

*Entwicklung eines Verfahrens zum Programmtest in der  
robotergestützten Montage*

Von der Fakultät für Maschinenbau, Elektrotechnik und Wirtschaftsingenieurwesen  
der Brandenburgischen Technischen Universität Cottbus zur Erlangung des  
akademischen Grades eines Doktor-Ingenieurs (Dr.-Ing.)

genehmigte

**Dissertation**

vorgelegt von

Diplom-Ingenieur

**Mario Roßdeutscher**

geboren am 03.02.1984  
in Beeskow, Deutschland

Vorsitzende: PD Dr. paed. Dr.-Ing. habil. Annette Hoppe  
Gutachter: Prof. Dr.-Ing. Ulrich Berger  
Gutachter: Prof. Dr.-Ing. Christian Diedrich  
Tag der mündlichen Prüfung: 08.07.2011



Berichte aus dem Lehrstuhl Automatisierungstechnik  
BTU Cottbus  
Herausgeber: Prof. Dr.-Ing. Ulrich Berger

**Mario Roßdeutscher**

**Entwicklung eines Verfahrens zum Programmtest  
in der robotergestützten Montage**

Shaker Verlag  
Aachen 2011

**Bibliografische Information der Deutschen Nationalbibliothek**

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Zugl.: Cottbus, BTU, Diss., 2011

Copyright Shaker Verlag 2011

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

Printed in Germany.

ISBN 978-3-8440-0464-9

ISSN 1864-5789

Shaker Verlag GmbH • Postfach 101818 • 52018 Aachen

Telefon: 02407 / 95 96 - 0 • Telefax: 02407 / 95 96 - 9

Internet: [www.shaker.de](http://www.shaker.de) • E-Mail: [info@shaker.de](mailto:info@shaker.de)

## Vorwort des Herausgebers

Die Automatisierungstechnik bildet eine Schlüsseltechnologie für die Steigerung der Produktinnovation und die Verbesserung von Wertschöpfungsprozessen. Als Konsequenz einer globalen Wirtschaftsstruktur müssen alle Unternehmensbereiche wie Entwicklung, Produktion und Güterverkehr in einen übergreifenden Kontext gestellt und behandelt werden. Dabei stellt die informationstechnische Verknüpfung verbundener Unternehmen und Unternehmensbereiche bei stetig veränderlichen Aufgabenstellungen und Randbedingungen eine besondere Herausforderung dar. Die Automatisierung des betrieblichen und betriebsübergreifenden Informationsflusses sowie die Einbindung des Menschen in allen Phasen des Entwicklungs- und Leistungsprozesses bildet daher die vordringliche Aufgabenstellung für Forschung und Entwicklung. Durch den zielgerichteten, systematischen Einsatz und die ständig aktualisierte Beurteilung und Bewertung automatisierungstechnischer Lösungen wird die schnelle Umsetzung und Einführung hochwertiger und zukunftsweisender Innovationen gesichert.

Ziel der Forschungsarbeiten am Lehrstuhl Automatisierungstechnik der Brandenburgischen Technischen Universität Cottbus (BTU) ist die kontinuierliche Verbesserung der automatisierungstechnischen Methoden und Verfahren im Hinblick auf fortgeschrittene Produktionsstrukturen. Ein weiterer Schwerpunkt ist die Entwicklung integrierter Fertigungs- und Montagesysteme unter Einsatz neuartiger Steuerungstechnik. Dazu werden leistungsfähige Entwurfswerkzeuge der *Digitalen Fabrik* erprobt und weiterentwickelt. Durch die Bereitstellung modernster Laborausstattung und die Zusammenarbeit mit industriellen und institutionellen Technologieführern wird der Stand der Technik in Wissenschaft und Forschung aktualisiert abgebildet. Nationale und internationale Forschungsarbeiten zu ganzheitlichen Automatisierungskonzepten, der *Industrial Life Cycle Automation*, runden das Aufgaben- und Leistungsspektrum des Lehrstuhls ab.

