verwendet werden.

Folglich werden die Aufgabe der Kommunikation (Transport und Synchronisation) einem Transputer zugeordnet, der direkt über ein Dual-Port-RAM mit dem Single-Board-PC in Verbindung steht. Ein weiterer Transputer führt die Verarbeitung der Sensorwerte durch. Er ist direkt mit den analogen Eingängen für die Sensoren verbunden. Eine weitgehend parallele und vom übrigen Ablauf unabhängige Kommunikation und Sensordatenvorverarbeitung ist somit möglich. Die Aufgaben der Programm- und Datenverwaltung werden von einem weiteren Transputer übernommen. Die Abarbeitung der Systemdienste wie das Starten/Stoppen von

Anwenderprogrammen und die Verwaltung von Parameterdaten sind seine charakteristischen Aufgaben.

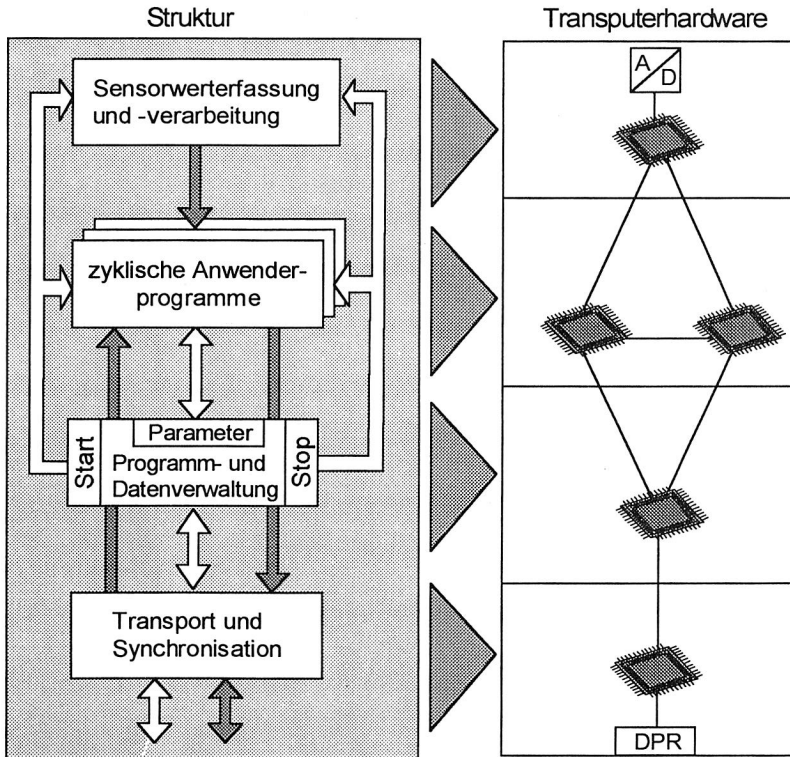


Bild 40: Abbildung der Softwarestruktur auf ein Transputercluster

Zur Bearbeitung der zyklischen Anwenderprogramme kann die Anzahl der verwendeten Transputer an Hand der geforderten Rechenleistung gewählt werden. Ausgewählte Funktionen können von vorn herein bestimmten Transputern zugeordnet werden. Bei den bisher durchgeführten Implementierungen erfüllen 2-3 Transputer die Abarbeitung der zyklischen Anwenderprogramme. Es lassen sich durchaus auch mehr als drei Transputer einsetzen. Der Anwender kann außerdem an Hand der Entwurfsregeln eine eigene, optimale Verteilung der Anwenderprogrammmodule auf den Transputern vornehmen. Er muß dabei aber beachten, daß die Schnittstellenkanäle entsprechend der Spezifikation bedient werden, da andernfalls die Anwenderprogrammsteuerung- und Datenverwaltung nicht mehr korrekt arbeitet.

4.3.3 Kopplung des Transputersystems mit der Sensorsteuerung SC

Für die Aufgaben der schnellen Sensorwertverarbeitung im Lageregeltakt der Robotersteuerung soll das Transputercluster mit der Robotersteuerung verbunden werden. Da das Transputercluster ein Subsystem des Single-Board-PCs darstellt, ist eine Anbindung des Transputernetzwerks mittels des Single-Board-PCs zweckmäßig. Die Aufgabe ist durch die nachfolgenden Anforderungen charakterisiert:

- ☐ Ein schneller Datentransfer zwischen der Robotersteuerung und dem Transputernetzwerk ist zwingend erforderlich. Die Verarbeitungszeit, die die Bearbeitungszeit des Anwenderprogramms sowie die Übertragungszeiten beinhaltet, darf Werte von typischerweise 2 Millisekunden nicht überschreiten.
- ☐ Die Anbindung des Transputerclusters soll weitgehend in das ZKS-System integrierbar sein, um eine einfache Programmverwaltung und Bedienung der Anwenderprogramme auf dem Transputer zu erzielen.
- ☐ Trotz der erforderlichen, geringen Übertragungszeiten müssen Mechanismen eine gesicherte und zuverlässige Datenübertragung gewährleisten.
- ☐ Beim Auftreten eines Fehlers muß ein Fehlersystem die Anlage in einen sicheren Zustand überführen und qualifizierte Fehlermeldungen versenden.

Strukturell gesehen, dient der Single-Board-PC zunächst als Transfer- und Umsetzstation bei der schnellen Sensorwertverarbeitung. Im wesentlichen müssen dazu unterschiedliche Daten über zwei Kommunikationsmedien bidirektional transferiert werden. Der Datentransfer muß die zwei nachfolgenden Grundfunktionalitäten erfüllen.

Die Programmsteuerung der Anwenderprogramme des Transputernetzwerks erfolgt zunächst über die „Zentrale Kommunikation und Steuerung“ der Sensorsteuerung SC. Hierfür wird ein zugehöriges, azyklisches Anwenderprogramm der Sensorsteuerung SC aktiviert, das die Transputerschnittstelle initialisiert und anschließend die zyklischen Anwenderprogramme des Transputerclusters abhängig von den entsprechenden Steuerkommandos der Robotersteuerung mit Parameterdaten versorgt und kontrolliert (Bild 41). Das azyklische Anwenderprogramm benutzt dazu die in Kapitel 4.3.1 vorgestellten Steuermechanismen auf dem Transputersystem.

Ferner sind für den Eingriff in die Lageregelung der Robotersteuerung Daten zyklisch zwischen dem Transputercluster und der Robotersteuerung zu transferieren. Diese Aufgabe muß von den Systemfunktionen und Mechanismen, die in der Transport- und Synchronisationsschicht der ZKS lokalisiert sind, übernommen werden. Sie setzen unmittelbar die jeweils empfangenen, zyklischen Daten auf die Kommunikationsmechanismen und Datenströme des anderen Kommunikationsmediums um. Damit kann eine sehr geringe Reaktions- und Übertragungszeit erzielt werden, da Umwege über höhere Hierarchiestufen vermieden werden. Durch eine entsprechende Konfiguration des Moduls ZKS kann die Bearbeitung der schnellen Sensorwertverarbeitung auch vollständig auf dem Single-Board-PC erfolgen.

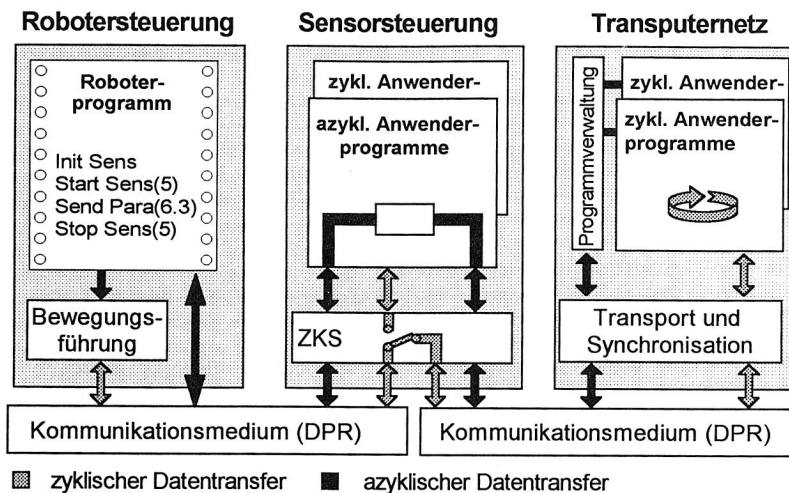


Bild 41: Datenfluß und Struktur der Kopplung zwischen dem Transputercluster und der Sensorsteuerung SC

Der gesamte Kommunikations- und Datenverkehr zwischen dem Transputersystem und dem Single-Board-PC weist die nachfolgenden Kennzeichen auf. Jede Übertragung von Daten wird durch eine Anforderungsmittelung an den Kommunikationspartner aktiviert und nach dem Erhalt der Daten quittiert. Zusätzlich wird der Datentransfer durch qualifizierte Quittungs- und Kennungsmechanismen, die die Gültigkeit von Zustandsdaten und die exakte, zeitliche Synchronisation signalisieren, zusätzlich gesichert. Fehler wie Synchronisationsfehler, die aus einer Zeitüberschreitung des zyklischen Anwenderprogramms auf dem Transputersystem resultieren, führen zu einer sicheren und deterministischen Systemreaktion. Infolgedessen wird die zyklische Kommunikation zur Robotersteuerung definiert abgebrochen, und in Folge wird eine Unterbrechung der Roboterbewegung ausgelöst. Zusätzlich wird eine qualifizierte Fehlermeldung unter Verwendung der azyklischen Datenübertragung an die Kommunikationspartner versendet. Schließlich konnten sehr geringe Reaktions- und Übertragungszeiten im Kommunikations- und Datenverkehr zwischen der Robotersteuerung und dem Transputercluster verwirklicht werden.

4.4 Programmentwicklung und Simulation des Programmablaufs

Zur Erprobung von Anwenderprogrammen, die sowohl Roboter- wie Sensorprogramme umfassen, wird exemplarisch ein Simulationssystem entworfen, das die Roboteranlage einschließlich ihres Kommunikationsverhaltens mit der Sensorsteuerung SC nachbildet. Zunächst wird

die Struktur des Simulationssystems und dessen Funktionseinheiten vorgestellt, sowie das Einsatzverhalten des Simulators näher beleuchtet. Ein Programmiersystem unterstützt ferner die Erstellung von Roboterprogrammen.

4.4.1 Strukturierung des Simulationssystems

Der Test von Programmen für das Sensor-Roboter-System kann erheblich erleichtert werden, wenn während der Programmentwicklung und -erprobung auf den Einsatz der Robotersteuerung verzichtet werden kann. Insbesondere läßt sich bei Fehlern in den Anwenderprogrammen die Gefahr einer Beschädigung der Roboteranlage und des Werkstücks gänzlich vermeiden. Diese Form der Erprobung von Anwenderprogrammen ist aber nur dann möglich, wenn die gesamte Roboteranlage hinreichend genau in einem Simulator abgebildet wurde und der Simulator die nachfolgenden Anforderungen erfüllt.

An ein Simulationssystem wird hauptsächlich die Forderung gestellt, daß es sich für die Erprobung des erwünschten Zusammenwirkens der Anwenderprogramme der Roboter- und der Sensorsteuerung eignet. Folglich muß das Simulationssystem zunächst das azyklische und zyklische Kommunikationsverhalten der Robotersteuerung exakt nachbilden und über das selbe oder ein vergleichbares Kommunikationsmedium mit dem SC-System verbunden sein. Der Simulator sollte weiterhin über die Grundfunktionen zur Abarbeitung von Roboterprogrammen, die die Steuerung des SC-Systems vornehmen, verfügen. Aus diesen Forderungen resultiert, daß der Simulator unter harten Echtzeitbedingungen an die Sensorsteuerung SC gekoppelt sein muß. Dies ist insbesondere erforderlich, um das zeitliche Verhalten von Anwenderprogrammen zu untersuchen und festzustellen, ob zyklische Anwenderprogramme die zulässigen Zykluszeiten einhalten. Ferner besteht die Notwendigkeit, daß der Roboter, seine Umwelt und die eingesetzten Sensorsysteme nachgebildet werden, um die virtuellen Sensorsignale, die die Sensorsteuerung SC als Eingangsgröße benötigt, zu gewinnen. Das dynamische Systemverhalten des Roboters und der Robotersteuerung soll außerdem im Simulator modelliert werden, um auch bei sensorgeregelten Anwendungen qualifizierte Simulationsergebnisse zu erhalten. Letztendlich müssen geeignete Protokolldaten zur Verfügung stehen, die im nachfolgenden Analyse- und Visualisierungssystem weiter ausgewertet werden können. Eine Analyse von verfügbaren Simulationssystemen ergab, daß keines der Systeme die obigen Forderungen erfüllen konnte. Daher war es notwendig, die Entwicklung des im folgenden vorgestellten Simulationssystems durchzuführen.

Die Basis des Entwurfs des Simulationssystems bildet die Kommunikationseinheit, die die Übertragung der Daten gemäß den Transfer- und Zugriffsmechanismen der Sensorsteuerung SC durchführt (Bild 42). Zusätzlich zur Kommunikationseinheit sind Module angeordnet, die die eigentlichen Funktionen der Robotersteuerung simulieren (Bild 43).

Die Interpretereinheit hat die Aufgabe, das Roboterprogramm zu analysieren und abhängig von der Programmsemantik Ereignisse zu aktivieren. Dies können z.B. Kommunikationsaufträge,

die eine Steuerung der Programme des SC-Systems bewirken, oder auch Bewegungsbefehle sein. Hierzu transferiert der Interpreter den entsprechenden Bewegungssatz an die Bahnplanung. Aus den Parametern des Bewegungssatzes berechnet die Bahnplanung das zugehörige Geschwindigkeitsprofil und die resultierenden Bahnstützstellen.

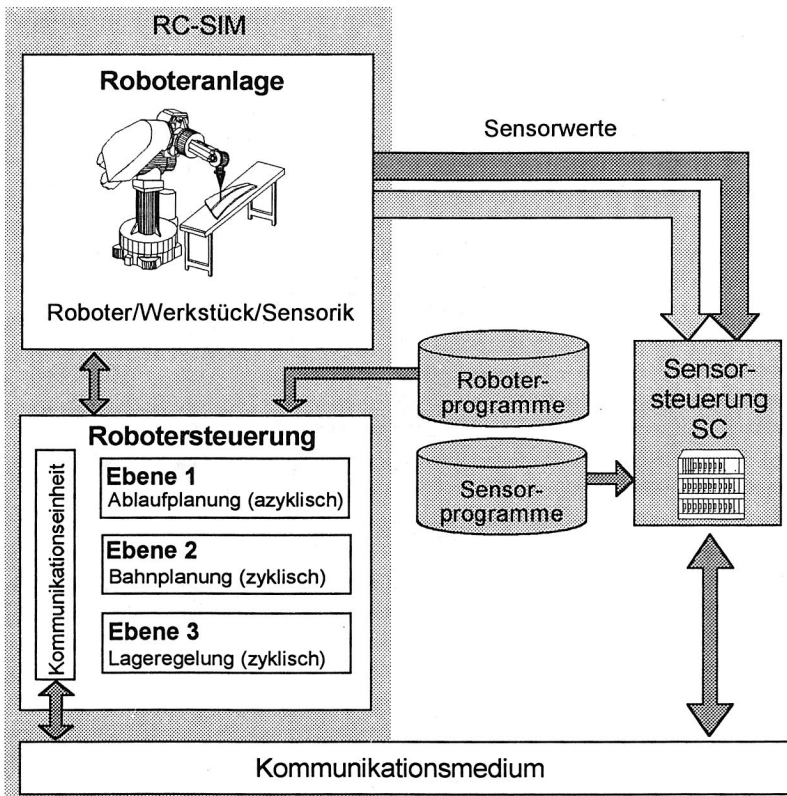


Bild 42: Aufgaben und Anbindung eines Robotersimulationssystems

Ferner besteht zwischen der Bahnplanung und der Kommunikationseinheit eine Verbindung, um abhängig vom Anwenderprogramm Korrekturwerte, die in die Bewegungsplanung eingreifen sollen, zu berücksichtigen. Eine sehr wesentliche Funktionseinheit des Simulationssystems stellt das Modul „Lageregelung und Roboterdynamik“ dar. Dieses Modul bildet die Rückwärtstransformation, die Dynamik der Lageregelung und der Robotermechanik sowie die Roboterkinematik nach. Als Ergebnis liefert das Modul die Position und Orientierung der virtuellen Roboterhand. Eine Verbindung zur Kommunikationseinheit ermöglicht die Übertragung

von Korrekturwerten, die von der Sensorsteuerung SC stammen, an das Modul „Lageregelung“. Aufgabe der Sensorwertbestimmung ist es nun, aus der simulierten Position der Roboterhand und einer geeigneten Beschreibung der Umwelt oder des Werkstücks den virtuellen Sensorwert zu berechnen. Der Sensorwerte muß dann über einen vereinbarten Datenkanal an die Sensorsteuerung SC übertragen werden. Für Analyse- und Visualisierungsaufgaben besteht die Möglichkeit, vorab die Zustands- und Ablaufgrößen festzulegen, die während eines Simulationslaufs protokolliert werden sollen.

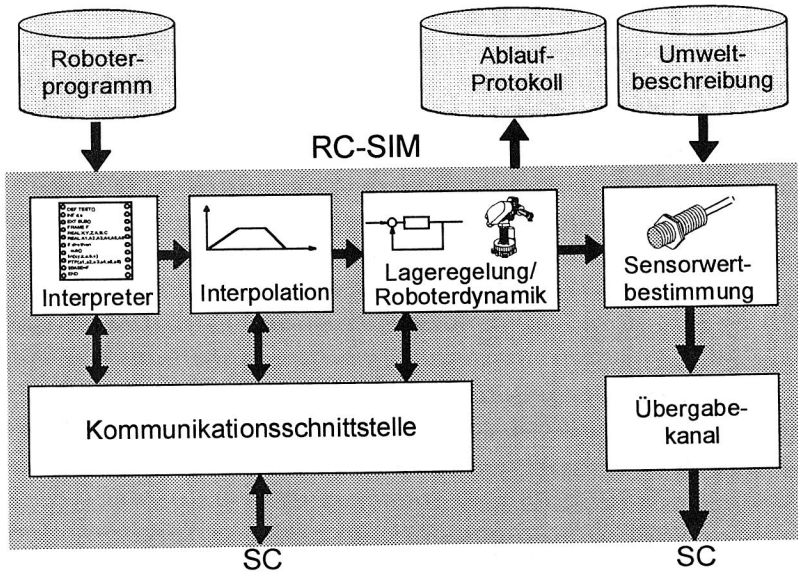


Bild 43: Struktur und Datenflüsse des Robotersimulationssystems

4.4.2 Merkmale der Funktionseinheiten

Das Simulationssystem ist in Funktionseinheiten gegliedert, die jeweils Teilkomponenten der Robotersteuerung nachbilden.

Interpretation:

Der Interpreter hat die Aufgabe, das Roboterprogramm zu interpretieren, davon abhängig entsprechende Aktion zu veranlassen und nachgeordnete Funktionseinheiten mit Daten zu versorgen. Im konzipierten Interpreter wurde exemplarisch eine Teilmenge des Sprachumfangs der standardisierten Programmiersprache IRL implementiert, so daß sensorgestützte Bewegungssätze und Kommunikationsanweisungen abgearbeitet werden können. Die nachfolgenden Funktionen zur Interpretation eines Programms waren hierzu notwendig:

- ☐ Interpretation von Konstanten und Variablen der Grundtypen von IRL
- ☐ Zuweisungen an Variablen sowie an Struktur- und Feldkomponenten
- ☐ Interpretation von Anweisungen, Funktionen und Bewegungssätzen
- ☐ Auswertung von Schlüsselausdrücken.

Der Interpreter wertet jeweils eine Zeile des Roboterprogramms aus. Dazu werden sukzessive die Wörter der Zeile mit den Wörtern in den Verzeichnissen verglichen, die sämtliche Schlüsselwörtern von IRL als auch die sich dynamisch ändernden Variablen und Konstanten des Programms enthalten, und die gespeicherten Aktionen ausgeführt (Bild 44).

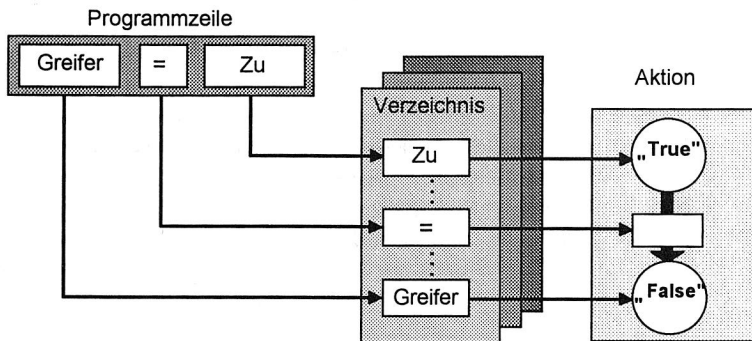


Bild 44: Abarbeitung eines Roboterprogramms durch den Interpreter

Der Interpretationsablauf wird angehalten, so bald der Interpreter auf ein Bewegungsbefehl oder eine Kommunikationsaufforderung stößt.

Interpolation:

Bewegungssätze werden vom Interpreter an den Interpolator übergeben, der die Stützpunkte der zu fahrenden Bewegungsbahn berechnet. Die Berechnung der Stützpunkte soll hier am Beispiel einer linearen Bewegung zwischen einem Start- und einem Zielpunkt erläutert werden (Bild 45).

Zunächst wird aus den einzelnen Komponenten des Start- und des Endpunktframes die notwendigen Translations-, Schwenkwinkel- und Drehwinkeländerungen berechnet, die erforderlich sind, um den Roboter an den Zielpunkt zu bewegen. An Hand der spezifizierten, maximalen Geschwindigkeiten und Beschleunigungen werden trapezförmige Geschwindigkeitsprofile der drei Teilbewegungen derart bestimmt, daß die Teilbewegungen synchron enden. Schließlich lassen sich die Bahnstützpunkte unter Verwendung der Geschwindigkeitsprofile wie folgt bestimmen:

$$\underline{x}(k+1) = \underline{x}(k) + \underline{v}(k)T_{\text{IPO}} \quad \text{Gl. 1}$$

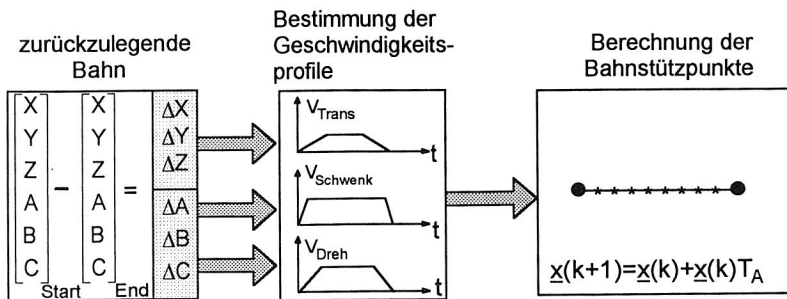


Bild 45: Bestimmung der Bahnstützpunkte durch ein Interpolationsverfahren

Kommunikation:

Das Simulationssystem bildet den gesamten Funktionsumfang der azyklischen und zyklischen Kommunikation der Robotersteuerung nach. Um eine möglichst hohe Flexibilität zu ermöglichen, wird die physikalische Schicht (Dual-Port-RAM), auf den normalen Arbeitsspeicher des Prozessors abgebildet, so daß die Anwenderprogramme auf jedem Standard-PC erprobt werden können. Außerdem muß die Eigenschaft der Robotersteuerung, Interrupts erzeugen zu können, im Simulationssystem nachgebildet werden. Dazu löst das Simulationssystem Ereignisse aus, die das Echtzeitsystem der Sensorsteuerung SC in der gleichen Weise wie die Interrupts der Robotersteuerung bearbeitet. Funktional ergeben sich dadurch keinerlei Unterschiede.

Lageregelung:

Zunächst müssen die berechneten Bahnstützpunkte des Interpolators in die entsprechenden Gelenkwinkel des Roboters transformiert werden. Die erforderliche Rückwärtstransformation stand zur Verfügung. Zur Abbildung des dynamischen Verhaltens der Lageregelkreise des Roboters und der Robotermechanik wird im Simulationssystem je Roboterachse die Möglichkeit zur Integration einer zeitdiskreten Systembeschreibung vorgesehen. Die Systembeschreibung erfolgt dabei in der Form einer diskreten Zustandsdarstellung für lineare, zeitinvariante Eingrößensysteme:

$$\begin{aligned} \underline{x}(k+1) &= \underline{A} \underline{x}(k) + \underline{b} u(k) \\ \underline{y}(k) &= \underline{c}^T \underline{x}(k) + d u(k) \end{aligned} \quad \text{Gl. 2}$$

Die Zustandsmatrix \underline{A} , der Steuervektor \underline{b} , der Ausgangsvektor \underline{c}^T und der Durchgangsskalar d können beim Starten des Simulationssystems entsprechend parametrisiert werden. Anschließend werden die Ausgangsgrößen, die Gelenkwinkel des Roboters, in die resultierende, kartesische Position der Roboterhand transformiert, die zur Berechnung des Sensorwert benötigt wird.

Sensorwertbestimmung:

Nachdem die Position und die Orientierung der Roboterhand simulativ ermittelt wurde, muß nun der virtuelle Sensorwert berechnet werden, den normalerweise der Sensor durch eine Messung zur Werkstückoberfläche liefert.

Zunächst soll die Werkstückoberfläche geeignet nachgebildet werden. Die Beschreibung einer Werkstückoberfläche in Form einer Punktmenge ist dabei am umfassendsten, da diese Beschreibungsform sowohl manuell einfach erstellt, als auch von Scannereinrichtungen und CAD-Systemen generiert werden kann. Zur Bestimmung von Werten zwischen Stützpunkten, sind in der Literatur mehrere Verfahren bekannt [64]. Die Interpolation von Oberflächenprofilen mittels einer einzigen Funktion ist hier ungeeignet, da sie aufgrund des hohen Polynomgrades stark zum Oszillieren neigt. Als Alternative bietet sich an, die gegebene Punktmenge in kleinere Teilintervalle zu zerlegen. Die so entstandenen Flächensegmente können durch Funktionen mit niedrigerem Polynomgrad interpoliert werden [64].

Wird als Funktion zwischen vier in einem Rechteck angeordneten Stützwerten ein Polynom dritten Grades gewählt, so kann eine Oberfläche nachgebildet werden, die über alle Flächensegmente sowohl stetig als auch stetig differenzierbar ist. Die Gesamtheit aller ermittelter Polynome wird als bikubische Splinefunktion bezeichnet [64].

Gesucht wird eine die Stützpunkte (x_i, y_j) interpolierende Splinefläche $f(x, y)$ mit den Teilfunktionen [64]

$$f_{ij}(x, y) = \sum_{k=0}^3 \sum_{l=0}^3 a_{ijkl} (x - x_i)^k (y - y_j)^l \quad \text{Gl. 3}$$

$$0 < i < n, 0 < j < m \quad i, j, n, m \in \mathbb{Z}.$$

a_{ijkl} bilden die gesuchten Koeffizienten für das Polynom des Flächensegments (i, j) . Weiterhin gelten oben genannte Stetigkeitsforderungen:

$$\frac{\partial}{\partial x} f_{ij}(x, y), \frac{\partial}{\partial y} f_{ij}(x, y), \frac{\partial^2}{\partial x \partial y} f_{ij}(x, y) \quad \text{sind stetig.} \quad \text{Gl. 4}$$

In Matrizenschreibweise lautet Gl. 3:

$$f_{ij}(x, y) = \begin{bmatrix} 1 & (x - x_i) & (x - x_i)^2 & (x - x_i)^3 \end{bmatrix} \begin{bmatrix} a_{ij00} & a_{ij01} & a_{ij02} & a_{ij03} \\ a_{ij10} & a_{ij11} & a_{ij12} & a_{ij13} \\ a_{ij20} & a_{ij21} & a_{ij22} & a_{ij23} \\ a_{ij30} & a_{ij31} & a_{ij32} & a_{ij33} \end{bmatrix} \begin{bmatrix} 1 \\ (y - y_j) \\ (y - y_j)^2 \\ (y - y_j)^3 \end{bmatrix} \quad \text{Gl. 5}$$

Bestimmt man

$$f_{ij}(x, y), \frac{\partial}{\partial x} f_{ij}(x, y), \frac{\partial}{\partial y} f_{ij}(x, y), \frac{\partial^2}{\partial x \partial y} f_{ij}(x, y) \quad \text{im Punkt } x = x_i, y = y_j,$$

so sind vier Polynomkoeffizienten sofort zu bestimmen:

$$z_{ij} := f(x_i, y_j) = a_{ij00} \quad \text{Gl. 6}$$

$$p_{ij} := \frac{\partial}{\partial x} f(x_i, y_j) = a_{ij10} \quad \text{Gl. 7}$$

$$q_{ij} := \frac{\partial}{\partial y} f(x_i, y_j) = a_{ij01} \quad \text{Gl. 8}$$

$$r_{ij} := \frac{\partial^2}{\partial x \partial y} f(x_i, y_j) = a_{ij11} \quad \text{Gl. 9}$$

Weiterhin folgt mit $\Delta x_i := x_{i+1} - x_i$, $y = y_j$ bzw. $\Delta y_j := y_{j+1} - y_j$, $x = x_i$

$$z_{i+1,j} = f(x_{i+1}, y_j) = \begin{bmatrix} 1 & \Delta x_i & \Delta x_i^2 & \Delta x_i^3 \end{bmatrix} \underline{A}_{ij} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \text{ bzw.} \quad \text{Gl. 10}$$

$$z_{i,j+1} = f(x_i, y_{j+1}) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \underline{A}_{ij} \begin{bmatrix} 1 \\ \Delta y_j \\ \Delta y_j^2 \\ \Delta y_j^3 \end{bmatrix}, \quad \underline{A}_{ij} = \begin{bmatrix} a_{ij00} & a_{ij01} & a_{ij02} & a_{ij03} \\ a_{ij10} & a_{ij11} & a_{ij12} & a_{ij13} \\ a_{ij20} & a_{ij21} & a_{ij22} & a_{ij23} \\ a_{ij30} & a_{ij31} & a_{ij32} & a_{ij33} \end{bmatrix}.$$

Setzt man diese Entwicklung für z, p, q und r für alle Indizes bis $i+1$ bzw. $j+1$ fort, so ergibt sich die Matrixdarstellung

$$\begin{bmatrix} z_{ij} & q_{ij} & z_{i,j+1} & q_{i,j+1} \\ p_{ij} & r_{ij} & p_{i,j+1} & r_{i,j+1} \\ z_{i+1,j} & q_{i+1,j} & z_{i+1,j+1} & q_{i+1,j+1} \\ p_{i+1,j} & r_{i+1,j} & p_{i+1,j+1} & r_{i+1,j+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & \Delta x_i & \Delta x_i^2 & \Delta x_i^3 \\ 0 & 1 & 2\Delta x_i & 3\Delta x_i^2 \end{bmatrix} \underline{A}_{ij} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & \Delta y_j & 1 \\ 0 & 0 & \Delta y_j & 2\Delta y_j \\ 0 & 0 & \Delta y_j^3 & 3\Delta y_j^2 \end{bmatrix}$$

oder kurz

$$\underline{W}_{i,j} = \underline{G}(x_i) \underline{A}_{i,j} \underline{G}^T(y_j). \quad \text{Gl. 11}$$

Die Splinekoeffizienten können in Abhängigkeit der gegebenen Werte z_{ij} und der noch zu bestimmenden partiellen Ableitungen p_{ij} , q_{ij} und r_{ij} durch die Formel

$$\underline{A}_{ij} = \underline{G}^{-1}(x_i) \underline{W}_{ij} \left(\underline{G}^T(y_j) \right)^{-1} \quad \text{Gl. 12}$$

berechnet werden.

p_{ij} , q_{ij} , und r_{ij} sind aufgrund der verbleibenden Freiheitsgrade für bikubische Flächeninterpolationen nicht eindeutig bestimmbar. Eine einfache Lösungsmöglichkeit besteht darin, bei kon-

stantem x bzw. y die partiellen Ableitungen gleich dem Mittelwert der Steigungen zu den beiden benachbarten Knoten zu setzen.

$$p_{ij} = \frac{\frac{(z_{ij} - z_{i-1,j})}{x_i - x_{i-1}} + \frac{(z_{i+1,j} - z_{ij})}{x_{i+1} - x_i}}{2}, \quad q_{ij} = \frac{\frac{(z_{ij} - z_{i,j-1})}{y_j - y_{j-1}} + \frac{(z_{i,j+1} - z_{ij})}{y_{j+1} - y_j}}{2} \quad \text{Gl. 13}$$

$$r_{ij} = \frac{p_{ij} + q_{ij}}{2}$$

Die Berechnung der Oberfläche kann off line vor dem eigentlichen Simulationslauf erfolgen. Aus der Oberflächenbeschreibung und der Position der Roboterhand muß nun der Sensor-meßwert bestimmt werden. Dies wird hier am Beispiel eines abstandmessenden Sensorsystems aufgezeigt. Um den Abstand zu berechnen, muß der Schnittpunkt des Sensorstrahls mit der Oberfläche bestimmt werden (Bild 46), d.h.

$$f_{\text{Sensorstrahl}}(x, y) = f_{\text{Oberfläche}}(x, y). \quad \text{Gl. 14}$$

Da es sich bei der Splinefunktion um eine segmentierte Funktion handelt, müßte die Schnittpunktberechnung für $(n+1) \times (n+1)$ Segmente der Oberfläche durchgeführt werden. Da die Projektion des Sensorstrahls auf die (x, y) -Parameterebene nur bestimmte Flächensegmente durchläuft, verringert sich die Anzahl der zu überprüfenden Splinepolynome erheblich (Bild 46).

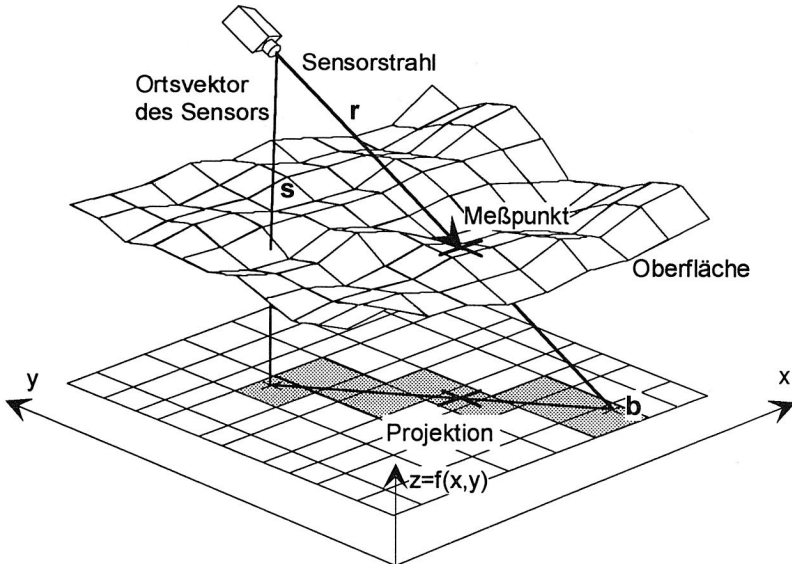


Bild 46: Berechnung des Sensormesspunkts auf einer Splineoberfläche

Zur Berechnung des Schnittpunktes des Sensorstrahls mit der Splineoberfläche muß letztendlich die Nullstelle von Gl. 14 bestimmt werden. Dazu wird das numerische Newtonsche Verfahren eingesetzt [14]. Dabei ist aber insbesondere zu beachten, daß das Verfahren durch eine geeignete Wahl der Startwerte bei Polynomen 3. Grades konvergiert.

Es werden sukzessive alle betroffenen Flächensegmente auf einen möglichen Schnittpunkte mit dem Sensorstrahl überprüft. Als Ergebnis erhält man den Meßpunkt des Sensor und kann daraus einfach den Sensormeßwert, hier den Abstand, bestimmen. Der Sensorwert muß anschließend über den spezifizierten Datenkanal an die Sensorsteuerung SC übertragen werden.

4.4.3 Einsatzverhalten des Simulationssystems

Das Simulationssystem ist v.a. durch seine Parametrierbarkeit gekennzeichnet. Die Werkstückoberfläche kann in Form einer Menge von Oberflächenstützpunkten flexibel konfiguriert werden. Analog läßt sich das Systemverhalten des Roboters in Form der charakteristischen Systemmatrizen an das Simulationsprogramm variabel übergeben. Weiter besteht die Möglichkeit, daß zu überprüfende Roboterprogramm frei vorzugeben und die Protokollfunktion flexibel zu parametrieren.

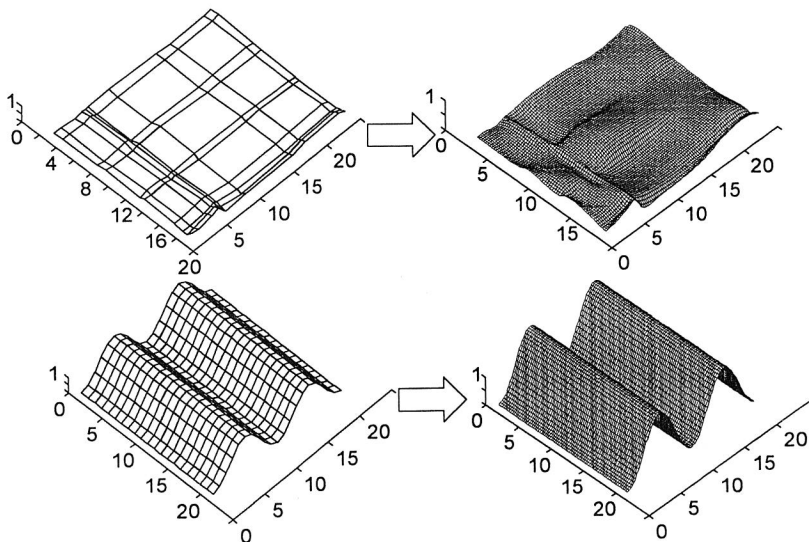


Bild 47: Interpolation von mit wenigen Stützpunkten beschriebenen Werkstückoberflächen durch bikubische Splinefunktionen des Simulationssystems

Das Simulationsprogramm weist akzeptable Simulationslaufzeiten auf. So können sämtliche Aktionen der Robotersteuerung und der Sensorsteuerung SC, die real einen Interpolationszy-

klus (32 ms) benötigen, innerhalb von 128 ms simuliert werden (CPU: 80486/33MHz). Ferner stellen die Echtzeitfunktionen des Simulationssystems sicher, daß die zyklischen Anwenderprogramme des SC-Systems auf die Einhaltung von spezifizierten maximalen Laufzeiten hin überwacht werden.

