Technik

Lukas Beyer

Entwicklung eines Konzepts zur realistischen Simulation des dynamischen Verhaltens von Industrierobotern

Diplomarbeit

Diplom.de

Bibliografische Information der Deutschen Nationalbibliothek:

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

ISBN: 9783832461843

Lukas Beyer

Entwicklung eines Konzepts zur realistischen Simulation des dynamischen Verhaltens von Industrierobotern

Diplom.de

Lukas Beyer

Entwicklung eines Konzepts zur realistischen Simulation des dynamischen Verhaltens von Industrierobotern

Diplomarbeit
an der Universität der Bundeswehr Hamburg
Fachbereich Maschinenbau
vier Monate Bearbeitungsdauer
April 1996 Abgabe

Diplom.de

Diplomica GmbH
Hermannstal 119k
22119 Hamburg

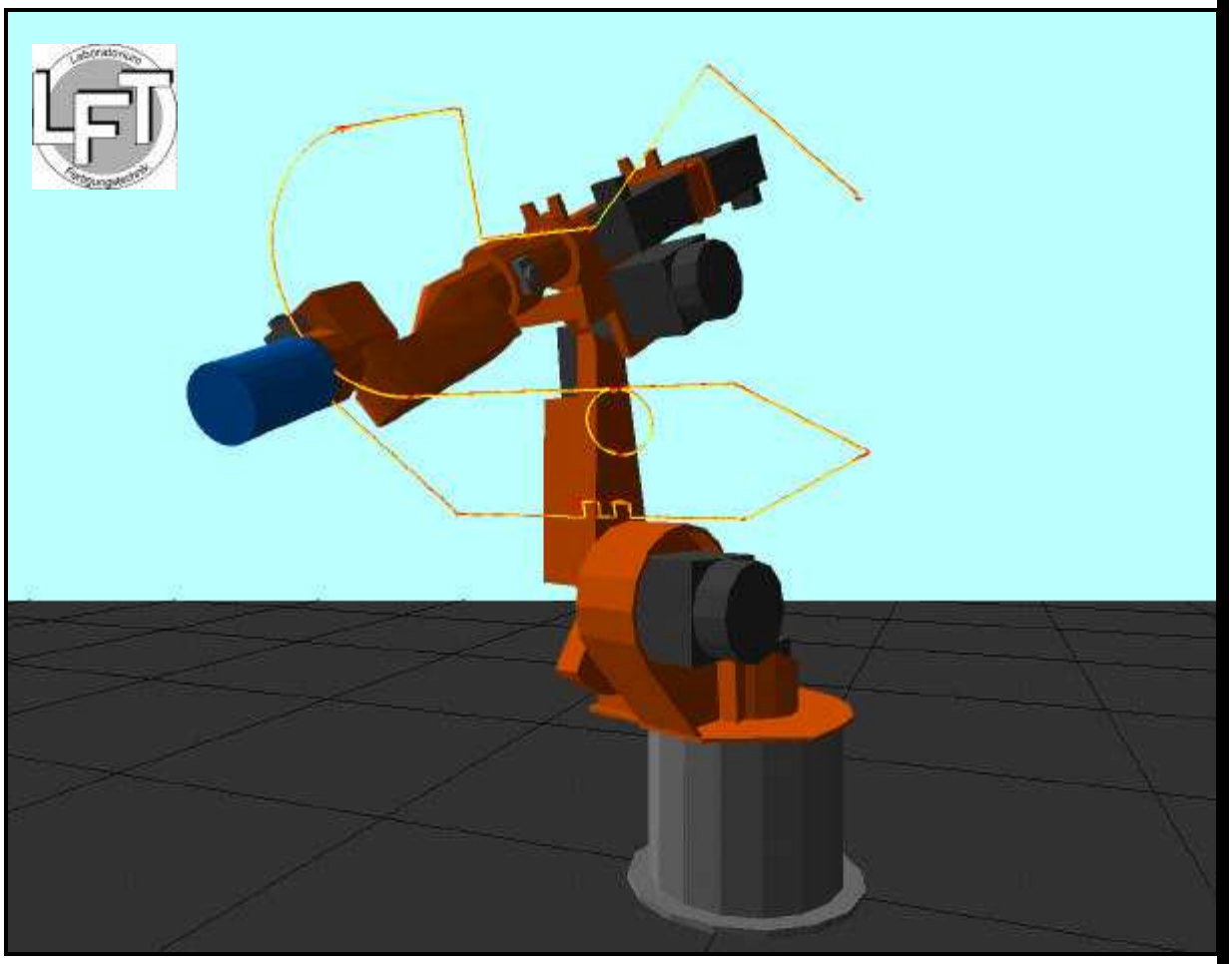
Fon: 040 / 655 99 20
Fax: 040 / 655 99 222

