

Stefan Anton

Inverse Kinematik am Robotersimulationsprogramm EASY-ROB™

Problemstellung

Bei der Betrachtung von Kinematiken von Industrierobotern (IR) stellen sich grundsätzlich zwei Probleme. Das Direkte kinematische Problem (KP), auch als Vorwärtstransformation bezeichnet, behandelt die Berechnung der Lage des Tool-Center-Points (TCP) in raumfesten Koordinaten aus den achsspezifischen Gelenkkoordinaten des Roboters. Bei dem Inversen kinematischen Problem (IKP), auch als inverse Transformation bezeichnet, handelt es sich um die Umkehrrelation, bei dem aus der Lage des TCP's die achsspezifischen Gelenkkoordinaten zu bestimmen sind. Im Gegensatz zur Vorwärtstransformation existieren für die inverse Transformationsproblematik keine allgemeingültigen Lösungskonzepte, mit denen explizite Bestimmungsgleichungen aufgestellt werden können [5]. Weiterhin ist die Lösung im Allgemeinen nicht eindeutig, so lassen sich beispielsweise für einen Knickarmroboter mit Zentralhand acht Gelenkwinkelstellungen berechnen, die zur gleichen Lage des TCP's führen.

Kinematiken in EASY-ROB™

Das Roboter-Simulationsprogramm EASY-ROB™ erlaubt es kinematische Ketten nach der bekannten Denavit-Hartenberg-Konvention (DH) [2] oder nach Universal-Koordinaten [6] zu beschreiben. Zwischen den Achsgelenken werden Achsverkopplungen bzw. beliebige mathematische Abhängigkeiten durch so genannte passive Achsgelenke beschrieben. Zur Lösung des Inversen kinematischen Problems sind für Standard-Kinematiken wie Knickarmroboter, SCARA- und Portal-Roboter, 5 achsige NC Maschinen mit C- und A-Achse, explizite Lösungen in geschlossener Form vorhanden. Für Sonder-Kinematiken sowie für neue Maschinenkonzepte steht ein Application Program Interface (API) zur Verfügung, das es ermöglicht mathematische Berechnungen in der Programmiersprache C bzw. C++ einzubinden. Diese C-Schnittstelle dient einerseits zur Lösung des IKP's als auch zur Lösung des direkten kinematischen Problems (KP), wie es

beispielsweise bei Parallelkinematiken (Tripoden, Hexapoden, Flexpicker) erforderlich ist. Berechnete Lösungen lassen sich unmittelbar in einer dreidimensionalen OpenGL™ Szene visualisieren und sind so leicht zu überprüfen und zu verifizieren.

Vorwärtstransformation

Kinematische Ketten können nach der Denavit-Hartenberg-Konvention [2,3] oder nach Universal-Koordinaten [6] beschrieben werden. Beide Konventionen haben gemeinsam, dass jedes Glied der kinematischen Kette mit einem körperfesten Koordinatensystem verbunden wird. Die Koordinatentransformation zwischen zwei benachbarten Gliedern hängt nur von der Koordinate des verbindenden Gelenks ab und wird mit homogenen Koordinaten \underline{T} bzw. \underline{D} beschrieben.

$$\underline{T}(\underline{q}) = \begin{pmatrix} \underline{n}(\underline{q}) & \underline{o}(\underline{q}) & \underline{a}(\underline{q}) & \underline{p}(\underline{q}) \\ 0 & 0 & 0 & 1 \end{pmatrix} = {}^0\underline{D} \cdot {}^1\underline{D} \cdot \dots \cdot {}^{N-2}\underline{D} \cdot {}^{N-1}\underline{D} \cdot {}^N\underline{D} \quad (1)$$

Vektor \underline{q} definiert die N achsspezifischen Gelenkkordinaten und $\underline{T}(\underline{q})$ die Lage des TCP's bezogen auf das raumfeste Koordinatensystem mit dem Index 0.