Die während eines Simulationslaufs protokollierten Daten können mit dem Analyse- und Visualisierungssystem, das die Funktionalität und die Systemumgebung des Mathematiksystems Matlab [98] verwendet, weiter ausgewertet werden. In Bild 47 kann man exemplarisch den visualisierten Ablauf einer Simulation erkennen, der mit der Vorgabe einer Werkstückoberfläche in Form einer „grobrastigen Punktwolke“ beginnt, die anschließend in eine Splinefläche umgesetzt wird.

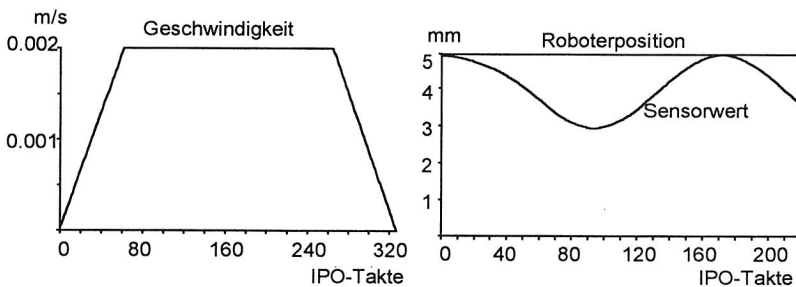


Bild 48: Simulierte Bahnkenngrößen und Sensorwerte beim „Überfahren“ der Splineoberfläche

Schließlich lassen sich die protokollierten Zustände der Robotersteuerung visualisieren, wobei in Bild 48 insbesondere das Bahngeschwindigkeitsprofil, die Position des Roboters und der vom Sensor gemessene Abstand beim „Überfahren“ eines sinusförmigen Werkstücks zu erkennen sind.

4.4.4 Erstellung von Roboterprogrammen

In das Gesamtsystem ISRS wurde das kommerzielle, textuelle Off-line-Programmiersystem APS [5], das für die eingesetzte Robotersteuerung ACR 20 entwickelt wurde, integriert. Es basiert auf der window-orientierten, grafischen Bedienoberfläche von MS-Windows und unterstützt die Erstellung, den Test und die Verwaltung von Anwenderprogrammen für die Robotersteuerung [6]. Mit Hilfe des Programmiersystems können off line Roboterprogramme für sensorgestützte Anwendungen erstellt und mit dem Compiler auf Syntaxfehler hin analysiert werden.

Bild 49 gibt ein typisches Beispiel eines Roboterprogramms für sensorgestützte Applikationen wieder. Eine dreigliedrige Struktur mit dem Starten, dem Parametrieren, dem Beenden des

Anwenderprogramms im SC-System sowie dem zwischengelagerten Bewegungsbefehl sind zu erkennen. Das syntaktisch richtige Roboterprogramm kann dann mit Hilfe einer seriellen Verbindung zur Robotersteuerung übertragen werden. Andererseits ist es auch möglich, das Roboterprogramm am Robotersimulator RC-SIM auszutesten.

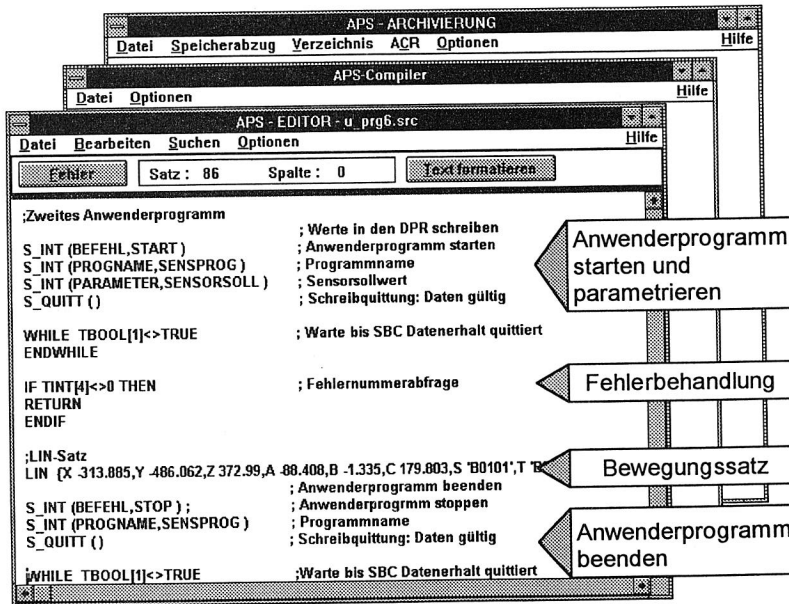


Bild 49: Roboterprogrammierung von sensorgestützten Anwendungen

4.5 Entwurf und Simulation von Sensorregelkreisen

Zum Entwurf und zur Simulation von Sensorsteuerungen und Sensorregelungen von Industrierobotern sowie zur Analyse und Visualisierung von Testläufen an der Roboteranlage beinhaltet das integrierte Gesamtsystem ISRS die Komponente „Dynamik-Simulator“. Die protokollierten Zustandsdaten der Sensorsteuerung SC können über eine Datenschnittstelle an den Dynamiksimulator übertragen werden.

Im wesentlichen basiert der Dynamik-Simulator auf dem Mathematikpaket Matlab und dem Visualisierungssystem Simulink [98], das um angepasste Werkzeuge zur Analyse und zur Simulation von Sensorregelkreisen erweitert wurde. Insbesondere stehen dafür folgende unterstützende Bibliotheken mit verschiedenen Verfahren zur Verfügung:

- ☐ Verfahren zur Identifikation von Strecken
- ☐ Sammlung identifizierter Strecken

- ☐ Bibliothek von geeigneten Regelalgorithmen
- ☐ Sammlung der Protokolldaten von Testläufen

Zusätzlich können Funktionen zur Bearbeitung und Verwaltung der Bibliotheken eingesetzt werden.

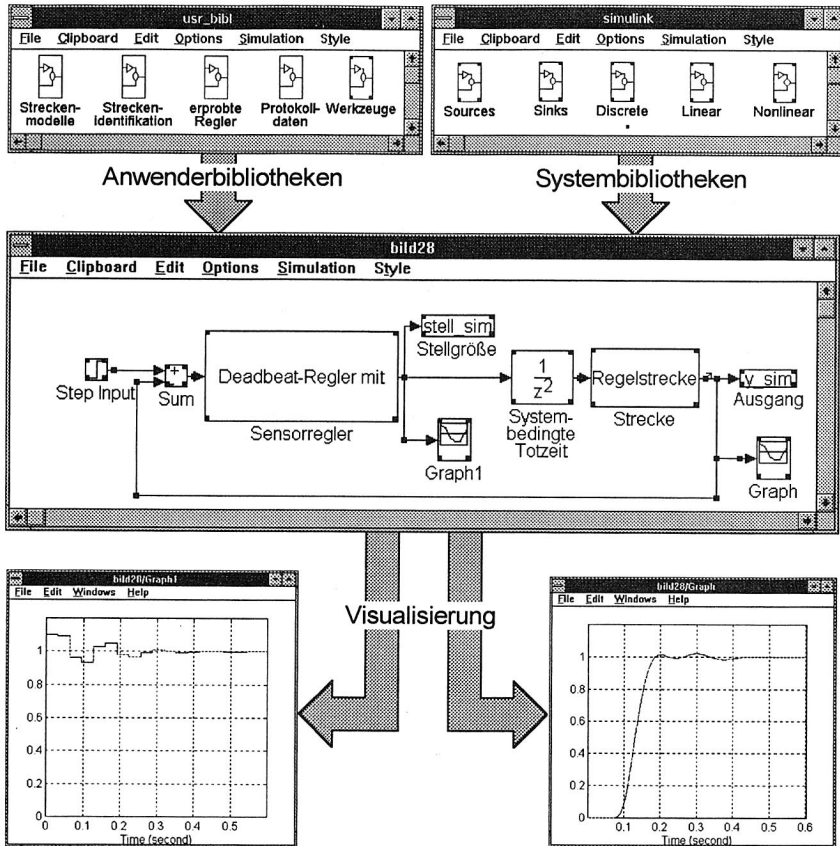


Bild 50: Simulation eines Sensorregelkreises unter Verwendung von Anwenderbibliotheken und integrierter Visualisierung

Zunächst wird mit Hilfe der Anwenderbibliotheken und der vorhandenen Systembibliotheken grafisch interaktiv das Blockschaltbild erstellt (Bild 50). Nach der Parametrierung der Blöcke und des Simulationslaufs kann die Simulation gestartet werden. Zeitgleich wird eine Visualisierung der spezifizierten Größen vorgenommen.

5. Adaption der Bewegungsbahn durch Sensorregelung im Interpolationstakt

Das in Kapitel 4 entworfene, integrierte Gesamtsystem wird nun für Untersuchungen, die eine Adaption der Bewegungsbahn eines Roboters durch Sensorregelung zum Ziel haben, eingesetzt. Damit läßt sich gleichzeitig die Flexibilität und Leistungsfähigkeit des entwickelten Konzepts exemplarisch an anspruchsvollen Aufgaben nachweisen.

Wie in Kapitel 2.2 und 2.3 aufgezeigt wurde, bestehen bisher hauptsächlich Defizite bei der On-line-Bahnadaption von Industrierobotern durch Sensorregelkreise. Insbesondere müssen effiziente Regelungsverfahren für die kartesische Sensorregelung entworfen und untersucht werden. Folglich werden zunächst Modelle der Regelstrecke identifiziert und parametrisiert. Der Entwurf effizienter, diskreter Regelungsverfahren und simulative Untersuchungen schließen sich an, um das erzielbare, dynamische Gesamtverhalten der Anlage quantifizieren und optimieren zu können. Die Regelungsalgorithmen sollen an der Modellanlage implementiert, validiert und optimiert werden.

5.1 Modellierung des kartesischen Sensorregelkreises

Um einen kartesischen Sensorregelkreis aufzubauen, gilt es zunächst, die Struktur des Regelkreises abzuleiten und nachfolgend eine Modellierung der Regelstrecke, die hier einen lagegeordneten Industrieroboter repräsentiert, durchzuführen.

5.1.1 Struktur des kartesischen Sensorregelkreises

Kartesische Sensorregelkreise nutzen die Möglichkeit kartesische Bahnkorrekturwerte im Interpolationstakt (10-40 ms) nach dem Interpolator in die Robotersteuerung einzubringen, und folglich eine Adaption der Bewegungsbahn des Roboters an vorherrschende Gegebenheiten in der Roboterumwelt zu bewirken. Der am Werkzeugbezugspunkt (TCP) des Roboters fixierte Sensor detektiert dazu, z.B. einen Versatz der programmierten Roboterbahn relativ zur Werkstückkontur. An Hand des Sensorsignals bestimmt der Sensorregler die notwendige Bahnkorrektur, die in die Robotersteuerung eingebracht wird und den Sensorregelkreis schließt.

Zunächst sollen die einzelnen Komponenten des Sensorregelkreises den Begriffs- und Modelldefinitionen nach DIN 19226 zugeordnet werden [22]. Die Bahnkorrekturwerte, die von der Sensorsteuerung SC an die Robotersteuerung übertragen werden, repräsentieren die Stellgröße des Regelkreises. Die Regelgröße stellt letztendlich die aktuell gefahrenen Bahn des Roboters dar, die möglichst exakt der Führungsgröße, die hier durch die Werkstückkontur zuzüglich einem festgelegten Abstandsvektor festgelegt ist, folgen soll.

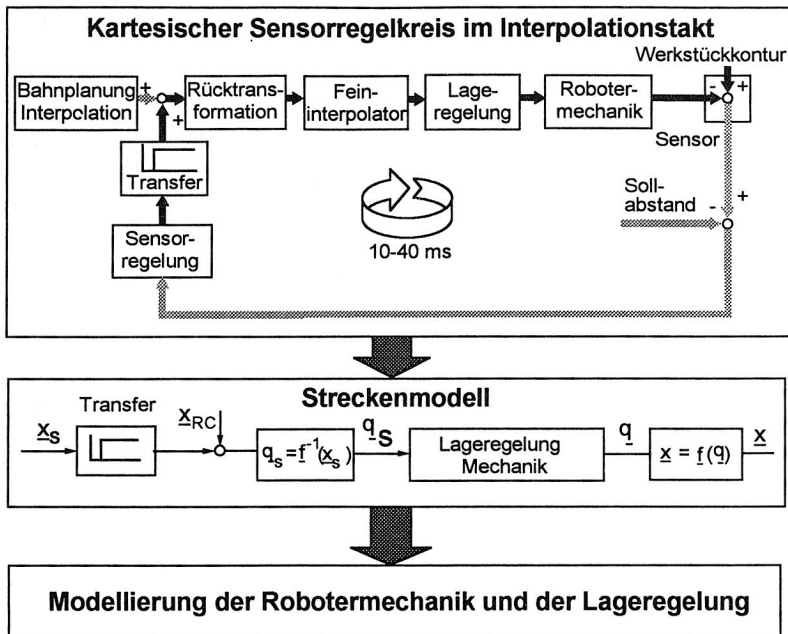


Bild 51: Grundlegende Struktur des kartesischen Sensorregelkreises und der zugrundeliegenden Regelstrecke

Folglich umfaßt die Regelstrecke die folgenden Komponenten der Robotersteuerung und des Roboters:

- ☐ **Datentransfer:** Das dynamische Systemverhalten der Datenübertragung zwischen dem Single-Board-PC und der Robotersteuerung kann durch ein entsprechendes Totzeitglied, das die Verzögerung durch den Datentransfer beschreibt, modelliert werden.
- ☐ **Rückwärtstransformation:** Die Rückwärtstransformation nimmt eine nichtlineare Abbildung der kartesischen Position und Orientierung des Roboters in das Achskoordinatensystem des Roboters vor, die später noch detailliert zu betrachten ist.
- ☐ **Feininterpolator:** Aufgabe des Feininterpolators ist es, zwischen zwei vorgegebenen, axialen Stützstellen weitere Stützstellen zu bestimmen, um die Lageregelung mit entsprechenden Sollwerten im Lageregeltakt zu bedienen.
- ☐ **Lageregelung/Mechanik:** Diese Komponente umfaßt alle Lageregelkreise sowie die Mechanik des Roboters, deren dynamisches Systemverhalten geeignet modelliert werden muß. Die Robotermechanik nimmt gleichzeitig die nichtlineare Abbildung der Achskoordinaten in kartesische Koordinaten vor.

Innerhalb der Regelstrecke wirkt zusätzlich eine Störgröße in Form der von der Bahnplanung und der Interpolation berechneten Bahnsollgröße ein. Der Sensor übernimmt gleichzeitig die Funktion der Meßeinrichtung und des Vergleichsglieds im Regelkreis. Er bildet aus der aktuellen Bahn des Roboters und der vorliegenden Werkstückkontur die Regeldifferenz, die vom Regelglied weiterverarbeitet wird. Der Sensor wird ausschließlich als Summationsglied modelliert, da die Ersatzzeitkonstante des Sensors hier kleiner ist als die Abtastzeit des Regelkreises. Nachdem die wesentlichen Komponenten des Regelkreises mit Ausnahme der Regelstrecke modelliert sind, sollen im folgenden Kapitel die Komponenten der Regelstrecke insbesondere die Lageregelung und die Robotermechanik weiter analysiert werden.

5.1.2 Modellierung eines lagegeregelten Industrieroboters

Industrieroboter sind konstruktiv so ausgeführt, daß sie aus mehreren durch Gelenke miteinander verbundenen Armen, die meist als starre Körper betrachtet werden, bestehen. Sie zählen zu den starren Mehrkörpersystemen (MKS) [162]. Das dynamische Systemverhalten dieser starren Mehrkörpersysteme läßt sich an Hand der folgenden, nichtlinearen und gekoppelten Bewegungsgleichungen darstellen [69]:

$$\underline{M}(\underline{q}) \cdot \ddot{\underline{q}} + \underline{H} \cdot \dot{\underline{q}} + \underline{f}(\underline{q}, \dot{\underline{q}}) + \underline{g}(\underline{q}) = \underline{F}(t) \quad \text{Gl. 15}$$

Dabei kennzeichnen

- \underline{M} die Trägheitsmatrix
- \underline{H} die Reibungsmatrix
- \underline{f} die Coriolis- und Zentrifugalkräfte
- \underline{g} die Gravitationskräfte
- \underline{F} die Antriebskräfte.

Zur Regelung von Mehrkörpersystemen wurde eine Vielzahl von verschiedenen, anspruchsvollen Regelungskonzepten vorgeschlagen. Es sei hier auf die einschlägige Literatur verwiesen [105, 108, 154].

Trotzdem haben sich in der Praxis zur Regelung von Industrierobotern einfache Regelstrukturen bewährt. Sie gehen dabei von einem reduzierten Mehrkörpersystem-Modell aus. Dazu werden die Bewegungen der Arme als kinetisch entkoppelt betrachtet und gleichwohl auftretende Terme in einem resultierenden Störmoment zusammengefaßt. Folglich reduziert sich Gl. 15 zu

$$\underline{M} \cdot \ddot{\underline{q}} + \underline{F}_{\text{Stör}} = \underline{F}(t). \quad \text{Gl. 16}$$

Dafür, daß sich diese Vorgehensweise trotz des nichtlinearen Systemverhaltens von Industrierobotern bewährt hat, können die nachfolgenden Gründe angeführt werden [108]:

- ☐ Die geringen Bahngeschwindigkeiten (<1m/s) bewirken nur geringe Zentrifugal- und Corioliskräfte.
- ☐ Die meist verwendeten, hochuntersetzenden Getriebe reduzieren den Einfluß der Armdynamik auf die Gesamtdynamik eines Gelenkregelkreises.
- ☐ Sehr robuste und leistungsfähige Gelenkregelungen.

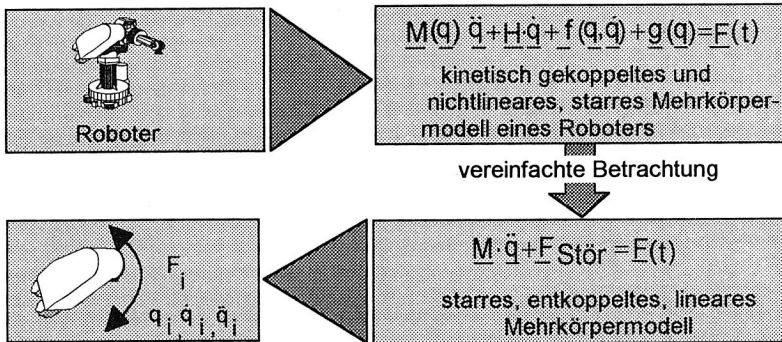


Bild 52: Vereinfachung des dynamischen Systemmodells eines Industrieroboters

Diese Regelungen zeichnen sich dadurch aus, daß jede Achse für sich betrachtet und geregelt wird und eine entsprechend hohe Robustheit gegenüber Störungen aufweist. Da die überwiegende Anzahl der heutigen, eingesetzten, kommerziell verfügbaren Industrierobotersteuerungen diese Lageregelstrukturen besitzen, sollen sie im weiteren betrachtet werden. Je Achsgelenk ist dabei ein kaskadierter Regelkreis, der aus einem Strom-, Drehzahl- und Lageregelkreis besteht, vorgesehen (Bild 53) [107]. Nach einer geeigneten Auslegung der Parameter des Strom- und Drehzahlreglers unter Verwendung des Entwurfsverfahrens des Symmetrischen Optimums kann die Drehzahlregelung durch ein resultierendes Verzögerungsglied 1. Ordnung angenähert werden. Eine detaillierte Abhandlung ist [107] zu entnehmen.

Um ein möglichst gleiches dynamisches Systemverhalten aller Achsen eines Roboters zu erzielen, wird in den Lageregelkreis oftmals ein Symmetrierglied, das ein Verzögerungsglied 1. Ordnung darstellt, eingefügt [113]. Das Symmetrierglied und der approximierte Drehzahlregelkreis lassen sich durch die Bildung der Summenzeitkonstanten als ein resultierendes Verzögerungsglied 1. Ordnung mit der Zeitkonstanten T_{rD} annähern [126]. Damit steht die Vorschrift zur Berechnung der Zeitkonstante T_{Si} des Symmetrierglieds fest:

$$T_{Si} = \max_{i=1..6} \{T_{Di}\} - T_{Di}$$

$$T_{rD} = T_{Si} + T_{Di} = \max_{i=1..6} \{T_{Di}\}$$

Gl. 17

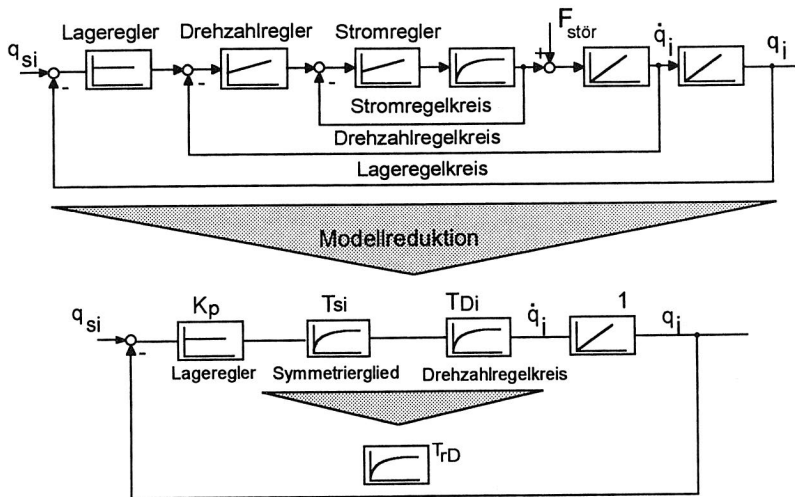


Bild 53: Reduktion des Systemmodells einer lagegeregelten Roboterachse

Folglich erhält man die Übertragungsfunktion des geschlossenen Lageregelkreises zu

$$F(s) = \frac{Q_i(s)}{Q_{si}(s)} = \frac{1}{1 + s \frac{1}{K_p} + s^2 \frac{T_{rD}}{K_p}} \quad \text{Gl. 18}$$

Um den Schleppfehler, der die bleibende Regelabweichung bei einer rampenförmigen Führungsgröße beschreibt, zu verringern, werden oft Lageregelkreise mit Geschwindigkeitsvorsteuerung eingesetzt [107]. Ausgehend vom Geschwindigkeitssollwert \dot{q}_s , der aus dem Lagesollwert durch Differentiation gewonnen wird, kann man zwei mögliche Pfade bis zur Lageabweichung e verfolgen (Bild 54). Die beiden Signalwege weisen die selbe Verzögerung auf, so daß sich im Idealfall keine Lageabweichung e ergibt. Durch die Annäherung des dynamischen Systemverhaltens des Drehzahlregelkreises durch ein Verzögerungsglied 1. Ordnung ist dieser Sachverhalt nur annähernd gegeben, jedoch kann der Schleppfehler meist um den Faktor 2 bis 3 gegenüber Lageregelungen ohne Vorsteuerung reduziert werden. Auf Grund der erforderlichen Bestimmung von \dot{q}_s ist auf eine geeignete stetige Führung des Sollwerts q_s zu achten [107].

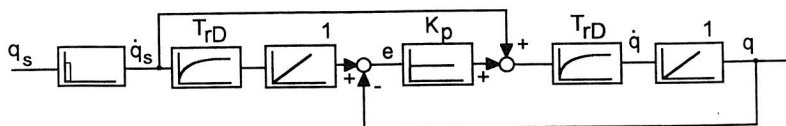


Bild 54: Lageregelkreis einer Roboterachse mit Geschwindigkeitsvorsteuerung

Im Idealfall ergibt sich die Übertragungsfunktion des Lageregelkreises mit Geschwindigkeitsvorsteuerung zu

$$F(s) = \frac{Q(s)}{Q_s(s)} = \frac{1}{1 + T_{rD}s} \quad \text{Gl. 19}$$

Die Lageregler und die notwendigen Glieder der Geschwindigkeitsvorsteuerung sowie das Symmetrierglied sind bei den meisten Robotersteuerungen digital ausgeführt. Der Drehzahl- und der Stromregelkreis sind oftmals analog implementiert. Es liegt folglich auf Grund der zeitdiskreten Arbeitsweise die Konfiguration einer Abtastregelung vor. Untersuchungen haben aber gezeigt [107], daß man den digitalen Lageregelkreis als quasikontinuierlichen Regelkreis betrachten kann, solange gilt:

$$T_A < \frac{1}{6} T_L, \quad T_A: \text{Abtastzeit, } T_L: \text{Ersatzzeitkonstante des Lageregelkreises} \quad \text{Gl. 20}$$

Diese Bedingung ist meist erfüllt, da die Ersatzzeitkonstanten der Lageregelkreise Werte zwischen 50-100 ms aufweisen und Abtastzeiten von 1-10 ms üblich sind. Die axialen Lageregelkreise sind mit Gl. 18 bzw. Gl. 19 hinreichend genau beschrieben. Da aber die Regelung im äußeren kartesischen Koordinatensystem erfolgen soll, muß auch die Kinematik des Roboters bei der Modellierung betrachtet werden. Dazu wird das dynamische Systemverhalten einer Roboterachse nach Gl. 18 an Hand einer inversen Laplace-Transformation in eine Differentialgleichung 2. Ordnung überführt:

$$\begin{aligned} \frac{T_{rD}}{K_p} \ddot{q}(t) + \frac{1}{K_p} \dot{q}(t) + q(t) &= q_s(t) \\ \Rightarrow \ddot{q}(t) + \frac{1}{T_{rD}} \dot{q}(t) + \frac{K_p}{T_{rD}} q(t) &= \frac{K_p}{T_{rD}} q_s(t) \end{aligned} \quad \text{Gl. 21}$$

Wird dies für alle Roboterachsen (hier: 6 Achsen) vorgenommen und führt man eine entsprechende Matrixschreibweise ein, so folgt:

$$\begin{aligned} \ddot{\underline{q}} + \underline{K} \cdot \dot{\underline{q}} + \underline{C} \cdot \underline{q} &= \underline{q}_s \\ \Rightarrow \ddot{\underline{q}} + \underline{K} \cdot \dot{\underline{q}} + \underline{C} \cdot (\underline{q} - \underline{q}_s) &= \underline{0} \\ \text{mit} \end{aligned} \quad \text{Gl. 22}$$

$$\ddot{\underline{q}} = \begin{bmatrix} \ddot{q}_1 \\ \vdots \\ \ddot{q}_6 \end{bmatrix}, \quad \dot{\underline{q}} = \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_6 \end{bmatrix}, \quad \underline{q} = \begin{bmatrix} q_1 \\ \vdots \\ q_6 \end{bmatrix}, \quad \underline{K} = \begin{bmatrix} \frac{1}{T_{rD1}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{T_{rD6}} \end{bmatrix}, \quad \underline{C} = \begin{bmatrix} \frac{K_{p1}}{T_{rD1}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{K_{p6}}{T_{rD6}} \end{bmatrix}$$

Unter Verwendung der Jacobi-Matrix \underline{J} , die eine linearisierte Beschreibung der Roboterkinematik darstellt [162], können die nachfolgenden Transformationsbeziehungen zwischen dem

Gelenkwinkelvektor \underline{q} und dem kartesischen Positions/Orientierungsvektor \underline{x} abgeleitet werden:

$$\begin{aligned}\dot{\underline{q}} &= \underline{J}^{-1} \cdot \dot{\underline{x}} \\ \Rightarrow \ddot{\underline{q}} &= \dot{\underline{J}}^{-1} \cdot \dot{\underline{x}} + \underline{J}^{-1} \cdot \ddot{\underline{x}}\end{aligned}\quad \text{Gl. 23}$$

Ist $\underline{q} - \underline{q}_s$ hinreichend klein, so folgt außerdem

$$\underline{q} - \underline{q}_s = \underline{J}^{-1}(\underline{x} - \underline{x}_s). \quad \text{Gl. 24}$$

Unter Einbeziehung der Transformationsbeziehungen von Gl. 23 und Gl. 24 läßt sich Gl. 22 umformen zu

$$\begin{aligned}\dot{\underline{J}}^{-1} \cdot \dot{\underline{x}} + \underline{J}^{-1} \cdot \ddot{\underline{x}} + \underline{K} \cdot \underline{J}^{-1} \cdot \dot{\underline{x}} + \underline{C} \cdot \underline{J}^{-1} \cdot (\underline{x} - \underline{x}_s) &= \underline{0} \\ \Rightarrow \underline{J} \cdot \dot{\underline{J}}^{-1} \cdot \dot{\underline{x}} + \underline{J} \cdot \underline{J}^{-1} \cdot \ddot{\underline{x}} + \underline{J} \cdot \underline{K} \cdot \underline{J}^{-1} \cdot \dot{\underline{x}} + \underline{J} \cdot \underline{C} \cdot \underline{J}^{-1} \cdot (\underline{x} - \underline{x}_s) &= \underline{0}.\end{aligned}\quad \text{Gl. 25}$$

Weisen die Drehzahlregelkreise aller Roboterachsen gleiche Zeitkonstanten T_{rD} und die gleiche Lagereglerversärkung K_p auf, so kann \underline{K} bzw. \underline{C} in die folgende Struktur überführt werden.

$$\underline{K} = k \cdot \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \dots & 0 & 1 \end{bmatrix} \quad \underline{C} = c \cdot \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \dots & 0 & 1 \end{bmatrix} \quad \text{mit } k = \frac{1}{T_{rD}}, c = \frac{K_p}{T_{rD}} \quad \text{Gl. 26}$$

Dieser Sachverhalt ist durch die Symmetrierung der Drehzahlregelkreise und dem Entwurf der Lageregler, wie vorher bereits dargestellt, gegeben. Die Gl. 25 vereinfacht sich folglich zu

$$\ddot{\underline{x}} + k \cdot \dot{\underline{x}} + c \cdot (\underline{x} - \underline{x}_s) + \underline{J} \cdot \dot{\underline{J}}^{-1} \cdot \dot{\underline{x}} = \underline{0}. \quad \text{Gl. 27}$$

Der letzte Term enthält Zentrifugal- und Coriolisbeschleunigungen, die bei geringen Geschwindigkeiten des Roboters vernachlässigt werden können, so daß aus Gl. 27 folgt:

$$\ddot{\underline{x}} + k \cdot \dot{\underline{x}} + c \cdot (\underline{x} - \underline{x}_s) = \underline{0} \quad \text{Gl. 28}$$

Wendet man Gl. 26 auf Gl. 22 an, so erhält man den analogen Ausdruck für das dynamische Systemverhalten in Achskoordinaten:

$$\ddot{\underline{q}} + k \cdot \dot{\underline{q}} + c \cdot (\underline{q} - \underline{q}_s) = \underline{0} \quad \text{Gl. 29}$$

Damit konnte gezeigt werden, daß bei kleinen Bahngeschwindigkeiten und bei gleichem dynamischen Verhalten aller lagegeregelter Roboterachsen die Dynamik jeder Roboterachse gleich der Dynamik in einer kartesischen Verfahrrichtung ist. Somit kann die kinematische Wirkung auf das dynamische Systemverhalten eines Roboters unter Berücksichtigung der obigen Voraussetzungen in 1. Näherung vernachlässigt werden.

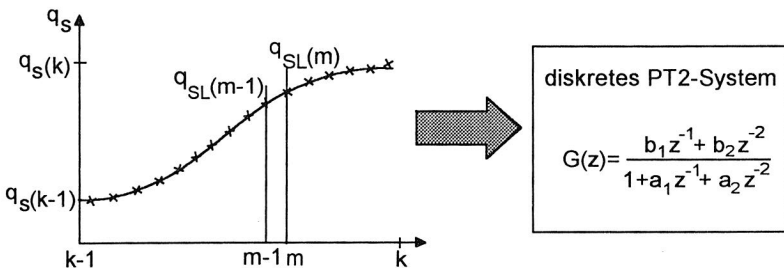


Bild 55: Aufgabe und dynamisches Systemverhalten des Feininterpolators

Vor den Lageregelkreisen einer Robotersteuerung befindet sich der Feininterpolator. Der Feininterpolator interpoliert in einem engeren Zeitraster weitere Stützpunkte zwischen zwei vom Interpolator vorgegebenen Stützpunkten und stellt gleichzeitig die Nahtstelle zwischen den im Interpolationstakt und den im Lageregeltakt arbeitenden Funktionsmodulen der Robotersteuerung dar. Im Modell wurde der Feininterpolator als diskretes System 2. Ordnung angenähert (Bild 55).

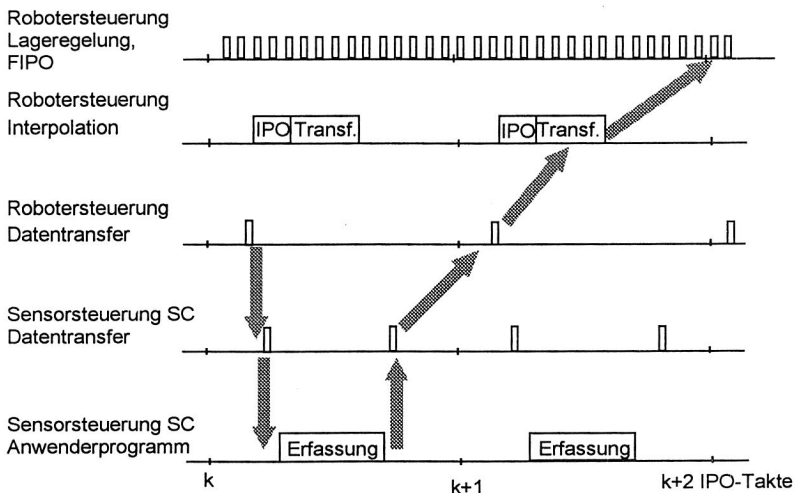


Bild 56: Zustandekommen von Verzugszeiten beim Transfer von Sensorkorrekturwerten

Zusätzlich muß im Streckenmodell eine Totzeit von zwei Interpolationstakten berücksichtigt werden. Zwischen der Anforderung von Korrekturdaten und dem „Anliegen“ dieser am Feininterpolator verstreichen prinzipbedingt zwei Interpolationstakte, die entsprechend als Totzeit zu berücksichtigen sind (Bild 56). Da bei den Übertragungsgliedern „Feininterpolator“ und „Datentransfer“ die Abtastzeitbedingungen von Gl. 20 nicht mehr erfüllt sind, können

diese Komponenten nur noch zeitdiskret modelliert werden. Folglich handelt es sich beim angestrebten, kartesischen Sensorregelkreis um eine Abtastregelung.

Im weiteren werden zwei Modelle der Regelstrecke, die sich wesentlich in der Ordnung des Modells unterscheiden, betrachtet (Bild 57). Für Zwecke der Simulation von kartesischen Sensorregelkreisen werden alle Komponenten der Robotersteuerung im vollen Detaillierungsgrad sowohl kontinuierlich wie zeitdiskret und ohne entsprechende Näherung im Simulationsmodell der Strecke berücksichtigt.

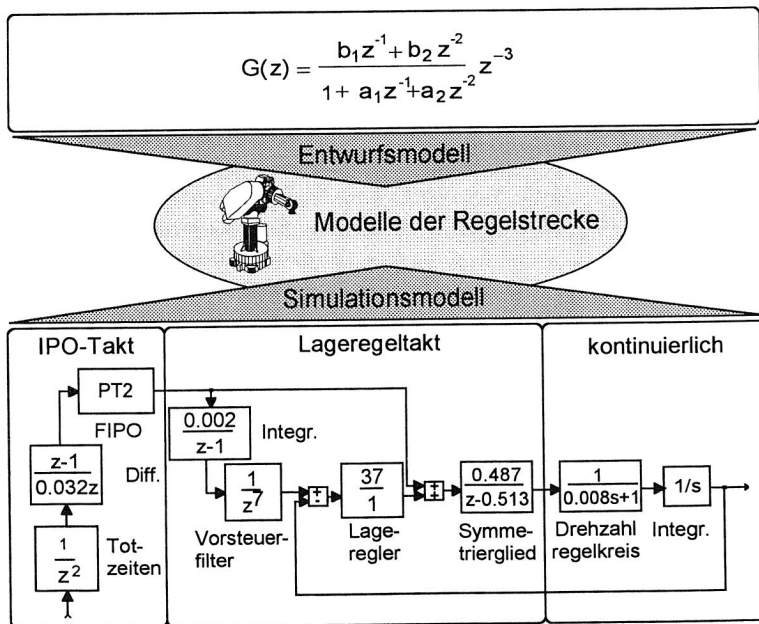


Bild 57: Streckenmodelle des kartesischen Sensorregelkreises

Für den Entwurf von Reglern soll zunächst ein einfaches, diskretes Streckenmodell identifiziert werden. Nachdem das dynamische Systemverhalten der Lageregelkreise näherungsweise durch ein Verzögerungsglied 1. Ordnung und das dynamische Systemverhalten des Feininterpolators entsprechend durch ein Verzögerungsglied 2. Ordnung beschrieben werden kann, läßt sich die Strecke zunächst als „Reihenschaltung“ der beiden Blöcke zuzüglich des prinzipbedingten Totzeitglieds annähern. Zur weiteren Vereinfachung wird das Systemverhalten des Lageregelkreises als Totzeitglied angenähert, so daß schließlich das Systemverhalten der Regelstrecke als diskretes Verzögerungsglied 2. Ordnung mit einer Totzeit von drei Interpolationstakten beschrieben werden kann. Begleitende Voruntersuchungen bestätigten diese Vorgehensweise.

$$G(z) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} z^{-3} = \frac{B(z)}{A(z)} \quad T_A = T_{IPO} \quad \text{Gl. 30}$$

Somit ergeben sich die in Bild 57 dargestellten Strukturen der Regelstrecke, deren Parameter im folgenden Kapitel zu bestimmen sind. Auf die Bausteine wurde zur Ableitung der diskreten Beschreibung im Frequenzbereich die sprunginvariante z-Transformation angewandt. Insbesondere kann bei Vorliegen der Laplace-Übertragungsfunktion über entsprechende Tabellenwerke direkt die z-Übertragungsfunktion bestimmt werden [67].

5.2 Identifikation der kartesischen Regelstrecke

Die Basis eines fundierten Regelungsentwurfs bildet ein ausreichend identifiziertes Modell der Regelstrecke. Nachdem im vorigen Kapitel eine qualitative Modellbildung durchgeführt wurde, werden nun die Parameter des Modells ermittelt. Dazu wird zunächst ein Verfahren zur Bestimmung von Modellparametern präsentiert.

