Arbeitshilfe B.14.1.4

Zentrale Aspekte der Robotik
Roboterprogrammierung I

Herstellerspezifische Programmiersprachen
Typischerweise verwendet jeder Roboterhersteller seine eigene Programmiersprache für Befehle an die Robotersteuerung. Diese herstellerspezifischen Sprachen zählen zu den „Domain Specific Language(s)“, ein Robotik Standard wurde in der zurückgezogenen Industrial Robot Language (IRL) [1] festgelegt. Befehle in einer herstellerspezifischen Sprache können die grundlegenden Roboterbewegungen PTP, LIN und CIRC enthalten, die im Verlauf in dieser Arbeitshilfe beschrieben werden, sowie einfache mathematische Operationen oder Programmablaufbefehle wie Schleifen oder Konditionen [2]. Im Folgenden ist eine alphabetische Auflistung der herstellerspezifischen Programmiersprachen einiger größerer Roboterhersteller gegeben:

- ABB - RAPID [3]
- Comau - PDL2 [4]
- Fanuc - KAREL [5]
- Kawasaki - AS [6]
- KUKA - KRL [7]
- Stäubli - VAL3 [8]
- Universal Robots - URScript [9]
- Yaskawa - INFORM [10]

Online und Offline Programmierung
Um ein Roboterprogramm zu schreiben, gibt es verschiedene Möglichkeiten. Zum einen können Programme direkt am Roboter („online“) geschrieben werden. Ein Beispiel dafür ist das „Teaching“, bei dem beispielsweise mithilfe des Handbediengerätes gewünschte Positionen des Roboters manuell angefahren und gespeichert werden, um sie bei der Ausführung in einer bestimmten Reihenfolge automatisch erneut anfahren zu können. Zum anderen können Programme direkt in einer Simulationsumgebung („offline“) geschrieben und dann zur Robotersteuerung transferiert werden (beispielsweise per USB Stick).

Roboterkinematik
In Arbeitshilfe B.11.1.1 „Zentrale Aspekte der Robotik“ wurden die Begriffe „Vorwärtskinematik“ und „Inverse Kinematik“ erwähnt, diese werden hier im Bezug auf die Roboterprogrammierung präzisiert. Der grundlegende Aufbau eines Industrieroboters ist dem einer herkömmlichen Produktionsmaschine sehr ähnlich. In beiden Fällen steuern Achsbefehle einer Steuerungseinheit einen physikalischen Aktor über ein Bus-System. Um eine bestimmte Achse eines Industrieroboters anzusteuern, enthält der Steuerungsbefehl das Verhalten des jeweiligen Motors. Neben der Position müssen dabei auch Geschwindigkeit, Beschleunigung und Abbremsung reguliert werden [2].
Allerdings wird für einen Roboterprozess, wie in Arbeitshilfe B.11.1.2 „Roboter Arbeitsräume“ beschrieben, im Regelfall die Bewegungsbahn vorgegeben. Diese besteht aus einer zeitlich bestimmten, geordneten Reihe von Posen, wobei die Posen beispielsweise durch Position und Orientierung des Endeffektors vorgegeben werden. Die Position, Geschwindigkeit, Beschleunigung und Abbremsung einzelner Achsmotoren muss daraus erst hergeleitet werden. Je mehr Gelenke die kinematische Kette eines Industrieroboters aufweist, desto komplexer sind die mathematischen Zusammenhänge zwischen den Achsbewegungen und dem Endeffektor.

Die **Vorwärtskinematik** wird aus der geometrischen Konfiguration eines Roboters abgeleitet, sie beschreibt die Position des Endeffektors durch eine vorwärtsgerichtete Verkettung der vorhandenen Achsen. Mathematisch bedeutet diese Verkettung eine Multiplikation von achsbezogenen Transformationsmatrizen, resultierend in einer Transformationsmatrix für den Endeffektor. Eine jegliche Roboterkonfiguration kann in Matrizenform modelliert werden, die resultierende Transformationsmatrix wird auch als Denavit-Hartenberg Matrix bezeichnet. Die Ableitung der Vorwärtskinematik nach der Zeit beschreibt die Geschwindigkeiten des Endeffektors, sie wird als die Jacobi-Matrix eines Roboters angegeben [2, 11].

