

**Inhalt:**

- 1 Geltungsbereich
- 2 Zweck
- 3 Offline-Programmierung von Industrierobotern
- 3.1 Programmierverfahren
- 3.2 Typischer Arbeitsablauf
- 3.3 Hardware- und Software-Umgebung
- 3.4 Anwendungen
- 4 Kriterienkatalog zur Systembeurteilung
- 5 Wirtschaftlichkeitsaspekte
- 6 Mitgeltende Normen und Richtlinien

**Begriffe**

|         |   |
|---------|---|
| CAD     | Computer Aided Design                                       |
| DXF     | Data Exchange Format (CAD-Schnittstelle)                    |
| IGES    | Initial Graphics Exchange Specification (CAD-Schnittstelle) |
| Offline | Programmierung an externem Rechner                          |
| OLP     | Offline Programmierung                                      |
| Online  | Programmierung mit Industrieroboter                         |
| OpenGL  | Open Graphical Language                                     |
| STEP    | Standard for the Exchange of Product Model Data (ISO 10303) |
| STL     | Stereolithographie-Language (CAD-Schnittstelle)             |
| VDAFS   | VDA-Flächenschnittstelle                                    |
| Viewer  | Software zur Visualisierung von 3D Szenen im VRML Format    |
| VRML    | Virtual Reality Modeling Language                           |

**1 Geltungsbereich**

Die in diesem Merkblatt aufgeführten Darstellungen, Begriffserläuterungen und Hinweise gelten für den Einsatz von Offline-Programmiersystemen beim Schweißen, Schneiden, Kleben und Beschichten mit Industrierobotern.

**2 Zweck**

Ziel dieses Merkblatts ist es, Begriffserläuterungen und nützliche Hinweise zum Einsatz von Offline-Programmiersystemen für Industrieroboter zu geben. Dabei ist beabsichtigt, den Leser über technische Anforderungen, mögliche Probleme und zu berücksichtigende Randbedingungen zu informieren. Ein Katalog mit wesentlichen Kriterien zur Beurteilung heute verfügbarer OLP-Systeme soll den gegenwärtigen Stand der Entwicklungen transparenter machen und mögliche Investitionsentscheidungen unterstützen.

Das Merkblatt versucht, einen umfassenden Überblick über den gegenwärtigen Entwicklungsstand von OLP-Systemen zu geben. Es erhebt jedoch keinen Anspruch auf Vollständigkeit.

Diese Veröffentlichung wurde von einer Gruppe erfahrener Fachleute in ehrenamtlicher Gemeinschaftsarbeit erstellt und wird als eine wichtige Erkenntnisquelle zur Beurteilung empfohlen. Der Anwender muss jeweils prüfen, wie weit der Inhalt auf seinen speziellen Fall anwendbar und ob die ihm vorliegende Fassung noch gültig ist. Eine Haftung des DVS und derjenigen, die an der Ausarbeitung beteiligt waren, ist ausgeschlossen.

**3 Offline-Programmierung von Industrierobotern**

**3.1 Programmierverfahren**

Bei der Programmierung von Industrierobotern unterscheidet man im Wesentlichen die Online-Verfahren und die Offline-Verfahren.

Die **Online Programmierung** erfolgt unmittelbar am realen Bauteil oder Objekt. Dabei ist der Roboter in den Programmierprozess eingebunden und kann während dieser Zeit nicht für Produktionsaufgaben genutzt werden. Das in diesem Zusammenhang in der Praxis am häufigsten verwendete Verfahren ist die „Teach-in“ Programmierung.

Bei der **Offline-Programmierung** wird die Erstellung von Roboterprogrammen auf einem externen Rechner abseits der Fertigungsstation durchgeführt. Dabei kann der Roboter – im Gegensatz zur Online-Programmierung – weiterhin für Produktionsaufgaben genutzt werden. Mehr oder weniger lange Stillstandszeiten, die bei einer Online-Programmierung unumgänglich sind, lassen sich durch die Offline-Programmierung auf ein Minimum reduzieren. Dieses ist besonders von Vorteil, wenn häufig neue Roboterprogramme zu erstellen sind oder bereits vorhandene Programme geändert bzw. an oft wechselnde Fertigungsaufgaben angepasst werden müssen.

Bei den heute bekannten Offline Programmierverfahren sind drei generelle Verfahrensvarianten zu unterscheiden.