Die in dieser Buchreihe erscheinenden Bände stammen aus den Forschungsarbeiten des Lehrstuhls Automatisierungstechnik der BTU Cottbus. In diesen Bänden werden neue Resultate und Erkenntnisse aus Forschung und Entwicklung veröffentlicht. Die Berichte aus dem Lehrstuhl Automatisierungstechnik sollen Forschung, Entwicklung und Anwendung zu automatisierungstechnischen Fragestellungen enger verknüpfen und daraus Potential für zukünftige Innovationen erzeugen.

Ulrich Berger



---

## Kurzfassung

Der Einsatz digitaler Werkzeuge zur effizienteren Produktionsplanung von Anlagen rückt hinsichtlich immer kürzer werdender Produktlebenszyklen zunehmend in den Vordergrund. Insbesondere die Inbetriebnahmephase von Roboterzellen kann durch den Einsatz von Offline-Programmierung verkürzt und der Anlagenanlauf beschleunigt werden. Im Bereich der Inbetriebnahme von robotergestützten Montageanlagen existieren Defizite hinsichtlich der vorzeitigen Erstellung und des Tests von Roboterprogrammen, was die Motivation der vorliegenden Arbeit begründet. Nachfolgend wird das Verfahren der virtuellen Roboterprogrammentwicklung (VIPE) vorgestellt. Unter dem Begriff virtuelle Roboterprogrammentwicklung wird die vollständige Programmerstellung und der systematische Test von Programmfunktionalitäten ohne Vorhandensein der realen Roboterzelle verstanden. Die virtuelle Roboterprogrammentwicklung wird im Rahmen dieser Arbeit wie folgt dargelegt:

In den Kapiteln 1 und 2 wird die Zielstellung erläutert und die eigentliche Problemstellung analysiert. Dies erfolgt mithilfe einer Roboterprogrammanalyse zur Identifikation montagespezifischer Programmstrategien, welche zum heutigen Stand mangels Simulationsansätzen nicht vorzeitig abgesichert werden. Dazu gehören Positionierungs-, Füge- und Fehlerbehandlungsstrategien. Basierend auf diesen Erkenntnissen wird in Kapitel 3 eine Anforderungsanalyse durchgeführt, welche die Basis zur Bewertung bestehender Lösungen des Standes der Technik und Forschung in Kapitel 4 darstellt.

In Kapitel 5 erfolgt die Entwicklung eines Testkonzeptes und der Entwurf eines Systems zur Erfüllung gestellter Anforderungen. Mithilfe der Informationen über die zu beobachtenden Montageaktionen wird ein ereignisbasierter funktionaler Test zur Prüfung beliebiger Montageabläufe unter Nutzung der Unified Modeling Language vorgesehen. Die Beschreibung von Testszenarien erfolgt mithilfe von Zustandsautomaten und Sequenzdiagrammen. Das Gesamtsystem erlaubt die Kopplung einer herstellereinspezifischen Steuerung und führt eine Starrkörpersimulation zur Erzeugung plausibler Sensorsignale und Aktorbewegungen durch, welche essenziell zur Absicherung der vorgestellten Strategien sind. Kapitel 6 weist die Konzepttauglichkeit anhand der Realisierung des Gesamtsystems in Form eines Softwareprototyps nach.

Anschließend wird in Kapitel 7 mithilfe von praxisnahen Anwendungsbeispielen der Funktionsnachweis erbracht. Anhand eines mechatronischen Zellenmodells zur Schraubenmontage werden drei exemplarische Strategien beschrieben. Mithilfe einer Positionierungsstrategie wird die Werkstückträgerposition durch Auswertung von Abstandsmessungen ermittelt. Eine Fügestrategie stellt die prozesssichere Schraubenmontage unter Einfluss von Bauteiltoleranzen sicher. Abschließend wird eine Fehlerbehandlungsstrategie für die Schraubenkommissionierung erläutert. Für diese Strategien werden Testszenarien modelliert und Ereignissequenzen abgeleitet.