agentur@diplom.de
www.diplom.de

ID 6184

ID 6184
Beyer, Lukas: Entwicklung eines Konzepts zur realistischen Simulation des dynamischen Verhaltens von Industrierobotern
Hamburg: Diplomica GmbH, 2002
Zugl.: Hamburg, Universität der Bundeswehr, Diplomarbeit, 1996

Diplomica GmbH
http://www.diplom.de, Hamburg 2002
Printed in Germany



Dynamische Robotersimulation

Inhaltsverzeichnis

1. Einleitung

Die vorliegende Arbeit ist das Ergebnis einer grundlegenden Untersuchung zur Dynamiksimulation von Industrierobotern (IR), die im Rahmen einer Diplomarbeit am Institut für Konstruktions- und Fertigungstechnik der Universität der Bundeswehr Hamburg durchgeführt wurde. Ziel dieser Untersuchung war es zu zeigen, daß es mit dem am Institut eingesetzten graphischen Simulationssystem zur Roboterprogrammierung möglich ist, das Bewegungsverhalten eines Industrieroboters sowie die Parameter seines realen Regelkreises unter Berücksichtigung der dynamischen Kenngrößen (wie beispielsweise Massen und Trägheiten) nachzubilden.

1.1. Themenüberblick

Durch steigenden Wettbewerbsdruck auf globalisierten Märkten sehen sich heute viele Industrieunternehmen dem Zwang ausgesetzt, ihre Produktivität und Flexibilität stetig zu erhöhen. Diese Bestrebungen weisen häufig in die Richtung der rechnerintegrierten Produktion (CIM), dem vollständigen Datenverbund vom Auftragseingang bis zum Vertrieb. Teil dieses Gesamtkonzepts ist die rechnerunterstützte Fertigung (CAM), deren Grundlage die NC-Technik im Werkzeugmaschinenbau bildet, welche heute einen hohen technischen Standard erreicht hat. In diesem Rahmen werden auch zunehmend moderne Industrieroboter eingesetzt, da sie - unter anderem wegen der rasanten Entwicklung der Mikroelektronik - an mehr und mehr Fertigungsaufgaben angepaßt werden können [6].

Die Bedeutung von Industrierobotern ist vor allem in Folge ihrer hohen Flexibilität in einem ständigen Wachstum begriffen. Aus diesem Grund, aber auch wegen ihrer sinkenden Preise wird der Anteil von Robotern in der industriellen Fertigung weiter ansteigen, wodurch der Mensch von monotonen und gefährlichen Arbeiten entlastet und die Produktivität der Betriebe durch Rationalisierung erhöht werden kann. Entwicklung und Einsatz von Industrierobotern können somit einer hochentwickelten und technologieorientierten Wirtschaft positive Impulse verleihen.

Industrieroboter werden auch *frei programmierbare Manipulatoren* bzw. *Handhabungsgeräte* genannt (Siehe Abbildung 1). Sie sind in den letzten Jahren stetig verbessert

und weiterentwickelt worden. Diese Innovationen beziehen sich nicht nur auf mechanische und elektronische Komponenten wie z.B. Sensorik und Motorik, sondern offenbaren sich auch in der Entwicklung neuer Verfahren zur Steuerung und Programmierung. Speziell auf diesen beiden Forschungsfeldern ist das dynamische Verhalten der Roboter Gegenstand der Untersuchungen: Zum einen erfordert die Entwicklung selbsteinstellender, modellgestützter Regelungsverfahren, welche die Bahngenauigkeit verbessern, den Entwurf eines kinetischen Modells des Roboters, zum anderen benötigt man dieses auch zur dynamischen Simulation eines IR in graphischen Off-line-Programmiersystemen.