Dieses sind die:

- textuelle Offline Programmierung
- graphisch-interaktive Offline Programmierung
- CAD-basierte automatische Programm-Erzeugung

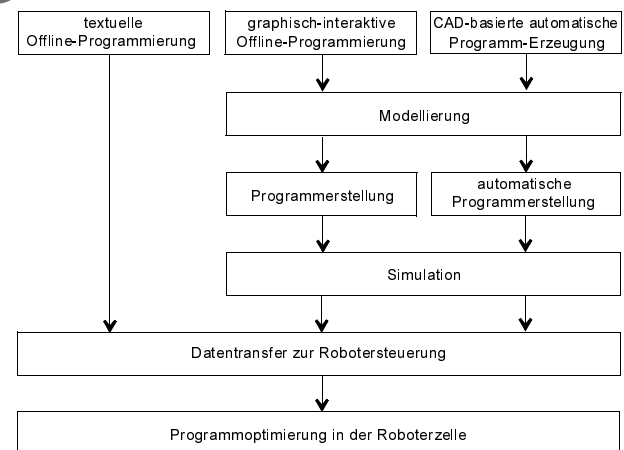


Bild 1. Offline-Programmierverfahren.

**3.1.1 Textuelle Offline-Programmierung**

Bei der textuellen Offline Programmierung wird das Roboterprogramm Befehl für Befehl in Textform erstellt. Dieses geschieht

an einem externen Rechner mit Hilfe eines Text-Editors. Zur Anwendung kommen dabei Programmiersprachen, die in der Regel herstellerabhängig sind.

Der Nutzer muss bei der textuellen Offline-Programmierung über genaue Kenntnisse der jeweiligen Sprachsyntax verfügen und gleichzeitig ein hohes Abstraktionsvermögen mitbringen, um den Programmablauf folgerichtig Schritt für Schritt beschreiben zu können.

Die Arbeitspositionen, die der Roboter anfahren soll, lassen sich bei dieser Methode in absoluten oder auch relativen Koordinatenangaben programmieren. Alternativ werden auch nur sogenannte Positionsmarken im Programmtext definiert. Die korrekten Raumpositionen müssen dann nachträglich durch „Teachen“ unmittelbar am Bauteil nachgetragen werden.

Die textuelle Offline-Programmierung ist zwar eine kostengünstige Variante. Sie setzt jedoch umfangreiche Programmierkenntnisse voraus.

### 3.1.2 Graphisch-interaktive Offline-Programmierung

Mit der Entwicklung der Computer-Grafik und der zunehmenden Leistungsfähigkeit der Rechner haben graphisch-interaktive Offline-Programmiersysteme in den letzten Jahren zunehmend an Bedeutung gewonnen. Sie ermöglichen heute mit Hilfe von 3D-Grafik-Funktionen eine vollständige Abbildung der realen Roboterzelle in ein dreidimensionales Simulationsmodell. Alle Bewegungen des Roboters und auch zugehöriger peripherer Achsen lassen sich am Simulationsmodell realitätsnah nachbilden. Damit sind die Voraussetzungen gegeben, um eine komplette Programmierung der Fertigungsaufgabe am Computer-Bildschirm zu ermöglichen.

Häufige Verwendung findet das sogenannte „graphische Teach-in“. Dabei wird das Robotermodell wie beim realen Programmieren durch „Teachen“ von Raum- bzw. Bahnpositionen und Orientierungen für die Werkzeugbewegung programmiert. Die dabei notwendigen Bewegungen des Robotermodells, eventuell zugeordneter Positionierer und auch eingesetzter weiterer externer Achsen werden zu diesem Zweck vom Computer animiert und als bewegte 3D-Grafik dargestellt.

Parametereingaben, z. B. zur Festlegung des Schweißprozesses, erfolgen meist über spezielle Eingabemasken. Wenn nötig, die Programmierstellung notwendige Funktionen lassen sich über entsprechende Menüs bzw. Menüleisten aktivieren.

Nach der Erstellung des Roboterprogramms am Computer-Bildschirm ermöglichen die Offline-Programmiersysteme heute auch eine Simulation der Abläufe. Mit ihrer Hilfe lassen sich die offline erstellten Roboterprogramme komplett in frei wählbaren Zeitrasteren auf ihre Funktionsfähigkeit, Erreichbarkeit der Bahnpunkte und Kollisionsfreiheit überprüfen. Systemabhängig sind auch Zeitstudien durchführbar. Auf diese Weise können bereits komplette Fertigungszeiten abgeschätzt und gegebenenfalls eine Optimierung der Abläufe vor am Bildschirm durchgeführt werden, ohne dass hierzu der reale Roboter benötigt wird.

Abhängig von ihrer Leistungsfähigkeit verfügen die angebotenen OLP-Systeme über einen mehr oder weniger komplexen Funktionsumfang. Die Handhabung setzt oft umfangreiche Systemkenntnisse voraus. Diese helfen vorab im Rahmen entsprechender Schulungsmaßnahmen erworben und in der praktischen Anwendung vertieft werden, um einen erfolgreichen Einsatz zu gewährleisten.