Die Denavit-Hartenberg-Konvention zeichnet sich dadurch aus, dass die Transformation vom Gelenk 'i-1' zum Gelenk 'i' mit den vier Parametern θ , d , a und α beschrieben werden kann.

$${}^{i-1}\underline{D} = Rot_{z_{i-1}}(\theta_i) Trans_{z_{i-1}}(d_i) Trans_{x_i}(a_i) Rot_{x_i}(\alpha_i) = \begin{pmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

Die homogene Matrix ${}^{i-1}\underline{D}$ wird als Denavit-Hartenberg Matrix bezeichnet. Wie die Gleichung (2) erkennen lässt sind nur Transformation und Rotation und Z- und X-Richtung erlaubt. Bei der Modellierung typischer kinematischer Ketten ist deshalb eine gewisse Übung erforderlich. Teilweise sind sogar Umrechnungen unumgänglich.

Bei der Modellierung in Universal-Koordinaten hingegen, stehen sechs Parameter (Translation in X, Y und Z und Rotation um X, Y und um Z) zur Verfügung um die feste Transformation von Gelenk 'i-1' zum Gelenk 'i' zu bestimmen. Somit können beispielsweise Knickarmroboter-Kinematiken 1:1 vom Datenblatt modelliert werden.

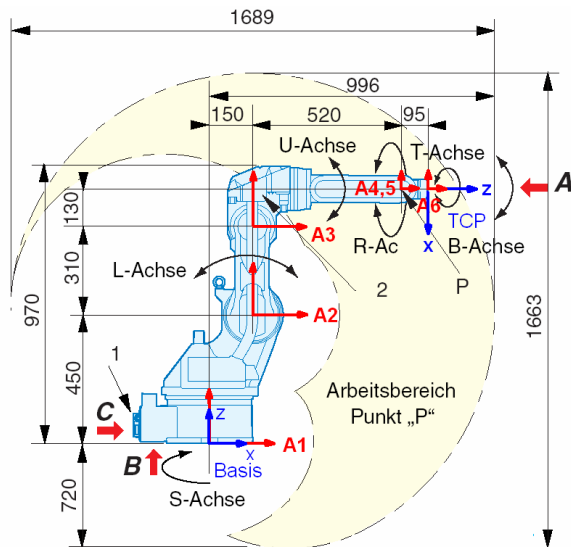


Bild 1: Datenblatt: MOTOMAN UP 6 Compact

Achse	T_x	T_y	T_z	R_x	R_y	R_z
A ₁	150	0	450	0	0	0
A ₂	0	0	310	0	0	0
A ₃	520	0	130	0	0	0
A ₄	0	0	0	0	0	0
A ₅	95	0	0	0	0	0
A ₆	0	0	0	0	90°	0

Tabelle 1: Transformationsmatrizen nach Universal- Koordinaten in Nullstellung

$$\text{Rot}_i(q_i) = R_{z1}; R_{y2}; R_{y3}; R_{x4}; R_{y5}; R_{x6}$$

$$D_i(A_i) = \text{Rot}_i(q_i) * T_{x_i} T_{y_i} T_{z_i} R_{x_i} R_{y_i} R_{z_i}$$

$${}^0T_6(\underline{q}) = {}^0D_1 D_2 D_3 D_4 D_5 D_6 \quad (3)$$

An dieser Stelle sei anzumerken, dass aus mathematischer Sicht beide Konventionen gleichwertig sind und für das direkte und inverse kinematische Problem keinerlei Bedeutung haben.

Inverse Transformation

Bei der Inversen Transformation werden aus einer gegebenen Lage des TCP's in einem raumfesten Koordinatensystem die Gelenkkoordinaten berechnet werden. Auf Grund der Mehrdeutigkeit des Problems ist zusätzlich die gewünschte Konfiguration vorzugeben. Bild 2) und Bild 3) zeigen einen sechs achsigen Knickarmroboter in drei von acht möglichen Konfigurationen.

Zur Lösung des Inversen kinematische Problems bieten sich drei Lösungsvarianten an.

1. Explizites und analytisches Lösungsverfahren
2. Numerisches Lösungsverfahren
3. Kombination aus explizitem und numerischem Lösungsverfahren

Welches Lösungsverfahren Anwendung findet hängt von der kinematischen Struktur des Roboters ab. Ein Knickarmroboter zeichnet sich beispielsweise dadurch aus, dass sich die

Handachsen in einem Punkt schneiden. Somit wird die TCP-Position nur durch die Grundachsen festgelegt und die TCP-Orientierung ausschließlich durch die Handachsen.