5.2.1 Verfahren zur Bestimmung der Modellparameter

Zur Identifikation und Parametrierung von dynamischen Systemen existieren eine Vielzahl von Verfahren, die sich jeweils an den Gegebenheiten und Erfordernissen des zu identifizierenden Systems orientieren. [66] gibt einen Überblick über heutige, gebräuchliche Verfahren zur Systemidentifikation. Insbesondere zur Identifikation des dynamischen Verhaltens von Robotern sind in der Literatur qualifizierte Verfahren bekannt [135]. Die Situation der Systemidentifikation ist hier dadurch charakterisiert, daß bei der vorhergehenden Systemanalyse die Struktur des System weitgehend spezifiziert wurde. Ferner sind die Mehrzahl der Parameter der Strecke bereits bekannt, da sie als Konfigurationsparameter der digital implementierten Steuer- und Regelalgorithmen der Robotersteuerung offen zugänglich und änderbar sind (Maschinendaten).

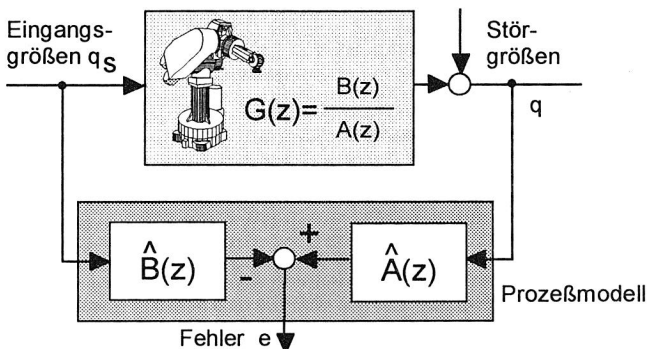


Bild 58: Blockschaltbild der nichtrekursiven Parameterschätzung nach der Methode der kleinsten Quadrate [66]

Folglich ist ein Identifikationsverfahren erforderlich, das eine Vorgabe der Systemstruktur und einiger Systemparameter zuläßt. Bild 58 zeigt das Blockschaltbild der Parameterschätzung nach der Methode der kleinsten Quadrate. Durch eine Minimierung des resultierenden Fehlers e sollen die Parameter des Modells, das strukturell in Form einer z -Übertragungsfunktion vorgegeben ist, bestimmt werden. $e(k)$ beschreibt den Fehler zwischen dem realen System und dem Modell, wobei der Fehler e linear abhängig ist von den zu schätzenden Parametern \hat{a}_i und \hat{b}_i .

$$\hat{A}(z) \cdot q(z) - \hat{B}(z) \cdot q_s(z) = e(z) \quad \text{Gl. 31}$$

In der Zeitbereichsdarstellung lautet Gl. 31, wie folgt, wobei m die Ordnung des Modells repräsentiert:

$$\begin{aligned} q(k) + \hat{a}_1(k-1) \cdot q(k-1) + \dots + \hat{a}_m(k-m) \cdot q(k-m) \\ - \hat{b}_1(k-1) \cdot q_s(k-1) - \dots - \hat{b}_m(k-m) \cdot q_s(k-m) = e(k) \end{aligned} \quad \text{Gl. 32}$$

Allgemeines Ziel einer Identifikation muß es sein, den Modellfehler e zu minimieren. Dies kann durch eine Minimierung der Funktion V , die den quadratischen Gesamtfehler beschreibt, erfolgen.

$$V = \sum_{k=1}^N e^2(k) \quad \text{Gl. 33}$$

Zur Bestimmung der je m Parameter der Parametervektoren $\hat{\underline{a}}$, $\hat{\underline{b}}$ sind $N > 2m$ Gleichungen notwendig. Abhängig von den gemessenen Eingangs- und Ausgangssignalen q_s und q können analytisch die Parametervektoren $\hat{\underline{a}}$, $\hat{\underline{b}}$ berechnet werden, so daß der quadratische Gesamtfehler minimal wird [66]. Sollen aber zusätzlich Nebenbedingungen wie die Einschränkung des Parameterraums eingebracht werden, ist dieses analytische Verfahren nicht mehr geeignet.

Zur Minimierung des quadratischen Gesamtfehlers wurde deshalb das aus der Optimierungstheorie bekannte Verfahren der Lagrange-Multiplikator-Methode verwendet [104]. Hierbei gilt es, die Lagrangefunktion L , die aus dem quadratischen Gesamtfehler V , der Nebenbedingung $g(x)$ und den Lagrangemultiplikatoren λ besteht, zu minimieren.

$$L(\hat{\underline{a}}, \hat{\underline{b}}, \lambda) = V(\hat{\underline{a}}, \hat{\underline{b}}) + \sum_{i=1}^m \lambda_i \cdot g(\hat{\underline{a}}, \hat{\underline{b}}) \quad \text{Gl. 34}$$

Bei Modellen höherer Ordnung wie in diesem Fall erfolgt die Optimierung iterativ unter Verwendung der Optimierungstoolbox von Matlab [98]. Als Vorteil dieses Verfahrens gegenüber anderen Verfahren kann angeführt werden, daß

- ☐ Nebenbedingungen eingebracht,
- ☐ Kenntnisse des zu identifizierenden System berücksichtigt und
- ☐ Identifikationen mit einfachen Testsignalen durchgeführt werden können.

5.2.2 Ergebnisse der Systemidentifikation

Als Testumgebung zur Untersuchung der kartesischen Sensorregelung stand ein Industrieroboter vom Typ Manutec r2 mit der Robotersteuerung SIROTEC ACR 20, der die notwendigen Voraussetzungen erfüllte, zur Verfügung. Bild 59 zeigt die verwendete Roboteranlage und das Testbett.

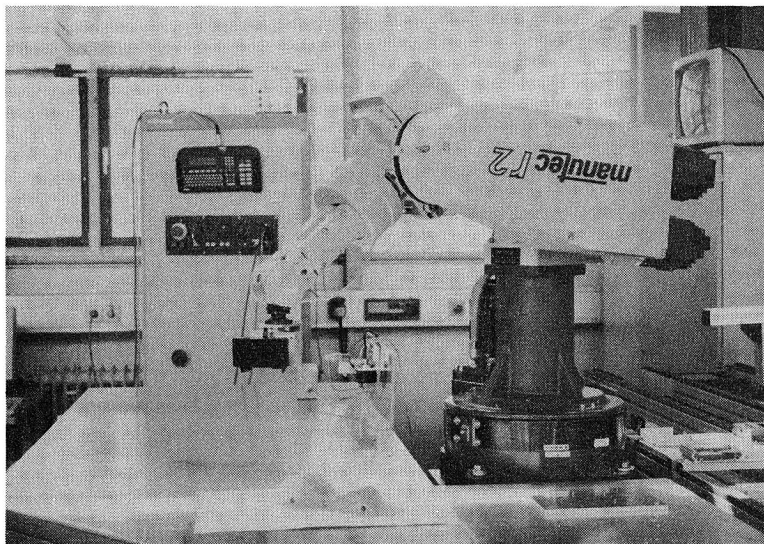


Bild 59: Eingesetzter Industrieroboter mit Triangulationssensor

Zur Anregung der kartesischen Regelstrecke am Industrieroboter wurde ein kartesischer Sprung in Richtung der z-Achse des Roboterfußkoordinatensystems aufgebracht. Die Reaktion des Roboters auf die Sprunganregung wurde protokolliert und ausgewertet. Dazu stand ein entsprechendes, zyklisches Anwenderprogramm in der Sensorsteuerung SC zur Verfügung, daß eine Sprunganregung des Roboters softwaremäßig bewerkstelligte. Als Meßsystem kam ein hochgenauer, abstandsmessender Triangulationssensor, der eine Genauigkeit von $50\text{ }\mu\text{m}$ besitzt, zum Einsatz. Die Identifikation der Regelstrecke erfolgte mit dem in Bild 60 dargestellten Meßaufbau. An Hand der im vorigen Kapitel vorgestellten Identifikationsmethode konnten die in Bild 57 dargestellten Streckenmodelle parametrisiert werden.

Wie in Bild 61 zu sehen ist, konnte ein fast deckungsgleicher Verlauf der Sprungantworten der parametrisierten Streckenmodelle und der Sprungantwort des realen Systems erreicht werden. Auftretende Modelldifferenzen können auf die Vernachlässigung der Kinematik des Roboters zurückgeführt werden.

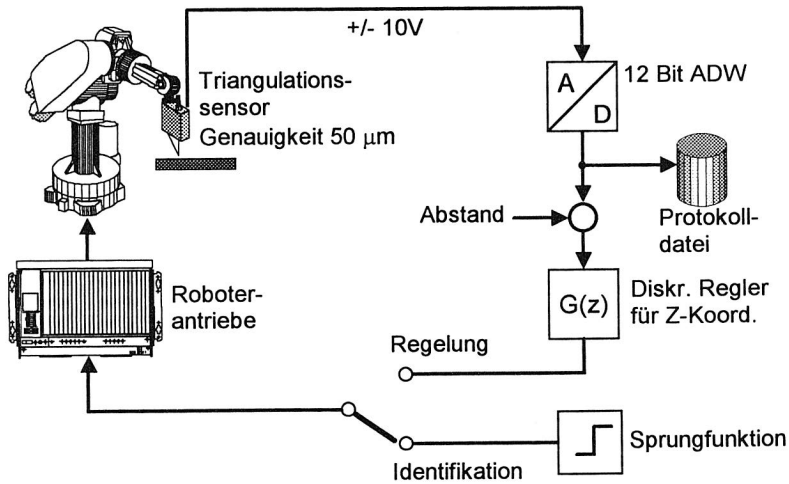


Bild 60: Aufbau des kartesischen Sensorregelkreises

Die Schwankungen in der Sprungantwort des realen Systems nach Erreichen des Sollwert lassen sich aus der Bahnungenauigkeit des Roboters ableiten. Die Ungenauigkeiten liegen hier durchaus im üblichen Bereich für Industrieroboter [113].

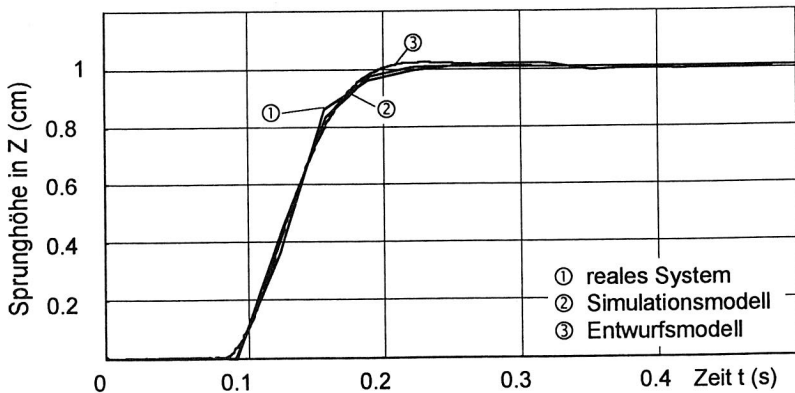


Bild 61: Sprungantworten der identifizierten Streckenmodelle und des realen Systems

Zunächst ist in der Sprungantwort deutlich die Totzeit von zwei Interpolationszykluszeiten (64 ms) zu erkennen, wobei eine nennenswerte Reaktion des Roboters erst ab ca. drei Interpolationszykluszeiten erfolgt. Danach steigt die Systemantwort sehr schnell (ca. 100 ms) auf den Endwert an.

Analysiert man die Pole des Entwurfsmodells der Strecke, so bestätigt sich das Zeitverhalten der Strecke. Eine dreifache Polstelle bei $z=0$, hervorgerufen von der Streckentotzeit, sowie ein konjugiert komplexes Polpaar in der rechten z -Ebene, das durch seine Lage nahe der reellen Achse ein gedämpftes, fast asymptotisches Verhalten der Strecke verursacht, charakterisieren das Streckenmodell.

Die Streckenidentifikation wurde in weiteren Arbeitspunkten und mit unterschiedlichen Sprunghöhen durchgeführt. Dabei kann nur eine sehr geringe Abhängigkeit der Sprungantworten vom jeweiligen Arbeitspunkt festgestellt werden. Die Arbeitspunkte, an denen die Untersuchungen vorgenommen wurden, repräsentieren durchwegs Stellungen des Roboters, die bei Bearbeitungsaufgaben typischerweise auftreten.

Zusammenfassend läßt sich festhalten, daß die Strecke ein stark totzeitbehaftetes Verhalten zeigt und nach einer Anregelzeit von 190 ms den Sollwert erreicht. Folglich muß beim nachfolgenden Reglerentwurf das Totzeitverhalten der Strecke besonders berücksichtigt werden.

5.3 Entwurf und Untersuchung von Regelungskonzepten

Verschiedene Randbedingungen, die von den Anwendungen vorgegeben werden, schränken die Freiheit beim Entwurf von Reglern ein. Folglich müssen prinzipiell geeignete Reglertypen, wobei nach [67] zwischen parameteroptimierten und strukturoptimalen Reglern unterschieden wird, auf ihre Eignung untersucht und einer Bewertung unterzogen werden. Die Untersuchung und Bewertung der Regelgüte erfolgt an Hand von Sprungantworten und deren Kenngrößen [22].

5.3.1 Zielsetzung und Randbedingungen

Die kartesische Sensorregelung verfolgt das Ziel, die kartesische Regelstrecke, die die jeweiligen, kartesischen Verfahrrichtungen des Roboters repräsentieren, unter Verwendung eines Sensorreglers geeignet zu beeinflussen, so daß die Regelgröße der Führungsgröße möglichst verzögerungsfrei nachfolgt. Insbesondere sollen sich bei zeitlich verändernden Führungsgrößen nur geringe Regeldifferenzen, die auch als Schleppfehler bezeichnet werden, ergeben. Insofern ist zunächst zu fordern, daß keine Regeldifferenzen bei sprungförmigen Führungsgrößen verbleiben.

Die eingesetzten Sensorregler müssen robust sein gegenüber Störsignalen, die die Strecke oder das Sensorsignal beeinflussen. Da bei der Identifikation der Regelstrecke gewisse Näherungen verwendet wurden, sollten die eingesetzten Regler den Regelkreis nicht in die Nähe des Stabilitätsrandes bringen.

Die kartesische Regelstrecke zeichnet sich insbesondere durch ein stark totzeitbehaftetes Verhalten aus. Da die Totzeit der Strecke die Bedingung

$$T_T > 0,8 \cdot T_G \quad T_G: \text{Ausgleichszeit der Strecke}$$

erfüllt, handelt es sich um ein System mit dominierender Totzeit T_T [55]. Ist die Totzeit dominant im Vergleich zur Prozeßdynamik, so ergeben sich einige Besonderheiten beim Einsatz und Entwurf von Reglern [67].

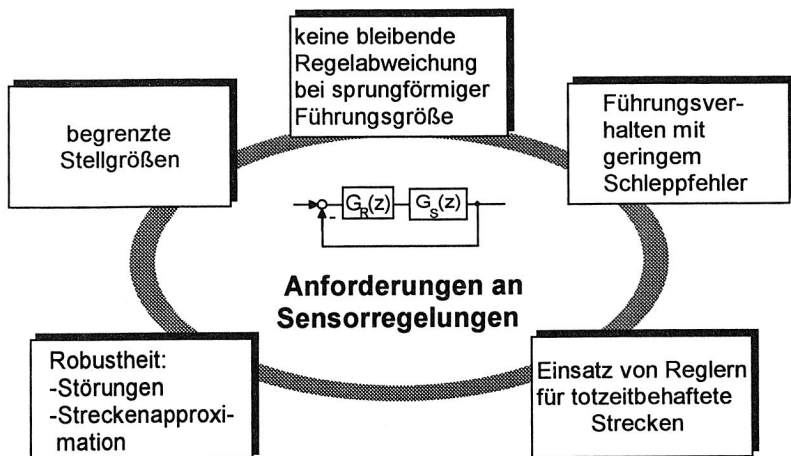


Bild 62: Wesentliche Anforderungen an die Sensorregelung von Industrierobotern

Zur Regelung von totzeitbehafteten, kontinuierlichen Prozessen kommen außer P- und I-Regler auch Prädiktor- und Deadbeat-Regler zum Einsatz. Folglich bietet es sich an, diese Regler hinsichtlich ihrer Tauglichkeit für diskrete Regelsysteme zu untersuchen. Diskrete Regler können zunächst grundsätzlich in parameteroptimierte und struktureoptimale Regler untergliedert werden [67]. Bei parameteroptimierten Reglern wird die Regelstruktur fest vorgegeben, und es werden nur noch die Reglerparameter an die Regelstrecke angepaßt. Typische Vertreter dieser Reglerklasse sind PID-Regler, die auch in ihren Abarten PI-, PD-, P-Regler auftreten können. Struktureoptimale Regler liegen dann vor, wenn sowohl die Regelstruktur als auch die Regelparameter an die Regelstrecke angepaßt werden. Kompensierende Regler wie Deadbeat-, Kompensations- und Zustandsregler sind typische Vertreter dieser Klasse. Nachdem Zustandsregler auf Grund des höheren Aufwands nicht weiter betrachtet werden, sollen im weiteren PID-, Prädiktor- und Deadbeat-Regler auf ihre Einsetzbarkeit simulativ und experimentell untersucht werden.

5.3.2 Untersuchung von parameteroptimierten Reglern

Im folgenden sollen verschiedene, parameteroptimierte Regler auf ihre Eignung hin untersucht werden. Da die Regelstrecke kein integrierendes Verhalten aufweist (Pol: $z=1$), muß der Regler einen Pol bei $z=1$ besitzen, um bleibende Regelabweichungen im Regelkreis bei sprungförmigen Führungsgrößen zu verhindern. Diese Tatsache kann unter Anwendung der

Grenzwertsätze der z-Transformation abgeleitet werden. Folglich beschränkt sich der Reglerentwurf zunächst auf PI- und PID-Regler.

PI-Regler:

Geht man von einem allgemeinen Regelalgorithmus 1. Ordnung aus, so erhält man den PI-Regelalgorithmus, indem man die Koeffizienten so wählt, daß sich ein Pol bei $z=1$ ergibt [67].

$$G_{PI}(z) = \frac{Y(z)}{E(z)} = \frac{d_0 z + d_1}{z - 1} = \frac{d_0 + d_1 z^{-1}}{1 - z^{-1}} \quad \text{Gl. 35}$$

Die Differenzengleichung ergibt sich zu

$$y(k) = d_0 e(k) + d_1 e(k-1) + y(k-1). \quad \text{Gl. 36}$$

Zur Bestimmung der freien Reglerparameter sind verschiedenste Verfahren bekannt [2, 67], wobei hier das bereits in Kapitel 5.2.1 verwendete Optimierungsverfahren eingesetzt wurde. Das Gütekriterium, das es zu minimieren gilt, beinhaltet den Betrag der Regelabweichung e und die Nebenbedingung, daß ab einem spezifizierten Zeitpunkt kein Überschwingen mehr erfolgen soll. Damit konnten sehr gute Ergebnisse bei der Auswahl der Parameter erzielt werden, die sich kaum von den Parametern unterschieden, die an Hand von Probierv Verfahren ermittelt wurden.

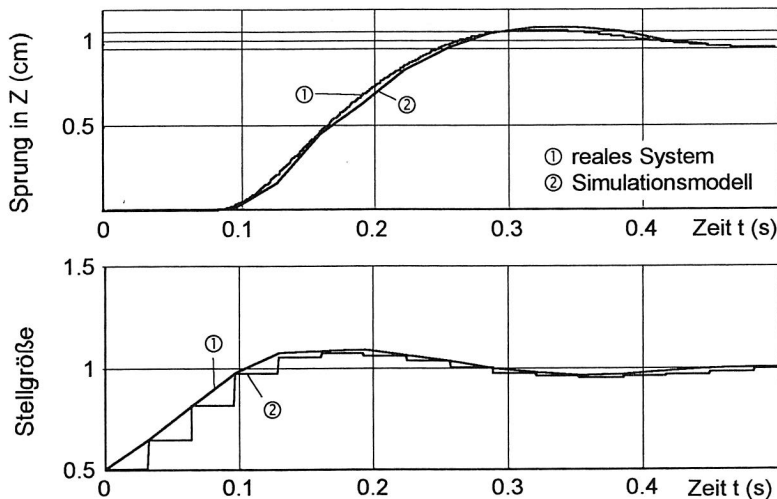


Bild 63: Sprungantwort und Stellgröße eines PI-geregelten, kartesischen Sensorregelkreises

Bild 63 zeigt sowohl die simulativ bestimmte Sprungantwort des Regelkreises als auch die gemessene Sprungantwort des Robotersystems. Dabei wurde jeweils eine Führungsgröße von

10 mm in Richtung der z-Achse des Roboterfußkoordinatensystems dem Regelkreis aufgebracht. Zunächst kann man eine gute Übereinstimmung zwischen den simulierten und den gemessenen Sprungantworten erkennen. Dies läßt darauf schließen, daß das Streckenmodell gut mit dem realen System übereinstimmt. Nach einer Anregelzeit von 267 ms erreicht die Regelgröße das Toleranzband um den Sollwert, das mit $\pm 5\%$ des Sollwerts festgelegt wurde, und verbleibt nach 346 ms innerhalb des Toleranzbands.

Auffällig ist, daß der Sensorregelkreis eine um ca. 120 ms höhere Ausregelzeit besitzt als die Regelstrecke. Die Stellgröße nimmt unwesentlich größere Werte an als die Anregungsfunktion, so daß Beschränkungen innerhalb der Robotersteuerung bei nicht zu groß gewählten Führungsgrößenprüngen kein Problem darstellen. An anderen Arbeitspunkten und mit anderen Bahngeschwindigkeiten durchgeführte Untersuchungen bestätigen die vorliegenden Ergebnisse.

PID-Regler:

Auf Grund des verzögerten Verhaltens der Sensorregelung unter Verwendung eines diskreten PI-Reglers, soll nun ein PID-Regler untersucht werden, der durch den D-Anteil ein schnelles Folgen der Führungsgröße ermöglicht. Die Übertragungsfunktion bzw. die Differenzengleichung eines PID-Reglers läßt sich vom allgemeinen Regelalgorithmus 2. Ordnung ableiten, wobei die Nennerparameter so gewählt werden, daß sich ein Pol bei $z=1$ ergibt. Diese Reglerpolstelle bewirkt ein integrierendes Verhalten des Reglers und stellt so sich, daß die bleibende Regelabweichung des Regelkreises bei sprungförmigen Führungsgrößen zu null wird.

$$G_{\text{PID}}(z) = \frac{Y(z)}{E(z)} = \frac{d_0 + d_1 z^{-1} + d_2 z^{-2}}{1 - z^{-1}} \quad \text{Gl. 37}$$

Die Differenzengleichung ergibt sich zu

$$y(k) = d_0 e(k) + d_1 e(k-1) + d_2 e(k-2) + y(k-1). \quad \text{Gl. 38}$$

Wie bereits beim PI-Regler erfolgt die Bestimmung der Reglerparameter d_0 , d_1 , d_2 unter Verwendung des Optimierungsverfahrens aus Kapitel 5.2.1.

Die in Bild 64 dargestellte Sprungantwort des simulierten und des wirklichen Regelkreises stimmen sehr gut überein. Wie erwartet, konnte die Anregelzeit des PID-geregelten Systems auf 215 ms reduziert werden. Folglich liegt die Anregelzeit hier in der Größenordnung der Anregelzeit der Strecke. Die dazu erforderliche Stellgröße ist unwesentlich größer, so daß Beschränkungen innerhalb der Robotersteuerung nicht überschritten werden. Weitere Untersuchungen zeigten, daß die Reaktion des Regelkreises auf Führungsgrößenprünge an anderen Arbeitspunkten annähernd gleich sind. Im weiteren soll der Einsatz von strukturoptimalen Reglern für die gegebene Aufgabenstellung untersucht werden.

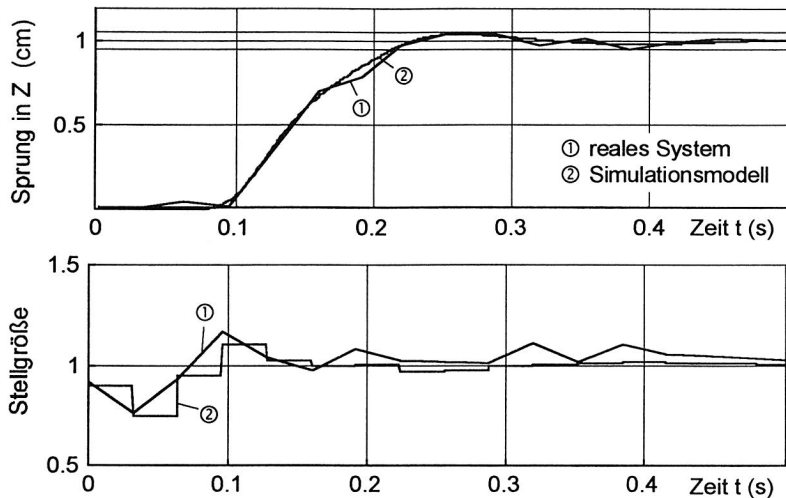


Bild 64: Sprungantwort und Stellgröße eines PID-geregelten kartesischen Sensorregelkreises

5.3.3 Untersuchung von strukturoptimalen Reglern

Strukturoptimale Regler zeichnen sich dadurch aus, daß die Reglerstruktur und die Reglerparameter vorab analytisch so festgelegt werden, daß der Regelkreis eine definierte Struktur und ein dynamisches Systemverhalten zeigt. Die wichtigsten Vertreter dieser Klasse sind die im weiteren näher untersuchten Prädiktor- und Deadbeat-Regler. Diese Reglertypen sind besonders für totzeitbehaftete Systeme geeignet [67].

Prädiktorregler:

Bei Regelstrecken mit dominanter Totzeit können Prädiktorregler eingesetzt werden, die unter der Voraussetzung einer guten Streckenkenntnis wesentlich bessere Ergebnisse liefern als Standardregler [55]. Der klassische Prädiktorregler besteht aus dem frei wählbaren Regler $R(z)$ und aus zwei Rückführzweigen, in die jeweils das Streckenmodell bzw. das totzeitfreie Streckenmodell eingehen (Bild 65).

Die Übertragungsfunktion des Prädiktorreglers lautet folglich

$$G_R(z) = \frac{Y(z)}{E(z)} = \frac{R(z)}{1 + G_s(z)(R(z) - R(z) \cdot z^{-d})} \quad \text{Gl. 39}$$

Bildet man nun die Übertragungsfunktion des geschlossenen Regelkreises

$$G_{RK}(z) = \frac{X(z)}{X_s(z)} = \frac{R(z) \cdot G_s(z)}{1 + R(z) \cdot G_s(z)} \cdot z^{-d}, \quad \text{Gl. 40}$$

so zeigt sich die Wirkung des Prädiktorreglers. Durch die spezielle Struktur des Prädiktorreglers befindet sich das separierte Totzeitglied der Strecke nicht mehr im Regelkreis sondern außerhalb (Bild 65).

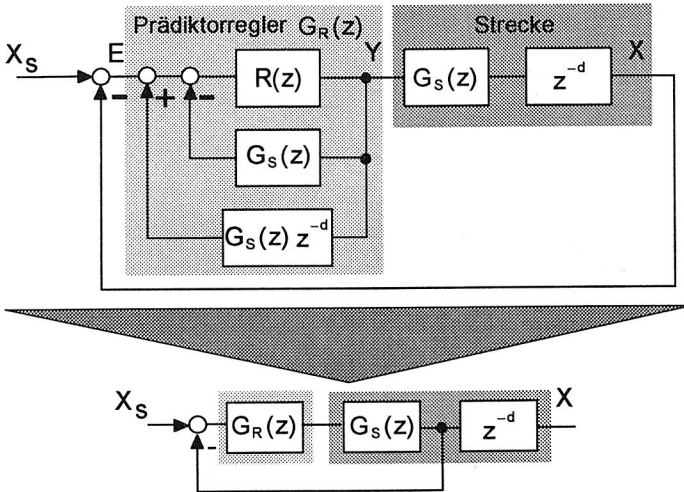


Bild 65: Struktur des Prädiktorreglers und dessen Wirkung auf den Regelkreis [55]

Folglich muß beim Entwurf und der Parametrierung des wählbaren Reglers $R(z)$ die Totzeit und ihre phasenabsenkende Wirkung im Regelkreis nicht mehr berücksichtigt werden. Meist kann der Regler $R(z)$ so parametrierung werden, daß eine bessere Dynamik des Regelkreises als beim Einsatz eines Standardreglers erzielt werden kann. Problematisch ist hierbei aber, daß eine genaue Streckenkenntnis insbesondere der Totzeit der Strecke vorhanden sein muß. Da diese Bedingung hier durchaus erfüllt ist, wird der Einsatz des Prädiktorreglers für die kartesische Sensorregelung näher untersucht.

Dazu wurde der Regler $R(z)$ als PI-Regler konfiguriert, um so bleibende Regelabweichungen nach Führungsgrößenprüngen zu verhindern. Folglich ergibt sich hier ein Prädiktorregler 8. Ordnung. Die beiden freien Parameter des Prädiktorreglers wurden mit Hilfe des für den PI- bzw. PID-Reglerentwurf bereits erprobten Optimierungsverfahrens bestimmt, so daß das Gütekriterium ein Minimum einnimmt. Das Gütekriterium beinhaltet den Betrag der Regelabweichung und eine Nebenbedingung, die das Überschwingen der Sprungantwort begrenzt.

Die Sprungantwort des realen und des simulierten Sensorregelkreises zeigen eine sehr gute Übereinstimmung. Die Anregelzeit konnte auf einen Wert von 165 ms verringert und somit auch die Anregelzeit der Strecke unterschritten werden. Auf Grund des einmaligen Überschwingens der Sprungantwort über das Toleranzband hinaus weist die Ausregelzeit einen Wert von 220 ms auf. Die erforderliche, maximale Stellgröße ist vergleichbar mit der maxima-

len Stellgröße des PID-geregelten Systems und überschreitet folglich keine Beschränkungen innerhalb der Robotersteuerung.

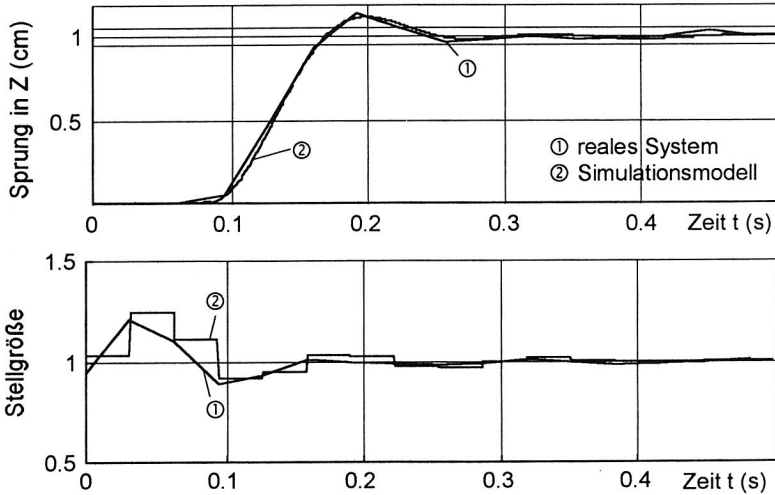


Bild 66: Sprungantwort und Stellgröße eines prädiktorgeregelten kartesischen Sensorregelkreises

Der Prädiktorregler zeigte eine höhere Empfindlichkeit gegenüber Störungen oder großen Änderungen des Arbeitspunktes als die untersuchten PI- und PID-Regler. Meist stellte sich dabei ein ungedämpfteres Systemverhalten ein. Dies ist im wesentlichen auf Diskrepanzen zwischen der realen Strecke und dem Streckenmodell zurückzuführen. Wird der Regler $R(z)$ als PID-Regler konfiguriert, so zeigen simulative Studien, daß im Vergleich zur PI-Konfiguration sich ein weniger gedämpftes Systemverhalten einstellt. Folglich ist eine PID-Konfiguration nicht geeignet.

Deadbeat-Regler:

Deadbeat-Regler weisen die Eigenschaft auf, daß sie die Strecke mit geeigneten Stellgrößen versorgen, so daß die Eingangs- und Ausgangssignale der Strecke nach einem Führungsgrößenprung von einer endlichen Einstellzeit ab sich in einem stationären Zustand befinden. Die Anwendung von Deadbeat-Reglern ist auf asymptotisch stabile Strecken beschränkt [67]. Deadbeat-Regler mit minimaler Einstellzeit weisen bei zu klein gewählter Abtastzeit oftmals sehr große Stellgrößen auf, die hier die Beschränkungen innerhalb der Robotersteuerung überschreiten würden. Für einen Deadbeat-Regler ist die Abtastzeit T_A nach [67] mit

$$\frac{T_A}{T_{An}} \geq 0,18, \text{ hier mit } \frac{T_A}{T_{An}} = \frac{32\text{ms}}{190\text{ms}} = 0,168 \quad \text{Gl. 41}$$

bereits zu klein gewählt. Um die erforderlichen Stellgrößen zu reduzieren, kann die Einstellzeit, z.B. um einen Abtastschritt, verlängert werden. Der sich dabei ergebende frei Entwurfsparameter repräsentiert einen Wert der Stellgröße. Da der 1. Stellgrößenwert im allgemeinen der GröÖte ist, wird dieser meist vorgegeben. Die nach [67] für diesen Fall gültige Beziehung

$$\frac{T_A}{T_{An}} \geq 0,11 \quad \text{Gl. 42}$$

ist erfüllt. Folglich werden im weiteren Deadbeat-Regler mit Stellgrößenvorgabe näher untersucht.

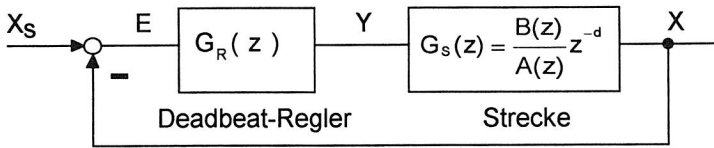


Bild 67: Regelkreisstruktur mit Deadbeat-Regler

Die Forderung nach einer endlichen Einstellzeit der Regelgröße x und der Stellgröße y nach einem Führungsgrößensprung x_s ist erfüllt, wenn nach [67] die folgenden Übertragungsfunktionen am Regelkreis die Beziehungen

$$\begin{aligned} \frac{X(z)}{X_s(z)} &= p_1 z^{-1} + \dots + p_{m+d} z^{-(m+d)} + p_{m+d+1} z^{-(m+d+1)} = P(z) \\ \frac{Y(z)}{X_s(z)} &= q_0 + q_1 z^{-1} + \dots + q_{m+1} z^{-(m+1)} = Q(z) \quad \text{mit} \quad \sum_{i=1}^{m+d+1} p_i = 1 \end{aligned} \quad \text{Gl. 43}$$

aufweisen. Die Strecke $(m+d)$. Ordnung mit einer Totzeit von d Abtastzyklen besitzt die allgemeine Übertragungsfunktion

$$G_s(z) = \frac{X(z)}{Y(z)} = \frac{b_1 z^{-(1+d)} + \dots + b_m z^{-(m+d)}}{1 + a_1 z^{-1} + \dots + a_m z^{-m}} \quad \text{Gl. 44}$$

Gleichzeitig gilt für die Streckenübertragungsfunktion, wenn Gl. 43 und Gl. 44 in ein Gleichungssystem zusammengefaßt werden.

$$G_s(z) = \frac{P(z)}{Q(z)} = \frac{p_1 z^{-1} + \dots + p_{m+d+1} z^{-(m+d+1)}}{q_0 + q_1 z^{-1} + \dots + q_{m+1} z^{-(m+1)}} = \frac{b_1 z^{-(1+d)} + \dots + b_m z^{-(m+d)}}{1 + a_1 z^{-1} + \dots + a_m z^{-m}} \quad \text{Gl. 45}$$

Damit die beiden Teile des Gleichungssystems die selbe Ordnung in z^{-1} aufweisen, müssen $P(z)$ und $Q(z)$ ein gemeinsames Polynom $(1-z^{-1}/c)$ besitzen. Folglich lautet das Gleichungssystem dann

$$\frac{(p'_1 z^{-1} + \dots + p'_{m+d} z^{-(m+d)})(1 - z^{-1}/c)}{(q'_0 + q'_1 z^{-1} + \dots + q'_m z^{-m})(1 - z^{-1}/c)} = \frac{b_1 z^{-(1+d)} + \dots + b_m z^{-(m+d)}}{1 + a_1 z^{-1} + \dots + a_m z^{-m}} \quad \text{Gl. 46}$$

Die Koeffizienten p'_i , q'_i können an Hand eines Koeffizientenvergleichs mit den Parametern a_i , b_i bestimmt werden. Analog lassen sich p_i , q_i nach Gl. 45 und Gl. 46 berechnen [67]. Die Übertragungsfunktion des Deadbeat-Reglers ergibt sich zu

$$G_R = \frac{Q(z)}{1-P(z)} = \frac{q_0 + q_1 z^{-1} + \dots + q_{m+1} z^{-(m+1)}}{1 - p_{d+1} z^{-(d+1)} + \dots + p_{m+d+1} z^{-(m+d+1)}} \quad \text{Gl. 47}$$

Auf Grund des nun frei wählbaren Parameters c kann der 1. Wert der Stellgröße $y(0) = q_0$ frei festgelegt werden. Der 1. Wert der Stellgröße darf nicht zu klein gewählt werden, da ansonst der 2. Wert der Stellgröße größer wird als der 1. Wert. [67] gibt hierzu geeignete Entwurfskriterien an. Um den maximalen Wert der Stellgröße gering zu halten, wurde der 1. Wert der Stellgröße so vorgegeben, daß er nur geringfügig größer ist als der 2. Wert.

Bild 68 zeigt die Regelgröße und die sich ergebende Stellgröße des simulierten und des tatsächlichen Sensorregelkreises bei einem Führungsgrößensprung in z -Richtung. Es konnte wiederum eine gute Übereinstimmung zwischen den Simulationsergebnissen und den Ergebnissen am realen System festgestellt werden. Im Vergleich zum PI-geregelten System weist die Sprungantwort kein Überschwingen und eine um 100 ms geringere Anregelzeit auf. Selbst im Vergleich mit der Strecke ist eine Verbesserung der Anregelzeit von 50 ms zu erkennen.