Konventionelle Simulationssysteme liefern beim Ablauf eines Bewegungsprogramms exakt die geplante, meist mit eigenen Bewegungsplanungsalgorithmen berechnete Bahn, ohne die Abweichungen zu berücksichtigen, die sich auf Grund der dynamischen Eigenschaften der Maschine ergeben und die sich in manchen Fällen bis in den Zentimeterbereich erstrecken können. Die wirklichkeitsnahe Simulation der Dynamik ist somit einer von zahlreichen Schritten auf dem Weg zu einer realistischen Simulation von Industrierobotern.

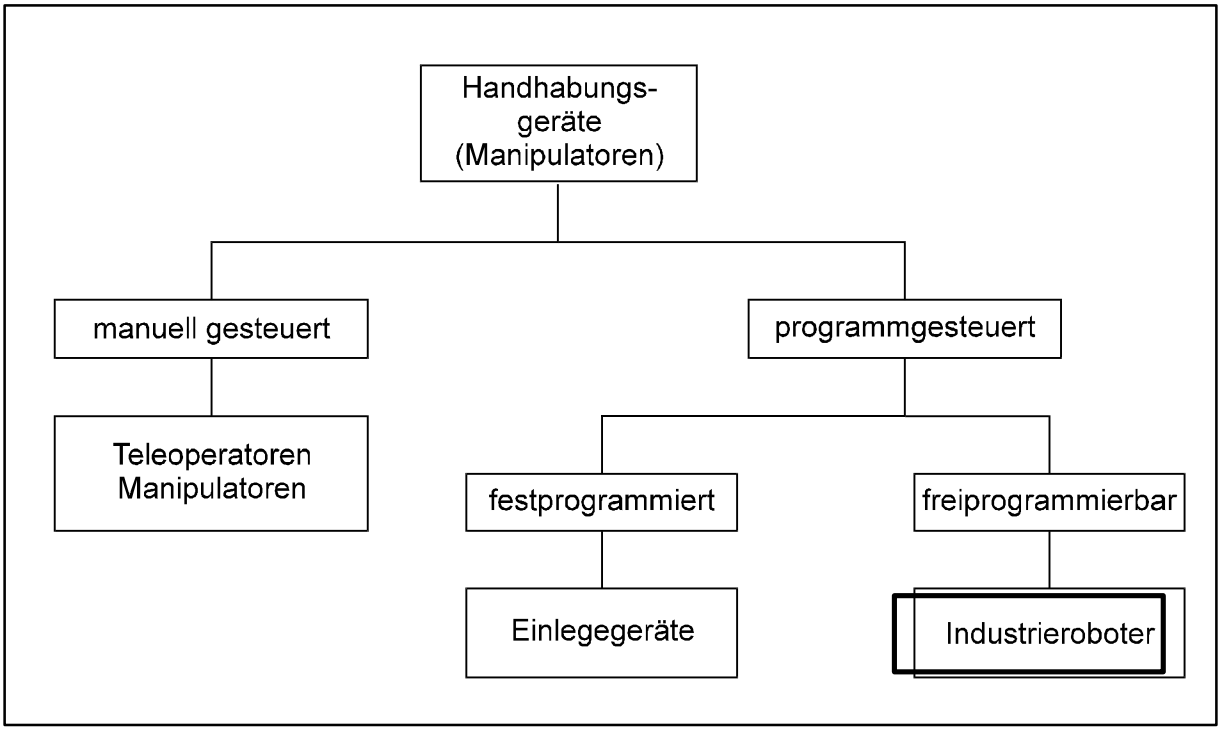


Abb. 1: Einteilung von Handhabungsgeräten [5]

1.2. Aufgabenstellung

Ausgehend von der vorgenommenen Themenabgrenzung ergibt sich nun das Kernproblem der Untersuchung. Dieses besteht zum einen darin, ein allgemein gültiges Modell des Industrieroboters zu beschreiben, welches dynamische Prozesse mit einschließt und die dazu notwendigen Parameter beinhaltet, zum anderen in der praktischen Implementierung der Dynamikkomponenten dieses Modells in Form lauffähiger Programmodule zur Einbindung in das Simulationssystem.

Bezüglich der *Modellbildung* ist zu klären, welche dynamischen Einflüsse im einzelnen berücksichtigt werden sollen. Da hier eine Vielzahl von möglichen Parametern existiert, muß das reale Geschehen in sinnvoller Weise abstrahiert werden, respektive der zur Verfügung stehenden Daten des realen Roboters.