Die **Inverse Kinematik** dreht die im vorigen Absatz erwähnte Verkettung um, aus einer vorgegebenen Position des Endeffektors werden die dafür notwendigen Achspositionen bestimmt. Allerdings ist die Lösung der Inversen Kinematik nicht immer eindeutig. Abbildung 1 zeigt unterschiedliche Gelenkstellungen, mit denen der gleiche Punkt durch den Endeffektor eines Industrieroboters erreicht wird. Ob der Roboterarm den Punkt von oben oder unten erreichen soll, kann beispielsweise durch zusätzliche Variablen vorgegeben werden [2].

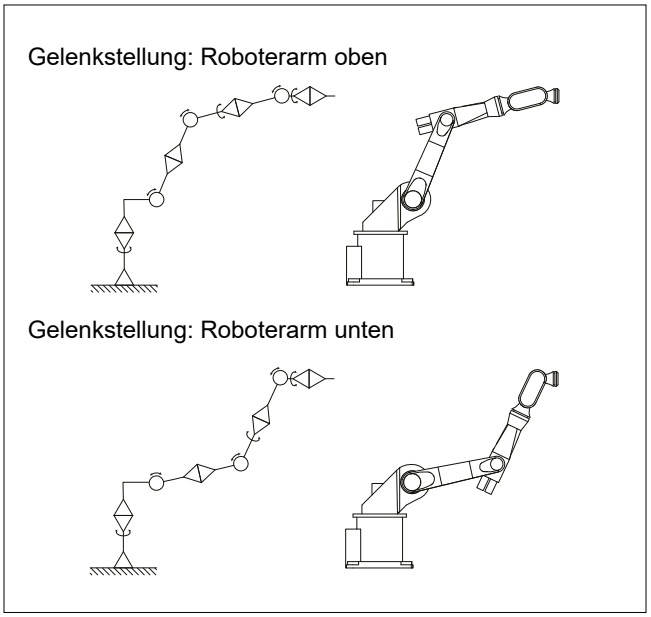

Abbildung 1 - Gelenkstellungen eines Roboters zum Erreichen eines Punktes

Wie in Arbeitshilfe B.11.1.2 „Roboter Arbeitsräume“ beschrieben, können **Singularitäten** für ungünstige Gelenkstellungen auftreten. Das Auftreten einer Singularität kann mathematisch durch die Bedingung ermittelt werden, dass die Determinante der Jacobi-Matrix null ergibt. In diesem Fall kann keine Inverse Kinematik bestimmt werden und es gibt unendlich viele mögliche Gelenkstellungen zum Erreichen eines Punktes. In einer Singularität wird außerdem die Geschwindigkeit des Endeffektors durch eine oder mehrere Achsen nicht beeinflusst. Daher können sehr große Achsgeschwindigkeiten auftreten, die es unbedingt zu vermeiden gilt [2, 11].

Grundlegende Roboterbewegungen

Wurde die Inverse Kinematik eines Industrieroboters erfolgreich bestimmt, kann das Verhalten (Position, Geschwindigkeit, Beschleunigung und Abbremsung) der einzelnen Achsmotoren zur Realisierung einer Bewegungsbahn ermittelt werden. Die Bewegungsbahn wiederum wird beispielsweise durch Punkte im kartesischen Raum festgelegt, in denen Position und Orientierung des Endeffektors sowie die Gelenkstellung (Roboterarm oben oder unten) bestimmt sind. Zwischen zwei Punkten im kartesischen Raum können verschiedene, grundlegende Roboterbewegungen vorgegeben werden, dessen Eigenschaften im Folgenden beschrieben werden.