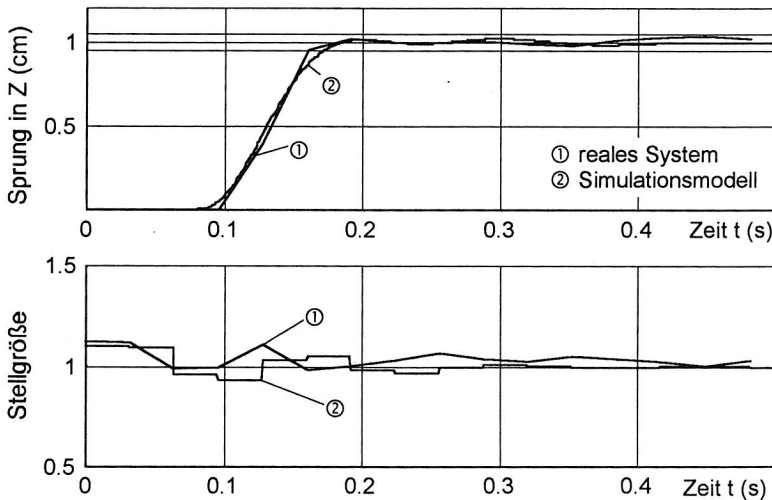


Bild 68: Sprungantwort und Stellgröße eines deadbeatgeregelten, kartesischen Sensorregelkreises

Der Deadbeat-Regler weist eine geringe Empfindlichkeit gegenüber Änderungen des Arbeitspunkts innerhalb des betrachteten Arbeitsraums auf. Weitere Untersuchungen zeigen, daß der Deadbeat-Regler sehr empfindlich gegenüber einer fehlerhaft gewählten Totzeit reagierte.

Dies wird in der Literatur bestätigt [67]. Da die Totzeit der betrachteten Regelstrecke genau bekannt ist, treten folglich aber keine Probleme auf.

5.3.4 Vergleich der untersuchten Regelungen

Die bisher untersuchten Regelungen mit den verschiedenen Typen von eingesetzten Reglern sind in der Tabelle 5 zusammengefaßt und können direkt verglichen werden. Als Bewertungsmaßstab sind im wesentlichen die Kenngrößen der jeweiligen Sprungantwort angegeben, wobei die Kenngrößen der Regelstrecke aus Referenzgründen mit aufgeführt sind.

	Regelstrecke	Regelung mit PI-Regler	Regelung mit PID-Regler	Regelung mit Prädiktorregler	Regelung mit Deadbeat-Regler
Anregelzeit (ms)	190	267	215	165	172
Ausregelzeit (ms)	190	346	215	220	172
Totzeit (ms)	96	96	96	96	96
Überschwingweite	3%	7%	4%	13%	2%
Empfindlichkeit gegen Streckentotzeitfehler	/	-	-	+	+
Empfindlichkeit gegen Arbeitspunktänderung	/	-	-	+	0

Tabelle 5: Vergleich von verschiedenen Reglern

Die Sensorregelung mit den parameteroptimierten PI- und PID-Reglern weisen auf Grund der dominanten Totzeit in der Regelstrecke höhere An- und Ausregelzeiten als die Strecke selbst auf. Demgegenüber kann die Dynamik der Sensorregelung beim Einsatz der strukturopimalen Prädiktor- und Deadbeat-Regler verbessert werden. Die Sensorregelung mit dem Prädiktorregler weist zwar die geringste Anregelzeit auf, besitzt aber eine hohe Überschwingweite von 13% des Sollwerts. Folglich stellt der Deadbeat-Regler mit Stellgrößenvorgabe die Beste der untersuchten Lösungen dar. Die Empfindlichkeit der betrachteten, parameteroptimierten Reg-

ler gegenüber Arbeitspunktänderungen oder fehlerhaft modellierten Streckentotzeiten ist durchwegs gering.

Die strukturopimalen Regler zeigen hingegen eine höhere Empfindlichkeit, da sie prinzipbedingt eine Kompensation von Streckenpolen und -nullstellen bewirken. Der Deadbeat-Regler ist zwar empfindlich gegenüber einer fehlerhaft modellierten Totzeit, weist aber ansonst in Summe die besten Kennwerte auf. Da die Totzeit der Strecke, die im wesentlichen durch Verzögerungszeiten während des Datentransfers verursacht wird, genau bekannt ist und kaum schwankt, tritt dieser Nachteil des Deadbeat-Reglers bei Sensorregelungen nicht auf. Demzufolge wird im weiteren die Sensorregelung mit Deadbeat-Reglern untersucht.

5.4 Bewertung der kartesischen Sensorregelung

Im folgenden werden die v.a. für die Bearbeitungstechnologien bedeutenden Kenngrößen wie der resultierende Schleppfehler der Sensorregelung bei rampenförmigen Werkstückverläufen (Führungsgrößenverläufen) analysiert und praktisch untersucht. Weitere Aussagen lassen sich an Hand des Amplituden- und Phasengangs der kartesischen Sensorregelung machen. Als Sensorregler werden bei allen Untersuchungen der Deadbeat-Regler mit Stellgrößenvorgabe und zu Referenzzwecken der PI-Regler eingesetzt.

5.4.1 Ableitung und Validierung des Schleppfehlers

Zur Charakterisierung des Schleppfehlers der Lageregelung bei rampenförmigen Führungsgrößen ist bei numerisch gesteuerten Werkzeugmaschinen und Robotern als Kenngröße die Geschwindigkeitsverstärkung K_V von Bedeutung (K_V -Wert). Oftmals soll die Bearbeitungsmaschine mit konstanter Geschwindigkeit v Bahnen abfahren. Dies ist gleichbedeutend mit der Vorgabe einer rampenförmigen Führungsgröße am Lageregelkreis. Hierbei stellt sich wegen des verwendeten Lagereglers ein Schleppfehler e ein, um den der Lageistwert hinter dem Lagesollwert zurückbleibt. Der K_V -Wert wird definiert als [107]

$$K_V = \frac{v}{e}. \quad \text{Gl. 48}$$

Der K_V -Wert ist dabei identisch mit der Verstärkung des Lagereglers [107]. V.a. für Anwender im Bereich der Bearbeitungstechnologien ist der K_V -Wert eine anschauliche Form zur Charakterisierung der Lageregelung einer Bearbeitungsmaschine. In Analogie soll deshalb der K_V -Wert für Sensorregelungen, im weiteren mit K_{VS} bezeichnet, definiert und abgeleitet werden.

Ergeben sich bei der sensorgeregelten Bearbeitung rampenförmige Abweichungen der Werkstückkontur gegenüber der programmierten Bahn, so bleibt die gefahrene Istbahn hinter der erforderlichen Sollbahn zurück, und es resultiert der Schleppfehler e (Bild 69).

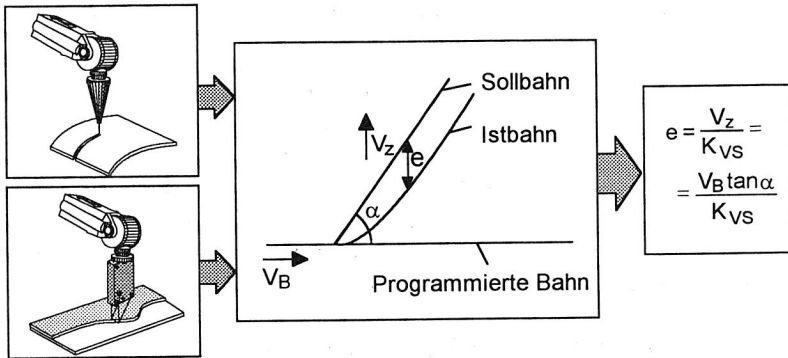


Bild 69: Entwicklung des Schleppfehlers bei sensorregelten Anwendungen

Der Schleppfehler wird von der Dynamik des Sensorregelkreises, die durch den K_{vs} -Wert charakterisiert wird, und von der erforderlichen Korrekturgeschwindigkeit v_z in Korrekturrichtung beeinflusst. Die Abhängigkeit des Schleppfehlers e vom K_{vs} -Wert und der Korrekturgeschwindigkeit v_z lässt sich so in Bild 70 graphisch darstellen.

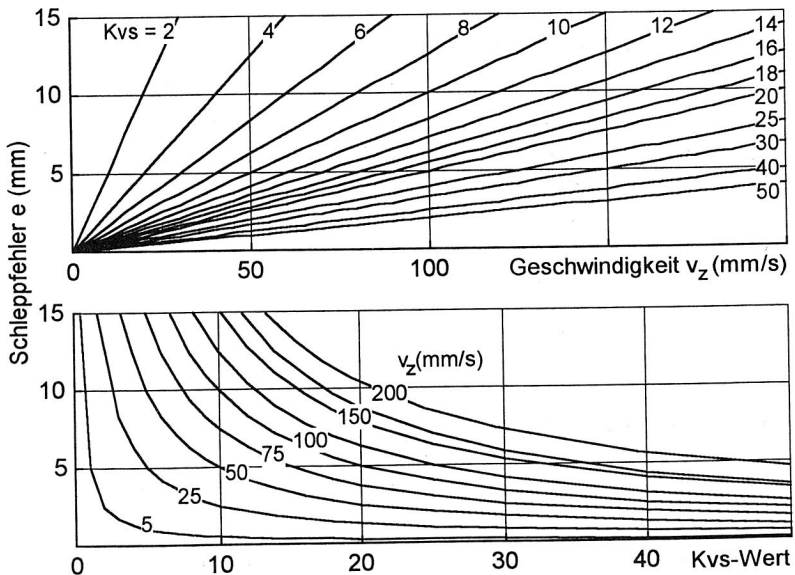


Bild 70: Resultierender Schleppfehler in Abhängigkeit der Korrekturgeschwindigkeit und des K_{vs} -Werts

Nachdem der K_{VS} -Wert von Sensorregelkreisen eine unbekannte Größe darstellt, soll der K_{VS} -Wert von PI- und deadbeatgeregelten Sensorregelkreisen allgemein abgeleitet und an Hand von experimentellen Untersuchungen validiert werden. Nach Gl. 43 und unter Berücksichtigung, daß die Parameter p_1 bis p_d den Wert null besitzen, lautet die Übertragungsfunktion eines Sensorregelkreises, der unter Verwendung eines Deadbeat-Reglers mit Vorgabe der 1. Stellgröße geregelt wird.

$$G(z) = \frac{X(z)}{X_s(z)} = p_{d+1}z^{-(d+1)} + \dots + p_{m+d+1}z^{-(m+d+1)} = P(z) \quad \text{mit} \quad \sum_{i=d+1}^{m+d+1} p_i = 1 \quad \text{Gl. 49}$$

Unter Berücksichtigung von Bild 67 kann man den Schleppfehler $E(z)$ ableiten zu

$$E(z) = X_s(z) \left[1 - \frac{X(z)}{X_s(z)} \right] = X_s(z) [1 - P(z)]. \quad \text{Gl. 50}$$

Bei Vorgabe einer rampenförmigen Führungsgröße X_s mit der Steigung v_z ergibt sich ein stationärer Schleppfehler E , der sich mit Hilfe der Endwertsätze der z -Transformation bestimmen läßt zu

$$\begin{aligned} \lim_{k \rightarrow \infty} e(k) &= \lim_{z \rightarrow 1} (z-1)E(z) = \lim_{z \rightarrow 1} (z-1)X_s(z)[1 - P(z)] \\ &= \lim_{z \rightarrow 1} (z-1) \frac{v_z T_A z}{(z-1)^2} \left[1 - p_{d+1}z^{-(d+1)} - \dots - p_{m+d+1}z^{-(m+d+1)} \right] \\ &= \lim_{z \rightarrow 1} \frac{v_z T_A z}{z-1} \left[(1-z^{-1})(1+z^{-1}+z^{-2}+\dots+z^{-d} + (1-p_{d+1}z^{-(d+1)}) + \right. \\ &\quad \left. + (1-p_{d+1}-p_{d+2})z^{-(d+2)} + \dots + (1-p_{d+1}-p_{d+2}-\dots-p_{m+d})z^{-(m+d)} \right) \Big] \quad \text{Gl. 51} \\ &= \lim_{z \rightarrow 1} v_z T_A \cdot \left[1 + z^{-1} + z^{-2} + \dots + z^{-d} + (1-p_{d+1})z^{-(d+1)} + \dots \right. \\ &\quad \left. + \dots + (1-p_{d+1}-p_{d+2}-\dots-p_{m+d})z^{-(m+d)} \right] \\ &= v_z T_A [d+1+m-m p_{d+1} - (m-1)p_{d+2} - \dots - 2p_{m+d+1} - p_{m+d}]. \end{aligned}$$

Folglich ergibt sich allgemein der K_{VS} -Wert an einer Sensorregelung, die unter Verwendung eines Deadbeat-Regler mit Vorgabe der 1. Stellgröße geregelt wird, zu

$$K_{VSDB} = \frac{v_z}{e} = \frac{1}{T_A (m+d+1 - \sum_{i=1}^m (m+1-i)p_{d+i})}. \quad \text{Gl. 52}$$

In der gleichen Weise läßt sich allgemein der Schleppfehler einer PI-geregelten Sensorregelung ableiten.

$$\begin{aligned}
 \lim_{k \rightarrow \infty} e(k) &= \lim_{z \rightarrow 1} (z-1) \frac{v_z T_A z}{(z-1)^2} \frac{1}{1 + G_{PI}(z)G_s(z)} = \\
 &= \lim_{z \rightarrow 1} \frac{v_z T_A z}{z-1} \frac{1}{1 + \frac{d_0 z + d_1}{z-1} \frac{b_1 z^{-1} + b_2 z^{-2} + \dots + b_m z^{-m}}{1 + a_1 z^{-1} + \dots + a_m z^{-m}}} z^{-d} \\
 &= \frac{(1 + \sum_{i=1}^m a_i) v_z T_A}{(d_0 + d_1) \sum_{i=1}^m b_i}
 \end{aligned} \quad \text{Gl. 53}$$

Der K_{VS} -Wert lautet dann

$$K_{VSPI} = \frac{(d_0 + d_1) \sum_{i=1}^m b_i}{(1 + \sum_{i=1}^m a_i) T_A} \quad \text{Gl. 54}$$

Setzt man die in Kapitel 5.2, 5.3.2 und 5.3.3 ermittelten Parameter der Strecke, des PI- und des Deadbeat-Regler ein, so erhält man die folgenden K_{VS} -Werte des PI-, bzw. deadbeatgeregelten Sensorregelkreises.

$$K_{VSPI} = 5,31 \frac{1}{s} \quad K_{VSDB} = 6,79 \frac{1}{s} \quad \text{Gl. 55}$$

Unter Verwendung der berechneten K_{VS} -Werte können die Schleppfehler der beiden betrachteten Sensorregelkreise in Abhängigkeit der programmierten Bahngeschwindigkeit v_B und des Steigungswinkels α der rampenförmigen Führungsgröße skizziert werden (Bild 71).

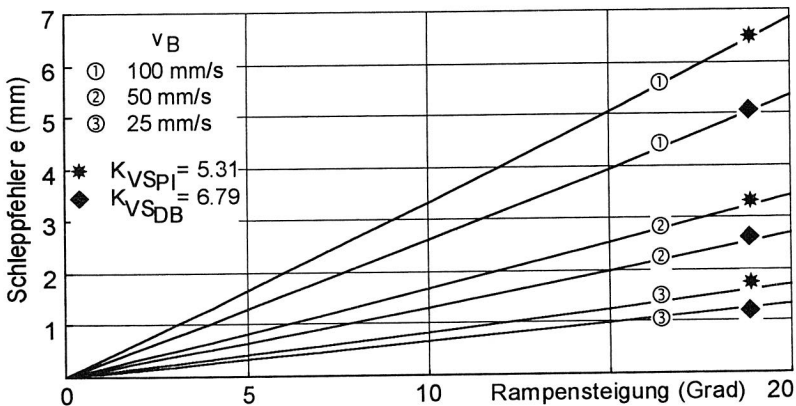


Bild 71: Schleppfehler von Sensorregelkreisen in Abhängigkeit der programmierten Bahngeschwindigkeit und des K_{VS} -Werts

Dabei ist insbesondere ersichtlich, daß die Sensorregelung mit Deadbeat-Regler auf Grund der höheren Dynamik einen geringeren Schleppfehler aufweist. Dies dokumentiert sich auch in den entsprechenden K_{VS} -Werten.

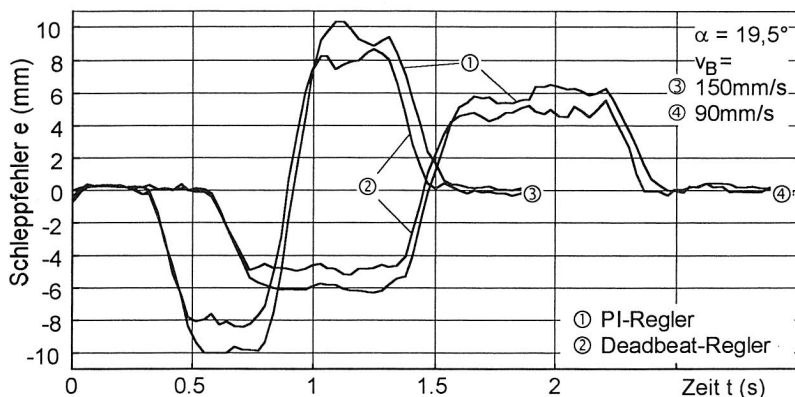


Bild 72: Gemessene Schleppfehlerverläufe von verschiedenen Sensorregelkreisen

Im folgenden werden die analytisch, rechnerisch bestimmten Schleppfehler mit den tatsächlich an der Roboteranlage gemessenen Schleppfehlern verglichen. Mit der Vorgabe einen konstanten Abstand zur Werkstückkontur einzunehmen, fährt der sensorgeregelte Roboter mit dem in Bild 60 skizzierten Aufbau über eine an- und absteigende rampenförmige Werkstückkontur. Je nach der programmierten Bahngeschwindigkeit des Roboters und dem eingesetzten Regler ergeben sich verschiedene Schleppfehlerverläufe (Bild 72).

Ein Vergleich der gemessenen und berechneten Schleppfehler bei verschiedenen programmierten Geschwindigkeiten wird in Tabelle 6 vorgenommen und zeigt eine sehr gute Übereinstimmung.

	berechneter Schleppfehler e (Rampe $\alpha=19,5^\circ$)		gemessener Schleppfehler e (Rampe $\alpha=19,5^\circ$)	
	$v_B=90\text{mm/s}$	$v_B=150\text{mm/s}$	$v_B=90\text{mm/s}$	$v_B=150\text{mm/s}$
Sensorregelkreis mit PI-Regler	6,00 mm	10,00 mm	≈ 6 mm	≈ 10 mm
Sensorregelkreis mit Deadbeat-Regler	4,67 mm	7,82 mm	$\approx 4,8$ mm	≈ 8 mm

Tabelle 6: Vergleich von gemessenen und berechneten Schleppfehlern

Schließlich soll der Amplituden- und Phasengang des deadbeatgeregelten Sensorregelkreises simulativ bestimmt und an der Roboteranlage validiert werden, um weitere Aussagen zum Einsatz und zur Verbesserung von Sensorregelkreisen zu erlangen.

Die simulative Bestimmung und graphische Darstellung des Bode-Diagramms des deadbeatgeregelten Sensorregelkreises kann sehr einfach unter Einsatz des Simulationssystems erfolgen (Bild 73). Zur Messung des Amplituden- und Phasengangs der kartesischen Sensorregelung an der Roboteranlage wird eine sinusförmige Führungsgröße (Abstand) in z-Richtung mit variabler Kreisfrequenz durch das Anwenderprogramm vorgegeben. Folglich ergibt sich für die Sensorregelung eine virtuelle sinusförmige Werkstückkontur mit veränderbarer Kreisfrequenz. Die sich ergebende Amplitudenänderung und Phasenverschiebung der Istbahn gegenüber der vorgegebenen Sollbahn wird bei verschiedenen Kreisfrequenzen gemessen (Bild 73).

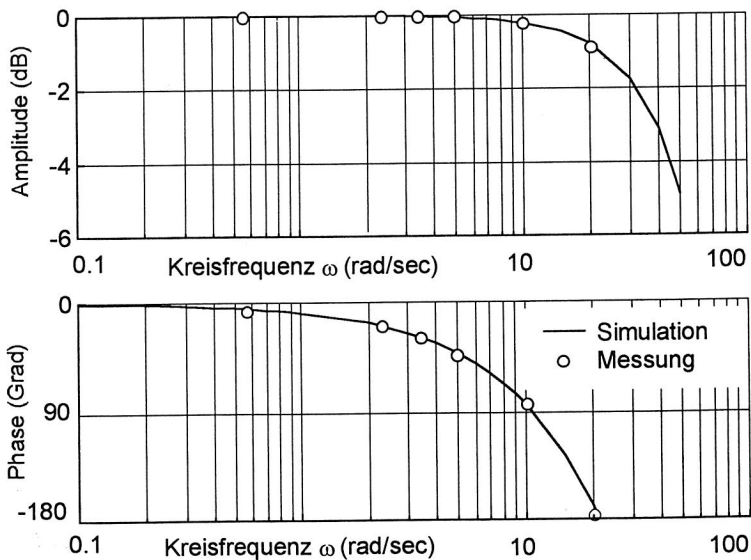


Bild 73: Amplituden- und Phasengang einer Sensorregelung mit Deadbeat-Regler

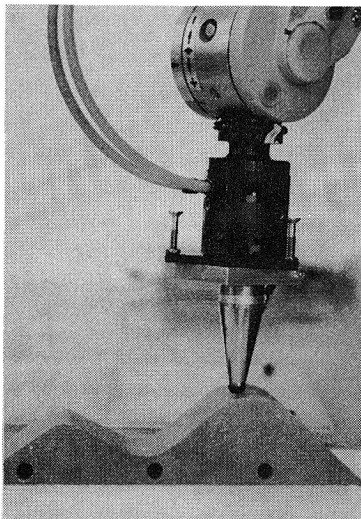
Aus Bild 73 lässt sich ablesen, daß eine Phasenverschiebung zwischen dem Ein- und Ausgangssignal am Sensorregelkreis bereits sehr früh einsetzt, wohingegen die Amplitude eine Absenkung erst ab einer Phasenverschiebung von 120° erfährt. Die Ursache hierfür sind hauptsächlich in der Totzeit der Regelstrecke zu suchen, die stark phasenabsenkend wirkt. Eine wirk-same, sensorgeregelte Bahnadaption kann grundsätzlich bis zu einer maximale Phasenabsenkung von 45° erwartet werden. Folglich lässt sich ablesen, daß bei PI- bzw. deadbeatgeregelten Sensorregelkreisen bis zu Führungsgrößenfrequenzen von ca. $f_{45^\circ} = 0,67$ Hz bzw. $0,87$ Hz zufriedenstellende Regelergebnisse zu erwarten sind.

Zusammenfassend kann man festhalten, daß sich der Schleppfehler durch den Einsatz von geeigneten Reglern reduzieren läßt. Der Hauptanteil des Schleppfehlers wird aber von der Totzeit der Regelstrecke verursacht, so daß eine Reduktion der Totzeiten für Anwendungen mit hohen Genauigkeits- und Bahngeschwindigkeitsanforderungen das Ziel sein muß. Diese Zielsetzung soll folglich in Kapitel 6 verfolgt werden.

5.4.2 Einsatzbeispiele der kartesischen Sensorregelung

Die kartesische Sensorregelung im Interpolationstakt läßt sich bei Bearbeitungstechnologien, die mittlere Genauigkeits- und Bahngeschwindigkeitsanforderungen stellen, gut einsetzen. Exemplarisch wird zunächst eine Sensorregelung unter Verwendung eines industriell verfügbaren, kapazitiven Sensors aus der Laserstrahlbearbeitung aufgebaut, mit dem Ziel einen vorgegebenen Abstand zur Werkstückoberfläche einzuhalten (Bild 74a). Mit dem Deadbeat-Regler lassen sich weitgehend die gleichen Ergebnisse erzielen wie bei den vorangegangenen Untersuchungen.

a)



b)

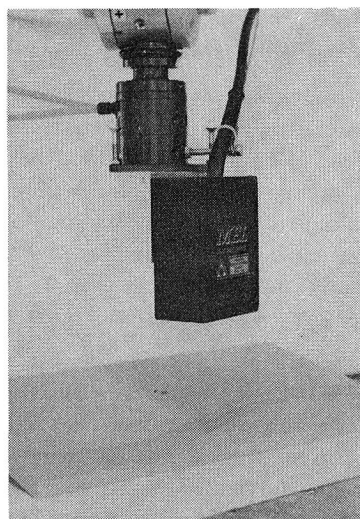


Bild 74: Beispielhafte Anwendungen für kartesische Sensorregelungen

In einem 2. Einsatzbeispiel wird der Roboter entlang einer Kante am Werkstück geführt und nimmt dabei gleichzeitig einen vorgegebenen Abstand zur Werkstückoberfläche ein (Bild 74b). Als Sensor dient ein Lasertriangulationsscanner, der sowohl den Abstand zur Werkstückoberfläche als auch die relative Lage der Kante bestimmt.

6. Schnelle Bahnkorrektur durch Sensorregelung im Lageregeltakt

Die Integration von Sensordaten im Lageregeltakt der Robotersteuerung verspricht gegenüber dem Eingriff im Interpolationstakt eine deutliche Reduktion der Totzeiten im Sensorregelkreis, so daß eine höhere Dynamik des Regelkreises resultiert [42]. Da die in sehr geringen Zykluszeiten durchzuführenden Berechnungen hohe Anforderungen an das Softwaresystem stellen, wird zunächst die Strukturierung des Transputersystems und notwendige Untersuchungen, die zur Sicherstellung des Ziels dienen, vorgestellt. Eine Untersuchung des dynamischen Systemverhaltens der Regelstrecke, der Entwurf geeigneter Regler und die Ableitung von aussagefähigen Kenngrößen zur Regelkreisdynamik folgen und werden entsprechend der Vorgehensweise in Kapitel 5 durchgeführt. Schließlich erfolgt eine vergleichende Bewertung der Ergebnisse des dynamischen Systemverhaltens von Sensorregelkreisen im Interpolations- und im Lageregeltakt.

6.1 Strukturierung des Transputersystems

Die Grundstruktur des Transputersoftwaresystems und die Mechanismen zur Anwenderprogrammsteuerung und -verwaltung auf dem Transputersystem wurde bereits in Kapitel 4.3 vorgestellt. Hier wird nun der Entwurf einer geeigneten Strukturierung der zyklischen Anwenderprogramme für die schnelle Sensorregelung im Lageregeltakt präsentiert. Im Fall der schnellen Sensorregelung muß das zyklische Anwenderprogramm eine zweckmäßige Struktur besitzen, die es ermöglicht, daß sämtliche Aufgaben wie die Erfassung von Sensorwerten, die Regelung und die Transformation von Koordinaten innerhalb der maximal zur Verfügung stehenden Zeit von weniger als einer Lageregeltaktzeit abgearbeitet werden.

Die schnelle Sensorregelung bedingt dabei grundsätzlich, die in Bild 75 dargestellte, funktionale Struktur und den skizzierten Datenfluß zwischen den Funktionsmodulen des Anwenderprogramms. Der Ablauf der zyklischen Berechnung eines neuen Sensorkorrekturwertes für die Adaption der Bewegungsbahn des Roboters beginnt mit der Übergabe der aktuellen Istwerte der Achswinkel des Roboters. Im 1. Schritt bestimmt das Modul „Vorwärtstransformation“ die kartesische Position und Orientierung des Werkzeug Bezugspunktes des Roboters aus den übertragenen Achswinkeln. Gleichzeitig wird aus den bereits vorliegenden Sensordaten und dem geforderten Sensorsollwert die Regelabweichung e ermittelt und an die Sensorregelung transferiert. Die von der Sensorregelung berechnete Stellgröße Δx ergibt zusammen mit der aktuellen Position des Roboters x_{ist} einen neuen kartesischen Sollwert der Roboterposition. Dieser Wert wird anschließend unter Verwendung der Rückwärtstransformation in die entsprechenden Sollwerte q_{korr} des Achskoordinatensystem transformiert. Da die Robotersteuerung Achswinkelkorrekturen relativ zu den aktuellen Achswinkeln fordert, werden die aktuel-

len Achswinkel subtrahiert. Abschließend werden die errechneten Korrekturwerte an die Steuerung übergeben.

Um eine effektive Verarbeitung dieser Teilaufgaben zu erreichen, ist es vor allem notwendig, die einzelnen Funktionen hinsichtlich ihrer Parallelisierbarkeit zu untersuchen. Sharkey [134] und Pantring [103] schlagen eine Verteilung auf der Ebene der Funktionen vor. Für die vorliegende Problemstellung war eine Verteilung auf Funktionenebene, d.h. Funktionen wie Transformation oder Regelung, mit Ausnahme der Sensorwerterfassung, die auf einem separaten Transputer abgearbeitet wird, nur bedingt möglich. Die Funktionen der Transformation und der Regelung lassen sich auch durch ein Pipeline-Verfahren [46] nicht parallelisieren, da sie streng sequentiell nacheinander innerhalb eines Zyklus vollständig durchgeführt werden müssen.

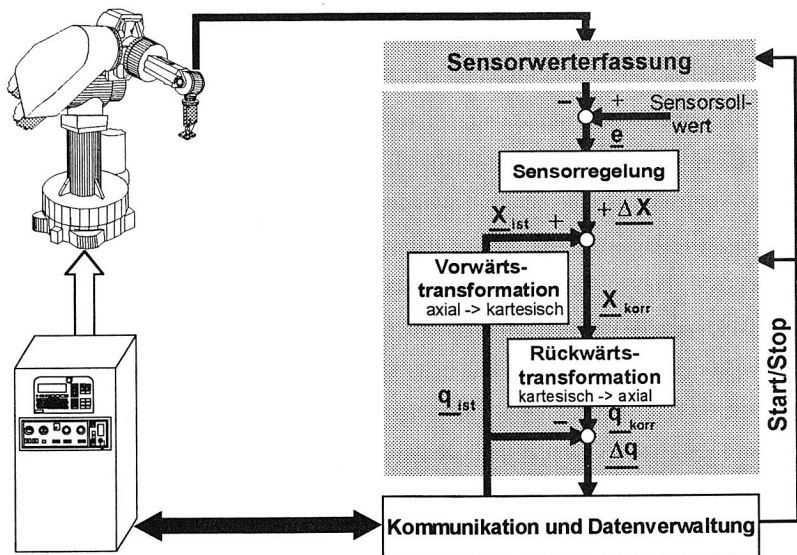


Bild 75: Funktionale Struktur des zyklischen Anwenderprogramms

Bei Untersuchungen zur Rechenzeit der Algorithmen konnte festgestellt werden, daß die Transformationen wesentlich die Gesamtrechenzeit bestimmen. Sie beinhalten den größten Aufwand an Berechnungen, der vor allem in umfangreichen, trigonometrischen Operationen besteht. Deshalb sind sie hinsichtlich ihrer Parallelisierbarkeit näher zu untersuchen. Alle anderen Funktionen umfassen nur wenige Multiplikations- und Additionsoperationen, so daß sie die Rechenzeit nur unwesentlich beeinflussen.

Die Vorwärtstransformation ermittelt aus der Stellung der Roboterachsen die Lage sowie die Orientierung des Werkzeug Bezugspunkts (TCP) bezüglich des Roboterfußkoordinatensystems.

Sie stellt eine eindeutige Abbildung der Achswinkelkoordinaten in die kartesischen Koordinaten des Werkzeugbezugspunkts her. Zur Bestimmung der Vorwärtstransformation wurde das von Reddig [106] vorgestellte Verfahren, das an den vorliegenden Robotertyp angepaßt wurde, verwendet. Dazu verschafft man sich zunächst ein vereinfachtes, geometrisches Modell des Roboters in der Nullstellung und heftet an das i . Roboter gelenk, das die Gelenke i und $i+1$ verbindet, ein Koordinatensystem K_i . Der Ursprung des Koordinatensystems liegt auf der i . Gelenkachse, und das Koordinatensystem ist parallel zum Roboterfußkoordinatensystem ausgerichtet.

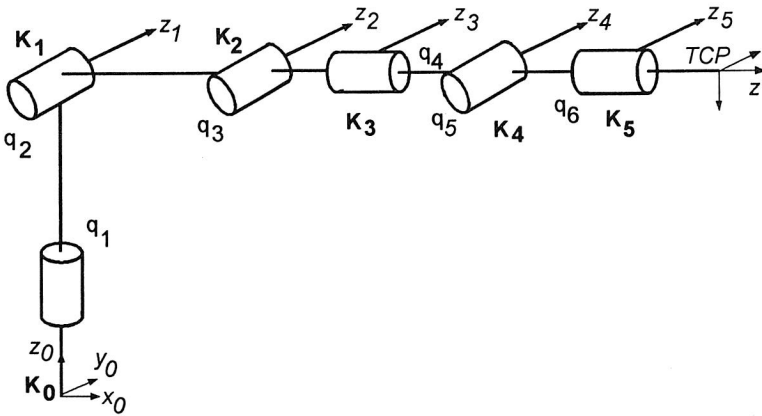


Bild 76: Definition der Koordinatensysteme des Roboters zur Ableitung der Vorwärtstransformation

Die Koordinatensysteme K_i und K_{i-1} können nun durch eine Folge von Translationen und Rotationen ineinander übergeführt werden. Diese Operationen lassen sich nach Denavit und Hartenberg [20] in der homogenen Transformationsmatrix ${}^{n-1}\underline{A}_n$ zusammenfassen. Die Position und Orientierung des Werkzeugbezugspunkts im Roboterfußkoordinatensystem erhält man durch die Multiplikation aller A-Matrizen des Roboters.

$${}^0\underline{A}_6 = \begin{bmatrix} \underline{n} & \underline{o} & \underline{a} & \underline{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^0\underline{A}_1 \cdot {}^1\underline{A}_2 \cdot {}^2\underline{A}_3 \cdot {}^3\underline{A}_4 \cdot {}^4\underline{A}_5 \cdot {}^5\underline{A}_6 \quad \text{Gl. 56}$$

Der Vektor \underline{p} beschreibt dabei die Translation, und die Vektoren \underline{n} , \underline{o} und \underline{a} repräsentieren die Orientierung des TCPs [38]. Folglich ist die Position und Orientierung des TCPs nur von den Achswinkeln abhängig und kann somit unabhängig voneinander, laufzeitparallel berechnet werden. An Hand von Voruntersuchungen, die das Ziel hatten, Rechenzeiten von trigonometrischen Funktionen und Transputerinterkommunikationszeiten zu bestimmen, wurde eine zeitoptimale Berechnung der Vorwärtstransformation angestrebt. Eine Analyse der erforderlichen Rechenoperationen der Vorwärtstransformation ergab, daß sich die Bestimmung einiger,

kartesischer Koordinaten in nur wenigen Operationen unterscheidet. Eine Abschätzung der Gesamtrechnenzeit der Vorwärtstransformation an Hand der vorab bestimmten Kommunikations- und Rechenzeiten für unterschiedliche Prozeßverteilungen auf verschiedenen Transputertopologien zeigte, daß eine Verteilung der Prozesse auf zwei Transputer die zeitoptimale Lösung des Problems darstellt. Die Gesamtrechnenzeit der Vorwärtstransformation betrug 350 µs, und die erreichte Rechenzeiterparnis gegenüber einer rein sequentiellen Abarbeitung lag bei ca. 29 % [43].

Die Rückwärtstransformation bestimmt aus einer vorgegebenen Position und Orientierung des Werkzeug Bezugspunkts die zugehörigen Achswinkel des Roboters. Die Rückwärtstransformation stellt im Gegensatz zur Vorwärtstransformation keine eindeutige Abbildung dar und kann nicht ohne weiteres aus einer Invertierung der nichtlinearen Vorwärtstransformation gewonnen werden. Zur Lösung des Problems werden sowohl analytische wie numerische Verfahren vorgeschlagen [38, 162].

Analytische Verfahren existieren meist nur für Roboter, deren Gelenkachsen parallel orientiert sind oder gemeinsame Schnittpunkte im Raum aufweisen, wohingegen numerische Verfahren sich grundsätzlich immer eignen. Numerische Verfahren zeigen aber große Ungenauigkeiten, falls der Roboter sich nahe einer singulären Stellung befindet. Unter einer singulären Stellung versteht man eine Konfiguration des Roboters, bei der der Roboter über einen seiner Freiheitsgrade nicht mehr verfügt [38]. Folglich wurde die Eignung von analytischen Verfahren und deren parallele Strukturierung näher untersucht. Analytische Verfahren weisen aber durchwegs eine sequentielle Struktur zur Berechnung der Achswinkel auf. So werden die Winkel rekursiv berechnet, d.h. der Winkel q_i ergibt sich aus dem vorhergehenden Winkel q_{i-1} und der kartesischen Pose \underline{x} .

$$q_i = f(q_{i-1}, \underline{x}) \quad \text{Gl. 57}$$

Somit benötigt jeder Berechnungsschritt die Ergebnisse der vorherigen Berechnung. Eine parallele Bearbeitung der Rechenoperationen wird folglich nur bedingt möglich sein. Diese Vorüberlegungen bestätigen sich bei der Implementierung, die an Hand der zur Vorwärtstransformation durchgeführten Voruntersuchungen optimiert wurde. Neben der geringen parallelen Struktur der Rückwärtstransformation, die hier auf zwei Transputer abgearbeitet wird, fällt besonders die unterschiedliche Auslastung der Prozessoren auf (Bild 78). Durch die Parallelisierung der Berechnung konnte die Laufzeit um 19% verringert werden. Die Werte liegen in den Bereichen, die bei ähnlichen Problemen erreicht wurden [115].

An Hand eines von Zhang [166] entwickelten Approximationsverfahrens kann die sequentielle Struktur von analytischen Rückwärtstransformationen „aufgebrochen“ werden. Unter der Annahme, daß sich die Achswinkel zwischen den Berechnungsschritten k und $k-1$ nicht stark unterscheiden und daß die Berechnung zyklisch durchgeführt wird, verwendet man für die Bestimmung der Achswinkel zum Zeitpunkt k die Achswinkel von $k-1$. Mit dieser Näherung ist die sequentielle Struktur aufgelöst, und jede Achswinkelberechnung kann parallel ausgeführt

werden. Außerdem können Berechnungen, die ausschließlich Terme mit $q(k-1)$ enthalten, vorab berechnet werden.

$$q_i(k) = f(q_{i-1}(k-1), \underline{x}(k))$$

Gl. 58

$$q_i(k) = f(q_{i-1}(k-1), \underline{x}(k)) + \Delta i$$

Darüber hinaus bietet dieses Verfahren die Möglichkeit einen weiteren Korrekturwert Δi zur Kompensation des Fehler zu verwenden. Dieser wird durch eine lineare Extrapolation des Gradienten der inversen, kinematischen Funktion gewonnen. Da die Berechnung der einzelnen Achswinkel nicht rekursiv erfolgt, akkumulieren sich die Fehler nicht. Zudem lassen die auftretenden Positionsfehler den Schluß zu, daß eine lineare Extrapolation für das Approximationsverfahren nur selten notwendig ist [166].