Im Hinblick auf die *Realisierung* der Dynamiksimulation lautet die Frage, wie das zuvor entworfene dynamische Modell am besten in einen Programmcode überführt werden kann. Hierbei sind vor allem die Anforderungen zu berücksichtigen, die das Simulationssystem an die zu entwickelnden Module stellt - ebenso sollten die Möglichkeiten, die es z.B. durch vorhandene Dynamikfunktionen bietet, umfassend ausgenutzt werden.

Konkret ergeben sich aus diesen allgemeinen Anforderungen an das zu realisierende System folgende Einzelaufgaben:

- Einarbeitung in die Theorie der Dynamiksimulation
- Vergleich verschiedener Reglerkonzepte
- Recherche der erforderlichen Parameter des realen Roboters
- Entwurf der Systemmodule zur Realisierung von Regler und physikalischem Modell des IR, aufbauend auf vorhandene Prototypen
- Implementierung der Module im System, Anpassung der Reglerparameter
- Durchführung und Auswertung verschiedener Tests

Ziel der Untersuchungen ist es, anhand des Simulationssystems vorwiegend *qualitative* Aussagen über die zu erwartenden Abweichungen eines Roboters von seiner programmierten Sollbahn treffen zu können. Präzise *quantitative* Ergebnisse sind unter den gegebenen Voraussetzungen (wie etwa den zur Verfügung stehenden Daten) noch nicht zu erwarten. Dennoch könnte die Dynamiksimulation in Zukunft vermehrt praktische Anwendung bei der Off-line-Programmierung bestimmter Applikationen finden, beispielsweise beim Laser-schweißen, da hierbei mit ihrer Hilfe deutliche Verbesserungen der Bahngenauigkeit erzielt werden können. Außerdem könnte das Verfahren bei der Optimierung von Reglereinstellungen Anwendung finden. Voraussetzung ist jedoch ein exaktes Modell der Kinetik und des (Lage- bzw. Geschwindigkeits-) Reglers des jeweiligen Roboters.

1.3. Gang der Untersuchung

Zunächst sollen in einem allgemeinen Teil (Kapitel 2) die Grundlagen der Programmierung von Industrierobotern erläutert werden, speziell die der Programmierung mittels graphischer Simulationssysteme. Auf diese Weise wird der Zusammenhang hergestellt, in dem die Dynamiksimulation mit der gesamten Thematik steht, denn sie stellt trotz ihrer Komplexität nur einen Teilaspekt in der Vielzahl der Einflußfaktoren auf die Roboterprogrammierung dar. Zusätzlich erfolgt eine Darstellung der verschiedenen Probleme, die sich bei der Robotersimulation im allgemeinen ergeben, d.h. die Ursachen der Abweichungen zwischen realem und simuliertem Verhalten werden erläutert. Außerdem werden mögliche Lösungsansätze vorgestellt.

Die der Dynamiksimulation zu Grunde liegende Theorie wird in Kapitel 3 behandelt. Dazu wird zunächst auf die Möglichkeiten des Verfahrens sowie auf gewisse Schwierigkeiten bei der Umsetzung eingegangen. Im folgenden Abschnitt wird dann das mathematische Modell für die Simulation entwickelt. Dafür wird in dieser Arbeit eine vereinfachte kinetische Nach-bildung eines Roboters herangezogen, die jedoch die wesentlichen Parameter berücksichtigt. Grundsätzlich wird auf bereits bekannte Arten der Modellierung zurückgegriffen.

Im vierten Kapitel wird die Umsetzung des theoretischen Modells in lauffähige Programmodule innerhalb des Simulationssystems dargestellt. Nach einer kurzen Erläuterung des benutzten Off-line-Programmier- und Simulationssystems sowie vorhandener Prototypen der

Dynamikmodule folgt die Beschreibung der eigentlichen Implementierung. Dabei wurde auf systematisches Vorgehen Wert gelegt. So kamen bei der Programmierung gewisse Grundsätze des Software Engineering zum Einsatz wie etwa strukturierte Programmierung, Top-Down-Methode und umfassende Dokumentation. Damit sollte eine möglichst hohe Transparenz der entwickelten Software sichergestellt werden, so daß Nachfolgeprojekte ohne einen allzu hohen Einarbeitungsaufwand in Angriff genommen werden können.