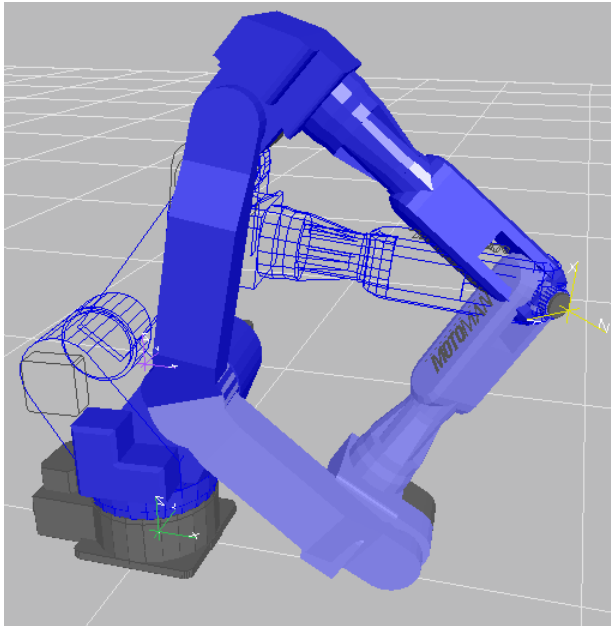


Bild 2: Knickarmroboter in drei von acht möglichen Konfigurationen

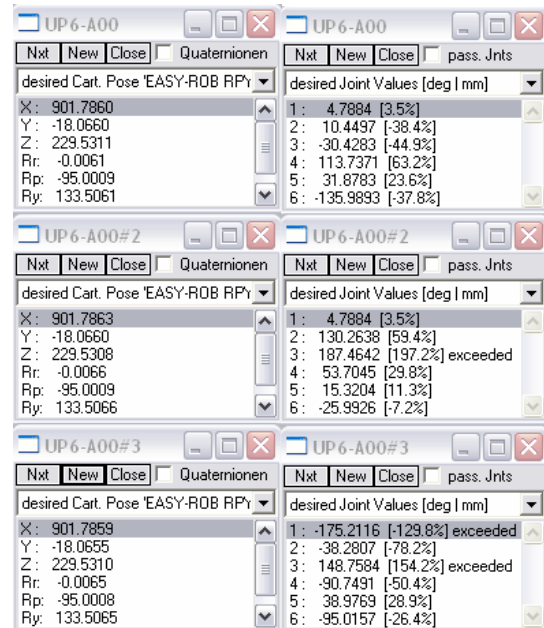


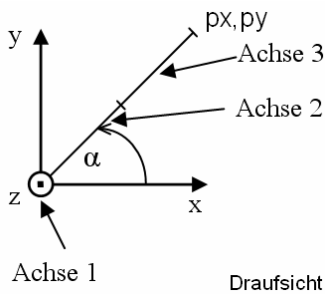
Bild 3: Identische TCP Lage und zugehörige Gelenkkordinaten je nach Konfiguration

Zur Berechnung der Grundachsen A_1 , A_2 und A_3 bietet sich die geometrische Lösung an. Die Lage des TCP's bezogen auf die Roboterbasis ergibt sich aus [6].

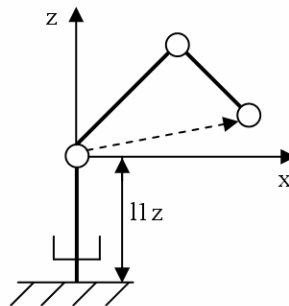
$${}^0T_6(\underline{a}) = {}^0T_3(q_1q_2q_3) * {}_3T_6(q_4q_5q_6) = [{}^0R_6(q_1q_2q_3q_4q_5q_6) \mid {}^0\underline{p}_6(q_1q_2q_3)] \quad (4)$$