---

---

## Abstract

The usage of digital software tools for a more efficient production planning of plants moves in respect of shortened product life cycles into the spotlight. In particular the commissioning time of robot cells can be shortened with the usage of offline-programming technology. Furthermore the ramp-up time is accelerated. In the field of commissioning of assembly plants, there are still existing deficits in terms of an early programming and testing of robot programs which causes the motivation of this work. Subsequently the technique of virtual robot program development (VIPD) is presented. The VIPD technique is understood as the complete program generation and the systematic test of assembly programs without the existence of the real robot cell. The aspects of virtual robot program development are presented in this work as following:

The chapters 1 and 2 describe the objectives and analyse the existing problem. This is done with a robot program analysis to identify assembly specific program strategies which can not be verified with existing software tools. This includes strategies for positioning, joining and error handling. Based on these results a requirement analysis is performed in chapter 3 which forms the basis for the evaluation of existing solutions of the state of research and technological development (see chapter 4).

Chapter 5 presents a testing concept and a system design according to the specific requirements. Using simulation information about the observed assembly process, an event-triggered functional test for the verification of any assembly states is conceptualized. It uses the Unified Modeling Language for the formal description of test cases in the form of state machines and sequence diagrams. The overall system provides an interface for the connection of virtual robot controls and runs a rigid body simulation for the generation of plausible sensor signals and actor movements which are essential for the verification of the presented strategies. Chapter 6 validates the concept with an implementation in form of a software prototype.

After that chapter 7 proves the system functionality with the help of practical application examples. At first a mechatronic cell model for the screw assembly is described. Based upon that three examples with different types of strategies are shown. A positioning strategy is used to determine the exact position of the workpiece carrier with distance measurements. A joining strategy ensures the process stable screw assembly under the influence of part tolerances. Finally, an error handling strategy for the screw transport is presented. For all strategies testscenarios were modeled and test sequences were derived.



---

## **Vorwort des Verfassers**

Die vorliegende Dissertation entstand während meiner Doktorandentätigkeit im Center Produktions- und Werkstofftechnik der Direktion Produktionsplanung im Geschäftsfeld Mercedes-Benz Cars, Daimler AG im Werk Untertürkheim.

Mein besonderer Dank gilt Herrn Prof. Dr.-Ing. Ulrich Berger, dem Inhaber des Lehrstuhls Automatisierungstechnik der Brandenburgischen Technischen Universität Cottbus, für die wissenschaftliche Betreuung, die stetige Motivation und das rege Interesse, welches er mir bei der Erstellung dieser Arbeit entgegenbrachte.

Herrn Prof. Dr.-Ing. Christian Diedrich, Inhaber des Lehrstuhls für integrierte Automation an der Otto von Guericke Universität Magdeburg, möchte ich herzlich für die Übernahme des Koreferats danken. Weiterhin danke ich Frau PD Dr. paed. Dr.-Ing. habil. Annette Hoppe für die Übernahme des Vorsitzes der Prüfungskommission.

Ebenso möchte ich meinem ehemaligen Teamleiter Herrn Dipl.-Ing. Willi Klumpp (Team flexible Montagekonzepte) für die Chance danken, über den Zeitraum von 4 Jahren von ihm lernen zu dürfen. Sowohl fachlich kritische Diskussionen als auch die zur Verfügung stehenden Möglichkeiten ermöglichten eine zügige Fertigstellung dieser Arbeit. Herrn Dipl.-Inf. Fabian Aichele möchte ich sehr für die intensive Projektarbeit bei der Implementierung des Simulationssystems danken. Auch bedanke ich mich bei allen Studenten, die während der Projektlaufzeit mit großem Engagement mitgearbeitet haben. Weiterhin möchte ich meinen Teamkollegen für die vielen fachlichen Diskussionen danken, die zum Gelingen dieser Arbeit beitrugen. Insbesondere Herrn Dipl.-Ing. Volker Zipter danke ich für wertvolle fachliche Hinweise und die Durchsicht der Dissertation.

Nicht zuletzt gilt mein besonderer Dank meiner Frau Anja, meinen Eltern und allen Freunden und Kollegen, die mich bei der Erstellung dieser Arbeit unterstützt haben und während dieser Zeit häufig auf mich verzichten mussten. Ich möchte Ihnen insbesondere in den schweren Phasen der Arbeit für die Motivation und Unterstützung danken.