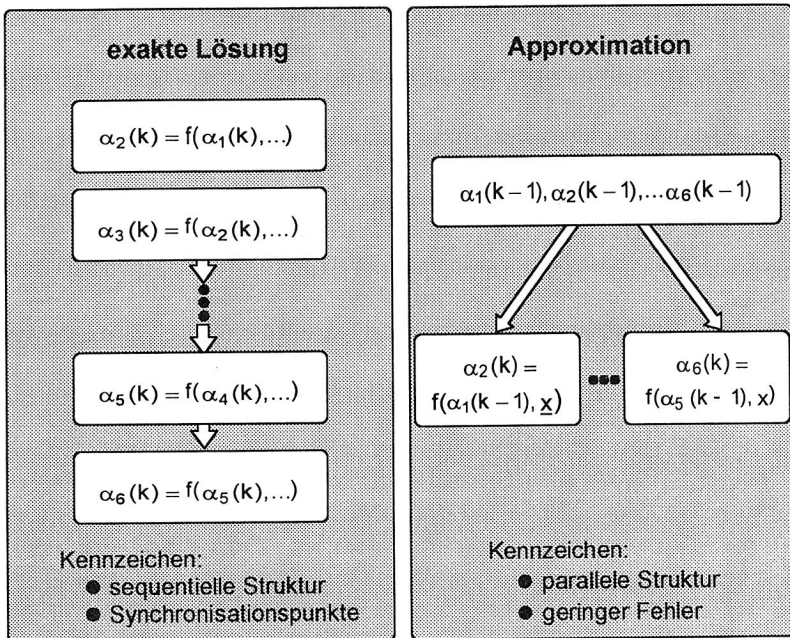


Bild 77: Eingesetzte Verfahren zur Berechnung der Rückwärtstransformation

Die Berechnung der Achswinkel wurde geeignet auf drei Transputer verteilt. Insbesondere die vorab durchgeführten Untersuchungen zum Rechenzeitbedarf von trigonometrischen Funktionen und von Transputerinterkommunikationen diente zur zeitoptimalen Verteilung der Prozesse (Bild 78). Die Gesamtrechenzeit der Rückwärtstransformation konnte durch den Einsatz der Approximationslösung von 529 μs auf 198 μs reduziert werden. Die Zeitersparnis teilt sich dabei auf in eine Zeit von 89 μs , die durch parallele Berechnung der Achswinkel erreicht wird,

und in eine Zeit von 142 μs , die durch die Verlegung bestimmter Teilberechnungen in Zeiten geringer Auslastung der Transputer eingespart wird. Die sich ergebenden Zeiten der Transformationen garantieren eine sichere Abarbeitung der verbleibenden Funktionen wie der Sensorwerterfassung und Sensorregelung innerhalb eines Lageregeltaktes.

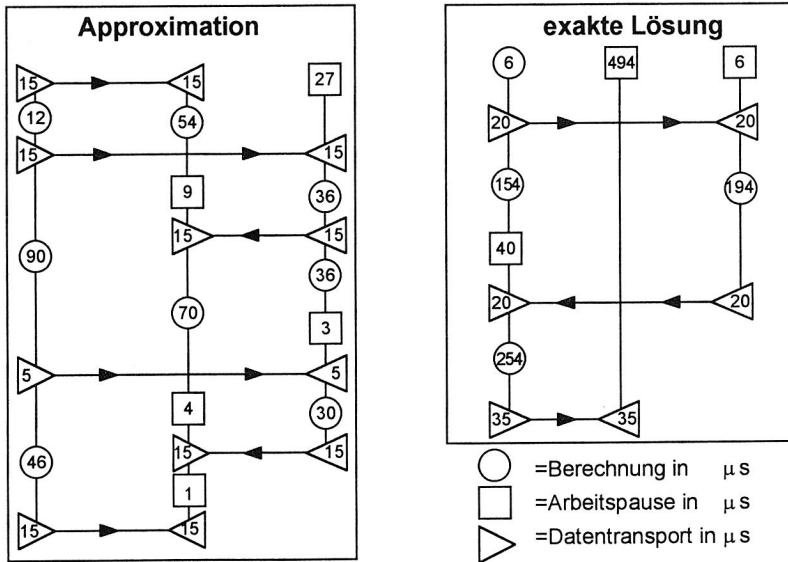


Bild 78: Zeitlicher Ablauf der parallelierten Rückwärtstransformationen

6.2 Modell des axialen Sensorregelkreises

Axiale Sensorregelkreise nutzen die Möglichkeit, die axialen Sollwerte der Lageregelkreise der Robotersteuerung direkt durch eine Übertragung von axialen Korrekturwerten im Lageregeltakt (1-10 ms) der Robotersteuerung zu beeinflussen und folglich eine Korrektur der Bewegungsbahn des Roboters zu bewirken. Dazu ist am Werkzeugbezugspunkt (TCP) des Roboters ein Sensor angebracht, der eine Ablage der programmierten Bahn relativ zur Werkstückkontur detektiert. An Hand der vom Sensor gemessenen, kartesischen Größe errechnet der Sensorregler die erforderliche Bahnkorrektur, die in die Robotersteuerung integriert wird und den Regelkreis schließt. Abhängig von der gewählten Regelungsstrategie arbeitet der Sensorregler in axialen oder kartesischen Koordinaten. Die Sensorregelung stellt eine Abtastregelung dar, deren Abtastzeit gleich der Lageregeltaktzeit der Robotersteuerung ist.

Der Sensorregelkreis beinhaltet die Lageregelung und die mechanischen Komponenten des Roboters sowie den Sensor und den Sensorregler. Außerdem ergibt sich prinzipbedingt eine

Trotz von einem Lageregelakt bei der Übertragung der Korrekturwerte an die Robotersteuerung. Die von der Interpolation berechneten, axialen Sollwerte der Lageregelung stellen hier eine Störgröße dar, die ggf. von der Sensorregelung kompensiert werden muß. Je nach Betrachtungsweise kann ein axiales und ein kartesisches Streckenmodell abgeleitet werden, die weitgehend identisch sind.

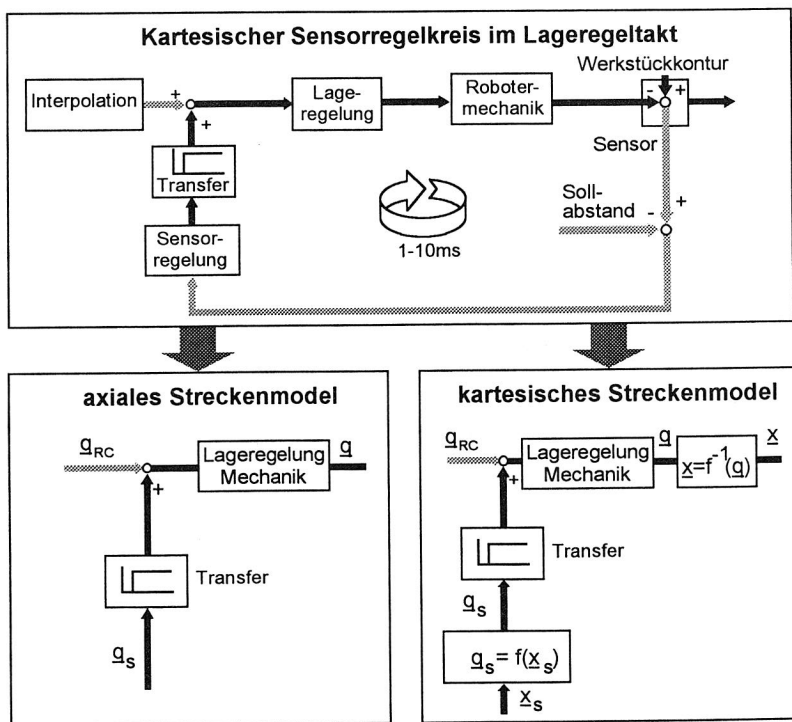


Bild 79: Struktur des schnellen Regelkreises im Lageregelakt und die zugrundeliegende Regelstrecke

Da die Struktur der Lageregelung und die Robotermechanik hier mit der in Kapitel 5.1.2 betrachteten Struktur übereinstimmt, läßt sich weitgehend das gleiche Modell einsetzen. Das starre, gekoppelte, nichtlineare Mehrkörpermodell des Roboters kann dabei durch ein starres, entkoppeltes und lineares Einkörpermodell angenähert werden. Je Achsgelenk ist ein kaskadierter Regelkreis, der aus einem Strom-, Drehzahl- und Lageregelkreis besteht, vorhanden, wobei der Drehzahl- und der Stromregelkreis in einem resultierenden Verzögerungsglied 1. Ordnung zusammengefaßt werden kann. Die Geschwindigkeitsvorsteuerung am Lageregelkreis ist im Gegensatz zu Kapitel 5.1.2 nicht vorgesehen, da eine „glatte“ Vorgabe von Lage-

sollwerten (Führungsgröße) hier nicht sichergestellt werden kann. Das Symmetrierglied zur dynamischen Symmetrierung aller Drehzahlregelkreise ist aber sehr wohl vorhanden.

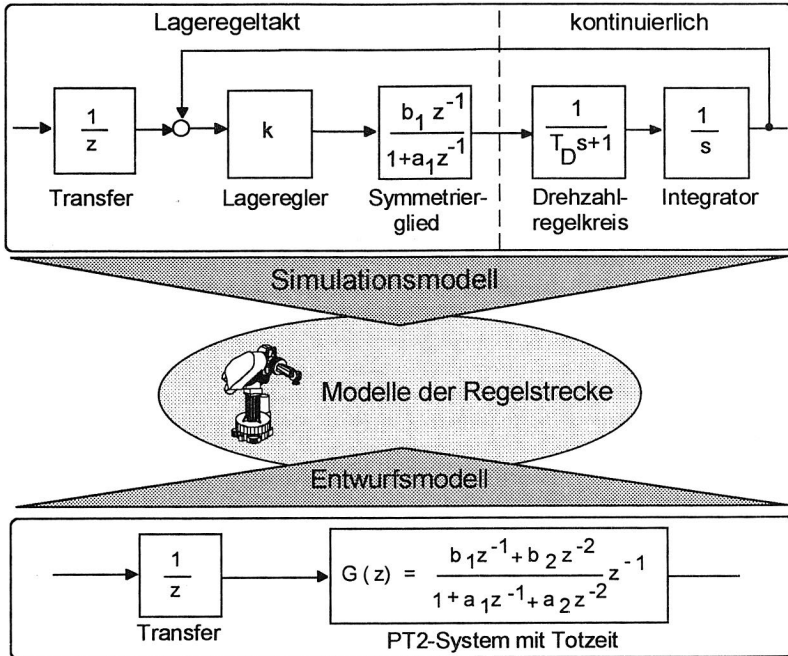


Bild 80: Entwurfs- und Simulationsmodell der Regelstrecke

Folglich resultiert das in Bild 80 dargestellte Simulationsmodell. Unter der Voraussetzung, daß die Drehzahlregelkreise der betrachteten Achsen die gleiche Ersatzzeitkonstante und die gleiche Lagereglerv Verstärkung besitzen, wurde in Kapitel 5.1.2 unter Verwendung der Jacobi-Matrix gezeigt, daß das dynamische Systemverhalten von der Kinematik des Roboters weitgehend nicht beeinflusst wird. Die Gültigkeit wurde zusätzlich zur theoretischen Ableitung in Kapitel 5.1.2 empirisch an Hand von axialen Sprungantworten der Lageregelkreise des betrachteten Roboters nachgewiesen. Folglich ist das dynamische Systemverhalten des axialen und kartesischen Streckenmodells von Bild 79 identisch.

Im weiteren wird eine kartesische Betrachtungsweise verwendet, da die Führungsgröße kartesisch vorliegt und eine kartesische Regelung im allgemeinen weniger Rechenaufwand erfordert. Eine Sensorregelung mit dem Ziel einen konstanten Abstand zur Werkstückoberfläche einzuhalten, erfordert nur einen kartesischen Sensorregler, der Korrekturen in z-Richtung berechnet, wohingegen bei einer axialen Regelung sechs Sensorregler notwendig sind.

Zwei unterschiedliche Modelle der Regelstrecke, die sich hauptsächlich im Detaillierungsgrad unterscheiden, werden weiter verwendet (Bild 80). Zur Simulation des dynamischen Systemverhaltens der Regelstrecke und des Sensorregelkreises kommt das Simulationsmodell, das alle Komponenten im vollen Detaillierungsgrad sowohl in kontinuierlicher als auch in zeitdiskreter Form und ohne entsprechende Näherung enthält, zum Einsatz. Für den Entwurf der Regler wird ein reduziertes Modell entworfen und identifiziert, das aus einem Verzögerungsglied 2. Ordnung und einem Totzeitglied mit einer Totzeit von zwei Lageregeltaktzeiten besteht. Voruntersuchungen bestätigten dabei die gewählte Struktur des Entwurfsmodells.

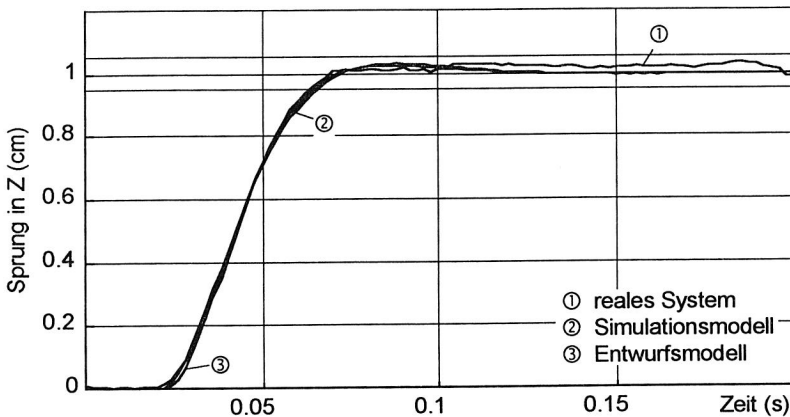


Bild 81: Sprungantworten der simulierten und der realen Regelstrecke

Zur Identifikation der Parameter des Streckenmodells wurde die Regelstrecke mit einem karthesischen Sprung, der durch ein entsprechendes Anwenderprogramm erzeugt wurde, in z-Richtung des Roboterfußkoordinatensystems angeregt. Gleichzeitig erfolgt die Protokollierung der Sprungantwort entsprechend dem Versuchsaufbau von Kapitel 5.2.2. Sämtliche Parameter des Entwurfsmodells sowie die nicht genau bekannte Ersatzzeitkonstante des Drehzahlregelkreises im Simulationsmodell konnte mit Hilfe des in Kapitel 5.2.1 vorgestellten Verfahrens zur Bestimmung der Modellparameter ermittelt werden. Dazu werden rekursiv die Sprungantworten des realen Systems sowie des Modells verglichen und anschließend die Parameter des Modells mit Hilfe des Optimierungsverfahrens sukzessive verändert, bis eine vorgegebene Modellfehlergrenze unterschritten wird. In Bild 81 sind dazu die Sprungantworten der Modelle und des realen Systems dargestellt. Dabei kann ein fast deckungsgleicher Verlauf der Sprungantworten beobachtet werden, daß auf eine sehr gute Übereinstimmung des identifizierten Modells mit dem realen System schließen läßt. Die An- und Ausregelzeiten weisen einen Wert von 65 ms auf.

6.3 Untersuchung des schnellen Sensorregelkreises

Die schnelle Sensorregelung verfolgt das Ziel, die Regelstrecke unter Verwendung eines Sensorreglers geeignet zu beeinflussen, so daß das System der Führungsgröße möglichst verzögerungsfrei nachfolgt. Insbesondere sollen bei sprungförmigen Führungsgrößenänderungen keine Regelabweichungen verbleiben und sich bei zeitlich ändernder Führungsgröße nur geringe Regeldifferenzen, die als Schleppfehler bezeichnet werden, ergeben.

Der Verwirklichung des geforderten Führungsverhaltens stehen aber bestimmte Einschränkungen, die zwingend zu berücksichtigen sind, entgegen. So müssen die Stellgrößen, die in Form von Bahnkorrekturen an die Robotersteuerung übertragen werden, begrenzt werden, um Beschränkungen in der Lageregelung der Robotersteuerung nicht zu verletzen. Insofern ist eine Begrenzung der geforderten Regelkreisdynamik auf Werte der Streckendynamik geeignet, Überlastungen der Robotermechanik wie Lager und Getriebe zu verhindern und einem erhöhten Verschleiß der Robotermechanik vorzubeugen. Dabei ist insbesondere zu berücksichtigen, daß die Lageregelung der Roboter meist werkseitig optimal eingestellt sind, um bei Bahnbewegungen optimale Ergebnisse zu erzielen [121]. Schließlich muß der Sensorregelkreis eine hohe Robustheit gegenüber Änderungen des Streckenverhaltens, das z.B. durch Parameteränderungen hervorgerufen wird, und gegenüber einwirkenden Störungen aufweisen. Falls sich Modellfehler einstellen, sollte der eingesetzte Regler den Regelkreis nicht in einen instabilen Zustand bringen.

Auf Grund dieser Anforderungen und der Eigenschaften der Regelstrecke muß der Regler ein integrierendes Verhalten (Polstelle: $z=1$) aufweisen. Folglich wird zunächst ein PI-Regler auf seine Eignung hin untersucht. Die Parameter des Reglers werden unter Verwendung des in Kapitel 5.2.1 bereits vorgestellten Optimierungsverfahrens ermittelt. Dabei gilt es, Parameter zu bestimmen, die eine Minimierung eines quadratischen Regelgütekriteriums unter der Nebenbedingung erfüllen, daß die Regelabweichung ab einem vorgegebenen Zeitpunkt einen definierten Maximalwert nicht mehr überschreitet.

In Bild 82 kann eine sehr gute Übereinstimmung der simulierten wie auch der realen Sprungantwort des Sensorregelkreises im Lageregeltakt beobachtet werden. Nach einer Anregelzeit von 70 ms tritt die Regelgröße in das Toleranzband um den Sollwert ein und verläßt diesen nicht mehr. Die Anregelzeit des Sensorregelkreises ist daher nur unwesentlich größer als die Streckenanregelzeit, womit eine Mehrzahl der Anforderungen mit einem PI-Regler bereits erfüllt werden kann. Die Stellgröße nimmt nur unwesentlich größere Werte an als die Führungsgröße, so daß Beschränkungen innerhalb der Robotersteuerung bei nicht zu groß gewählten Führungsgrößenprüngen nicht überschritten werden.

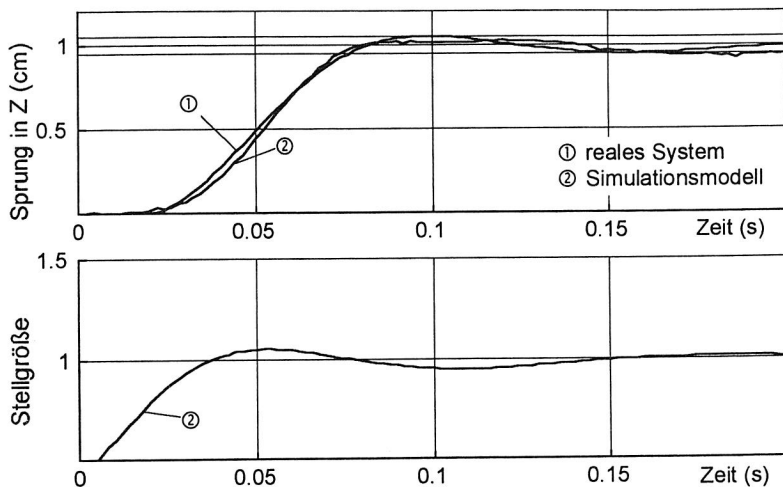


Bild 82: Gemessene und simulierte Sprungantwort des Sensorregelkreises mit einem PI-Regler

An anderen Arbeitspunkten durchgeführte Untersuchungen bestätigen die vorgestellten Ergebnisse. Die untersuchten Arbeitspunkte stellen dabei typische Positionen des Roboters bei Bearbeitungsaufgaben im Arbeitsraum des Roboters dar. Begleitend durchgeführte Untersuchungen, die eine axiale Sensorregelung mit je einem axialen Sensorregler für jeden Lageregelkreis des Roboters zu Ziel haben, führen zu annähernd deckungsgleichen, kartesischen Sprungantworten. Damit kann die Tragfähigkeit des gewählten, einfachen Ansatzes mit einem kartesischen Sensorregler praktisch nachgewiesen werden.

Zur Charakterisierung der Dynamik von Sensorregelkreisen bei rampenförmigen Führungsgrößen wurde in Kapitel 5.4.1 der K_{VS} -Wert definiert und analytisch für deadbeat- und PI-geregelte, diskrete Systeme mit Streckentotzeit bestimmt. An Hand der Reglerparameter und der Streckenparameter des Entwurfsmodells erhält man den K_{VS} -Wert der betrachteten Sensorregelung zu

$$K_{VS} = 21,4 \frac{1}{s} \quad \text{Gl. 59}$$

Der aus dem K_{VS} -Wert resultierende Schleppfehler der Sensorregelung wird an Hand von Schleppfehleruntersuchungen an der Roboteranlage validiert.

Dazu wird im Softwaresystem ein rampenförmig ansteigender Abstandsollwert bezüglich einer ebenen, waagrechten Werkstückkontur vorgegeben und der resultierende Schleppfehler bestimmt. Um eine Vergleichbarkeit der gemessenen Schleppfehler mit den Ergebnissen in Kapitel 5.4.2 zu gewährleisten, wurden die Messungen jeweils bei den gleichen Bahngeschwindig-

keiten und Rampensteigungen durchgeführt. In Bild 83 ist der sich ergebende Schleppfehler, der mit Störsignalen überlagert ist, dargestellt. Ein Vergleich der berechneten und der gemessenen Schleppfehler im Lageregeltakt zeigt Tabelle 7, wobei eine sehr gute Übereinstimmung der Werte festgestellt werden kann.

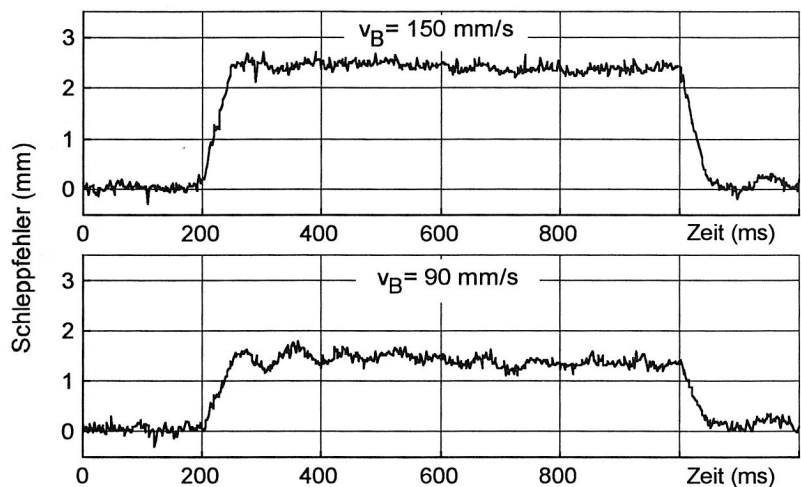


Bild 83: Schleppfehler des Sensorregelkreises bei unterschiedlichen Bahngeschwindigkeiten und Rampensteigungen

Um eine umfassende Aussage zum dynamischen Systemverhalten des Sensorregelkreises zu erhalten, wird schließlich der Amplituden- und Phasengang des Sensorregelkreises simulativ bestimmt und anschließend an der Roboteranlage an einigen Meßpunkten validiert. Simulativ kann das Bodediagramm des Regelkreises sehr einfach ermittelt werden. Die Bestimmung des Amplituden- und Phasengangs an der Roboteranlage erfolgt durch Vorgabe einer sinusförmigen, kartesischen Führungsgröße und der Messung der resultierenden Amplituden- und Phasenänderung der Istbahn gegenüber der Führungsgröße. Die Messung wird bei unterschiedlichen Kreisfrequenzen der sinusförmigen Führungsgröße durchgeführt.

	berechneter Schleppfehler e (Rampe: $\alpha=19,5^\circ$)		gemessener Schleppfehler e (Rampe: $\alpha=19,5^\circ$)	
	$v_B=90\text{mm/s}$	$v_B=150\text{mm/s}$	$v_B=90\text{mm/s}$	$v_B=150\text{mm/s}$
Sensorregelkreis mit PI-Regler	1,49 mm	2,48 mm	$\approx 1,5 \text{ mm}$	$\approx 2,5 \text{ mm}$

Tabelle 7: Vergleich von gemessenen und berechneten Schleppfehlern

Wie man in Bild 84 erkennen kann, besteht eine weitgehende Übereinstimmung des simulierten Bodediagramms mit den Meßpunkten, die an der Roboteranlage ermittelt wurden. Dies dokumentiert wiederum die Tragfähigkeit des verwendeten Streckenmodells. Das Bodediagramm ist weitgehend gekennzeichnet vom charakteristischen Verhalten eines Verzögerungsglieds 2. Ordnung mit einer Kennkreisfrequenz $\omega_0 \approx 45 \text{ rad/s}$ (vgl. Entwurfsmodell). Die Kennkreisfrequenz stellt gleichzeitig die Eckfrequenz im Bodediagramm dar, ab der der Amplitudengang gegen eine Asymptote mit einer Steigung von -40 dB/Dekade strebt.

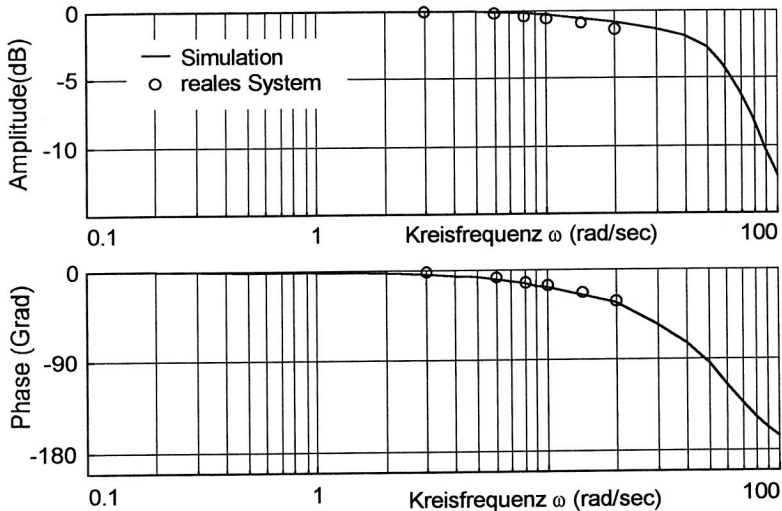


Bild 84: Amplituden- und Phasengang des schnellen Sensorregelkreises im Lageregeltakt

6.4 Vergleich und Bewertung der Sensorregelung im Interpolations- und Lageregeltakt

Die Sensorregelung im Interpolationstakt unterscheidet sich von der Sensorregelung im Lageregeltakt hauptsächlich dadurch, daß das Verhalten der Strecke im Interpolationstakt von einer dominanten Totzeit geprägt ist. So besitzt die Strecke im Interpolationstakt eine um den Faktor 2,9 größere Anregelzeit als die Strecke im Lageregeltakt.

Die Streckentotzeit im Interpolationstakt ist prinzipbedingt und kann nur durch eine Verringerung der Interpolationstaktzeit gesenkt werden, was aber aus Gründen der erforderlichen Rechenzeit der Bahnplanung der Robotersteuerung nicht möglich ist. Der im Interpolationstakt verwendete Deadbeat-Regler kann zwar die Anregelzeit der Sensorregelung deutlich verringern, es verbleibt aber letztendlich eine um den Faktor 2,5 größer Anregelzeit gegenüber der

Sensorregelung mit PI-Regler im Lageregeltakt. Die Verbesserung des dynamischen Verhaltens wird dabei durch ein schnelleres Ansteigen der Sprungantwort nach dem hauptsächlich von der Streckentotzeit geprägten Zeitraum bis 100 ms erreicht.

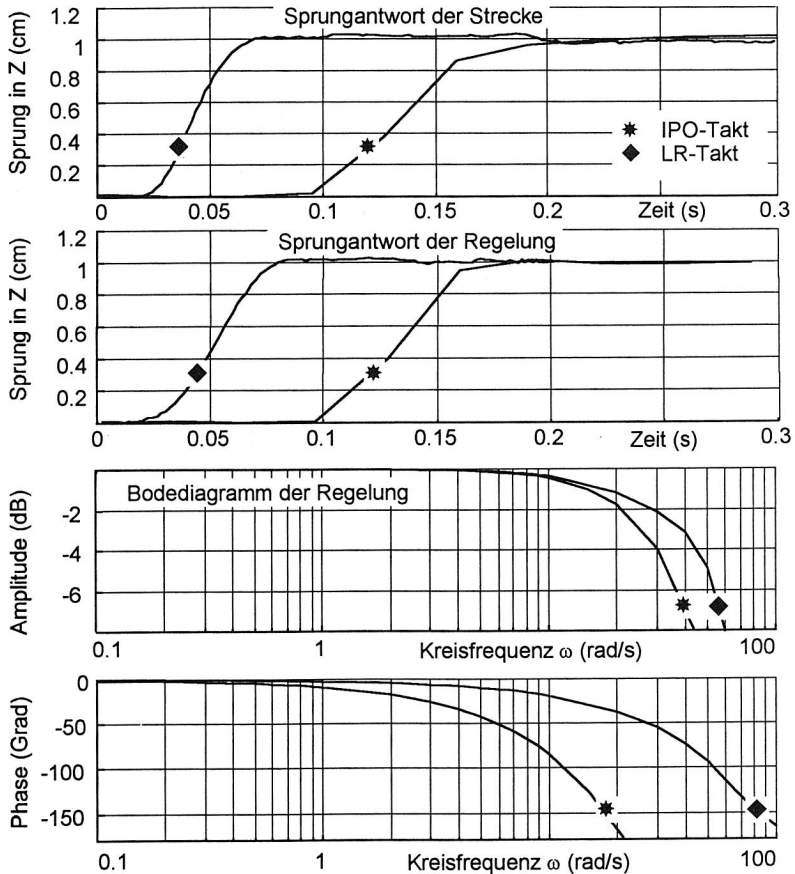


Bild 85: Vergleich des dynamischen Verhaltens von Sensorregelungen im Interpolations- und im Lageregeltakt

Die Wirkung der Streckentotzeit wird insbesondere im Amplituden- und Phasengang der sensorregelten Systeme deutlich. Man kann hierbei besonders die stark phasenabsenkende Wirkung der Streckentotzeit im Interpolationstakt erkennen. Der schnelle Abfall der Phase im Interpolationstakt bedingt einen höheren Schleppfehler beim Folgen einer vorgegebenen Kontur gegenüber der Sensorregelung im Lageregeltakt. Die Erhöhung des Schleppfehlers wird deut-

lich, betrachtet man den um den Faktor 3,2 höheren K_{VS} -Wert der Sensorregelung im Interpolationstakt.

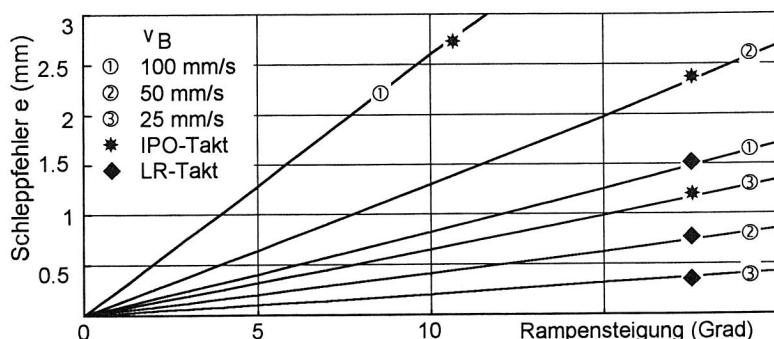


Bild 86: Schleppfehler der Sensorregelung im Interpolations- und im Lageregeltakt in Abhängigkeit der Geschwindigkeit und der Rampensteigung

Bild 86 gibt dazu einen Überblick zu den resultierenden Schleppfehlern der sensorgeregelten Systeme. Der Schleppfehler stellt somit ein entscheidendes Kriterium dar, um festzustellen, ob eine Aufgabenstellung durch Sensorregelung realisierbar ist. Folglich werden nun die Anforderungen der Bearbeitungstechnologien hinsichtlich der Bahngeschwindigkeit und der Genauigkeit, wie sie in Kapitel 3 spezifiziert wurden, den Kennwerten der Sensorregelung im Interpolations- und Lageregeltakt gegenüber gestellt. Dazu wird in Tabelle 8 die maximal mögliche Rampensteigung der Werkstückkontur angegeben, die von den betrachteten sensorgeregelten Systemen bei der angegebenen Bahngeschwindigkeit mit der erforderlichen Genauigkeit „abgefahren“ werden können. Die Bahngeschwindigkeit und Genauigkeit stellen die typischen Werte der betrachteten Bearbeitungstechnologien dar (vgl. Kapitel 3). Zur Veranschaulichung ist in Tabelle 8 auch die maximale Höhe y der Rampe bei einer vorgegebenen Basislänge x aufgeführt.

Anwendungen und deren Kriterien	Bahngeschwindigkeiten (m/min)	Genauigkeiten (mm)	Sensorregelung			
			im IPO-Takt: max. Steigung (Grad)	im IPO-Takt: max. Anstieg: y (mm), x=50mm	im LR-Takt: max. Steigung (Grad)	im LR-Takt: max. Anstieg: y (mm), x=50mm
Laserstrahlschneiden	1 - 15	0,2	4,7-0,3	4-0,3	14,4-1,0	13-0,9
Laserstrahlschweißen	1 - 6	0,2	4,7-0,8	4-0,7	14,4-2,5	13-2,1
Laserstrahlbiegen	1 - 10	0,2	4,7-0,5	4-0,4	14,4-1,5	13-1,3
Bahnschweißen	0,5 - 3	0,8	33,1-6,2	32,6-5,4	64,0-18,9	102,7-17,1
Kleberauftrag	10-30	0,7	1,6-0,5	1,4-0,5	5,1-1,7	4,5-1,5
Entgraten v. Kunststoff	5 - 15	0,5	2,3-0,8	2,0-0,7	7,3-2,5	6,4-2,1
Entgraten v. Keramik	0,1 - 1	0,3	50,7-7,0	61,1-6,1	75,4-21,1	192,6-19,3

Tabelle 8: Realisierbarkeit von Anforderungen der Bearbeitungstechnologien

In Tabelle 9 sind abschließend die wesentlichen Kenngrößen der Sensorregelung im Interpolations- und Lageregeltakt vergleichend zusammengefaßt.

Den Vorteilen der Sensorregelung im Lageregeltakt gegenüber der Sensorregelung im Interpolationstakt steht der Nachteil eines deutlich höheren Rechenaufwands zur Bestimmung der Stellgrößen für die Robotersteuerung gegenüber. Insbesondere die notwendige Berechnung der Vorwärts- und Rückwärtstransformation des Roboters auf dem Transputernetzwerk erfordert eine entsprechende Rechenleistung und bedingt höhere Anschaffungskosten als auch höhere Kosten bei der Programmerstellung. Dieser Nachteil dürfte sich aber in den nächsten Jahre stark abschwächen, da gerade für das PC-Basissystem leistungsfähige Prozessoren in der Entwicklung sind. Zusammenfassend besitzt folglich die Sensorregelung im Lageregeltakt das höchste Entwicklungspotential.

	Sensorregelkreis im Interpolationstakt mit Deadbeat-Regler	Sensorregelkreis im Lageregeltakt mit PI-Regler
Anregelzeit der Strecke (ms)	190	65
Anregelzeit der Regelung (ms)	172	70
K_{VS} -Wert (1/s)	6,79	21,4
Schleppfehler, $\alpha=19,5^\circ$, $v_B=90\text{mm/s}$ (mm)	4,8	1,5
f_{45° (Hz)	0,87	4,2

Tabelle 9: Vergleich von verschiedenen Kenngrößen der Sensorregelung im Interpolations- und im Lageregeltakt

7. Flexible Integration multisensorieller Daten

Oftmals soll an einer Robotersteuerung nicht nur ein einzelnes Sensorsystem integriert werden sondern eine Vielzahl von Sensoren. Bisher bestehen aber Defizite bei der Entwicklung von Multisensorsystemen. Folglich werden zunächst kurz die wesentlichen Problemstellungen beleuchtet und eine allgemeine Strukturierung von Multisensorsystemen vorgestellt. Daraus wird der Entwurf eines flexibel konfigurierbaren Systems zur Integration und Fusion multisensorieller Daten abgeleitet und im Anschluß näher beleuchtet. Als Basis zur Integration von multisensoriellen Daten dient das in Kapitel 4 vorgestellte Gesamtsystem zur Sensorintegration.

7.1 Motivation und Defizite der Integration multisensorieller Daten

Zunächst soll eine Abgrenzung der Begriffe „Integration“ und „Fusion“ von Multisensorinformationen nach Luo [85] vorgenommen werden. Danach beschreibt die Multisensorintegration den synergetischen Gebrauch von Multisensorinformationen durch ein System, um die Erfüllung einer Aufgabe zu unterstützen. Im Gegensatz dazu umfaßt die Fusion multisensorieller Daten den eigentlichen Verschmelzungsprozeß, der durch eine Verknüpfung der Einzelinformationen zu einer resultierenden Information charakterisiert ist. Diese Definition hat hauptsächlich zum Ziel, die allgemeinen Gesichtspunkte der Multisensorintegration wie die System- und Steuerungsstruktur von den sehr spezifischen Eigenschaften der Fusionsmechanismen abzugrenzen [85].

Multisensorsysteme werden eingesetzt, um durch Redundanz (mehrere, gleichartige Sensoren) oder durch Diversifikation (mehrere Sensoren mit unterschiedlichen Meßergebnissen) die Zuverlässigkeit und Genauigkeit bei der Aufnahme von Meßgrößen zu erhöhen. Jeder einzelne Sensor dieses Systems liefert eine Teilmeldung, aus der der gesamte nutzbare Informationsgehalt durch eine geeignete Verknüpfung der Einzelinformationen gewonnen wird. Die potentiellen Vorteile der Integration und Fusion von Informationen mehrerer Sensoren liegen vor allem darin, daß die Informationen in kürzerer Zeit genauer und billiger erworben werden können. Die Bedeutung und der potentielle Nutzen der Integration und Fusion von multisensoriellen Daten läßt sich folglich in die vier wesentlichen, folgenden Teilaspekte untergliedern [57]:

- ❑ **Redundante Information** wird von einer Gruppe von gleichartigen Sensoren geliefert, wobei jeder Sensor möglicherweise mit unterschiedlicher Genauigkeit das gleiche Merkmal in der Umgebung mißt. Die Fusion dieser redundanten Informationen kann durch einen Datenvergleich den Unsicherheitsfaktor reduzieren, der bei realen Messungen durch mögliche physikalische Ungenauigkeiten des Aufbaus und durch Störungen immer zu berücksichtigen ist [8, 52].