Orientierung: ${}^0R_6 = [\underline{n} \ \underline{o} \ \underline{a}]$

Position: ${}^0\underline{p}_6 = [x \ y \ z]^T \quad (5,6)$

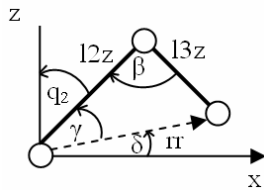


$$q_1 = \text{atan2}(p_y, p_x) \quad (7)$$



$$r = \sqrt{p_x^2 + p_y^2 + (p_z - l_{1z})^2} \quad (8)$$

$$q_3 = \pi - \beta ; \beta = \text{arccos} \left(\frac{r}{l_{2z} + l_{3z}} \right) \quad (9)$$



$$\gamma = \text{sssa} (l_{3z}, rr, l_{2z}) \quad (10)$$

$$\delta = \text{asin} ((p_z - l_{1z})/rr) \quad (11)$$

$$q_2 = \pi/2 - \gamma - \delta \quad (12)$$

Obige Berechnungen sollen nur das Prinzip der geometrischen Lösung zeigen. Erreichbarkeitsabfragen, der Einfluss von Gelenkoffsets, Verfahrrichtungen und verschiedene Konfigurationen sind zur besseren Übersicht nicht enthalten (siehe [6]). Die Kosinussatz API-Funktion $\alpha = \text{sssa} (l_1, l_2, l_3)$ berechnet den Winkel α gegenüber der Seite l_1 .

Berechnung der Handachsen A_4, A_5 und A_6 mit (4, 7, 9 und 12)

$${}_3R_6 (q_4 q_5 q_6) = {}_0R_3^T (q_1 q_2 q_3) * {}_0R_6 (q) \quad (13)$$

$$[q_4 \ q_5 \ q_6]^T = \text{frame_to_vector_idx} ({}_3R_6, \text{Rot_xyx}) \quad (14)$$

Die API-Funktion `frame_to_vector(T, Rot_idx)` berechnet in Abhängigkeit der Orientierungsvorschrift oder des Handgelenksmodus, in unserem Beispiel $Rx_4Ry_5Rx_6$ (3), die Handgelenkwinkel aus.

Einen Sonderfall bilden Kinematiken mit weniger als sechs Freiheitsgrade, z.B. SCARA-Roboter die nur vier Gelenke haben. Sind die Zielpositionen im Tech-In Verfahren erzeugt worden, ist gewährleistet, dass die Positionen stets erreichbar sind. Im Fall, dass die Zielpositionen Offline erzeugt wurden oder aus einem anderen Programm kommen oder das Werkzeug gewechselt wurde, ist bei der inverseren Transformation ein sogenanntes Tool fitting durchzuführen. Das Tool fitting hängt unmittelbar von der kinematischen Struktur ab.

Numerisches Lösungsverfahren zur inversen Transformation

Explizite und geschlossene Lösung sind bei komplexen kinematischen Strukturen schwer oder zum Teil gar nicht zu finden. In solchen Fällen bieten sich numerische Lösungsverfahren an. Das Prinzip besteht darin die kinematische Struktur in einem Arbeitspunkt, in der Regel ist das die aktuelle Robotergelenkstellung, zu linearisieren. Die Lösung des Kinematischen Problems lässt sich aus dem folgenden m-dimensionalen

Gleichungssystem eindeutig bestimmen.

$$\underline{x} = \underline{f}(\underline{q}) \quad (15)$$

\underline{x} beschreibt die Position und Orientierung im raumfesten Koordinatensystem und \underline{q} die m achsspezifischen Gelenkkordinaten. Das Inverse Kinematische Problem

$$\underline{q} = \underline{g}(\underline{x}) \quad (16)$$

führt im allgemeinen Fall auf ein System nichtlinearer transzendenter Gleichungen für die keine allgemeingültigen Lösungsverfahren existieren. Allgemeingültige Lösungskonzepte lassen sich nur durch iterative numerische Verfahren realisieren [5].