Mario Roßdeutscher

Stuttgart/ Trebatsch, im Juni 2011



---

## Inhaltsverzeichnis

<b>Verzeichnis der Abkürzungen</b>	<b>XI</b>
<b>Verzeichnis der Formelzeichen und Symbole</b>	<b>XIII</b>
<b>Verzeichnis der Abbildungen</b>	<b>XIV</b>
<b>Verzeichnis der Tabellen</b>	<b>XVI</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Herausforderungen für die Produktion in der Automobilindustrie . . . . .	1
1.2 Zielstellung der Arbeit . . . . .	3
1.3 Struktur der Arbeit . . . . .	4
<b>2 Motivation und Problemstellung</b>	<b>6</b>
2.1 Besonderheiten von Roboterprogrammen für die Montage . . . . .	7
2.1.1 Roboterprogrammanalyse . . . . .	9
2.1.2 Phasen der Programmentwicklung . . . . .	14
2.1.3 Herausforderungen beim Programmtest . . . . .	16
2.2 Potenziale durch den Einsatz digitaler Werkzeuge . . . . .	20
2.3 Zusammenfassung und Fazit . . . . .	23
<b>3 Anforderungsanalyse</b>	<b>24</b>
3.1 Anforderungen an die Simulationsumgebung zur Programmausführung . . . . .	24
3.1.1 Nutzung der herstellerepezifischen Steuerungsumgebung . . . . .	24
3.1.2 Plausibles Verhalten der roboternahen Peripherie . . . . .	25
3.1.3 Prozessvisualisierung . . . . .	26
3.1.4 Leistungsanforderungen . . . . .	27
3.1.5 Nutzung bestehender Daten und Integration in den Anlagenentstehungsprozess	27
3.2 Anforderungen an die Testumgebung zum Programmtest . . . . .	29
3.2.1 Test von Programmfunktionalitäten . . . . .	30
3.2.2 Beschreibung von Testscenarien . . . . .	30
3.2.3 Testdurchführung . . . . .	31
3.2.4 Testdatenverwaltung . . . . .	31
3.3 Zusammenfassung und Priorisierung . . . . .	31
<b>4 Stand der Technik und Ableitung des Handlungsbedarfes</b>	<b>34</b>
4.1 Verfahren zur Programmierung von Industrierobotern . . . . .	34
4.1.1 Überblick und Eingrenzung . . . . .	34
4.1.2 Prinzipien der Offline-Programmierung . . . . .	36
4.1.3 Kommerzielle CAR-Werkzeuge . . . . .	39

---

4.1.4	Forschungsansätze . . . . .	42
4.1.5	Bewertung und Fazit . . . . .	43
4.2	Methoden und Ansätze zur Montagesimulation . . . . .	45
4.2.1	Überblick und Eingrenzung . . . . .	45
4.2.2	Grundlagen der interaktiven Dynamiksimulation . . . . .	48
4.2.3	Einsatz von Physik-Engine Bibliotheken . . . . .	52
4.2.4	Montagespezifische Ansätze aus Technik und Forschung . . . . .	53
4.2.5	Bewertung und Fazit . . . . .	54
4.3	Methoden und Ansätze zum Softwaretest . . . . .	55
4.3.1	Überblick und Eingrenzung . . . . .	55
4.3.2	Ansätze zum Test von Steuerungsprogrammen . . . . .	57
4.3.3	Modellbasiertes Testen mithilfe von CASE-Werkzeugen . . . . .	59
4.3.4	Bewertung und Fazit . . . . .	60
4.4	Datenaustausch im Anlagenentstehungsprozess . . . . .	61
4.5	Bewertung bestehender Ansätze und Ableitung des Handlungsbedarfes . . . . .	63
4.6	Zusammenfassung . . . . .	67
<b>5</b>	<b>Konzeption und Systementwurf</b>	<b>69</b>
5.1	Konzept für den systematischen und funktionalen Programmtest . . . . .	69
5.2	Entwurf des Gesamtsystems . . . . .	72
5.3	Simulationsumgebung zur Programmausführung . . . . .	75
5.3.1	Physik-Engine zur Starrkörpersimulation . . . . .	75
5.3.2	Verhaltenssimulation . . . . .	76
5.3.3	3D-Visualisierung . . . . .	78
5.4	Testumgebung . . . . .	78
5.4.1	Bausteinsystem für die Beobachtung des Montageablaufes . . . . .	78
5.4.1.1	Prüfung von Montagezuständen . . . . .	79
5.4.1.2	Kollisionsüberwachung zwischen Objektgruppen . . . . .	80
5.4.1.3	Unzulässige Kontaktsituationen . . . . .	82
5.4.1.4	Verhalten bei Störungen . . . . .	82
5.4.1.5	Kollisionsfreie Erreichung von Teachpunkten . . . . .	84
5.4.1.6	Zustände der roboternahen Peripherie . . . . .	85
5.4.1.7	Verhalten bei toleranzabhängigen Abläufen . . . . .	85
5.4.2	Ablauf während der Testdurchführung . . . . .	85
5.4.2.1	Testverwaltung . . . . .	86
5.4.2.2	Testtreiber . . . . .	86
5.4.2.3	Roboterprogrammkontrolle . . . . .	88
5.4.2.4	Testüberwachung . . . . .	89
5.4.2.5	Ereigniserzeugung durch Bausteinmodul . . . . .	89