Waren graphisch-interaktive Offline-Programmier- und Simulationssysteme für Roboter früher ausschließlich nur auf Hochleistungsrechner, wie z. B. Workstations, lauffähig, so erobern heute zusehends PC-basierte Systeme den Markt. Angeboten werden die Programmiersysteme sowohl von Roboterherstellern für ihre speziellen Produkte, aber auch von Software-Häusern. Sie stellen dem Nutzer spezielle Robotermodell-Bibliotheken und zugehörige Funktionssoftware für die flexible Anwendung zur Verfügung.

### 3.1.3 CAD-basierte automatische Programm-Erzeugung

Die automatische CAD-basierte Programm-Erzeugung ermöglicht eine Generierung kompletter Bearbeitungsprogramme auf der Grundlage von Konstruktionsdaten und Simulationsmodellen (z. B. für Schweiß- und Schneidaufgaben). Dabei dient zunächst das CAD-Bauteilmodell zur vollautomatischen Generierung der Bahnstützpunktkoordinaten und des Quellcodes des Steuerungsprogrammes. In einem weiteren Schritt können unter Berücksichtigung des Zellenmodells Bewegungsvorgänge für die Steuerung von externen roboter- oder werkstückführenden Zusatzachsen berechnet werden. Das vollständige Steuerungsprogramm für die Bauteilbearbeitung wird ohne Benutzereingriff automatisch simuliert. Beim Auftreten von Erreichbarkeits- oder Kollisionsproblemen werden entsprechende Fehlermeldungen in einer Protokolldatei hinterlegt und die zugehörigen Programme aus dem automatischen Ablauf ausgeschleust. Die weitere Vorgehensweise zur Behebung dieser Probleme entspricht der graphisch-interaktiven Offline-Programmierung, wie im vorherigen Abschnitt beschrieben. Die automatische CAD-basierte Programm-Erzeugung erlaubt eine Reduzierung der Programmierdauer bis zu einem Verhältnis von 1:100 und dadurch die automatisierte Fertigung kleinster Losgrößen mit hoher Wirtschaftlichkeit.

Technologiemodule ermöglichen die vollautomatische Programmgenerierung durch Verknüpfung des Simulationsmodells mit Regeln zur aufgabenspezifischen Bewegungsführung. Diese Regeln erlauben es, vollständige Bewegungsbahnen des Roboters aus Elementen des Bauteil-Geometriemodells abzuleiten. Das Regelwerk ist innerhalb gewisser Grenzen an die Aufgabenstellung anpassbar und lässt sich dadurch jeweils auf eine Gruppe ähnlicher Fertigungsverfahren anwenden (z. B. Plasma-, Laser-, Wasserstrahl-schneiden oder Fräsen einer Kantenkontur). Bei entsprechender Gestaltung des Technologiemoduls und geeigneter Bereitstellung der Geometrie- und Fertigungsdaten des Bauteils lässt sich sogar ein Automatisierungsgrad von 100% bei der Generierung der Roboterprogramme selbst für komplexe Aufgabenstellungen erreichen.

### 3.2 Typischer Arbeitsablauf

Da die Mehrzahl der heute eingesetzten Offline-Programmiersysteme über graphisch-interaktive Programmier- und Simulationsfunktionen verfügen, werden sich die nachfolgenden Abschnitte ausschließlich mit dieser Systemvariante befassen.

Die Erstellung von Roboterprogrammen mit graphisch-interaktiven Offline-Programmiersystemen verlangt eine Reihe von wichtigen Arbeitsschritten. Zu ihnen zählen in der zeitlichen Reihenfolge die:

- **Modellierung** der Fertigungszelle mit Übernahmen der Bauteildaten
- **Graphisch-interaktive Programmierung** von Werkzeugbewegungen und Prozessabläufen am Computerbildschirm
- **Simulation** des erstellten Programms am Computerbildschirm mit Kollisionsuntersuchungen und Taktzeitermittlungen.
- **Optimierung** des erstellten Roboterprogramms im Rahmen von mehreren Simulationsstudien mit Kollisionsüberprüfungen und Taktzeitvergleichen
- **Umsetzung** des erstellten Programms in eine roboterspezifische Programmiersprache. Je nach Systemtyp wird alternativ das Programm zunächst in einer robotertyp-unabhängigen Meta-Sprache erstellt. Mit Hilfe eines entsprechenden Post-Processors erfolgt dann die Überführung der Meta-Daten in den jeweiligen roboterspezifischen Maschinen-Code.
- **Übertragung (Download)** des in der Simulation optimierten und im Maschinencode vorliegenden Programms zur Robotersteuerung in der Fertigung.
- **Nachbearbeitung** des offline erstellten Programms in der Roboterzelle mit Optimierung der Bewegungsabläufe. Eine Kalibrierung ist notwendig, da trotz leistungsfähiger Computertechnik das Modell der Fertigungszelle mit den realen Verhältnissen nicht exakt übereinstimmt. Lässt sich die Bauteilposition z. B. nicht durch exaktes Spannen festlegen, ist einen Ver-