Differentiation von (15) nach der Zeit ergibt

$$\dot{\underline{x}} = \frac{\partial f(q)}{\partial q} * \frac{\partial q}{\partial t} = J(q) * \dot{\underline{q}} \quad (17)$$

mit $\underline{J}(\underline{q})$ als Jacobi-Matrix im Endeffektor bezogen auf das raumfeste Koordinatensystem. Die Jacobi-Matrix eines Roboterarms beschreibt die Abbildung von Gelenkgeschwindigkeiten auf die Geschwindigkeiten des TCP im kartesischen Raum und den zeitlichen Änderungen der Orientierung des Endeffektors bezogen auf ein raumfestes Koordinatensystem, z.B. die Roboterbasis.

Zur Bestimmung der Jacobi-Matrix sind drei Verfahren erwähnt.

1. Bestimmung durch analytische Ableitung:

Änderungen der kartesischen Lage x als Funktion der Änderungen der

Gelenkwinkel q $\frac{\partial x_i}{\partial q_j} = \sum_{j=1}^n \frac{\partial f_i(q)}{\partial q_j} \partial q_j$. Je nach Komplexität der nichtlinearen

Funktionen, bzw. der kinematischen Kette des Roboterarms, ist es möglich die Jacobi-Matrix mit mehr oder weniger Aufwand durch analytische Ableitung zu berechnen.

2. Geometrische Bestimmung mittels Vektorprodukt:

Dieser gängige Ansatz geht von linearen und rotatorischen Geschwindigkeiten von starren Körpern aus. Bei gängigen Industrieroboter-Kinematiken sind die einzelnen Achsen entkoppelt, so dass jede 4x4 Matrix der 6 Gelenke eine Rotation um die x,y oder z-Achse bzw. eine Translation in x,y oder z beschreiben. In diesem Fall bietet sich die Berechnung der Jacobi-Matrix durch geometrische Berechnung mittels Vektorprodukt an und kann in stark optimierter Form als Algorithmus abgelegt werden.

3. Allgemeines Verfahren zur Bestimmung der Jacobi-Matrix

Komplexere Kinematiken können innerhalb der kinematischen Kette beliebige Kopplungen und Unstetigkeiten enthalten. In EASY-ROB™ ist ein allgemeines Verfahren realisiert, dass zur Berechnung der Jacobi-Matrix nur die Vorwärtstransformation $\underline{x} = \underline{f}(\underline{q})$ gemäß (15) verwendet.

Wurde die Jacobi-Matrix bestimmt gibt es die verschiedensten Lösungsstrategien. Eine wichtige Rolle spielen hierbei die Freiheitsgrade des Roboters. Für den Fall $m = n = 6$ ergibt sich immer eine eindeutige Lösung, sofern diese existiert. Als Lösungsstrategie bietet sich das Gauß'sche Eliminationsverfahren an. Im Fall $m > 6$ ist das Gleichungssystem unterbestimmt und die Lösung ist mehrdeutig. Als Lösungsstrategien bietet sich die Singulärwert-Zerlegung an, welche in EASY-ROB™ implementiert ist, oder die Verwendung der Pseudoinversen oder die verallgemeinerte Lösung.

Unabhängig vom Verfahren zur Bestimmung der Jacobi-Matrix und von der gewählten Lösungsstrategie läuft der iterative Algorithmus prinzipiell in den folgenden Schritten ab.

1. Gegeben $\underline{x}_{\text{soll}}$, gesucht ist $\underline{q}_{\text{soll}}$, so dass gilt $\underline{x}_{\text{soll}} = \underline{f}(\underline{q}_{\text{soll}})$
2. Bestimmung $\underline{x}_{\text{ist}} = \underline{f}(\underline{q}_{\text{ist}})$, und der Jacobi-Matrix $\underline{J}(\underline{q}_{\text{ist}})$
3. Fehlerberechnung $\Delta \underline{x} = \underline{x}_{\text{soll}} - \underline{x}_{\text{ist}}$
4. Lösungsstrategie für $\underline{J}(\underline{q}_{\text{ist}}) \Delta \underline{q} = \Delta \underline{x}$, Begrenzung von $\Delta \underline{q}$
5. Neue Gelenkkoordinaten bestimmen $\underline{q}_{\text{ist}} = \underline{q}_{\text{ist}} + \Delta \underline{q}$, $\underline{x}_{\text{ist}} = \underline{f}(\underline{q}_{\text{ist}})$
6. Fehler prüfen. Ist $|\underline{x}_{\text{soll}} - \underline{x}_{\text{ist}}| < \varepsilon$ Ziel erreicht, sonst Schritt 2).