---

5.5	Komponenteninteraktion im Gesamtsystem . . . . .	90
5.6	Datenmodell und Modellerstellung . . . . .	91
5.6.1	Datenmodell des Gesamtsystems . . . . .	92
5.6.1.1	Aufbau des mechatronischen Zellenmodells . . . . .	92
5.6.1.2	Bestandteile eines Testfalls . . . . .	94
5.6.2	Erweitertes Modell für die Dynamiksimulation . . . . .	95
5.6.2.1	Geometriemodell . . . . .	95
5.6.2.2	Kinematikmodell . . . . .	96
5.6.2.3	Physikmodell . . . . .	97
5.6.3	Verhaltensmodellierung . . . . .	97
5.6.4	Signalverknüpfungen . . . . .	99
5.6.5	Virtuelle Sensoren . . . . .	100
5.6.5.1	Positionsbasierter Ansatz . . . . .	101
5.6.5.2	Kollisionsbasierter Ansatz . . . . .	101
5.6.5.3	Speicherung von Sensorikinformationen . . . . .	102
5.6.6	Bausteinkonfiguration . . . . .	103
5.6.7	Erstellung von Testszenarien . . . . .	105
5.6.8	Erstellung von Ereignissequenzen . . . . .	106
5.7	Zusammenfassung und Diskussion des Konzepts . . . . .	107
<b>6</b>	<b>Technische Realisierung</b>	<b>110</b>
6.1	Komponentenauswahl und Systemarchitektur . . . . .	110
6.2	Softwareprototyp PhySim . . . . .	112
6.2.1	Starrkörpersimulation . . . . .	113
6.2.2	Bausteinmodul und Bausteinkonfiguration . . . . .	114
6.2.2.1	Prüfung von Montagezuständen . . . . .	116
6.2.2.2	Kollisionsüberwachung zwischen Objektgruppen . . . . .	117
6.2.2.3	Unzulässige Kontaktsituationen . . . . .	117
6.2.2.4	Kollisionsfreie Erreichung von Teachpunkten . . . . .	118
6.2.2.5	Objektmanipulation . . . . .	119
6.2.3	3D-Visualisierung . . . . .	119
6.3	CASE-Werkzeug für Verhaltenssimulation und Testumgebung . . . . .	120
6.3.1	Verhaltenssimulation . . . . .	121
6.3.2	Testüberwachung und Testverwaltung . . . . .	122
6.3.3	Testtreiber . . . . .	123
6.3.4	Roboterprogrammkontrolle . . . . .	126
6.4	Modellerstellung . . . . .	127
6.4.1	Import von Geometrie- und Kinematikmodell . . . . .	128
6.4.2	Umsetzung des Physikmodells . . . . .	128