Einige Aspekte des Themas mußten im Rahmen des Projekts auch ausgespart bleiben, da entsprechende Ressourcen nicht zur Verfügung standen. So konnten z.B. die Getriebe-elastizitäten nicht modelliert werden, da keine entsprechenden Daten vorlagen. Aus Zeitgründen mußte auch Test und Analyse des fertigen Systems auf einige Fälle begrenzt werden, die sicherlich nicht alle möglichen abdecken. Für eine qualitative Beurteilung sind die erzielten Ergebnisse jedoch zufriedenstellend; eine quantitative Untersuchung kann in einer eventuellen Anschlußarbeit erfolgen.

2. Programmierung von Industrierobotern

In diesem Kapitel erfolgt eine generelle Darstellung der verschiedenen Techniken zur Roboterprogrammierung, wodurch der Bezug zwischen dem speziellen Thema der Arbeit und der allgemeinen Thematik des Forschungsgebietes hergestellt wird. Dazu sollen zunächst die unterschiedlichen Programmierverfahren kurz erläutert werden. In einem weiteren Abschnitt wird dann näher auf die Programmierung mittels Robotersimulation eingegangen. Als letzter Unterpunkt folgt schließlich eine Beschreibung und Beurteilung der dabei auftretenden Probleme.

2.1. Überblick über die verschiedenen Programmierverfahren

Ein IR läßt sich abstrahiert als Blockschaltbild darstellen, welches aus Steuerung (Rechner), Meßglied (Sensoren) und Stellglied (Antriebe) besteht. Der Steuerung werden von außen Sollwerte vorgegeben, d.h. Positionen, Geschwindigkeiten etc., welche die Motoren realisieren sollen. Diese Vorgabe geschieht in Form eines Programms, das in einer bestimmten Programmiersprache in die Steuerung eingegeben wird.

Es existiert hier keine einheitliche Roboterprogrammiersprache, sondern viele verschiedene, die sich jedoch in drei Gruppen einteilen lassen: zum ersten gibt es ältere, häufig aus Sprachen zur Programmierung von NC-Maschinen (z.B. UNC) entwickelte Systeme, zum zweiten existieren anwendungsspezifische Sprachen (z.B. AL, AUTOPASS, ROBEX, etc.), die jeweils für bestimmte Aufgaben (beispielsweise zum Schweißen) eingesetzt werden. Drittens finden seit mehreren Jahren höhere Programmiersprachen wie FORTRAN, PASCAL, C oder BASIC bzw. Derivate davon Anwendung (z.B. S-IRL bei Siemens-Steuerungen). Solche höheren Sprachen gewinnen heute immer mehr an Bedeutung, weil damit erstellte Programme gut strukturiert und leicht lesbar sind.

Als größter gemeinsamer Nenner dieser Vielfalt und als Ansatz zu einem einheitlichen Code wurde bereits 1985 IRDATA (Industrial Robot Data) konzipiert, eine sehr abstrakte Sprache, die mit Assembler vergleichbar ist und die für die Übersetzung des Einsatzes von Pre- und Postprozessoren bedarf. IRDATA ist genormt und erfüllt die Funktion eines Schnittstellencodes [5].

Die Programmiersprachen sind jedoch nur die Werkzeuge zur Steuerung eines Industrieroboters. Demgegenüber gibt es verschiedene Methoden der Programmierung, auf die im folgenden eingegangen werden soll. Man unterscheidet dabei prinzipiell zwischen *direkter* und *indirekter* Programmierung, je nachdem, ob die Implementierung des Programms am IR selbst oder unabhängig davon vorgenommen wird, z.B. in einer zentralen Programmier-abteilung. Das folgende Diagramm stellt die verschiedenen Verfahren anschaulich dar.