- ❑ **Ergänzende Informationen** verschiedener Sensoren ermöglichen das Erfassen von Merkmalen in der Arbeitsumgebung, die von einem einzelnen Sensor nicht wahrgenommen werden können. Dabei reicht es aus, daß jeder Sensor nur einen Teil dieses Merkmals mißt [15].
- ❑ Im Vergleich zu Einzelsensorsystemen können mehrere Sensoren in **kürzerer Zeit** die benötigte Information liefern. Dies begründet sich zunächst in einer höheren, aktuellen Meßgeschwindigkeit jedes einzelnen Sensors und der möglichen gleichzeitigen, parallelen Ausführung der Messungen durch die einzelnen Sensoren [57].
- ❑ Bei Multisensorsystemen spricht man oft von **kostengünstiger Information**, falls man die Information mit einem geringeren Kostenaufwand erwerben kann [57]. Für komplexe Aufgaben stehen zwar manchmal hochspezialisierte und extra entwickelte, einzelne Sensorsysteme zur Verfügung, jedoch fallen hier sehr hohe Investitionskosten an, die oft durch einen kostengünstigen Multisensorikansatz vermieden werden könnten. Dazu werden einzelne, kostengünstige Standardsensoren verwendet, deren beschränkte Informationsinhalte zur erforderlichen Nutzinformation fusioniert werden.

Bisher wurden eine Vielzahl von unterschiedlichen Konzepten von Multisensorsystemen vorgeschlagen, jedoch eignen sich nur wenige Systeme, um multisensorielle Information in Echtzeit in die entsprechende Steuerung zu integrieren [57]. Die oftmals verwendeten, komplexen und kostenintensiven Sensorsysteme, wie z.B. Bildverarbeitungssysteme oder taktile Feldsensoren, erschweren zusätzlich eine Echtzeitintegration von Daten in die unteren Hierarchieebenen von Steuerungen.

Außerdem sind komplexe Sensorsysteme in industriellen Umgebungen nur beschränkt einsetzbar, da die Investitionskosten durchaus die Größenordnung eines Industrieroboters annehmen können. Folglich sollen insbesondere Multisensorsysteme entwickelt werden, die aus preiswerten Sensoren bestehen, die über keine bzw. eine geringe, eigene Intelligenz verfügen [44].

Eine Vielzahl der entworfenen Multisensorsysteme stellen spezifische Lösungen für sehr spezielle Aufgabenstellungen dar, die insbesondere durch eine geringe Anpaßflexibilität und dem Fehlen einer offenen Systemarchitektur charakterisiert sind [15]. Vor allen Dingen bestehen innerhalb eines Multisensorsystems bisher kaum einheitliche Vorgehensweisen bei der Sensordatenverarbeitung und -fusion, so daß die entsprechend erforderlichen Programmodule unter jeweils hohem Kostenaufwand spezifisch für die einzelnen Anwendungen erstellt werden müssen. [56]. Folglich erfordert ein Multisensorsystem eine weitgehende Unterstützung bei der Konfiguration und dem Aufbau des Systems. Der Aufwand für die Installation, das „Rüsten“ und „Umrüsten“ des Multisensorsystems sollte möglichst gering sein und einfach und schnell vorgenommen werden können [44].

7.2 Allgemeine Strukturierung von Multisensorsystemen

Zunächst wird das allgemeine Modell eines Systems zur Integration und Fusion von multisensoriellen Daten vorgestellt, an dem die wesentlichen Teilaspekte eines Entwurfs von Multisensorsystemen diskutiert werden (Bild 87).

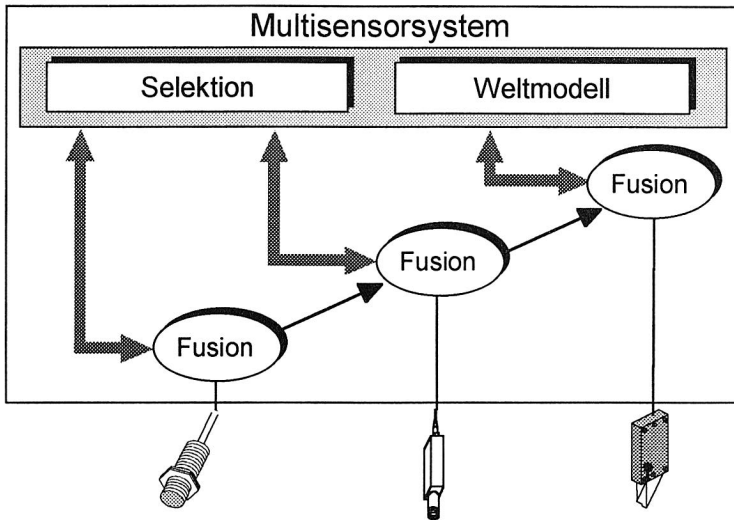


Bild 87: Allgemeines Modell zur Multisensorintegration und Fusion [85]

Die gemessenen Daten der Sensoren werden jeweils entsprechend ihrem Repräsentationsniveau in den Fusionsknoten der unterschiedlichen Ebenen zu neuen Daten mit höherem Repräsentationsniveau zusammengefaßt. Dieser Vorgang läßt sich sukzessive in höheren Abstraktions- und Merkmalsebenen fortsetzen, bis die gewünschte Zielinformation erzeugt ist. Die Form der Datenfusion soll zunächst nicht näher festgelegt werden und ist meist applikationsabhängig.

Die Steuerung des Integrationsprozesses übernehmen die Integrationskomponenten „Sensorselektion“ und „Weltmodell“. Die Aufgabe der Sensorselektion ist die Auswahl und Aktivierung des jeweils geeigneten Sensor- und Fusionsknotens zur Lösung des gestellten Problems. Dazu kann sich die Sensorselektion des Weltmodells bedienen, indem die jeweils gewonnenen Sensor- und Fusionsdaten abgelegt sind. In den meisten Multisensorintegrationsprozessen steigt das Repräsentationsniveau an, während sich die Information von unten nach oben durch die abgebildete Struktur bewegt (Bild 87). Dabei werden auf dem niedrigsten Niveau die Rohdaten der Sensoren noch in Signale umgewandelt, doch nach einigen Fusionschritten werden diese Signale in immer abstraktere, numerische oder symbolische Darstellungsformen transformiert.

Die Integration mehrerer Sensoren kann an Hand von unterschiedlichen Integrationsstrategien und Strukturierungsschemen erfolgen, wobei im folgenden die gebräuchlichsten Strategien kurz beschrieben sind (Bild 88).

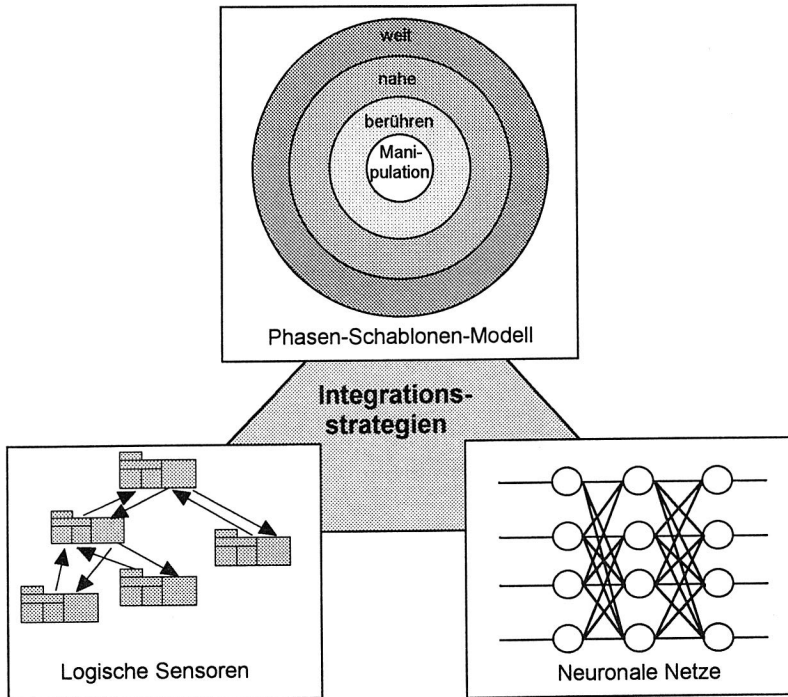


Bild 88: Strategien zur Integration von multisensoriellen Daten

Phasen-Schablonen-Modell:

Luo und Lin [86] haben ein allgemeines Integrationsmuster entwickelt, das auf vier temporal getrennten Phasen im Prozeß zum Informationserwerb basiert. Abhängig von der jeweiligen Entfernung vom zu vermessenden Objekt und ihrer Zugehörigkeit zu einer Phase kommen verschiedene Sensoren zum Einsatz. Die gemessenen Daten der jeweiligen Phase werden in getrennten Schablonen abgelegt und mit den Daten der außerhalb liegenden Schablonen fusioniert.

Neuronale Netze:

Neuronale Netze liefern einen sehr guten Formalismus, um den Multisensorintegrationsprozeß zu modellieren. Die einzelnen Neuronen können trainiert werden, Sensorinformationen dar-

zustellen und durch einen „assoziativen Rückruf“ komplexe Neuronenkombinationen auf Sensorsignale zu aktivieren [85].

Logische Sensoren:

Ein logischer Sensor stellt eine abstrakte Definition eines Sensors dar. Durch die abstrakte Definition des logischen Sensors können unnötige Details über den aktuellen, physikalischen Sensor ausgeklammert und nur sein funktionaler Nutzen betrachtet werden. Diese Eigenschaft des logischen Sensors erhöht die Portabilität und Anpaßbarkeit des Systems an sich verändernde Aufgabenstellungen. Unter dem Einsatz des Strukturelements logischer Sensor lassen sich hierarchisch strukturierte Netze aufbauen [85].

Zur Steuerung des Integrationsprozesses muß an verschiedensten Stellen des Multisensorsystems entschieden werden, welcher Sensor zur optimalen Lösung der augenblicklichen Problemstellung eingesetzt werden soll. Diese Aufgabe wird im allgemeinen Modell von der Sensorselektion wahrgenommen, wobei prinzipiell zwei Sensorselektionsstrategien zu unterscheiden sind [85].

Bei der **Prä-Selektion** findet die Sensorauswahl zur optimalen Lösung der Aufgabenstellung während des Programmentwurfs oder der Programminitialisierung statt. Im Gegensatz dazu erfolgt bei der **Real-Time-Selektion** die Auswahl des zweckmäßigsten Sensors zur Laufzeit des Systems unter der Beachtung von sich ändernden System- oder Umgebungsbedingungen.

Zur Fusion von Sensordaten sind eine Vielzahl von Verfahren bekannt. Da die Fusionsmethoden oftmals stark applikationsabhängig sind und für den Entwurf eines Multisensorsystems eine untergeordnete Rolle spielen, sei hier auf die einschlägige Literatur verwiesen [34, 84, 85, 133].

7.3 Entwurf eines Systems zur Integration multisensorieller Daten

Die Anforderungen an ein Multisensorsystem münden in ein Konzept für ein flexibel konfigurierbares System zur Integration multisensorieller Daten. Im folgenden wird die Struktur, die Datenflüsse und die Schnittstellen des Systems vorgestellt und abschließend die besonderen Aspekte der Einbindung in ein Echtzeitsystem näher beleuchtet.

7.3.1 Anforderungsprofil eines Multisensorsystems

Die in Kapitel 7.1 abgeleiteten Defizite von Multisensorsystemen fordern Systeme, die es erlauben, Sensoranwendungen kostengünstig zu entwerfen. Sie sind insbesondere so zu strukturieren, daß sie in modifizierten Anwendungen einfach angepaßt und bestehende Module weitgehend wiederverwendet werden können. Dazu ist eine hohe Anpaß- und Änderungsflexibilität der Systeme notwendig [128]. Eine Vielzahl von Anwendungen erfordern Multisen-

sorsysteme, die eine Integration und Fusion von multisensoriellen Daten innerhalb eines fest vorgegebenen Zeitraums oder zyklisch mit einer definierten Zykluszeit durchführen. Ein harter Echtzeitbetrieb des Multisensorsystems ist daher unumgänglich. Dies gilt besonders für Sensorsysteme, die mit einfachen Sensoren mit geringer Intelligenz arbeiten, und bevorzugt Aufgaben im Bereich der Sensorregelung von Industrierobotern wahrnehmen. Der Entwurf eines Multisensorsystems erfordert,

- ☐ eine modulare Strukturierung des Systems, die Änderungen am System auf die Modifikation weniger Teilmodule des Systems begrenzt.
- ☐ daß die Anpaßflexibilität des Systems weitgehend durch eine geeignete Konfiguration des Systems erzielt wird.
- ☐ Schnittstellen zwischen den Modulen des Multisensorsystems, die den Austausch von Modulen erleichtern.
- ☐ in den unteren Ebenen des Systems eine zeitlich sehr effiziente und deterministische Integration und Fusion von Sensordaten.

7.3.2 Hierarchische Strukturierung

Zusätzlich ist es oftmals zweckmäßig, das Multisensorsystem um Module zur Rückführung von Sensordaten auf die aktiven Geräte einer Fertigungsanlage zu ergänzen. Für Aufgaben im Bereich der Sensorregelung von Industrierobotern ist die Rückführung der Sensorsignale eine zwingende Voraussetzung zur Erfüllung der Aufgabenstellung. Demzufolge soll hier die Rückführung zusätzlich betrachtet werden. Die Forderungen der Multisensorintegration gelten weitgehend in analoger Form für die Rückführung der Sensorinformation an die aktiven Geräte der Roboteranlage.

Zur Strukturierung von Multisensorsystemen werden zunächst die Formen der Rückführung von Sensorsignalen an Roboteranlagen analysiert (Bild 89). Danach treten zwei Arten von Rückführungen auf. Die 1. Form ist dadurch charakterisiert, daß die Bewegungsbahn des Roboters durch multisensorielle Daten direkt on line während der Bewegung beeinflusst wird. Die Steuerung von Aktionen ist das Ziel der anderen Form von Rückführungen. Bei beiden Rückführarten muß grundsätzlich eine Erfassung und Vorverarbeitung der Sensordaten durch das Multisensorsystem erfolgen. Die direkten Beeinflussung der Bewegungsbahn des Roboters setzt meist eine Fusion von Sensordaten und die Bestimmung von Merkmalen der Sensordaten voraus. Dies kann z.B. die Bestimmung eines Abstandsvektors zur Sensorregelung eines Roboters beim Schutzgasschweißen sein. Zur Steuerung von Aktionen am Roboter muß das Sensorsystem vollständige Meßaufgaben mit einer Klassifikation der Merkmale des Sensorsignals durchführen. Das Ergebnis beeinflusst dann den weiteren Ablauf der Anwendung.

Das Anforderungsprofil des Multisensorsystems und eine Analyse von Meßwerterfassung- und Positionieraufgaben führte bei der Konzeption des Multisensorsystems zu einer modularen,

hierarchischen Struktur (Bild 90). Entsprechend den verschiedenen Arten von Rückführungen bei Sensor-Roboter-Systemen unterteilen sich die drei Hauptmodule in drei Strukturebenen [44]. Die Ebene 1, die hier als Sensor/Aktor-Ebene bezeichnet wird, umfaßt im Bereich der Aktorik die Ausführung von Bewegungen. Die Sensorwerterfassung und -vorverarbeitung, die die Erfassung von Sensordaten und deren Normieren und Filterung beinhaltet, ist die zentrale Aufgabe der Ebene 1 im Bereich der Sensorik.

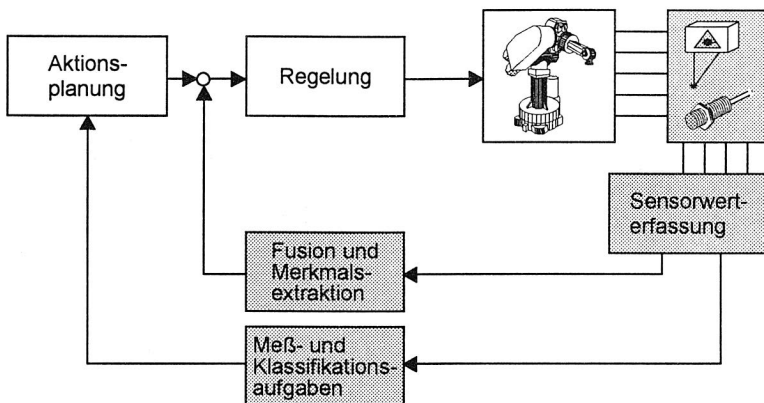


Bild 89: Prinzipielle Struktur der Rückführung bei multisensoriellen Sensor-Roboter-Systemen

In der Ebene 2, die Ebene der Merkmals- und Bewegungsgenerierung, werden die Bewegungen errechnet, Sensordaten fusioniert und die erforderlichen Merkmale der Sensorsignale extrahiert. Die Bewegungsplanung kann z.B. mit Hilfe eines Bahninterpolators oder auch durch den Regler einer Sensorregelung erfolgen. Die Merkmalsgenerierung umfaßt dabei die Fusion von mehreren Sensorsignalen zu einem resultierenden Nutzsignal und der Extraktion eines Merkmals. In der Weise können z.B. Abstandsvektoren relativ zu einer Oberfläche ermittelt werden.

Die 3. Ebene übernimmt die Aufgabe der Koordinierung und Steuerung des Systems. Dazu ist eine Aufgabenbeschreibung hinterlegt, die festlegt, in welcher Reihenfolge und mit welchen Abhängigkeiten Elementaraktionen, Meß- und Klassifizierungsaufgaben ausgeführt werden. Die Elementaraktionen steuern die Bewegungsgenerierung und lösen Meß- und Klassifizierungsaufgaben aus. Nachdem die erforderlichen Meßergebnisse vorliegen, können weitere Aktionen ausgelöst werden.

Das Weltmodell dient zum Austausch von Daten zwischen dem Aktorik- und dem Sensorik-Bereich innerhalb der jeweiligen Hierarchieebene. Insbesondere werden Sensordaten und Meß-

ergebnisse zum Aktorik-Bereich transferiert. Außerdem stehen Parameter, die von den Modulen des Multisensorsystems benötigt werden, im Weltmodell zur Verfügung.

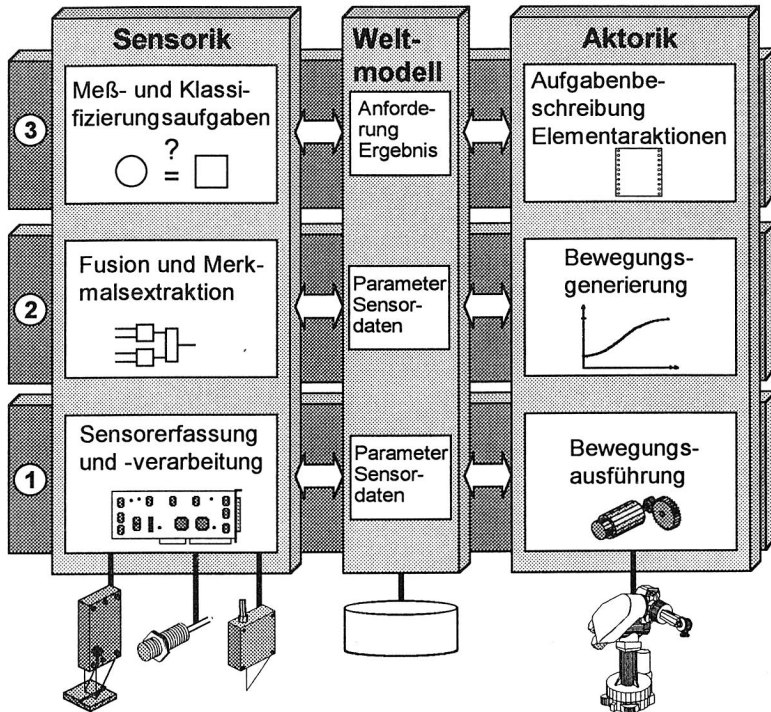


Bild 90: Modulare, hierarchische Struktur des Multisensorsystems

Die Übertragung von Daten innerhalb einer Ebene des Systems ermöglicht einen schnellen Transfer von Daten gleichen Abstraktionsniveaus zwischen der Sensorik und der Aktorik. Bei sensorgeregelten Anwendungen müssen die fusionierten Sensordaten keinen Umweg über höhere Hierarchiestufen nehmen, sondern können direkt in der Ebene übertragen werden.

7.3.3 Schnittstellen und Datenflüsse

Um den Austausch von Modulen des Multisensorsystems zu erleichtern, müssen die Schnittstellen zwischen den einzelnen Ebenen spezifiziert und die Struktur des Datenflusses weiter detailliert werden. Die Ebene 1 im Bereich Sensorik ist durch eine lineare Struktur des Datenflusses charakterisiert, d.h. jedes Ausgangssignal dieser Ebene wird von genau einem Eingangssignal beeinflusst. Innerhalb der Ebene ist je ein Block zur Normierung und Filterung der Sensordaten vorgesehen. Die erforderlichen Parameter der Filter- und Normierungsfunktionen

können dazu dem Weltmodell entnommen werden. Die Schnittstelle zwischen der Ebene 1 und 2 stellen die Minikanäle dar, wobei ein Minikanal jeweils die vorverarbeiteten Daten eines Sensors liefert. Folglich ist ein Minikanal durch den Sensor und die eingesetzten Filter- und Normierungsfunktionen charakterisiert. Von höheren Ebenen des Systems aus betrachtet, stellen die Minikanäle eine Abstraktion der erfaßten Sensorsignale dar, da insbesondere Details wie die verwendeten Sensorverarbeitungsroutinen für die weitere Bearbeitung unerheblich sind. Die aktuellen Daten der Minikanäle stehen außerdem über das Weltmodell der Aktorik zur Verfügung.

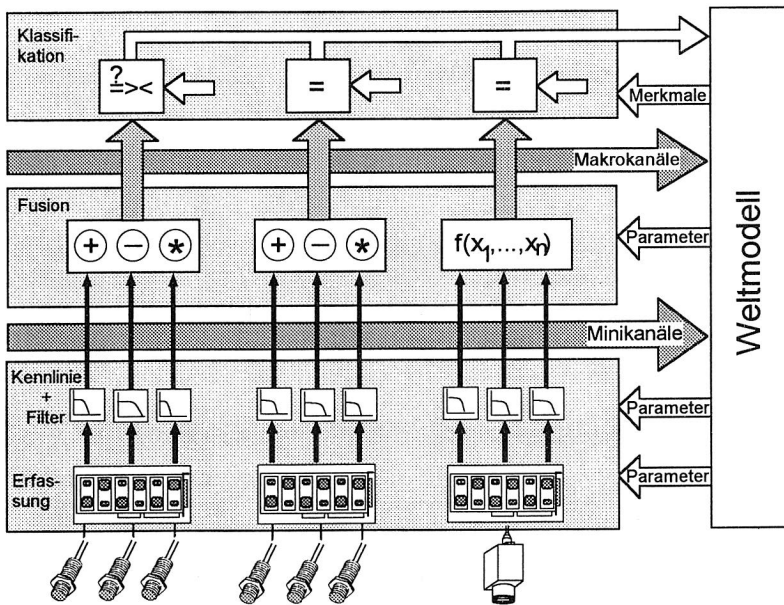


Bild 91: Schnittstellen und Datenflußstruktur des Multisensorsystems

Die Ebene 2 besitzt eine baumartige Struktur des Datenflusses, d.h. mehrere Minikanäle beeinflussen je einen Ausgang der Ebene 2. An den Knotenstellen des Baums werden die Daten der Minikanäle fusioniert, wobei unterschiedlich komplexe Operatoren eingesetzt werden. Die Ergebnisse der Verschmelzungsoperationen werden über die Makrokanäle an die Ebene 3 übergeben. Die Makrokanäle stellen eine weitere Abstraktionsebene im System dar, da insbesondere Details wie die eingesetzten Fusionsoperatoren für die weitere Bearbeitung in der Ebene 3 ohne Bedeutung sind. Zusätzlich stehen die Daten der Makrokanäle im Weltmodell der Aktorik zur Verfügung.

Die Ebene 3 hat die Aufgabe Meßaufträge, die von der Aktorik vorgegeben werden, durchzuführen und dazu die Aktionen der unterlagerten Schichten zu steuern. Die Aktivierung oder die Terminierung der zyklischen Arbeitsweise der unterlagerten Ebenen sind typische Steueranweisungen. Außerdem werden in der Ebene Klassifikationsaufgaben ausgeführt, die z.B. durch einen Vergleich von im Weltmodell hinterlegten Referenzmerkmalen mit den Daten des Makrokanals gekennzeichnet sind. Das Ergebnis eines Klassifikationsauftrags wird an das Weltmodell transferiert, so daß es der aufrufenden Elementaraktion zur Verfügung steht.

Da der Bereich der Aktorik eine sehr ähnliche Datenfluß- und Schnittstellenstruktur aufweist, soll sie im weiteren nicht detailliert werden.

7.3.4 Modellierung mit logischen Einheiten

Zur Strukturierung und Modellierung des Multisensorsystems eignen sich von den in Kapitel 7.2 vorgestellten Verfahren besonders das Strukturelement des „Logischen Sensors“. Ein logischer Sensor wird über seinen Namen und entsprechende Kommandobefehle angesprochen. Er ist grundsätzlich untergliedert in kommando- und datenverarbeitende Einheiten. Die ankommenden Kommandobefehle werden interpretiert, und resultierende Befehle werden an die logischen Sensoren in niederen Ebenen weitergesendet, bzw. an die eigene Datenverarbeitungseinheit weitergegeben. Die eintreffenden Sensorsignale werden mit Hilfe der in den logischen Sensoren hinterlegten Programmen verarbeitet und von der Sensorselektion am Ausgang bereitgestellt [85].

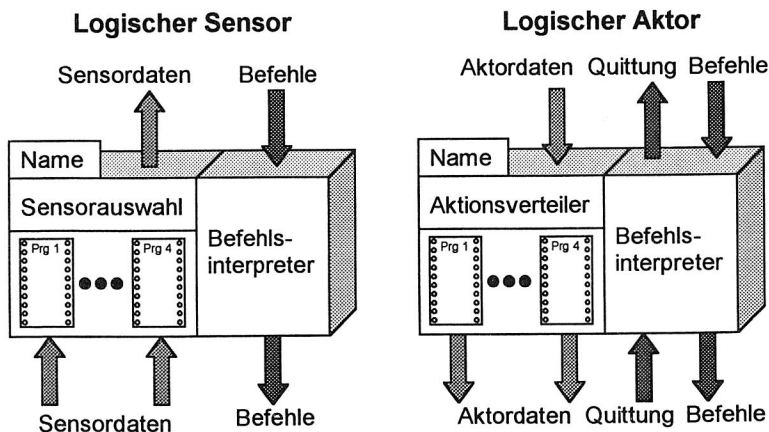


Bild 92: Allgemeiner Aufbau der Strukturelemente „Logischer Sensor und Aktor“

In Ergänzung des Strukturelements „Logischer Sensor“ wird das Strukturelement „Logischer Aktor“ eingeführt, der die Rückführung der Sensorinformation zu den aktiven Komponenten

der Fertigungsanlage modelliert. Logische Aktoren werden über Namen angesprochen und übernehmen die Ergebnisse logischer Sensoren oder Befehle und Daten von überlagerten, logischen Aktoren. Zur Bearbeitung von Befehlen steht ein Befehlsinterpretierer zur Verfügung, der wiederum Befehle bzw. Daten an logische Aktoren oder logische Sensoren versendet. Im Unterschied zu den logischen Sensoren wird die Ausführung von Befehlen bei den logischen Aktoren der überlagerten Schicht mit einer Quittung mitgeteilt. Dies ist erforderlich, da im Gegensatz zu den logischen Sensoren eine Quittierung nicht implizit durch die Übertragung von Daten an die höheren Schichten erfolgt.

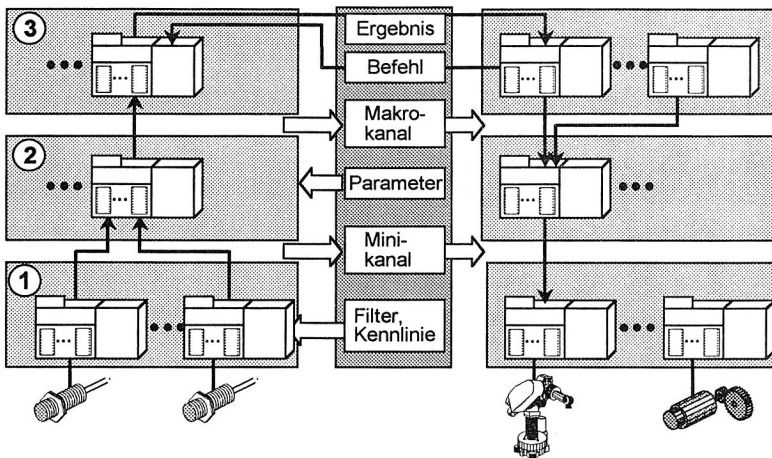


Bild 93: Modell einer Multisensoranwendung unter Verwendung von logischen Sensoren und Aktoren

Eine Multisensorapplikation und deren Einordnung in das hierarchisch aufgebaute Multisensorsystem läßt sich mit diesen beiden Strukturelementen vollständig modellieren. Die Schnittstellenkomponente „Minikanal“ bzw. „Makrokanal“ stellen dabei jeweils den Datenausgang eines logischen Sensors dar. Je Mini- bzw. Makrokanal ergibt sich folglich ein logischer Sensor, der die innerhalb der Ebene 1 bzw. der Ebene 2 erforderlichen Operationen zur Bestimmung der Kanalinformation nachbildet. Die logischen Sensoren der Ebene 1 und 3 weisen die Besonderheit auf, daß sie jeweils nur über einen Dateneingang und -ausgang verfügen (Bild 93). In der selben Weise kann die Aktorik mit Hilfe von logischen Aktoren modelliert werden. Faßt man alle logischen Einheiten einer Anwendung zu einer Knotenmenge zusammen und repräsentieren die von allen logischen Einheiten ausgehenden Auftragsbeziehungen eine Kantenmenge, so erhält man einen gerichteten Graph der Anwendung. Dieser Graph ist insbesondere ein qualifiziertes Hilfsmittel zur Konfiguration und Parametrierung des Multisensorsystems, das in Kapitel 7.4 vorgestellt wird.

Die Aufgabenbeschreibung der Ebene 3 ist in einem logischen Aktor hinterlegt, der den Gesamt Ablauf der Anwendung steuert. Die Struktur des Systems und der Ablauf wird während der Initialisierung statisch festgelegt und ändert sich nicht mehr während der Laufzeit. Diese Festlegung ist eine Grundvoraussetzung, um ein zeitlich deterministisches Verhalten des Multisensorsystems sicherzustellen. Die statische Struktur des Multisensorsystems hat zur Folge, daß sich das Weltmodell einfach strukturieren läßt und während der Initialisierung des Systems automatisch aufgebaut wird. Außerdem erfolgt die Auswahl des zweckmäßigsten Sensors zur Lösung des Meßproblems off line während der Konfiguration des Multisensorsystem. Insofern kommt hier eine Prä-Selektionsstrategie zum Einsatz.

7.3.5 Einbindung des Systems in ein Echtzeitsystem

Sensorapplikationen erfordern meist ein zeitlich deterministisches Verhalten. Folglich muß ein Multisensorsystem in ein zweckmäßiges Echtzeitsystem integriert werden. Auf der Basis der in Kapitel 4 entworfenen und implementierten, echtzeitfähigen Sensorsteuerung SC, die eine Bahnadaptation eines Roboters ermöglicht, wurde das vorgestellte Multisensorsystem prototypisch umgesetzt.

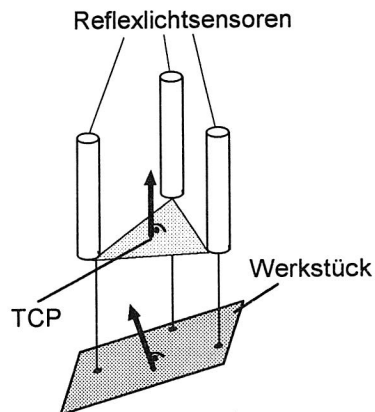
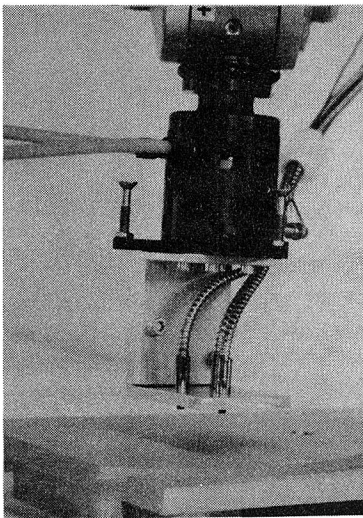


Bild 94: Orientierungsbestimmung des TCPs mit Reflexlichtsensoren

Am Beispiel der in Bild 94 dargestellten Orientierungsführung des Werkzeugbezugspunkts (TCP), die mit drei abstandmessenden Sensoren relativ zu einer Werkstückoberfläche erfolgt,

soll hier die Einbettung des skizzierten Multisensorsystems in das Echtzeitsystem der Sensorsteuerung SC gezeigt werden (Bild 95).

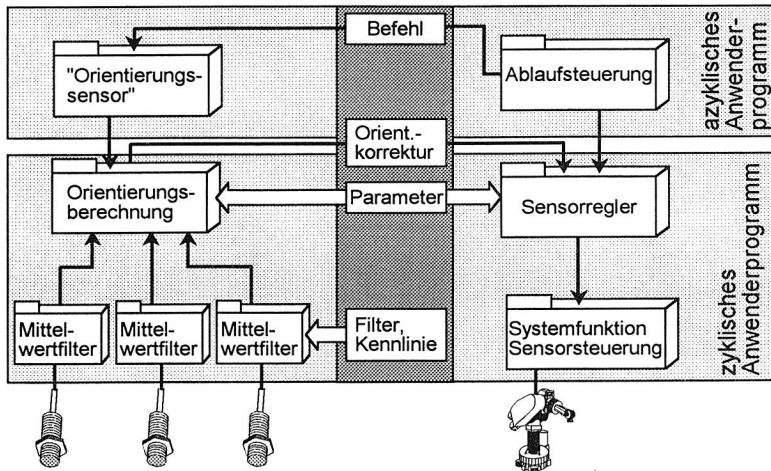


Bild 95: Eingliederung des Multisensorsystems in das Echtzeitsystem der Sensorsteuerung SC

Auf Grund der oftmals erforderlichen, zyklischen Bearbeitung der Module der Ebene 1 und 2 des Multisensorsystems ist eine taktgetriebene Arbeitsweise gewählt worden. Folglich konnten die Ebenen 1 und 2 in ein zyklisches Anwenderprogramm der Sensorsteuerung SC integriert werden, womit ein zeitlich deterministisches Verhalten der Ebenen 1 und 2 sichergestellt ist. Im Gegensatz dazu weisen die Module der Ebene 3 des Multisensorsystems eine ereignisgesteuerte Struktur auf, die sich in ein azyklisches Anwenderprogramm eingliedern läßt.

7.4 Flexible Konfiguration des Multisensorsystems

In Kapitel 7.1 wurde als Anforderung an ein Multisensorsystem eine einfache Anpaßbarkeit an sich ändernde Anwendungen genannt. Da der Aufwand für Änderungen am Softwaresystem möglichst gering sein soll, ist es erforderlich, einfache und anwendungsorientierte Beschreibungsmechanismen zur Konfiguration und Parametrierung des Multisensorsystems zur Verfügung zu stellen. Die in Kapitel 7.3 entworfenen Strukturen und Datenschnittstellen ermöglichen eine flexible Konfiguration der Verbindungstopologie der Verarbeitungsroutinen der drei Ebenen, der Verarbeitungsmethoden selbst und deren Parameter. Demzufolge muß zunächst die betrachtete Anwendung hinsichtlich der erforderlichen Verbindungstopologie der drei Ebenen analysiert werden. In einem 2. Schritt ist es notwendig, die einzelnen Verarbeitungsooperatoren und deren Parameter zu spezifizieren. Die Beschreibung der Verbindungstopologie der Anwendung und der verwendeten Verarbeitungsooperatoren und -parameter erfolgt mit einer

speziell entwickelten, problemangepaßten Beschreibungssprache in Backus-Naur-Form und wird in vorgegebenen Konfigurationsdateien spezifiziert (Bild 96).

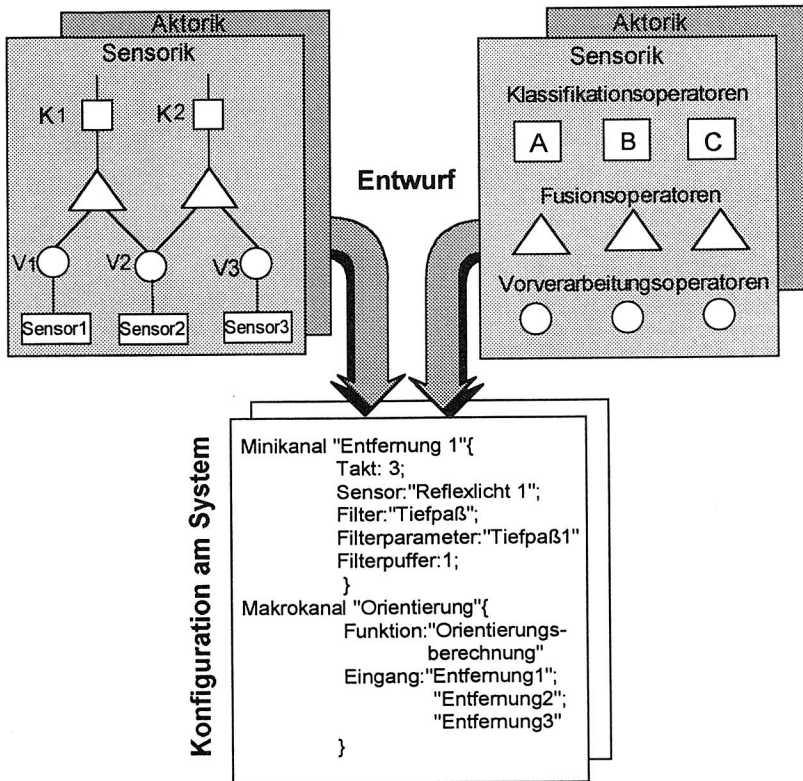


Bild 96: Flexible Konfiguration der Verbindungstopologie und Verarbeitungsooperatoren des Multisensorsystems

Die Einleseroutinen, die mit Hilfe der vom Betriebssystem UNIX bekannten Werkzeuge „lex“ und „yacc“ erzeugt werden, lesen die Konfigurationsdateien ein [88]. Die Konfigurationsdatei kann in definierter Grammatik auf korrekte Syntax überprüft werden, und der Inhalt der Dateien wird in Hauptspeicherstrukturen dem Multisensorsystem zur Verfügung gestellt. Zusammen mit den Systembibliotheken der Sensorsteuerung SC und der eingelesenen Konfiguration läßt sich ein Multisensorsystem erzeugen, das sich in der gewünschten Form konfiguriert. Damit kann das System einfach und effizient für die verschiedenen Aufgaben konfiguriert und aufgebaut werden. Eine einfache und schnelle Umrüstung des Systems bei sich ändernder Aufgabenstellung ist damit möglich.