---

---

6.4.3	Integration von Sensoren . . . . .	130
6.5	Erstellung von Testszenarioszenarien und Ableitung von Ereignissequenzen . . . . .	134
6.5.1	Erstellung von Testszenarioszenarien . . . . .	134
6.5.2	Erstellung von Ereignissequenzen . . . . .	135
6.6	Zusammenfassung und Fazit . . . . .	136
<b>7</b>	<b>Anwendungsbeispiele</b>	<b>138</b>
7.1	Modell- und Aufgabenbeschreibung . . . . .	138
7.1.1	Positionierungsstrategie Werkstückträger . . . . .	140
7.1.2	Fügestrategie "Schraubenmontage" . . . . .	142
7.1.3	Fehlerbehandlungsstrategie "Schraube kommissionieren" . . . . .	144
7.2	Testszenarioszenarien und Ereignissequenzen . . . . .	145
7.2.1	Positionierungsstrategie Werkstückträger . . . . .	145
7.2.2	Fügestrategie "Schraubenmontage" . . . . .	147
7.2.3	Fehlerbehandlungsstrategie "Schraube kommissionieren" . . . . .	150
<b>8</b>	<b>Zusammenfassung und Ausblick</b>	<b>154</b>
8.1	Zusammenfassung und Fazit . . . . .	154
8.2	Ausblick . . . . .	156
<b>9</b>	<b>Literaturverzeichnis</b>	<b>158</b>

---

## Verzeichnis der Abkürzungen

AutomationML ..	Automation Markup Language
BMBF .....	Bundesministerium für Bildung und Forschung
CAD .....	Computer Aided Design
CAEX .....	Computer Aided Engineering Exchange
CAR .....	Computer Aided Robotics
CASE .....	Computer Aided Software Engineering
CATIA .....	Computer Aided Three-Dimensional Interactive Application
CIRC .....	Kreisbewegung
COM .....	Component Object Model
CPF .....	Conditioner Pipeline Framework
CSG .....	Constructive Solid Geometries
DELMIA .....	Digital Enterprise Lean Manufacturing Interactive Application
E/A .....	Eingangs-/ Ausgangssignal der Robotersteuerung
ENV .....	Environment
GPU .....	Graphical Processing Unit
GUI .....	Graphical User Interface
HS .....	Hilfssensor
HTML .....	Hypertext Markup Language
JT .....	Jupiter Tessellation
KDE .....	K Desktop Environment
KR .....	KUKA Roboter
KS .....	Koordinatensystem
LCP .....	Lineares Komplementaritätsproblem
LIN .....	Linearbewegung
NCP .....	Nichtlineares Komplementaritätsproblem
ODE .....	Open Dynamics Engine
OEE .....	overall equipment effectiveness
PA .....	Programmausführung
PLM .....	Product Lifecycle Management
PPU .....	Physical Processing Unit
PT .....	Programmtest
PTP .....	Point-to-point Bewegung
QT .....	Quasar Technologies
RB .....	Rigid Body (Starrkörper)
RC .....	Robot Control (Robotersteuerung)
RCS .....	Roboter Control Simulation

---

RRS .....	Realistic Robot Simulation
RT .....	Reflexaster
SK .....	Setzkasten
SKM .....	Starrkörpersimulation
SPS .....	speicherprogrammierbare Steuerung
STEP .....	Standard for the exchange of product model data
TCP .....	Tool Center Point
TP .....	Testprogramm
TRAMES .....	Transport-Prozesse in der mechatronischen Simulation
TU .....	Testumgebung
UML2 .....	Unified Modeling Language 2
UML2TP .....	UML2 Testing Profile
V .....	Visualisierung
VADE .....	Virtual Assembly Design Environment
VIPE .....	Virtuelle Roboterprogrammentwicklung
VRC .....	Virtual Robot Control
VRML .....	virtual reality modeling language
VS .....	Verhaltenssimulation
XML .....	Extensible Markup Language