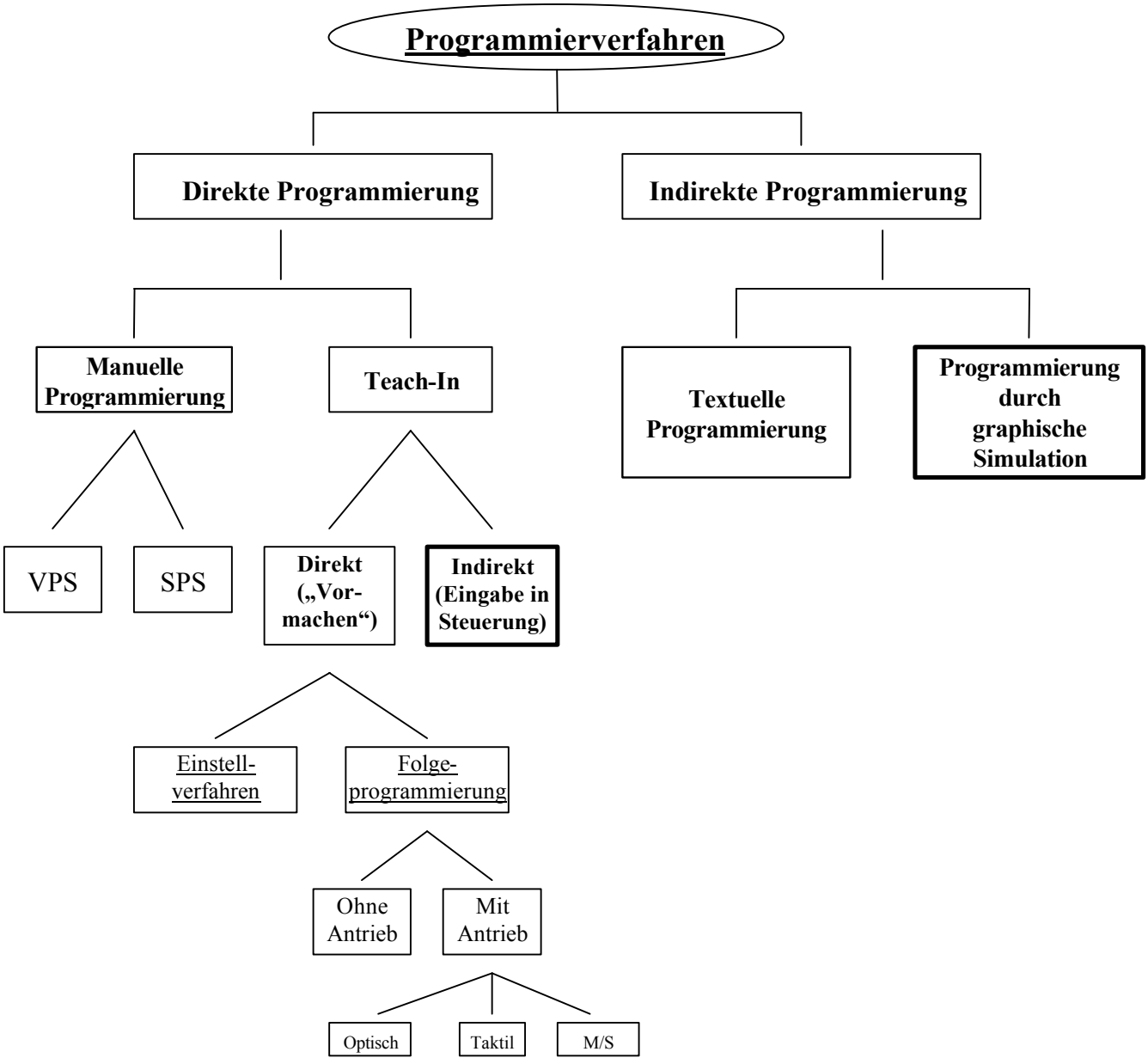


Abb. 2: Schema der Programmierverfahren für Industrieroboter

2.1.1. Direkte Programmierung

Diese Methode wird auch *On-line-Programmierung* genannt, da der IR an seinem Einsatzort programmiert wird und dabei eingeschaltet ist. Die modernen On-line-Verfahren (Teach-In-Verfahren) arbeiten alle auf Softwarebasis. Der Vorteil der On-line-Programmierung liegt darin, daß die Programme sofort getestet und korrigiert werden können. Allerdings verursacht sie teure Stillstandszeiten, da die Fertigungszelle währenddessen nicht für die Produktion zur Verfügung steht.

Direktes Teach-In

Beim Einstellverfahren werden die Bremsen der Roboterachsen gelöst und der Roboterarm von Hand (ohne Antrieb) in die gewünschte Position und Orientierung gebracht, die dann durch Betätigen von Funktionstasten abgespeichert wird. So läßt sich eine große Anzahl von Haltepunkten einlesen, was allerdings entsprechenden Speicherplatz und Lagesensoren an den Roboterachsen erforderlich macht. Vorteil dieses Verfahrens ist die einfache Handhabung und die Sicherheit des Bedieners vor unkontrollierten Bewegungen des Industrieroboters.