PTP (“point to point”) Bewegung - Für die vorgegebenen Punkte einer Bewegungsbahn werden entsprechende Achswerte mithilfe der Inversen Kinematik berechnet. Die Bewegung zwischen den vorgegebenen Punkten wird, soweit nicht anders spezifiziert, durch eine Optimierung hin zu möglichst geringen Achsbewegungen bestimmt. Bei einem Industrieroboter, dessen Kinematik aus rein rotatorischen Achsen aufgebaut ist, folgt diese Bewegung daher logischerweise keiner Geraden, wie in Abbildung 2 zu sehen ist [2].

LIN („linear“) und CIRC („circular“) Bewegung - Mit diesen grundlegenden Roboterbewegungen ist es möglich, dass der Endeffektor einem statisch vorgegebenen Pfad zwischen zwei Punkten folgt. Zwischen zwei vorgegebenen Punkten im kartesischen Raum müssen dazu weitere Punkte interpoliert werden, sowie für jeden dieser Punkte Position und Orientierung des Endeffektors. Die Interpolation an sich ist nicht problematisch, allerdings kann eine geringe Bewegung des Endeffektors entlang einer Geraden große Bewegungen einzelner Achsen zur Folge haben, selbiges gilt für Geschwindigkeit und Beschleunigung. Auch Änderungen der Gelenkstellung des Roboters (oben / unten) können während einer solchen Bewegung schwer durchgeführt werden. Abbildung 3 zeigt eine LIN Bewegung und Abbildung 4 eine CIRC Bewegung für einen Industrieroboter [2].