8. Rechnergestützte Planung und Auswahl von Sensoren

Vor der eigentlichen, systemtechnischen Integration von Sensoren in Fertigungsanlagen müssen zunächst qualifizierte Sensoren zur Lösung der jeweiligen Aufgabenstellung ausgewählt werden. Insbesondere bei der Planung von Sensoren für Aufgaben der Sensorregelung bestehen Defizite, die durch einen Mangel an rechnergestützten Planungssystemen gekennzeichnet sind. Eine Analyse der aktuellen Problemstellungen bei der Auswahl von Sensoren führte zum Entwurf einer Auswahlssystematik, die in ein rechnergestütztes Planungssystem umgesetzt wurde, das im Anschluß näher beleuchtet wird.

8.1 Problematik der Sensorauswahl und -planung

Die Auswahl eines qualifizierten Sensors zur Lösung eines Meßproblems ist eine komplexe Aufgabe mit einem nicht zu unterschätzenden Investitionsrisiko [118, 146, 147]. Meist stehen verschiedenste Sensoren, die sich oft der unterschiedlichsten Meßprinzipien bedienen, zur Verfügung, um eine Meßgröße zu erfassen. Zusätzlich werden Sensoren von einer großen Zahl von Herstellern angeboten, deren Kataloge uneinheitlich strukturiert sind. Das „Durchsehen“ der Herstellerkataloge ist zudem sehr mühsam und zeitaufwendig, so daß die Suche oftmals auf einige wenige Herstellerkataloge und auf meist vorher bereits bekannte Sensoren beschränkt bleibt [87].

Bisher existieren außerdem keine einheitlichen Beurteilungskriterien für Sensoren und deren Auswahl. In den Herstellerkatalogen sind demnach die unterschiedlichsten Kenngrößen der Sensoren quantifiziert [87]. Damit ergeben sich insbesondere für einen unerfahrenen Projektierer große Probleme, einen qualifizierten Sensor für die Meßaufgabe auszuwählen.

Bislang bestehende Entwürfe zur Sensorauswahl beschränken sich auf die Darstellung eines abstrakten Schemas zu einer strukturierten Vorgehensweise bei der Sensorplanung [16, 40]. Andererseits wurden Sensordatenbanken zur Sensorplanung entworfen und implementiert. Der Auswahlvorgang wird dabei aber nur durch einen Suchvorgang nach einer spezifizierten Kenngröße des Sensors in einer relationalen Sensordatenbank unterstützt [87, 147]. Die Berücksichtigung von vorhandenem Erfahrungswissen während des Auswahlvorgangs ist dabei kaum möglich. Eine Unterstützung des Auswahlvorgangs unter Verwendung von Expertenwissen verspricht aber eine Qualifizierung des Planungsvorgangs.

8.2 Systematik zur Auswahl von Sensoren

Zunächst soll eine systematische Methode zur Sensorplanung und -auswahl vorgestellt werden, die sich an die Systematik von Classe [16] anlehnt. Der Auswahlvorgang weist eine durchgängig herstellerneutrale Vorgehensweise auf und kann insbesondere in der Auswahlphase durch geeignete, noch zu entwerfende, rechnergestützte Hilfsmittel erleichtert werden (Bild 97).

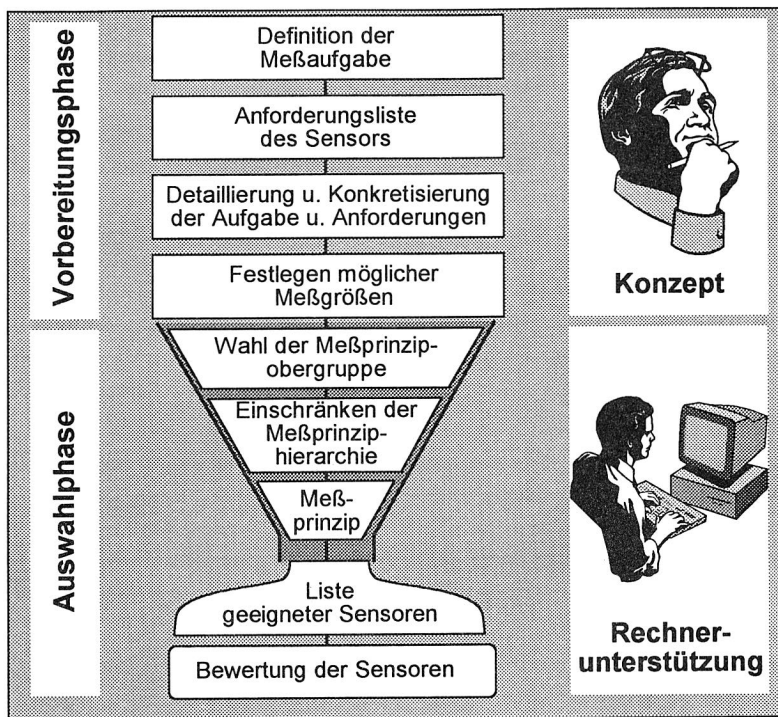


Bild 97: Modell der rechnergestützten Sensorauswahl

Das Auswahlverfahren ist grundsätzlich in die Vorbereitungsphase, die zur Detaillierung der Anforderungen an die Sensoren und zur Definition möglicher zu messender Meßgrößen führt, und in die Auswahlphase untergliedert. Während der Auswahlphase sollen an Hand der konkreten Meßgröße und den festgelegten Anforderungen die geeignetsten Sensoren bestimmt werden.

In der Vorbereitungsphase wird zunächst die Meßaufgabe definiert. Dazu muß die Aufgabenstellung hinsichtlich der erforderlichen Sensorfunktionen und der Umgebungsbedingungen, die besonders zu berücksichtigen sind, analysiert werden. Zur Systematisierung können die Anforderungen in einer Anforderungsliste zusammengefaßt werden, um zu vermeiden, daß möglichst keine wichtigen Aspekte des Sensors unberücksichtigt bleiben. Dabei werden die Anforderungen, die unbedingt erfüllt werden müssen, im Gegensatz zu Anforderungen mit Wunschcharakter extra gekennzeichnet. Die Aufgabenstellung und die Anforderungen an den Sensor sind weiter zu detaillieren und zu konkretisieren mit dem Ziel, mögliche Meßgrößen, die der Lösung der Meßaufgabe dienen, festzulegen. In der Vorbereitungsphase ist darauf zu achten, daß die Aufgabendefinition nicht von vornherein Lösungswege vorgibt, die andere günstige Lösungen bereits im Vorfeld unberücksichtigt läßt.

Zur Bestimmung des Werts einer Meßgröße können unterschiedlichste Meßprinzipien verwendet werden. In vielen Fällen ist aber gerade das Meßprinzip ausschlaggebend für den Erfolg in der speziellen Meßaufgabe. Folglich wird in der Auswahlphase sukzessiv die Zahl möglicher Meßprinzipien an Hand einer Meßprinziphierarchie, die in Kapitel 8.3 vorgestellt wird, eingeschränkt, bis ein geeignetes Meßprinzip gefunden ist. Da die Meßprinziphierarchie eine geeignete Strukturierung des Problems darstellt und die Anforderungen an den Sensor quantitativ spezifiziert sind, kann die Auswahlphase weitgehend rechnergestützt durchlaufen werden. Wurde ein geeignetes Meßprinzip gefunden, so kann eine Liste möglicher Sensoren angegeben werden, die die Anforderungen erfüllt. Dabei erfolgt gleichzeitig ein Übergang vom herstellerneutralen zum herstellerspezifischen Teil der Auswahlsystematik. Schließlich muß ein geeigneter Sensor aus dem Angebot der verschiedenen Hersteller ausgewählt werden.

8.3 Meßprinziphierarchie zur Strukturierung des Auswahlvorgangs

Die Auswahlphase des in Kapitel 8.2 vorgestellten Auswahlvorgangs ist geprägt von einer sukzessiven Konkretisierung geeigneter Meßprinzipien zur Erfüllung der Meßaufgabe. Dieser Vorgang beginnt mit der Spezifikation prinzipiell geeigneter Meßprinzipien zur Erfassung der Meßgröße. Durch eine Detaillierung der Anforderungen des Anwenders wird die Zahl der in Frage kommenden Meßprinzipien Schritt für Schritt eingeschränkt, bis nur mehr das geeignetste Meßprinzip verbleibt. Wurde dieses Meßprinzip bestimmt, so stehen durchwegs einige, qualifizierte Sensoren zur Auswahl.

Eine durchgeführte Analyse, die den Zusammenhang von Sensoren eines Meßprinzips und deren charakteristischen Auswahlparametern wie Meßbereich und Genauigkeit untersuchte, zeigte, daß eine hohe Übereinstimmung zwischen den kennzeichnenden Parametern eines Sensors und dessen Zugehörigkeit zu einem Meßprinzip besteht. Folglich lassen sich die charakteristischen Kenngrößen von Sensoren, die nach dem gleichen Meßprinzip konzipiert sind, in

einem resultierenden Parametervektor zusammenfassen. Der resultierende Parametervektor stellt eine Obermenge und Abstraktion der Parametervektoren der betrachteten Sensoren dar.

Zu jeder Meßgröße läßt sich eine Klasse von Meßprinzipien angeben, die hierarchisch strukturiert werden kann (Bild 98). Innerhalb dieser Meßprinziphierarchie kann je Ebene und zugehörigem Element ein aus der untergeordneten Ebene abstrahierter Parametervektor gebildet werden. In der Folge erhält man eine baumartige Meßprinziphierarchie mit jeweils charakteristischen Parametern für jede Ebene und jedes Element (Bild 98). Die Blätter des Hierarchiebaums stellen die konkreten Sensoren dar. Für die anderen Meßgrößen lassen sich analog die Hierarchiebäume der Meßprinzipien ableiten und mit Parametern versehen.

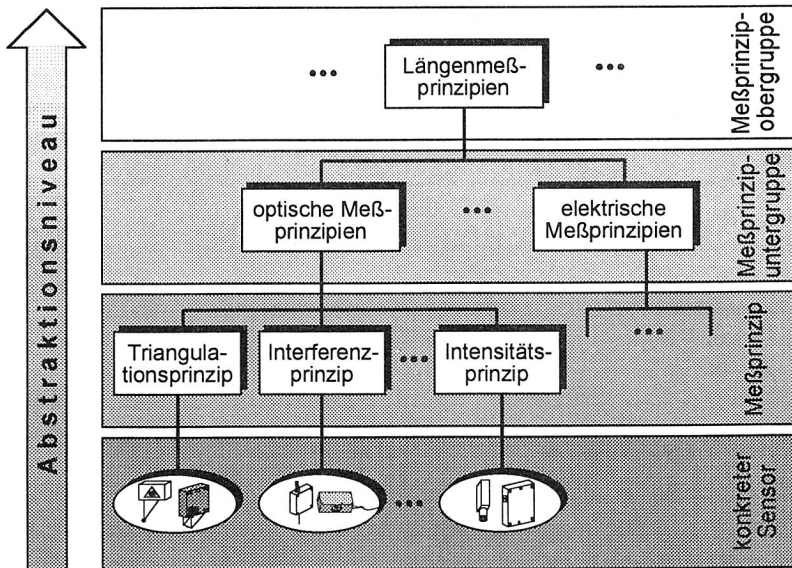


Bild 98: Meßprinziphierarchie zur Strukturierung des Auswahlvorgangs

8.4 Rechnergestütztes Auswahlssystem

Die in Kapitel 8.3 vorgestellte Meßprinziphierarchie zur Strukturierung des Auswahlvorgangs stellt die Basis des rechnergestützten Auswahlsystems dar. Zunächst wird die Struktur des Auswahlsystems vorgestellt und anschließend werden dessen Merkmale näher beleuchtet.

8.4.1 Struktur des Auswahlsystems

Das Auswahlsystem soll den Anwender, der für eine zu leistende Meßaufgabe einen geeigneten Sensor sucht, unter Verwendung des vorgestellten Auswahlverfahrens in einem interaktiven Dialog unterstützen. Dazu ist ein rechnergestütztes Auswahlssystem entstanden, das in die folgenden drei Hauptmodule unterteilt ist:

- ❑ **Anwenderschnittstelle:** Dieses Modul stellt die Mensch-Maschine-Schnittstelle des Auswahlsystems dar und dient zur Spezifikation des Anforderungsprofils eines Sensors durch den Anwender. Dazu wird in einem interaktiven Dialog das Anforderungsprofil sukzessiv konkretisiert.
- ❑ **Entscheidungsfindung:** Das Modul „Entscheidungsfindung“ hat die Aufgabe, daß geeignetste Meßprinzip zur Lösung der Aufgabenstellung auszuwählen. Dazu muß das jeweils spezifizierte Anforderungsprofil des Anwenders mit den in der Wissensbasis abgelegten Meßprinzipprofilen verglichen werden.
- ❑ **Wissensbasis:** In der Wissensbasis, die in Form einer Datenbank organisiert ist, sind die jeweiligen Profile der Meßprinzipien und die Kenngrößen der einzelnen Sensoren an Hand eines Datenmodells abgelegt.

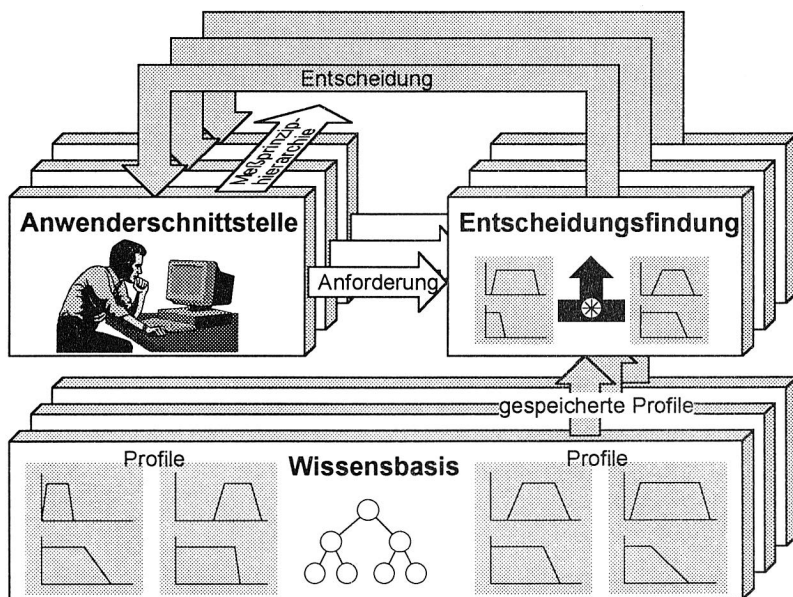


Bild 99: Struktur des Sensorauswahlsystems

Die Hauptmodule weisen eine dreischichtige Struktur auf, die sich an der Struktur der Meßprinziphierarchie orientiert. Dementsprechend ist je Meßprinziphierarchieebene ein Entscheidungsprozeß vorgesehen, wobei zur Entscheidungsfindung die Informationen der Benutzerschnittstelle und der Wissensbasis verwendet werden. Folglich ergibt sich ein dreistufiger, interaktiver Entscheidungsprozeß. Die jeweilige Entscheidung wird dem Anwender mitgeteilt und beeinflußt den weiteren Entscheidungsprozeß. Der erfahrene Benutzer kann aber auch den Suchlauf in einem Schritt durchführen. Das Anforderungsprofil muß dabei detailliert spezifiziert werden.

8.4.2 Entscheidungsfindung

Zunächst muß ein geeignetes Modell zur Abbildung der Anforderungsprofile des Benutzers entwickelt werden, um auf Basis dieser Modellierung einen Entscheidungsprozeß ausführen zu können. Die Anforderungsprofile sind dadurch charakterisiert,

- ☐ daß eine Kenngröße, wie z.B. das Sensorvolumen, einen vorgegebenen Wert nicht überschreitet.
- ☐ daß eine Kenngröße, wie z.B. der Meßbereich, innerhalb des vorgegebenen Bereichs liegt.
- ☐ daß eine Kenngröße, wie z.B. die Meßmethode, eine Ja/Nein-Bedingung erfüllt.

Diese sehr unterschiedlichen Kenngrößen lassen sich aber anschaulich an Hand sogenannter Zugehörigkeitsfunktionen, die aus der Fuzzy-Set-Theorie bekannt sind, abbilden [143].

Die Zugehörigkeitsfunktion einer Kenngröße nimmt dabei den Wert eins an, solange z.B. der typische Meßbereich des Meßprinzips erfüllt ist. Außerhalb des typischen Meßbereichs besitzt die Zugehörigkeitsfunktion den Wert null (Bild 100). Damit kann man außerdem die oftmals nicht exakt formulierbaren Kenngrößengrenzen abbilden, indem die Grenze durch eine mehr oder weniger flach abfallende Flanke beschrieben wird. Mit diesem Modell lassen sich sämtliche, relevanten Kenngrößen sowohl der Anforderungsprofile als auch der Meßprinzipprofile darstellen.

Die Entscheidungsfindung stellt an Hand der Zugehörigkeitsfunktionen fest, welches Meßprinzipprofil am besten mit dem Anforderungsprofil des Anwenders übereinstimmt. Zur Lösung dieser Problemstellung können prinzipiell unterschiedliche Strategien wie neuronale Netze, Expertensysteme und Fuzzy Logik, die aus der wissensbasierten Datenverarbeitung [78] bekannt sind, in Betracht gezogen werden.

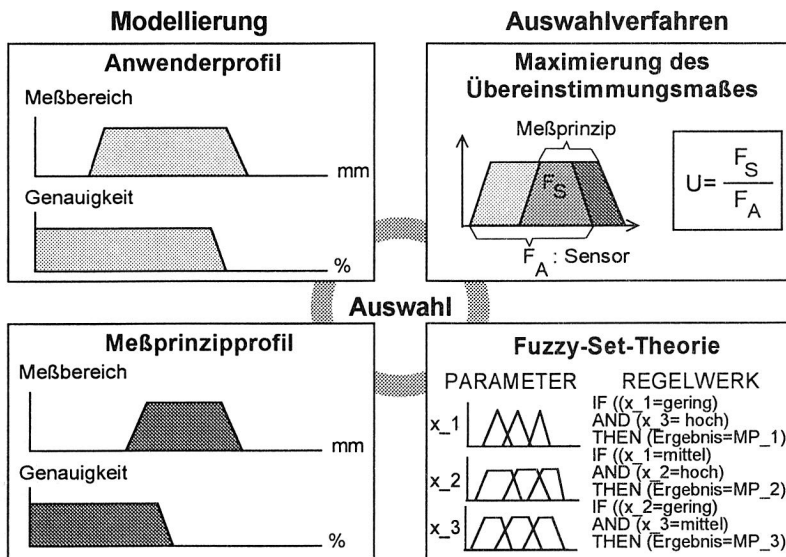


Bild 100: Modellierung und Verfahren zur Entscheidungsfindung

Da die Modellierung der Anwender- und Meßprinzipprofile bereits in einer zur Fuzzy Logik konformen Weise ausgeführt wurde, bietet es sich an, die Entscheidungsfindung unter Anwendung der Fuzzy-Set-Theorie vorzunehmen. Dazu wird folgender Ansatz vorgestellt. Die Anforderungskenngrößen lassen sich als linguistische Variablen auffassen. Beispielsweise kann man die Meßgenauigkeit als Variable mit den drei Werten niedrig, mittel und hoch darstellen. Jedem linguistischen Wert wird eine unscharfe Menge, beschrieben durch eine Zugehörigkeitsfunktion, zugeordnet. In den meisten Fällen sind dafür stückweise, lineare Funktionen wie Trapez- und Dreiecksfunktionen ausreichend. Mehrere linguistische Variablen, wie z.B. der Meßbereich, können, durch geeignete Operatoren verknüpft, als Eingang einer Wenn-Dann-Regel dienen, die durch unscharfes Schließen einen Ausgangswert, wie z.B. ein Meßprinzip, ermittelt (Bild 100). Werden mehrere dieser Regeln zusammengefügt, entsteht eine Regelbasis. Nachteil des vorgestellten Verfahrens ist, daß eine große Anzahl von Regeln zur Abbildung des Erfahrungswissens notwendig ist. Aus diesem Grund soll ein anschaulicher Ansatz, der auf die geometrische Übereinstimmung des Anforderungsprofils mit dem Meßprinzipprofil aufbaut, vorgestellt werden. Wenn die Fläche, die das Anforderungsprofil des Anwenders repräsentiert, mit F_A und die Schnittfläche mit F_S bezeichnet wird, liefert

$$U = \frac{F_S}{F_A}$$

Gl. 60

ein relatives Maß für die Übereinstimmung von Anforderung und Meßprinzip. Ist das Flächenstück, das das Anforderungsprofil darstellt, vollständig enthalten im Flächenstück, das das Meßprinzipprofil repräsentiert, so nimmt der Übereinstimmungskoeffizient den maximalen Wert eins an. Die Berechnung des Übereinstimmungsmaßes läßt sich für alle n Anforderungen des Anwenders und m Meßprinzipklassen durchführen. Um den Einfluß einer bestimmten Kenngröße auf das Gesamtergebn zu gewichten, kann das zugehörige Übereinstimmungsmaß U_{ij} mit einem Gewicht G_j multipliziert werden. Die Summe der gewichteten Faktoren ergibt dann das Gesamtergebn für eine einzelne Meßprinzipklasse:

$$E_i = U_{i1} \cdot G_1 + U_{i2} \cdot G_2 + \dots + U_{in} \cdot G_n = \sum_{j=1}^n U_{ij} \cdot G_j \quad \text{Gl. 61}$$

Die Meßprinzipklasse, deren Profile am besten mit den Anwenderprofilen übereinstimmen, weist anschaulich die maximale Summe der gewichteten Übereinstimmungsmaße auf.

$$E_k = \max_i (E_i) \quad \text{Gl. 62}$$

Im Auswahlssystem wurde der zuletzt skizzierte Algorithmus implementiert.

8.4.3 Wissensbasis

Die Wissensbasis enthält die jeweiligen Profile der Meßprinzipklassen und der einzelnen Sensoren. Die Entscheidungsfindung führt an Hand der in der Wissensbasis abgelegten Profile die Auswahl des geeignetsten Sensors aus. Es wird zunächst ein qualifiziertes Datenmodell konzipiert, um die Daten strukturiert abbilden zu können. Da die objektorientierte Programmierung gegenüber der herkömmlichen, funktionalen Programmierung mehrere Vorteile wie Vererbung und Polymorphismus besitzt, wurde hier eine objektorientierte Strukturierung des Datenmodells verwendet. Die Vererbung ermöglicht dabei die Wiederverwendung von bereits erstellter Software durch Erzeugen von Klassen, die Spezialisierungen von bereits existierenden Klassen sind. Die so erzeugten Objekte erben alle Eigenschaften der Oberklassen. Außerdem verstehen alle Objekte, die von einer gemeinsamen Basisklasse abgeleitet sind, eine gemeinsame Menge von Nachrichten, auf die sie aber ggf. unterschiedlich reagieren können. Diese Eigenschaften können den Programmieraufwand erheblich reduzieren [141]. Die entworfene, objektorientierte Datenstruktur zeichnet sich durch das Vorhandensein einer Klassenhierarchie aus, die in Bild 101 dargestellt ist. Jede Klasse übernimmt alle Funktionen und Variablen der darüberliegenden Klasse und kann außerdem noch zusätzliche enthalten [141].

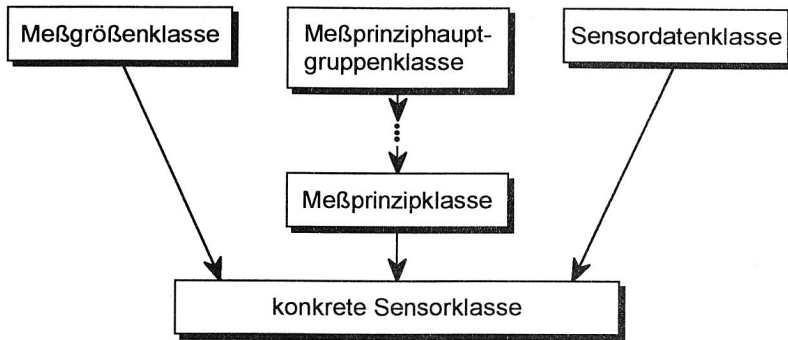


Bild 101: Klassenhierarchie der entworfenen Datenstruktur

Zur Organisation und Verwaltung von Daten eignen sich grundsätzlich Datenbanken, wobei zwischen hierarchischen, netzwerkförmigen, relationalen und objektorientierten Datenbankmodellen unterschieden wird. Auf deren Eigenschaften soll jedoch nicht weiter eingegangen werden [4, 27, 145]. Der Vergleich unterschiedlicher Kriterien wie beispielsweise die Einfachheit des Datenmodells, Programm- und Datenunabhängigkeit, die Schnelligkeit, die Ausbaumöglichkeiten sowie das vorhandene objektorientierte Datenmodell haben dazu geführt, die Profile der Meßprinzipklassen und der Sensoren in einem objektorientierten Datenbanksystem zu hinterlegen [4, 27].

8.4.4 Anwenderschnittstelle

In einem interaktiven Dialog mit dem Auswahlssystem konkretisiert der Ausführende sukzessiv das Anforderungsprofil. Der Auswahlvorgang kann in drei unterschiedlichen Betriebsarten vorgenommen werden, wobei im Einzelschrittmodus in drei Hauptdialogen zunächst die Meßgröße und Schritt für Schritt die Meßprinzipien an Hand der vorgegebenen Anforderungen des Anwenders spezifiziert werden.

Schließlich erfolgt der Zugriff auf die Datenbank, um unter Einsatz der Auswahlalgorithmen den geeignetsten Sensor zu ermitteln.

Im Gegensatz dazu wird im Automatikmodus das Anforderungsprofil nicht schrittweise vom Auswahlssystem interaktiv erfragt, sondern in einem Hauptdialog vollständig detailliert übergeben. Das mittlere Fenster von Bild 102 zeigt den Hauptdialog im Automatikmodus, wobei über die Schaltflächen Spezifikationsdialoge, die z.B. zur Angabe des Sensormeßbereichs erforderlich sind (oberes Fenster), angewählt werden können. An Hand der Spezifikationen ermittelt der Auswahlalgorithmus den optimalen Sensor. In Bild 102 ist im unteren Fenster die textuelle Ergebnisausgabe des Sensorauswahlsystems zu sehen.

Eingabe - Meßparameter

Bereiche

mm-Bereich
cm-Bereich
dm-Bereich
m-Bereich

Genauigkeit (in % des Messbereichendwertes)

Meßbereich in mm

von bis

*

Meßmethode

☐ berührend

☐ berührungslos

☒ egal

Automatikmodus / Dateneingabe

Parameter

Sensorkopffläche

Empfindlichkeit

Sensorkopfvolumen

Dynamik

Anschaffungskosten

Störeinflüsse

Installationsaufwand

OK

Abbruch

C:\SDB\OUTSENS\OUTSENS.EXE

```

Meßgrößennummer: 4
Meßprinzipgruppennummer: 1
Meßprinzipnummer: 1
Sensornummer: 1
Messbereich: 3.4000 - 6.9000
Genauigkeit: 0.8000
Temperaturempfindlichkeit: 3
Lichtempfindlichkeit: 1
Vibrationsempfindlichkeit: 2
Staubempfindlichkeit: 3
          
```

Bild 102: Anwenderschnittstelle des Auswahlsystems im Automatikmodus

Für den erfahrenen Anwender steht außerdem der Sprungmodus zur Verfügung, der einen direkten Sprung auf die jeweilige Meßprinziphierarchieebene erlaubt, um Sensoren mit definiertem Meßprinzip auswählen zu können. Das in der Datenbank hinterlegte Wissen kann außerdem über eigene, zugangsgeschützte Dialogfenster gewartet und erweitert werden. In der selben Weise stehen Dialogfenster zur Parametrierung der Auswahlalgorithmen zur Verfügung.

9. Zusammenfassung

Eine oftmals hohe Variantenvielfalt, geringe Produktlebenszeiten und eine hohe Produktqualität kennzeichnen gegenwärtig die Situation der produzierenden Unternehmen in der Bundesrepublik Deutschland. Vor dem Hintergrund der hohen Lohnkosten bestehen keinerlei Alternativen zu einer weiteren, flexiblen Automatisierung, um Qualitäts- und Produktivitätssteigerungen zu erzielen. Gerade Industrieroboter, die eine flexible Anpassung an wechselnde Fertigungsbedingungen ermöglichen und außerdem ein hohes Automatisierungspotential besitzen, können dazu einen wichtigen Beitrag leisten. Letztendlich entscheiden aber nicht alleine die Fähigkeiten des Industrieroboters über den Erfolg oder Mißerfolg von Automatisierungsbestrebungen, sondern die Roboterperipherie und Programmierwerkzeuge tragen wesentlich dazu bei. Insbesondere bei der Integration von externen Sensoren an Roboteranlagen bestehen weitreichende Defizite, die den Einsatz von Industrierobotern in einigen Bereichen einschränken, wenn nicht ganz verhindern. So existieren bisher nur eingeschränkte Möglichkeiten zur flexiblen und schnellen Integration externer Sensordaten, um die Bewegungsbahn von Industrierobotern an die aktuellen Form- und Lagetoleranzen des Werkstück anzupassen.

Im Rahmen der vorliegenden Arbeit wird deshalb ein Konzept entwickelt und realisiert, daß eine konsequente Unterstützung sowohl bei der Auswahl von qualifizierten Sensoren als auch deren schnelle und konfigurierbare Integration in eine Robotersteuerung beinhaltet. Hauptziel ist der Entwurf eines integrierten Gesamtsystems, das den Anwender insbesondere beim Entwurf und der Durchführung einer schnellen Sensorregelung eines Industrieroboters weitreichend und effizient unterstützt. Verschiedene Sensorregelungs- und Integrationskonzepte werden hinsichtlich ihres dynamischen Systemverhaltens untersucht, um Aussagen zur Leistungsfähigkeit von Sensorregelungen bei Bearbeitungsverfahren zu erlangen.

Zunächst wird die Situation der Sensorintegration an heutigen, robotergestützten Fertigungsanlagen analysiert und die Notwendigkeit und deren technologisch-ökonomischer Nutzen beleuchtet. Die bekannten Strategien zur Sensorintegration und Sensorregelung lassen sich systematisch strukturieren und bewerten. Eine Diskussion der Defizite bestehender Lösungen runden die Darstellung ab.

In einem 2. Schritt werden die Anforderungen an ein konfigurierbares, integriertes Gesamtsystem abgeleitet. Es gilt dabei insbesondere Fragen zu technologischen Anforderungen zu analysieren und zu strukturieren. Die technologischen Zieldefinitionen lassen sich auf die entsprechenden Komponenten eines Gesamtsystems projizieren und bilden die Basis zur Definition des Anforderungsprofils.

Diese Überlegungen münden in das Konzept des integrierten Gesamtsystems ISRS, das den Ausführenden bei der Programmerstellung, dem Entwurf von Sensorregelungen und dem Test von Sensorintegrationslösungen weitgehend unterstützt. Der Kern des Gesamtsystems ISRS stellt die flexibel parametrierbare Sensorsteuerung SC dar, die maßgeblich die Sensor-

wertverarbeitung, die Übertragung von Sensordaten zur Robotersteuerung und die Kommunikationssteuerung ausführt, sowie weitgehend von der Robotersteuerung aus bedienbar ist. Ein speziell auf die Belange der schnellen Sensorregelung optimiertes Transputersystem dient zur optionalen Erweiterung der Sensorsteuerung SC für den Fall der schnellen Sensorintegration im Lageregeltakt der Robotersteuerung. Der Robotersimulator RC-SIM, eine weitere Komponente des integrierten Gesamtsystems ISRS, erlaubt zudem eine Emulation der Robotersteuerung und deren Kommunikationsfunktionalität, um Sensorprogramme off line ohne die Gefahr einer Beschädigung der Anlage zu testen.

Auf der Basis des integrierten Gesamtsystems werden Untersuchungen, die eine Adaption der Bewegungsbahn des Roboters mit Hilfe unterschiedlicher Sensorregelungs- und Integrationskonzepte zum Ziel haben, durchgeführt. Verschiedene Sensorregler können dabei das dynamische Systemverhalten der Sensorregelung im Interpolationstakt verbessern, jedoch wird das dynamische Verhalten von einer totzeitdominanten Regelstrecke geprägt. Im Gegensatz dazu läßt sich mit einfachen Reglern bei der Sensorregelung im Lageregeltakt ein sehr gutes dynamisches Verhalten erzielen, das nur mehr durch die verbleibende Dynamik des Roboterantriebs begrenzt ist. Die resultierenden Schleppfehler bei unterschiedlichen Regelungs- und Integrationsstrategien werden analytisch wie praktisch ermittelt und ein Bezug zu den entsprechenden Anforderungen der Bearbeitungstechnologien hergestellt.

Das Gesamtsystem wird zudem um die Fähigkeit zur Integration und Fusion multisensorieller Daten erweitert. Eine Analyse der Defizite bestehender Systeme mündet in ein Konzept, daß eine flexible und je nach den bestehenden Erfordernissen frei konfigurierbare Integration von multisensoriellen Daten innerhalb des integrierten Gesamtsystems erlaubt. An Hand einer exemplarischen Anwendung kann der Nachweis erbracht werden, daß sich das Konzept auch für sensorgeregelte Applikationen eignet.

Vor der eigentlichen Integration von Sensoren in eine Roboteranlage müssen zunächst in einem mühsamen und zeitaufwendigen Prozeß qualifizierte Sensoren ausgewählt werden. Eine Analyse des Auswahlvorgangs und der bestehenden Defizite münden in ein Konzept für eine strukturierte Auswahlssystematik. Auf der Basis der Auswahlssystematik wird ein rechnergestütztes Planungssystem entworfen, das den Ausführenden bei Auswahl eines geeigneten Sensors unterstützt.

Literaturverzeichnis

1. Abidi, M.A.; Gonzalez, R.C.:
The Use of Multisensor Data for Robotic Application.
IEEE Transactions on Robotics and Automation, 6 (1990) Nr. 2, S. 159-177.
2. Ackermann, J.:
Abtastregelung.
Springer Verlag, Berlin, Heidelberg, New York, 1988.
3. ANSI:
ANSI / RIA 15.05-1-1990, Robot Systems, Point to Point and Static Performance
Characteristics, New York, 1989.
4. Atkinson, M.; DeWitt, D.; Maier, D.; Bancilhon, F.; Dittrich, K.; Zdonik, S.:
The Object-Oriented Database System Manifesto.
Proc. of DOOD, S. 28-36, Kyoto, 1989.
5. Bartelt, R.; Burkhardt, G.:
Mit Vollgas in die Kurven.
Roboter (1994) Juli/August, S. 30-32.
6. Bartelt, R.; Zinnow, U.:
Zukunftsweisendes Steuerungskonzept für Roboter und Handhabungssysteme.
Engineering & Automation 13 (1991) Nr. 2, S. 14-17.
7. Barthel, K.; Trunzer, W.:
Sensor verfolgt die Schweißbahn.
Laser-Praxis, Supplement zu Hanser Fachzeitschriften, S. 18-21, 1994.
8. Basir, A.O.; Shen, H.C.:
Sensory Data Integration : A Team Consensus Approach.
Proc. of Int. Conf. on Robotics and Automation, S. 1683-1688, Nice, 1992.
9. Berg, J.O.:
Path and Orientation Accuracy of Industrial Robots.
Internat. Journal of Advanced Manufacturing Technology (1993) Nr. 8, S. 29-33.
10. Berg, J.O.:
Robot Accuracy: A Matter of Programming.
Internat. Journal of Advanced Manufacturing Technology (1992) Nr. 7, S. 193-197.
11. Bernhardt, R.; Jacobi, A.; Schreck, G.; Willnow, C.:
Realistische Simulation von Industrierobotern.
ZwF 89 (1994) Nr. 4, S. 159-162.
12. Bernhardt, R.; Landvogt, W.; Schreck, G.; Leichsenring, O.:
Bahnschweißen mit Industrierobotern bei den Automobilzulieferern.
ZwF 89 (1994) Nr. 9, S. 419-422.
13. Buchmann, R.:
Bedienen mit 32 Bit.
Roboter (1994) April, S. 48-50.
14. Burg, K.; Haf, H.; Wille, F.:
Höhere Mathematik für Ingenieure.
Teubner Verlag, Stuttgart, 1985.

15. Buttazzo, G.:
HAREMS: Hierarchical Architecture for Robotics. Experiments with Multiple Sensors.
Proc. of 5th Int. Conf. on Advanced Robotics (ICAR), Vol.1, S. 43-48, Pisa, 1991.
16. Classe, D.:
Beitrag zur Steigerung der Flexibilität automatisierter Montagesysteme durch
Sensorintegration und erweiterte Steuerungskonzepte.
Carl Hanser Verlag, München, Wien, 1988.
17. Dausinger, F.; Hügel, H.:
Nutzungspotentiale von Lasern in der Blechverarbeitung.
Bänder Bleche Rohre (1992) Nr. 7, S. 47-50.
18. Dausinger, F.; Wahl, R.:
Robotergeführtes CO₂-Laserstrahlschweißen im Dünnblechbereich.
DVS-Berichte Nr. 135, S. 7-11.
19. Davies, C.:
New Frontiers for Real-Time Architectures.
Kongreß-Vortrags-Band, Echtzeit'90, S. 289-305, 1990.
20. Denavit, J.; Hartenberg, R.S.:
A Kinematic Notation for Lower-Pair Mechanisms based on Matrices.
ASME Journal of Applied Mechanics (1955) Nr. 22, S. 215-221.
21. Dillmann, R; Huck, M.:
Informationsverarbeitung in der Robotik.
Springer Verlag, Berlin, Heidelberg, New York, 1991.
22. DIN:
DIN 19226: Regelungs- und Steuerungstechnik Teil 1.
Beuth Verlag, Berlin, 1986.
23. DIN:
DIN 19245: Profibus.
Beuth Verlag, Berlin, 1991.
24. DIN:
DIN 44300: Informationsverarbeitung - Begriffe: Allgemeine Begriffe.
Beuth Verlag, Berlin, 1988.
25. DIN:
DIN V 66311, Teil 1: Sensorschnittstelle für Roboter- und Fertigungssysteme.
Beuth Verlag, Berlin, 1987.
26. DIN:
DIN 66312 Teil 1: Industrial Robot Language (IRL).
Beuth Verlag, Berlin, 1993.
27. Dittrich, K.:
Objektorientierte Datenmodelle als Basis komplexer Anwendungen.
Wirtschaftsinformatik 32 (1990) Nr. 3, S. 228-237.
28. Dornfeld, David A.:
Intelligent Deburring of Precision Components.
Proc. of Int. Conf. on Industrial Electronics, Control, Instrumentation and Automation,
S. 953-960, San Diego, 1992.