---

## Verzeichnis der Formelzeichen und Symbole

$\alpha$ .....	Neigungswinkel
$\vec{d}$ .....	Offsetvektor
$O$ .....	Rechenaufwand
ch .....	Präfix für Methoden, welche durch Flowports aufgerufen werden
EP .....	Eckpunkt
ev .....	Kennzeichnung eines Ereignisses (ev für Event)
evT .....	Ereignisse, welche an das Testszenario gesendet werden
evTF .....	Ereignisse, welche an die Testumgebung gesendet werden (Testframework)
its .....	Präfix für die Benennung instanzierter Objekte
MP .....	Messpunkt
PHY_IN .....	Präfix für Eingangsparameter der Starrkörpersimulation (Kräfte)
PHY_OUT .....	Präfix für Ausgangsparameter der Starrkörpersimulation (Positionen)
R_IN .....	Präfix für Robotereingangssignale auf Seite der Simulationsumgebung
R_OUT .....	Präfix für Roboterausgangssignale auf Seite der Simulationsumgebung
V_IN .....	Präfix für Konfigurationsparameter der Verhaltensmodelle
V_LOK .....	Präfix für interne Parameter für Verhaltensmodelle
VP .....	Vorpunkt
X .....	kartesisches Koordinatensystem, X-Achse
Y .....	kartesisches Koordinatensystem, Y-Achse
Z .....	kartesisches Koordinatensystem, Z-Achse
ZP .....	Zwischenpunkt

---

## Verzeichnis der Abbildungen

1	Herausforderungen produzierender Unternehmen am Beispiel der Automobilindustrie	2
2	Aufbau und Struktur der vorliegenden Arbeit . . . . .	4
3	Weltweit installierte Industrieroboter nach Anwendungsgebiet (vgl. [IFR 10]) . . . . .	6
4	Bedeutung der roboternahen Peripherie für die Montageautomatisierung . . . . .	8
5	Relative Anweisungshäufigkeit in montagespezifischer Programmanalyse . . . . .	10
6	Positionierungsstrategie am Beispiel eines quadratischen Werkstückträgers . . . . .	11
7	Fügestrategien nach SCHWEIGERT (vgl. [Schw92]) . . . . .	12
8	Intensität der Funktionsbeeinträchtigung nach VOSSLOH . . . . .	13
9	Vorgehen zur Entwicklung von Montageroboterprogrammen (vgl. [Hein93]) . . . . .	15
10	Angepasstes “magisches” Dreieck in Anlehnung an HOFFMANN (vgl. [Hoff08]) . . . . .	17
11	Bewertung von Systemzuständen in Roboterprogrammen (vgl. [Mert04]) . . . . .	18
12	Fehlerklassen im realisierten Programm (vgl. [Mert04]) . . . . .	19
13	Reifegraderhöhung und Vorverlagerung von Programmieraufwänden . . . . .	21
14	Verringerung der Fehlerkosten durch VIPE (vgl. [Pfei08]) . . . . .	22
15	Entstehungsprozess von Montageanlagen ohne VIPE . . . . .	28
16	Zukünftiger Entstehungsprozess von Montageanlagen mit VIPE . . . . .	29
17	Programmierungsverfahren in Anlehnung an BERGER (vgl. [Berg06a]) . . . . .	35
18	Arten der Montagesimulation (vgl. [Eger08, Lenn00]) . . . . .	47
19	Simulationszyklus einer Starrkörpersimulation (in Anlehnung an [Pare02]) . . . . .	49
20	Einordnung bestehender Lösungen des Standes der Technik und Forschung . . . . .	63
21	Überblick des Kapitels Konzeption und Systementwurf . . . . .	69
22	Funktionaler Programmtest auf Basis beobachtbarer Programmzustände . . . . .	71
23	Gesamtsystem zur virtuellen Roboterprogrammentwicklung . . . . .	73
24	Simulationsablauf innerhalb einer Iteration in der Simulationsumgebung . . . . .	75
25	Bereitstellung des Signalabbildes durch die Verhaltenssimulation . . . . .	77
26	Zustandsprüfung “Schraube gefügt” durch virtuelle Montagesensoren . . . . .	80
27	Kollisionskonfigurationen aus verschiedenen Objektgruppen . . . . .	81
28	Verklemmungssituation während Montageaufgabe “Schraube fügen” . . . . .	82
29	Koordinatensysteme zur Instanziierung virtueller Hilfssensoren . . . . .	84
30	Zustandsautomat zur Testdurchführung . . . . .	87
31	Roboterprogrammkontrolle für die Testdurchführung . . . . .	88
32	Komponenteninteraktion im Gesamtsystem . . . . .	90
33	Komponenteninteraktion der Simulationsumgebung . . . . .	91
34	Verhaltensmodell eines Parallelbackengreifers . . . . .	98
35	Signalverknüpfung eines induktiven Linearwegensensors und Robotersteuerung . . . . .	100
36	Linearwegensensor zur Bestimmung der Backengreiferstellung . . . . .	101
37	Kollisionsbasierter Ansatz für virtuelle Sensoren . . . . .	102