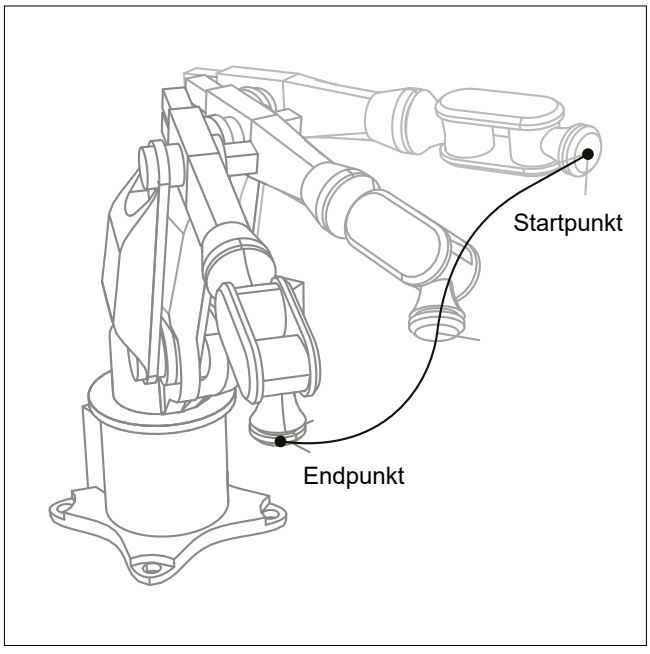

Abbildung 2 - Grundlegende Rotorbewegung PTP

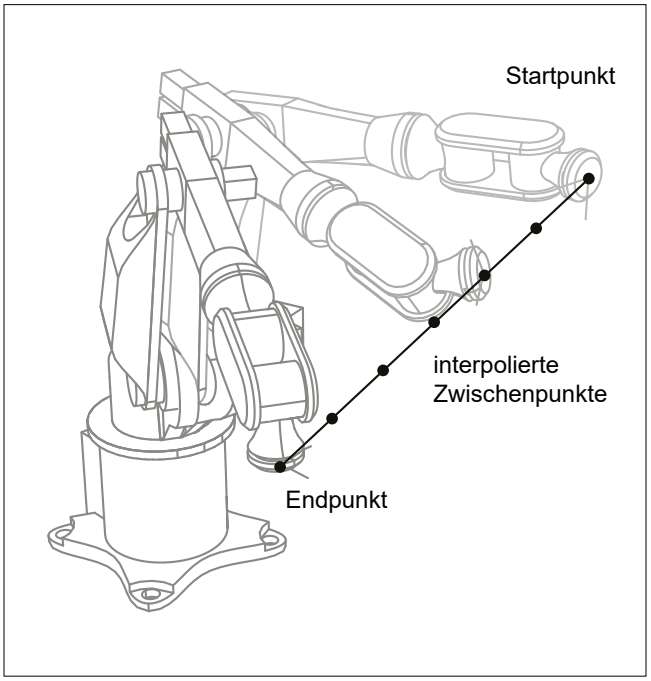

Abbildung 3 - Grundlegende Roboterbewegung LIN

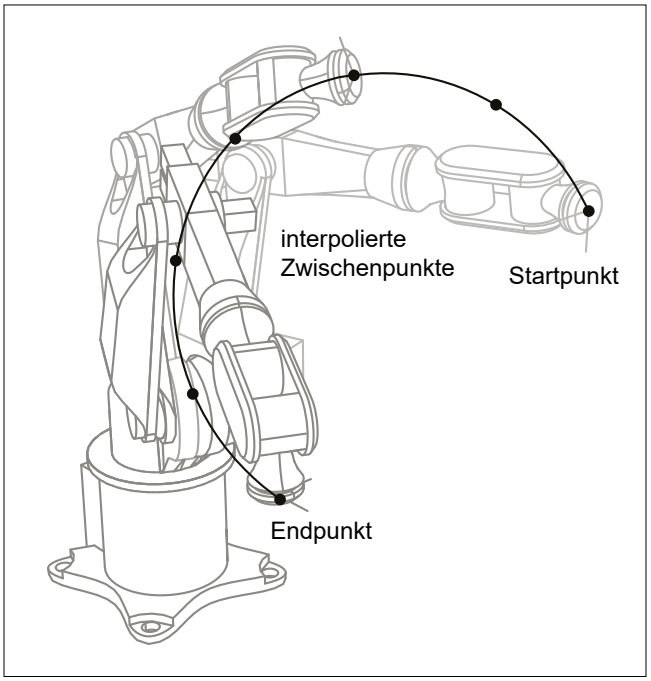

Abbildung 4 - Grundlegende Rotorbewegung CIRC

Aufbau eines Roboterprogramms

Der Aufbau eines Roboterprogramms wird im Folgenden beispielhaft anhand der herstellerspezifischen Programmiersprache Kuka Robot Language (KRL) [7] für einen Industrieroboter mit 6 Achsen beschrieben. KRL baut auf der Programmiersprache Pascal auf. Für Befehle an die Robotersteuerung werden zwei Dateien übertragen, Datenliste (.dat) und Bewegungskommandos (.src). Die grundlegenden, unterstützen Datentypen sind:

- **INT** (Ganze Zahlen -2^{31}-1 bis 2^{31}-1)
- **REAL** (Gleitkommazahlen $\pm 1{,}1e^{-38}$ bis $\pm 3{,}4e^{-38}$)
- **BOOL** (Logische Zustände TRUE und FALSE)
- **CHAR** (Schriftzeichen / ASCII Zeichen)

Für die Bewegungsplanung in KRL sind zudem erweiterte Datentypen in der Programmiersprache vordefiniert [7]:

- **STRUC AXIS REAL A1, A2, A3, A4, A5, A6** enthält Werte für die einzelnen rotatorischen oder translatorischen Achsen.
- **STRUC E6AXIS REAL A1, A2, A3, A4, A5, A6, E1, E2, E3, E4, E5, E6** enthält zusätzlich Werte für etwaige externe Achsen (beispielsweise für die in Arbeitshilfe B.11.1.2 „Roboter Arbeitsräume“ beschriebenen Lineareinheiten).
- **STRUC FRAME REAL X, Y, Z, A, B, C** enthält Position (X, Y, Z) und Orientierung (A, B, C) des Endeffektors (vgl. Arbeitshilfe B.11.1.2 „Roboter Arbeitsräume“).
- **STRUC POS REAL X, Y, Z, A, B, C, INT S, T** enthält zusätzliche Variablen (Status und Turn), die die Gelenkstellung des Roboters angeben.
- **STRUC E6POS REAL X, Y, Z, A, B, C, E1, E2, E3, E4, E5, E6, INT S, T** enthält zusätzlich Werte für etwaige externe Achsen

Die Robotersteuerung kann daraufhin die simultane Bewegung der einzelnen Achsen berechnen. Wie vielleicht aufgefallen, werden die Datentypen „STRUC AXIS REAL“ und „STRUC E6AXIS REAL“ für Bewegungsbefehle der Vorwärtskinematik verwendet, da achsspezifische Werte vorgegeben werden, aus denen sich die Position und Orientierung des Endeffektors ergibt. „STRUC FRAME REAL“, „STRUC POS REAL“ und „STRUC E6POS REAL“ werden hingegen für Bewegungsbefehle der Inversen Kinematik verwendet, in denen die achsspezifischen Werte aus der Position und Orientierung des Endeffektors berechnet werden.

Ausblick

Die Abschnitte dieser Arbeitshilfe geben einen Überblick über grundlegende, relevante Themenfelder für die Roboterprogrammierung. Allerdings wird sich in der Praxis eher der Roboterhersteller als der Roboteranwender beispielsweise mit Problemen der Inversen Kinematik auseinandersetzen. Auch das Erstellen von Roboterprogrammen wird mit der Entwicklung entsprechender Software zunehmend vereinfacht, beispielsweise durch die Einbindung von CAD-Umgebungen. Verschiedene Möglichkeiten zum Erstellen von Roboterprogrammen werden in der folgenden Arbeitshilfe B.11.1.5 „Roboterprogrammierung II“ behandelt.

Literatur

[1] Industrieroboter - Programmiersprache Industrial Robot Language (IRL). Standard, Normenausschuß Maschinenbau (NAM), Berlin, 1996.

[2] Stumm, S. C., Beetz, J., & Brell-Cokcan, S. - Interconnecting design knowledge and construction by utilizing adaptability and configurability in robotics: mediating digital information from architectural design to construction through parametric design intent based robot programming, 2019.

[3] ABB Robotics - Technical reference manual: RAPID Instructions, Functions and Data types, 2004.

[4] Comau Robotics - PDL2: Programming Language Manual System Software Rel. 3.1x, 2005

[5] FANUC America Corporation - System R-30iA and R-30iB Controller KAREL Reference Manual, 2014.

[6] Kawasaki - AS Language Reference Manual: Kawasaki Robot Controller E Series, 2015.

[7] KUKA Roboter GmbH - KR C1/KR C2/KR C3: Reference Guide, Release 4.1.

[8] Stäubli - VAL3 Reference Manual: Version 53, 2006.

[9] Universal Robots - The URScript Programming Language, 2018.

[10] YASKAWA - Instructions for INFORM Language, 2014.

[11] Niku, S. B. - Introduction to robotics: analysis, control, applications. John Wiley & Sons, 2020.

Sohnstraße 65 | 40237 Düsseldorf
T: +49(0)211.54012.080 | zentrale@bauforumstahl.de | www.bauforumstahl.de

Lehrstuhl für Individualisierte Bauproduktion | Prof. Sigrid Brell-Cokcan | M.Sc. Eike Stührenberg
RWTH Aachen | Campus-Boulvard 30 | 52074 Aachen
T: +49 241 80 95005 | office@ip.rwth-aachen.de | www.ip.rwth-aachen.de