29. Dresselhaus, M.:
Automatic Program Adaptation with Teach-in Accuracy: An Important New Control Function.
Proc. of 25th Int Symp. on Industrial Robots, S. 665-670, Hannover, 1994.
30. Drews, P.; Galyga, B.; Maupas, A.:
Design of a Neutral Open Interface for Robot/Sensor Communication.
Proc. of 8th CIM-Europe Annual Conference; CIME Applications and Benefits, S. 98-109, Springer- Verlag, Birmingham, 1992.
31. Drews, P.; Willms, K.; Arnold, St.:
Sensoren zur Werkzeugführung bei der Laserbearbeitung.
Kongreßband zur Sensor '93, Vol. IV, S. 323-330, Nürnberg, 1993.
32. Drews, P.; Zunker, L.:
Echtzeit-Bahnplanung eines Roboters unter Sensoreinsatz.
Robotersysteme 5 (1989), S. 213-218.
33. Dummermuth, R.:
Eine Alternative zum Kauf von Echtzeit-Kernen.
Elektronik (1992) Nr. 18, S. 129-132.
34. Durrant-Whyte, H.; Rao, B.; Hu, H.:
Toward a fully decentralized architecture for multisensor data fusion.
Proc. IEEE Int. Conf. on Robotics and Automation, S. 1331-1336, Cincinnati, 1990.
35. Eccles, J.; Herd, T.; Duffy, N.D.:
Multi-Sensor Integration for Robotic Assembly.
Proc. of 5th Int. Conf. on Advanced Robotics (ICAR), Vol.2, S. 1295-1298, Pisa, 1991.
36. Eisenträger, G.:
Europäische Sensordatenbank.
Marketing-Kongreß zur Sensor' 91, S. 76-78, Nürnberg, 1991.
37. Engel, G.:
Stand der internationalen Normung zu Kenngrößen für Industrieroboter.
VDI-Berichte 921, S. 1-14, VDI-Verlag, Düsseldorf, 1991.
38. Eppinger, M.; Kreuzer, E.:
Systematischer Vergleich von Verfahren zur Rückwärtstransformation bei Industrierobotern.
Robotersysteme (1989) Nr. 5, S. 219-228.
39. Espiau, B.:
An Overview of Local Environment Sensing in Robotics Applications.
NATO ASI Series, Sensors and Sensory Systems for Advanced Robots, Vol. F.43, S. 125-151, Springer -Verlag, Berlin, 1988.
40. Eversheim, W.; Hausmann, A.:
Planung des Sensoreinsatzes für flexibel automatisierte Montagesysteme mit Industrierobotern.
VDI-Z 127 (1985) Nr. 1/2, S. 37-40.
41. Feldmann, K.:
Rechnergestützte Planung und Steuerung von automatisierten Montageanlagen.
Siemens-Zeitschrift 64 (1990) Nr. 5, S. 4-8.
42. Feldmann, K.; Schönherr, U. ; Zeller, J.:
Fast Sensory Feedback for Demanding Robot Applications in New Technologies
Proc. of the 25th Int. Symp. on Industrial Robots, S. 549-554, Hannover, 1994.

43. Feldmann, K.; Schönherr, U.; Zeller, J.:
Sensory Feedback for Industrial Robots - Based on a Distributed System.
Reprints of the IFAC Conf. on Integrated Systems Engineering, S. 99-105, Baden-Baden, 1994.
44. Feldmann, K.; Schönherr, U.; Zeller, J.:
Multisensor Integration for Sensor Guided Robots.
IEEE/RSJ/GI Int. Conf. on Intelligent Robots and Systems (IROS), S. 1730-1735, München, 1994.
45. Feldmann, K.; Schönherr, U.; Zeller, J.:
Programming and Control of Laser Beam Processing by Fast Sensor Guided Robots.
Proc. of the LANE'94: „Laser Assisted Net shape Engineering“, S. 749-758, Erlangen, 1994.
46. Föhr, R.:
Verteilte Echtzeitsysteme.
Elektronik (1992) Nr. 5, S. 154-164.
47. Franke, E.A.; Sturdivant, V.R.; Nedungadi, A.:
Sensor Integrated Control for Manipulators.
Proc. of Int. Conf. on Systems, Man and Cybernetics, S. 791-796, Los Angeles, 1990.
48. Franke, H.; Jacubasch, A.; Kuntze, H.-B.; Salaba, M.; Becker, P.J.:
Sensor Based Programming of Robot Motion.
Proc. of Int. Conf. on Robot Vision and Sensory Controls, S. 67-76, 1988.
49. Freund, E.; Roßmann, J.; Uthoff, J.; van der Valk, U.:
Towards Realistic Simulation of Robotic Workcells.
Proc. of IEEE/RSJ/GI Int. Conf. on Intelligent Robots and Systems (IROS), S. 39-46, München, 1994.
50. Friemert, H.:
Ein Sensor für alle Fälle.
Der Praktiker (1988) Nr. 11, 1988.
51. Fuchs, K.:
Sensor-Based Robots in Computer-Aided Manufacturing.
Proc. of 6th IFAC/IFORS/IMACS Symposium on Information Control Problems in Manufacturing Technology, S. 609-611, Madrid, 1990.
52. Fukuda, T.; Shimojima, K.; Arai, F.; Matsuura, H.:
Multi-Sensor Integration system based on Fuzzy Inference and Neural Network.
Information Sciences 71 (1993) Nr. 1-2, S. 27-41.
53. Garnich, F.; Schwarz, H.:
Laser führt Laser.
Roboter (1990) Nr. 5, S. 14-18.
54. Gehlert K.:
Crash ohne Risiko.
Robotertechnik, S. 40-44, 1991.
55. Gräser, A.:
Erweiterung des Smith-Prädiktors bei Störgrößenaufschaltung.
Automatisierungstechnische Praxis 36 (1994) S. 46-52.
56. Hagg, E.:
Realisierung von Multisensoranwendungen mit vernetzten logischen Sensoren und Aktoren.
Shaker Verlag, Aachen, 1993.

57. Harashima, F.; Hashimoto, H.; Kubota, T.:
Sensor based Robot Control Systems.
Proc. of IEEE Int. Workshop on Intelligent Motion Control, Vol.1, S. PL 1-PL10, 1990.
58. Hartwich, R.G.; Hommel, G.:
Kooperation und Konkurrenz.
Springer Verlag, Berlin, 1989.
59. Hesseler, M.:
Off-line-Programmierung von Industrierobotern.
wt-Produktion und Management 84 (1994), S. 479-482.
60. Hirzinger, G.; Sanchez, D.A.; Lange, F.:
Neuronale Netze in der Robotik.
Informationstechnik 33 (1991) Nr. 6, S. 317-322.
61. Holling, G.H.:
Abschätzung von Bahnfehlern in Robotersystemen.
Vieweg Verlag, Braunschweig, 1990.
62. Hopf, B.; Mlitzko, D.; Woher, C.:
Applications of Industrial Robots at Mercedes-Benz AG, Sindelfingen Plant - Numbers of
Usage, Technology and Cost Aspects.
Proc. of the 25th Int. Symp. on Industrial Robots, S. 23-29, Hannover, 1994.
63. Horn, A.:
Optische Sensorik zur Bahnführung von Industrierobotern mit hohen Bahngeschwindigkeiten.
Springer Verlag, Berlin, 1994.
64. Hoschek, J.; Lasser, D.:
Grundlagen der geometrischen Datenverarbeitung.
Teubner Verlag, Stuttgart, 1989.
65. Hsieh, L.-H.; Schuler, H.:
Profildatenverarbeitung in der sensorunterstützten Montage.
ZwF 87 (1992) Nr. 1, S. 25-29.
66. Isermann, R.:
Identifikation dynamischer Systeme.
Springer-Verlag, Berlin, Heidelberg, New York, 1988.
67. Isermann, R.:
Digitale Regelsysteme.
Band I, Springer Verlag, Berlin, Heidelberg, New York, 1988.
68. ISO:
ISO 9283, Manipulating industrial robots, Performance criteria and related test methods,
Genf, 1988.
69. Jacobasch, A.; Kuntze, H.-B.; Arver, C.; Richalet, J.:
Anwendung eines neuen Verfahrens zur schnellen und robusten Positionsregelung von
Industrierobotern.
Robotersysteme (1987) Nr. 3, S. 129-138.
70. Keferstein, C.; Frick, O.:
Applikationsspezifischer Abgleich und Kalibrierung eines sechssachsigen
Knickarmpräzisionsroboters.
VDI-Berichte 921, S. 33-42, VDI-Verlag, Düsseldorf, 1991.

71. Keppke, A.:
Just in Time - Teil 1.
Computertechnik (1992) Nr. 8, S. 52-59.
72. Keppke, A.:
Just in Time - Teil 2.
Computertechnik (1992) Nr.9, S. 202-210.
73. Kern, R.:
Prozeßauswahl und Ablaufplanung in Echtzeit-Systemen.
Elektronik (1992) Nr. 14, S. 26-32.
74. Kief, H. B.:
FFS-Handbuch '92/93: Einführung in Flexible Fertigungssysteme CNC, DNC, CAD, CAM,
FFS, FMS, CAQ.
Carl Hanser Verlag, München, Wien, 1992.
75. Kim K.H.; Kunz U.:
Sensorgeführte Industrierobotersteuerung mit einem Tranputernetzwerk unter MS-Windows.
VDI-Berichte 1094, "Intelligente Steuerung und Regelung von Robotern", S. 403-412,
Langen, 1993.
76. Kinas, J.; Rieseler, H.:
Robotersimulation-Einfluß auf Entwicklung und Produktion in der Automobilindustrie.
CIM Management 10 (1994) Nr. 3, S. 6-9.
77. Kram, R.; Lang, E.:
Schnelle Sensordatenrückkopplung mit Hilfe liearisierter Transformationsrechnungen.
Intelligente Sensorsysteme in der Fertigungstechnik, S. 234-240, Springer Verlag, Berlin,
1989.
78. Kratzer, K.-P.:
Neuronale Netze: Grundlagen und Anwendungen.
Hanser Verlag, München, Wien, 1990.
79. Krazer, M.:
Automatisiertes Auftragen von Kleb- und Dichtstoff.
Chip Plus (1989) Nr. 6, S. 10-14.
80. Kreienkamp, U.; Schmittele, M.:
Standardisierung der Sensorschnittstelle.
Intelligente Sensorsysteme in der Fertigungstechnik, S. 275-280, Springer Verlag, Berlin,
Heidelberg, New York, 1989.
81. Kroth, E.:
Einbindung von Sensorfunktionen.
Robotertechnik (1991), S. 50-51.
82. Lange, F.; Hirzinger, G.:
Learning to Improve the Path Accuracy of Position Controlled Robots.
Proc. of Int. Conf. on Intelligent Robots and Systems, S. 494-501, München, 1994.
83. Lauer, P.M.; Kunz, U.:
Transputer network controls robot axes.
Proc. of IFAC Conf. on Robot Control, S. 423-427, Wien, 1991.
84. Linn, R.J.; Hall, D.L.:
A Survey of Multi-Sensor Data Fusion Systems.
Proc. of the SPIE Vol. 1470, S. 13-29, 1991.

85. Luo, R.C.; Kay, M.G.:
Multisensor Integration and Fusion in Intelligent Systems.
IEEE Transactions on System, Man, and Cybernetics, Vol. 19 (1989) Nr. 5, S. 901-931.
86. Luo, R.; Lin, M.:
Hierarchical Robot Multi-Sensor Data Fusion System.
NATO ASI Series, Highly Redundant Sensing in Robotic Systems, Vol. F58, S. 67-86, 1990.
87. Mäkinen, P.; Siivola, E.; Aumala, O.:
Sensor Selection System.
Kongreßband IV zur Sensor'93, S. 185-190, Nürnberg, 1993.
88. Mason, T.; Brown, D.:
lex & yacc.
O'Reilly & Associates, 1990.
89. McKeever, J.D.M.; Holton, D.R.W.; McKeag, R.M.:
Using transputers in a robot programming and control system.
Microprocessing and Microprogramming (1992) Nr. 34, S. 117-120.
90. Milberg, J.:
Der Laser als Teil der <<Neuen Fabrik>>.
Blech Rohre Profile (1993) Nr. 4, S. 297-302.
91. Mohr, U.:
Sensible Reaktion.
Roboter (1990) Oktober, S. 22-24.
92. Mohr, W.; Bernshaus, P.:
Eine Nasenlänge Vorsprung.
Robotertechnik, S. 37-38, 1991.
93. Münch, H.; Sorowka, D.:
Sensorgestützte, lage- und kraftgeregelte Automatisierung von Montageaufgaben.
ZwF 89 (1994) Nr. 9, S. 430-432.
94. Nabhan, T.M.; Zymaya, A.Y.:
Towards efficient scheduling for robotic computation.
Proc. of Int. Conf. on Robotics, Vision and Parallel Processing for Industrial Automation (ROVPIA' 94), S. 364-370, Ipoh, Malaysia, 1994.
95. Nitsch, H.; Kaierle, S.:
Der Schweißnaht auf der Spur.
Roboter (1994) Nr. 4, S. 24-28.
96. N.N.:
So kommen die Räder an's Auto.
Roboter (1994) Nr. 2, S. 10-12.
97. N.N.:
Entgraten aus dem Baukasten.
Flexible Automation (1990) Nr. 6, S. 31-34.
98. N.N.:
MATLAB, User's Guide, The MathWorks, Inc., Natick, 1993.
99. N.N.:
Neue Robotersysteme für schlanke Produktion.
Produktionsautomatisierung (1994) Nr. 1+2, S. 27-28.

100. N.N.:
RTKern - RealTime Multitasking Kernel für Programmiersprache C.
Benutzerhandbuch Version 3.0, On Time Informatik GmbH, Hamburg, 1992.
101. Nunes, U.; Faia, P.; Araujo, R.; deAlmeida, A.T.:
Hierarchical Architecture for Multi-Sensor Robot Cell Operation.
Proc. of Int. Symp. on Intelligent Control, S. 416-422, Glasgow, 1992.
102. Olomski, J.:
Bahnplanung und Bahnführung von Industrierobotern.
Fortschritte der Robotik 4, Vieweg Verlag, 1989.
103. Pantring, J.:
Einsatz von Parallelverarbeitung in der Robotik.
VDI-Berichte 1094, Intelligente Steuerung und Regelung von Robotern, S. 701-710, Langen, 1993.
104. Papageorgiou, M:
Optimierung.
Oldenbourg Verlag, München 1991.
105. Paul, R.P.:
Robot Manipulators: Mathematics, Programming and Control.
MIT Press, 1981.
106. Reddig, M.; Stelzer, J.:
Iterative Methoden der Koordinatentransformation am Beispiel eines 6-Achsen-
Gelenkroboters mit Winkelhand.
Robotersysteme (1986) Nr. 2, S. 138-142.
107. Pfaff, G.; Meier, C.:
Regelung elektrischer Antriebe II.
Oldenbourg Verlag, München, Wien, 1988.
108. Pfeiffer, F.; Reithmeier, E.:
Roboterdynamik.
Teubner Verlag, Stuttgart, 1987.
109. Prassana, V.K.; Rao, A.S.:
Parallel Algorithms for Robotics - A Survey.
Computer Science and Informatics 22 (1992) Nr.1, S. 1-18.
110. Pritschow, G.; Horn, A.:
Dynamik derzeitiger Sensorregelkreise für Industrieroboter.
Robotersysteme (1991) Nr. 7, S. 178-184.
111. Pritschow, G.; Horn, A.:
Optical Sensor Systems for On-line Guiding of Laser Tools at High Path Velocities.
Proc. of Int. Conf. Laser Assisted Net shape Engineering (LANE), S. 597-606, Erlangen, 1994.
112. Pritschow, G.; Horn, A.; Grefen, K.:
Dynamisches Verhalten und Grenzen sensorgestützter Industrieroboter mit vorausblickendem Sensor.
Robotersysteme 8 (1992), S. 155-161.
113. Pritschow, G.; Klingel, H.; Bauder, M.; Horn, A.:
Erhöhung der Bahngeauigkeit von Industrierobotern.
Robotersysteme 8 (1992), S. 162-170.

114. Qu, Z.; Kauppila, J.; Moisio, T.:
A Seam Tracking system for Sheet Metal Pipe Fabrication by Laser Beam Welding.
Proc. of Int. Conf. Laser Assisted Net shape Engineering (LANE), S. 741-748, Erlangen, 1994.
115. Rajagopalan, R.; Cheng, R.M.H.; Poon, S.:
Parallel computation of inverse dynamic of robot employing transputers.
Proc. of ISMM Int. Conf. on Parallel and Distributed Computing and Systems, S. 314-318, Anaheim, 1990.
116. Reinhart, G.; Lindl, H.; Trunzer, W.:
Planning and Integration of Laser Systems into Flexible Production Structures.
Proc. of Laser Assisted Net shape Engineering (LANE), S. 687-704, Erlangen, 1994.
117. Rzehak, H.:
Grundlagen und Methoden für die Praxis.
Elektronik (1991) Nr. 5, S. 202-206.
118. Rogos, J.:
Sensoren in der Fertigungstechnik.
Shah, R (Hrsg.), Sensoren 86/87, Düsseldorf, S. 66-73, 1986.
119. Schmid, D.:
Sensor simuliert Werkzeug.
Roboter (1990) Februar, S. 14-17.
120. Schmid, D.:
Sensorführung von Robotern.
VDI-Berichte Nr. 747, S. 235-268, VDI-Verlag, Düsseldorf, 1989.
121. Schmid, D.:
Dynamik sensorgeführter Roboter.
Intelligente Sensorsysteme in der Fertigungstechnik, S. 220-233, Springer-Verlag, Berlin, 1989.
122. Schmid, D.; Hardter, H.; Sichler, K.:
Zusatzachsen verbessern die Sensorführung von Robotern.
Robotersysteme 5 (1989), S. 247-251.
123. Schmid, D.; Michalak, E.; Sichler, K.:
Präzise auf Distanz - Ringsensor hält Schweißroboter im Griff.
Roboter (1994) Nr. 7/8, S. 18-20.
124. Schmid, D.; Michalak, E.; Zobel, R.:
Autonom sensorgeführter Roboter zur Schleifbearbeitung von Behältersegmenten.
Robotersysteme 6 (1990), S. 99-102.
125. Schmid, D.; Sichler, K.; Michalak, E.:
Sensorgestützte Bahnprogrammierung beim Laserschweißen mit Roboter.
Robotersysteme (1992) Nr. 8, S. 21-24.
126. Schmidt, G.:
Grundlagen der Regelungstechnik.
Springer Verlag, Berlin, Heidelberg, New York, 1984.
127. Schmidt, J.; Bott, K.:
Hartes sanft entgratet- Flexible Fertigungszelle zum Entgraten von Keramik.
Roboter (1991) Nr. 8, S. 28-30.

128. Schönherr, U.; Zeller, J.:
Bestimmung von Werkstückparametern mit Hilfe vektoriell angeordneter Basissensoren.
Kongreßband zur SENSOR'95, S. 795-798, Nürnberg, 1995.
129. Schwarz, H.:
Simulationsgestützte CAD/CAM-Kopplung für die 3D-Laserstrahlbearbeitung mit integrierter Sensorik.
Forschungsberichte 68, Springer-Verlag, 1994.
130. Schweigert, U.:
Sensor-Guided Assembly.
Sensor Review 12 (1992) Nr. 4, S. 23-27.
131. Schweizer, M.:
Roboterbranche mit Rückenwind.
Roboter (1993) Nr. 6, S. 10-12.
132. Schweinzer, H.:
Fast Sensor Correction of Robot Motion Paths Processed in Real-Time by Control Algorithms Running in Parallel.
The Journal of Real-Time Systems (1989) Nr. 1, S. 207-220.
133. Shafer, G.:
A Mathematical Theory of Evidence.
Princeton University Press, 1976.
134. Sharkey, M.P.; Daniel, R.W.; Elosegui, P.:
Transputer based real time robot control.
Proc. of 29th Conf. on Decision and Control, S. 1161- 1162, Honolulu, Hawaii, 1990.
135. Seeger, G.:
Selbststeuerende, modellgestützte Regelung eines Industrieroboters.
Fortschritte der Robotik 13, Vieweg Verlag, 1992.
136. Seidel, M.:
Schnelle, sensorgesteuerte Konturbearbeitung mit Industrierobotern.
VDI Fortschrittsberichte Reihe 2 Nr. 277, VDI-Verlag, Düsseldorf, 1992.
137. Sendler, U.:
Offline-programmiertes Laserschneiden im Automobilbau.
VDI-Z 136 (1994) Nr. 1/2, S. 28-30.
138. Sommer, E.:
Multiprozessorsteuerung für kooperierende Industrieroboter in Montagezellen.
Carl Hanser Verlag, München Wien, 1992.
139. Swik, R.:
Multitasking zum Kennenlernen.
Elektronik (1992) Nr. 12, S. 42-47.
140. Spur, G.:
The Change in Production Technology.
Proc. of Laser Assisted Net shape Engineering (LANE), S. 3-20, Erlangen, 1994.
141. Stroustrup, B.:
Die C++-Programmiersprache.
Addison-Wesley, Bonn, 1992.
142. Thater, G.; Behrens, A.; Roos, E.:
Reproducible Comparison Tests for Industrial Robots.
Proc. of 25th Int. Symp. on Industrial Robots, S. 491-497, Hannover, 1994.

143. Tilly, T.:
Fuzzy-Logik Grundlagen, Anwendungen, Hard- und Software.
Francis-Verlag, 1991.
144. Tradt H.R.:
Stand der Meßtechnik für die Ermittlung der Kenngrößen von Industrierobotern.
VDI-Berichte 921, S. 43-56, VDI-Verlag, Düsseldorf, 1991.
145. Trautloft, R.; Lindner, U.:
Datenbank Entwurf und Anwendung.
Verlag Technik GmbH, Berlin, 1991.
146. Tschulena, G.R.:
Sensortechnik: Übersicht und allgemeine Trends.
Shah, R (Hrsg.), Sensoren 86/87, Düsseldorf, S. 8-19, 1986.
147. Tschulena, G.R.:
Sensordatenbank: Beschreibung und Analyse der Angaben.
Shah, R (Hrsg.), Sensoren 86/87, Düsseldorf, S. 156-171, 1986.
148. Union Technique de l'Electricité (UTE):
NF C 46-602 Standard - Application Layer Periodic and Aperiodic Services, 1990.
149. Union Technique de l'Electricité (UTE):
NF Pr C 46-603 Draft Standard - Data Link Layer, 1990.
150. VDI:
VDI-Richtlinie 2861, Blatt 1-3, Kenngrößen für Industrieroboter.
VDI-Verlag, Düsseldorf, 1988.
151. Visser, A.; Kühn, W.; Kuang, L.:
Vermessen von Gelenkarmrobotern mit Hilfe von Theodoliten.
Technisches Messen 61 (1994) Nr. 2, S. 75-81.
152. v. Dungern, O.; Freyberger, F.; Schmidt, G.:
Eine modulare Grundsteuerung für flexible Montagezellen.
Automatisierungstechnische Praxis 32 (1990) Nr. 1, S. 22-29.
153. v. Trotha, L.:
Technologie nicht nur für die Zukunft.
Laser Praxis, Supplement zu Hanser Fachzeitschriften, LS 86-LS 90, 1992.
154. Vukobratovic, M.:
Introduction to Robotics.
Springer Verlag, Berlin, Heidelberg, New York, 1989.
155. Wahrburg, J.:
Control Concepts for Industrial Robots Equipped with Multiple and Redundant Sensors.
NATO ASI Series, Highly Redundant Sensing in Robotic System, Vol. F58, S. 277-291,
Springer Verlag, Berlin, 1990.
156. Wahrburg, J.:
Design Concepts of Sophisticated Sensor Systems to Improve the Flexibility of Industrial Robots.
Proc. of 21th Int. Symp. on Industrial Robots (ISIR), S. 249-254, Kopenhagen, 1990.
157. Wenzel, M.:
The importance of the Control for Robotic Path Welding.
Proc. of the 25th Int. Symp. on Industrial Robots, S. 435-440, Hannover, 1994.

158. Westkämper, E.; Wietfeld, T.:
Grafisch-interaktive Systeme zur Offline-Programmierung von Industrierobotern.
CIM Management 10 (1994) Nr. 3, S. 13-16.
159. Winz, R.:
Fallbeispiele für die Auswahl von Echtzeit-Betriebssystemen.
Elektronik (1992) Nr. 11, S. 206-216.
160. Wijesoma, S.W.; Wofe, D.F.H.; Richards, R.J.:
Eye-to-Hand Coordination for Vision-Guided Robot Control Applications.
The International Journal of Robotics Research 12 (1993) Nr. 1, S. 65-77.
161. Wilson, W.J.:
Vision Sensor Integration for Direct Manipulator End-Point Control.
NATO ASI, Vol. F 57, S. 281-299.
162. Wloka, D.W.:
Robotersysteme 1, Technische Grundlagen.
Springer-Verlag, Berlin, Heidelberg, New York, 1992.
163. Woenckhaus, C.; Kugelman, D.:
USIS - An Integrated 3D-Tool for Planning Production Cells.
Proc. of 25th Int. Symp. on Industrial Robots (ISIR), S. 31-38, Hannover, 1994.
164. Zang, Y.:
Object Oriented Modeling for Sensor-Guided Real-Time Robot Control.
Proc. of the 3rd North American Transputer Users Group, S. 331-342, Amsterdam, 1990.
165. Zeller, J.; Schönherr, U.:
Sensorintegration in moderne Robotersteuerungen.
VDI-Berichte 1094 "Intelligente Steuerung und Regelung von Robotern", S. 691-700, Langen,
VDI-Verlag, Düsseldorf, 1993.
166. Zhang, H.; Paul, R.P.:
A parallel inverse kinematics solution for robot manipulators based on multiprocessing and
linear extrapolation.
IEEE Transactions on Robotics and Automation 7 (1991) Nr. 5, S. 660-669.
167. Zhao, W.:
Sensorgeführte Industrieroboter zur Bahnverfolgung.
Hanser Verlag, München, Wien, 1990.

Lebenslauf

Persönliches

Franz-Josef Zeller

geb. am 31. Januar 1964 in Immenstadt im Allgäu

verheiratet mit Dr. Birgit Zeller, geb. Funk, 2 Kinder

Eltern: Roman Zeller, Frieda Zeller, geb. Müller

Schulausbildung

1970 - 74 Grundschule in Burgberg im Allgäu

1974 - 83 mathematisch-naturwissenschaftliches Gymnasium Sonthofen, Allgäu

Abschluß: Abitur

Wehrdienst

1983 - 84 Grundwehrdienst: Gebirgsinstandsetzungsbataillon 8, Sonthofen

Studium

1984 - 89 Studium der Elektrotechnik an der Technischen Universität München,

Studienrichtung: Automatisierungstechnik

Abschluß: Dipl.-Ing. Univ

Berufstätigkeit

1990 - 95 wissenschaftlicher Assistent in der Gruppe für Steuerungs- und Sensortechnik

am Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik der

Friedrich-Alexander-Universität Erlangen-Nürnberg

Leiter: Prof. Dr.-Ing. K. Feldmann

Reihe Fertigungstechnik Erlangen

Band 1

Andreas Hemberger

**Innovationspotentiale in der rechnerintegrierten Produktion durch
wissensbasierte Systeme**

208 Seiten, 107 Bilder. 1988. Kartonierte.

Band 2

Detlef Classe

**Beitrag zur Steigerung der Flexibilität automatisierter Montage-
systeme durch Sensorintegration und erweiterte Steuerungskonzepte**

194 Seiten, 70 Bilder. 1988. Kartonierte.

Band 3

Friedrich-Wilhelm Nolting

Projektiertung von Montagesystemen

201 Seiten, 107 Bilder, 1 Tabelle. 1989.

Kartonierte.

Band 4

Karsten Schlüter

**Nutzungsgradsteigerung von Montagesystemen durch den Einsatz
der Simulationstechnik**

177 Seiten, 97 Bilder. 1989. Kartonierte.

Band 5

Shir-Kuan Lin

Aufbau von Modellen zur Lageregelung von Industrierobotern

168 Seiten, 46 Bilder. 1989. Kartonierte.

Band 6

Rudolf Nuss

**Untersuchungen zur Bearbeitungsqualität im Fertigungssystem
Laserstrahlschneiden**

206 Seiten, 115 Bilder, 6 Tabellen. 1989. Kartonierte.

Band 7

Wolfgang Scholz

**Modell zur datenbankgestützten Planung automatisierter
Montageanlagen**

194 Seiten, 89 Bilder. 1989. Kartonierte.

Band 8

Hans-Jürgen Wißmeier

**Beitrag zur Beurteilung des Bruchverhaltens von Hartmetall-
Fließpreßmatrizen**

179 Seiten, 99 Bilder, 9 Tabellen. 1989. Kartonierte.

Band 9

Rainer Eisele

**Konzeption und Wirtschaftlichkeit von Planungssystemen in der
Produktion**

183 Seiten, 86 Bilder. 1990. Kartonierte.

Band 10

Rolf Pfeiffer

Technologisch orientierte Montageplanung am Beispiel der Schraubtechnik

216 Seiten, 102 Bilder, 16 Tabellen. 1990. Kartoniert.

Band 11

Herbert Fischer

Verteilte Planungssysteme zur Flexibilitätssteigerung der rechnerintegrierten Teillefertigung

201 Seiten, 82 Bilder. 1990. Kartoniert.

Band 12

Gerhard Kleineidam

CAD/CAP: Rechnergestützte Montagefeinplanung

203 Seiten, 107 Bilder. 1990. Kartoniert.

Band 13

Frank Vollertsen

Pulvermetallurgische Verarbeitung eines übereutektoiden verschleißfesten Stahls

XIII + 217 Seiten, 67 Bilder, 34 Tabellen. 1990. Kartoniert.

Band 14

Stephan Biermann

Untersuchungen zur Anlagen- und Prozeßdiagnostik für das Schneiden mit CO₂-Hochleistungslasern

VIII + 170 Seiten, 93 Bilder, 4 Tabellen. 1991. Kartoniert.

Band 15

Uwe Geißler

Material- und Datenfluß in einer flexiblen Blechbearbeitungszelle

124 Seiten, 41 Bilder, 7 Tabellen. 1991. Kartoniert.

Band 16

Frank Oswald Hake

Entwicklung eines rechnergestützten Diagnosesystems für automatisierte Montagezellen

XIV + 166 Seiten, 77 Bilder. 1991. Kartoniert.

Band 17

Herbert Reichel

Optimierung der Werkzeugbereitstellung durch rechnergestützte Arbeitsfolgenbestimmung

198 Seiten, 73 Bilder, 2 Tabellen. 1991. Kartoniert.

Band 18

Josef Scheller

Modellierung und Einsatz von Softwaresystemen für rechnergeführte Montagezellen

198 Seiten, 65 Bilder. 1991. Kartoniert.

Band 19

Arnold vom Ende

Untersuchungen zum Biegeumformen mit elastischer Matrize

166 Seiten, 55 Bilder, 13 Tabellen. 1991. Kartoniert.

Band 20

Joachim Schmid

Beitrag zum automatisierten Bearbeiten von Keramikguß mit Industrierobotern

XIV + 176 Seiten, 111 Bilder, 6 Tabellen. 1991. Kartoniert.

Band 21

Egon Sommer

**Multiprozessorsteuerung für kooperierende
Industrieroboter in Montagezellen**

188 Seiten, 102 Bilder. 1991. Kartoniert.

Band 22

Georg Geyer

**Entwicklung problemspezifischer Verfahrensketten
in der Montage**

192 Seiten, 112 Bilder. 1991. Kartoniert.

Band 23

Rainer Flohr

**Beitrag zur optimalen Verbindungstechnik in der
Oberflächenmontage (SMT)**

186 Seiten, 79 Bilder. 1991. Kartoniert.

Band 24

Alfons Rief

**Untersuchungen zur Verfahrensfolge Laserstrahlschneiden
und -schweißen in der Rohkarosserieherstellung**

VI + 145 Seiten, 58 Bilder, 5 Tabellen. 1991. Kartoniert.

Band 25

Christoph Thim

**Rechnerunterstützte Optimierung von Materialflußstrukturen
in der Elektronikmontage durch Simulation**

188 Seiten, 74 Bilder. 1992. Kartoniert.

Band 26

Roland Müller

CO₂-Laserstrahlschneiden von kurzglasverstärkten Verbundwerkstoffen

141 Seiten, 107 Bilder, 4 Tabellen. 1992. Kartoniert.

Band 27

Günther Schäfer

Integrierte Informationsverarbeitung bei der Montageplanung

195 Seiten, 76 Bilder. 1992. Kartoniert.

Band 28

Martin Hoffmann

**Entwicklung einer CAD/CAM-Prozesskette für die Herstellung
von Blechbiegeteilen**

149 Seiten, 89 Bilder. 1992. Kartoniert.

Band 29

Peter Hoffmann

**Verfahrensfolge Laserstrahlschneiden und -schweißen :
Prozeführung und Systemtechnik in der 3D-Laserstrahlbearbeitung von Blech-
formteilen**

186 Seiten, 92 Bilder, 10 Tabellen. 1992. Kartoniert.

Band 30

Olaf Schrödel

Flexible Werkstattsteuerung mit objektorientierten Softwarestrukturen

180 Seiten, 84 Bilder. 1992. Kartoniert.

Band 31

Hubert Reinisch

**Planungs- und Steuerungswerkzeuge zur impliziten
Geräteprogrammierung in Roboterzellen**

XI + 212 Seiten, 112 Bilder. 1992. Kartoniert.

Band 32

Brigitte Bärnreuther

**Ein Beitrag zur Bewertung des Kommunikationsverhaltens
von Automatisierungsgeräten in flexiblen Produktionszellen**
XI + 179 Seiten, 71 Bilder. 1992. Kartoniert.

Band 33

Joachim Hutfless

**Laserstrahlregelung und Optikdiagnostik in der Strahlführung
einer CO₂-Hochleistungslaseranlage**
175 Seiten, 70 Bilder, 17 Tabellen. 1993. Kartoniert.

Band 34

Uwe Günzel

**Entwicklung und Einsatz eines Simulationsverfahrens für operative
und strategische Probleme der Produktionsplanung und -steuerung**
XIV + 170 Seiten, 66 Bilder, 5 Tabellen. 1993. Kartoniert.

Band 35

Bertram Ehmann

**Operatives Fertigungscontrolling durch Optimierung auftragsbezogener
Bearbeitungsabläufe in der Elektronikfertigung**
XV + 167 Seiten, 114 Bilder. 1993. Kartoniert.

Band 36

Harald Kolléra

**Entwicklung eines benutzerorientierten Werkstattprogrammiersystems
für das Laserstrahlschneiden**
129 Seiten, 66 Bilder, 1 Tabelle. 1993. Kartoniert.

Band 37

Stephanie Abels

**Modellierung und Optimierung von Montageanlagen
in einem integrierten Simulationssystem**
188 Seiten, 88 Bilder. 1993. Kartoniert.

Band 38

Robert Schmidt-Hebbel

**Laserstrahlbohren durchflußbestimmender
Durchgangslöcher**
145 Seiten, 63 Bilder, 11 Tabellen. 1993. Kartoniert.

Band 39

Norbert Lutz

**Oberflächenfeinbearbeitung keramischer Werkstoffe mit
XeCl-Excimerlaserstrahlung**
187 Seiten, 98 Bilder, 29 Tabellen. 1994. Kartoniert.

Band 40

Konrad Grampp

**Rechnerunterstützung bei Test und Schulung an
Steuerungssoftware von SMD-Bestücklinien**
178 Seiten, 88 Bilder. 1995. Kartoniert.

Band 41

Martin Koch

**Wissensbasierte Unterstützung der Angebotsbearbeitung
in der Investitionsgüterindustrie**
169 Seiten, 68 Bilder. 1995. Kartoniert.

Band 42

Armin Gropp

**Anlagen- und Prozeßdiagnostik beim Schneiden mit einem
gepulsten Nd:YAG-Laser**
160 Seiten, 88 Bilder, 7 Tabellen. 1995. Kartoniert.

Band 43

Werner Heckel

**Optische 3D-Konturerfassung und on-line Biege winkelmessung
mit dem Lichtschnittverfahren**

149 Seiten, 43 Bilder, 11 Tabellen. 1995. Kartoni ert.

Band 44

Armin Rothhaupt

**Modulares Planungssystem zur Optimierung
der Elektronikfertigung**

180 Seiten, 101 Bilder. 1995. Kartoni ert.

Band 45

Bernd Zöllner

Adaptive Diagnose in der Elektronikproduktion

195 Seiten, 74 Bilder, 3 Tabellen. 1995. Kartoni ert.

Band 46

Bodo Vormann

**Beitrag zur automatisierten Handhabungsplanung
komplexer Blechbiegeteile**

126 Seiten, 89 Bilder, 3 Tabellen. 1995. Kartoni ert.

Band 47

Peter Schnepf

Zielkostenorientierte Montageplanung

144 Seiten, 75 Bilder. 1995. Kartoni ert.

Band 48

Rainer Klotzbücher

**Konzept zur rechnerintegrierten Materialversorgung
in flexiblen Fertigungssystemen**

156 Seiten, 62 Bilder. 1995. Kartoni ert.

Band 49

Wolfgang Greska

Wissensbasierte Analyse und Klassifizierung von Blechteilen

144 Seiten, 96 Bilder. 1995. Kartoni ert.

Band 50

Jörg Franke

**Integrierte Entwicklung neuer Produkt- und Produktionstechnologien
für räumliche spritzgegossene Schaltungsträger (3-D MID)**

196 Seiten, 86 Bilder, 4 Tabellen. 1995. Kartoni ert.

Band 51

Franz-Josef Zeller

Sensorplanung und schnelle Sensorregelung für Industrieroboter

190 Seiten, 102 Bilder, 9 Tabellen. 1995. Kartoni ert.