Kombination aus explizitem und numerischem Lösungsverfahren

Ein numerisches Lösungsverfahren ermittelt zwar die Lösung zu einer gegebenen TCP-Lage, jedoch kann immer nur eine Konfiguration berechnet werden. Stellen wir uns einen Knickarmroboter vor, jedoch ohne Zentralhand so dass sich die Achsen 4 und 5 nicht in einem Punkt schneiden. Ein Grund für diese Kinematik wäre die Idee, dass die Achse 4 eine Kamera trägt die ein permanentes Bild des Greifers liefert. Die Abweichung aus der Zentralhand sind nur wenige Zentimeter, so dass auch diese Kinematik in mehreren Konfigurationen verfahren werden soll. Bei der Kombination aus explizitem und numerischem Lösungsverfahren wird folgendermaßen vorgegangen.

1. Berechnung der expliziten Knickarmroboter-Lösung für eine gewünschte Konfiguration, in dem der Versatz in der Zentralhand zurückgesetzt wird. Die explizite Lösung berechnet eine Startwertvorgabe q_{ist} .
2. Bestimmung der exakten Lösung q_{soll} durch Verwendung des Nachgeschalteten numerischen Lösungsverfahrens.
3. Auswahl einer weiteren Konfiguration und weiter mit Schritt 1)

Bei der Kombinationslösung gilt es im ersten Schritt einen kinematischen Näherungstyp zu finden, für den eine explizite Lösung bereits existiert. In der Regel ist die Startwertvorgabe durch die Näherungslösung gut genug, so dass die Anzahl der Iterationen im Nachgeschalteten numerischen Lösungsverfahren recht klein ist.

Fazit

Glücklicherweise und Dank des Fortschritts in der Mikroelektronik ist die verfügbare Rechenleistung so hoch, dass insbesondere numerische Verfahren die Interaktion zwischen dem Bediener und dem Simulationsprogramm kaum beeinträchtigen. Gerade bei der Entwicklung neuer Roboterkinematiken und Maschinenkonzepte ist es eine wesentliche Zeitersparnis, sich nicht um die Inverse Transformation kümmern zu müssen. Erst wenn das Maschinenkonzept vollständig verifiziert ist, ist es sinnvoll nach einer eventuell vorhandenen expliziten- oder kombinierten Lösung zu suchen.

Literatur

- [1] Craig, J.J.: Introduction to Robotics. Addison-Wesley, 2nd. Ed, 1989
- [2] Paul, R.P.: Robot Manipulators, Mathematics, Programming and Control MIT Press 1981
- [3] Prof. Wolfgang Weber: Industrieroboter: „Methoden der Steuerung und Regelung“ Mai 2002, ISBN 3-446-21604-9

- [4] Prof. Thomas Horsch: Introduction to Robotics: „Module Trajectory generation and robot programming“
FH Darmstadt, Summer 2000, <http://www.easy-rob.com/data/LectureRobotics.pdf>
- [5] Prof. Heribert Münch, Dipl.-Ing. J. Bargfrede: „Universelle Koordinatentransformation für Industrieroboter“
Hochschule Magdeburg-Stendal, August 2001, http://www.easy-rob.com/data/paper_unitrans.pdf
- [6] Janko Härtig: Praktikumsarbeit, Roboterkinematiken „Direkte- und inverse Koordinatentransformation“
Hochschule Mittweida (FH), Juli 2004, <http://www.easy-rob.com/data/BPS-Inverse-Kinematik-JHaertig.pdf>

Autorenangabe:

Dipl. Ing. Stefan Anton
EASY-ROB, Kennedyallee 109
60596 Frankfurt/Main
Tel.: 069 677 24 287
Fax: 069 677 24 320
E-mail: stefan.anton@easy-rob.com