---

38	Speicherung von Sensorikinformationen mithilfe AutomationML . . . . .	103
39	Abstrahierte Darstellung eines Test szenarios als Zustandsautomat . . . . .	106
40	Ereignissequenz als Sequenzdiagramm . . . . .	107
41	Aufbau des Softwareprototyps . . . . .	111
42	Arbeitsplatz für die virtuelle Roboterprogrammentwicklung . . . . .	112
43	Bausteinkonfiguration in PhySim . . . . .	115
44	Konfiguration und Visualisierung von Montagesensoren . . . . .	117
45	Aktiviere Hilssensoren an Toolpunkt und Teachpunkten . . . . .	118
46	Parallele Ansichten der Szene zur verbesserten Benutzerwahrnehmung . . . . .	120
47	Zustandsautomaten während der Simulation im CASE-Werkzeug Rhapsody . . . . .	121
48	Ablauf zur Abarbeitung der Testfallliste durch PhySim . . . . .	122
49	Baumdarstellung Testverwaltung in Rhapsody . . . . .	123
50	Realisierung des Testtreibers als Zustandsautomat . . . . .	124
51	Port Konzept für Kommunikation mit Testtreiber . . . . .	125
52	Zustandsautomat “Automatik Extern” für die Roboterprogrammkontrolle . . . . .	126
53	Export von Geometrie- und Kinematikmodell mittels Graphic CPF . . . . .	128
54	Materialeigenschaften am Beispiel einer Schraube . . . . .	129
55	Starrkörperliste und physikalische Eigenschaften am Beispiel einer Schraube . . . . .	129
56	Reflextaster a) Realisierung mit der Bullet Bibliothek, b) Verhaltensmodell . . . . .	131
57	Linearwegsensor a) Geometriemodell, b) Verhaltensmodell . . . . .	133
58	Test szenario als Zustandsautomat am Beispiel “Teil greifen” . . . . .	135
59	Soll-Ereignissequenz für das Beispiel “Teil greifen” . . . . .	136
60	Zellenmodell für Anwendungsbeispiel Schraubenmontage . . . . .	139
61	Ablaufbeschreibung für Positionierungsstrategie Werkstückträger . . . . .	141
62	Fügestrategie Schraubenmontage . . . . .	143
63	Fehlerbehandlungsstrategie Schrauben kommissionieren . . . . .	144
64	Test szenario Bestimmung der Werkstückträgerposition . . . . .	146
65	Ereignissequenz Bestimmung der Werkstückträgerposition . . . . .	147
66	Test szenario für Schraubenmontage . . . . .	148
67	Ereignissequenz für Schraubenmontage . . . . .	150
68	Test szenario für Teilablauf “Schraube kommissionieren” . . . . .	151
69	Ereignissequenz Idealfall - Schraube wird kommissioniert . . . . .	152
70	Ereignissequenz Fehlerfall - Schraube wird in Abwurfkasten transportiert . . . . .	153

---

## Verzeichnis der Tabellen

1	Anforderungen an die Programmausführung . . . . .	32
2	Anforderungen an den Programmtest . . . . .	32
3	Vergleich verschiedener Datenaustauschformate bezüglich VIPE-Eignung . . . . .	61
4	Bewertung bestehender Lösungen zur virtuellen Roboterprogrammentwicklung . . . . .	66
5	Mögliche Teststufen während der Programmentwicklung . . . . .	72
6	Störungsursachen und Störungsverhalten mechatronischer Komponenten . . . . .	83
7	Informationsarten und deren Verwendung in der Simulationsumgebung . . . . .	93
8	Informationsarten und -verwendung für die Testumgebung . . . . .	95
9	Notwendige Informationen für die Konfiguration der Bausteine . . . . .	104
10	Konfigurationsparameter und Ereignisse der realisierten Bausteine . . . . .	116