Die Folgeprogrammierung funktioniert ähnlich wie das Einstellverfahren und wird auch Playback genannt. Allerdings werden hier nicht nur einzelne Haltepunkte gespeichert, vielmehr wird die abgefahrene Bahn mit einer bestimmten Frequenz abgetastet. Die gesamte Bewegung kann so festgehalten und jederzeit (eventuell mit höherer Geschwindigkeit) wiederholt werden. Beim Playback ohne eigenen Antrieb führt der Programmierer den Roboter „an der Hand“, wobei dessen Bremsen gelöst sind. Diese Bewegung wird von der Steuerung registriert und gespeichert. Bei Playback mit eingeschaltetem Antrieb verfährt der IR dagegen aus eigener Kraft. Er folgt dabei den Bewegungen eines kleineren und leichteren Roboterarms (Master/Slave-Methode) oder mit einem an seinem Werkzeug angebrachten Sensor einem Signal des Programmierers. Man unterscheidet dabei zwischen optischer und taktiler Folgeprogrammierung: Bei der ersten Methode reagiert der Sensor auf eine Lichtquelle, die vom Programmierer geführt wird, bei der zweiten kommt ein Tastsensor zum Einsatz, der manuelle Signale des Bedieners registriert. Die Folgeprogrammierung ist ein einfaches und schnelles Verfahren, das z.B. beim Lackieren eingesetzt wird.

Indirektes Teach-In

Das Indirekte Teach-In ist das zur Zeit am häufigsten angewandte Programmierverfahren. Mittels einer handlichen Tastatur (Programmierhandgerät, PHG) wird der IR in einem kartesischen Koordinatensystem oder auch achsspezifisch an die Sollpositionen herangefahren, die meistens über Musterwerkstücke bestimmt und dann in Form eines lesbaren Programms abgelegt werden. Die Bewegung kann dabei durch Punkt-zu-Punkt-Steuerung (point to point, PTP) oder durch Bahnsteuerung (continuous path, CP) beschrieben werden: Im einen Fall fährt der IR die Positionen auf beliebigem Weg an, im anderen auf einer genau vorgegebenen, durch einen Linear- oder Zirkularinterpolator erzeugten Bahn. Zusätzlich zu den Wegpunkten können weitere Informationen und Anweisungen eingegeben werden, z.B. in Bezug auf Geschwindigkeiten, Zeitspannen, Zählwerk, Greiferbetätigung, Sensoren, Unterprogrammaufrufe und ähnliches. So entsteht ein Programm aus Achskoordinaten und Zusatzinformationen, das jederzeit geladen und mit einer hohen Wiederholgenauigkeit ausgeführt werden kann. Die Vorteile des Teach-In sind, daß der IR sehr genau positioniert werden kann und daß zusätzliche Anweisungen in die Steuerung eingegeben werden können. Damit entspricht dieses On-line-Verfahren am ehesten dem, was man gemeinhin unter Programmierung versteht.

2.1.2. Indirekte Programmierung

Die Verfahren der direkten Programmierung haben allesamt den Nachteil, daß der IR während der Programmerstellung für die Produktion ausfällt und somit Fertigungszellen oder ganze Produktionslinien für Tage oder sogar Wochen stillgelegt werden müssen. Dies stellt einen wichtigen Kostenfaktor dar.

Deshalb wurde schon frühzeitig versucht, die Programmierung in einen Bereich außerhalb der Fertigung zu verlagern, z.B. in die Arbeitsvorbereitung (*Off-line-Programmierung*). Dies wurde mit der Einführung höherer Programmiersprachen im größeren Rahmen möglich und führte zunächst zur textuellen Programmierung.

Textuelle Programmierung

Unter textueller Programmierung versteht man die symbolische Beschreibung von Operationen und Daten in Form von Zeichenfolgen, d.h. der Programmierer erstellt einen lesbaren Text in einer bestimmten Programmiersprache, der von einem Programm (Compiler) gelesen und codiert werden muß. Vielfach wird dabei auch eine Syntaxüberprüfung durchgeführt.