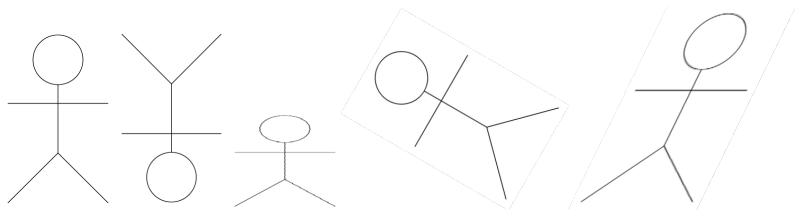


Koordinaten, Transformationen und Roboter

Dipl.-Inform. Wolfgang Globke

Institut für Algebra und Geometrie
Arbeitsgruppe Differentialgeometrie
Universität Karlsruhe

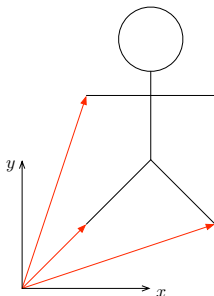
Seit Anbeginn der Zeit strebten die Menschen danach, Dinge zu transformieren...



... aber wie beschreibt man diese Transformationen?

Koordinaten in der Ebene und um Raum

Geometrische Objekte werden durch Punkte \mathbf{x} in der Ebene \mathbb{R}^2 bzw. im Raum \mathbb{R}^3 beschrieben.



Geometrische Transformationen sind Abbildungen

$$\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^n, \quad \mathbf{x} \mapsto A\mathbf{x},$$

die durch Multiplikation eines Vektors $\mathbf{x} \in \mathbb{R}^n$ mit einer Matrix $A \in \mathbb{R}^{n \times n}$ gegeben sind.

Die wesentlichen Transformationen sind

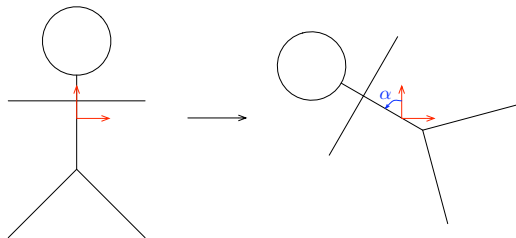
- Drehungen,
- Streckungen,
- Spiegelungen,
- Scherungen,
- Projektionen,
- Verschiebungen (nicht linear).

Drehungen im \mathbb{R}^2

Eine **Drehung** um den Nullpunkt der Ebene \mathbb{R}^2 um den Winkel α gegen den Uhrzeigersinn wird durch die Matrix

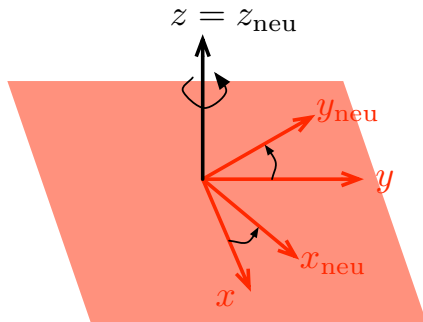
$$\begin{pmatrix} x_{\text{neu}} \\ y_{\text{neu}} \end{pmatrix} = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

beschrieben.



Drehungen im \mathbb{R}^3

Drehungen im Raum \mathbb{R}^3 können in einer beliebigen **Drehebene** stattfinden, auf der die **Drehachse** senkrecht steht.



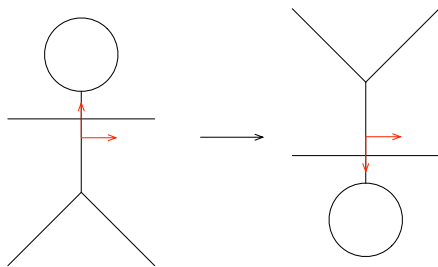
Die Drehachse bleibt bei der Drehung unverändert, die Drehung in der Drehebene funktioniert wie im \mathbb{R}^2 .

Die Drehachse bestimmt eindeutig die Drehebene, und umgekehrt. Die Drehungen um die x -, y - bzw. z -Achse etwa sind durch folgende Matrizen gegeben:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix}, \begin{pmatrix} \cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 1 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) \end{pmatrix}, \\ \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Drehungen um beliebige Ebenen sind komplizierter darzustellen, lassen sich aber auf diese drei zurückführen.

Bei einer **Spiegelung** werden eine oder mehrere Koordinatenrichtungen umgekehrt.



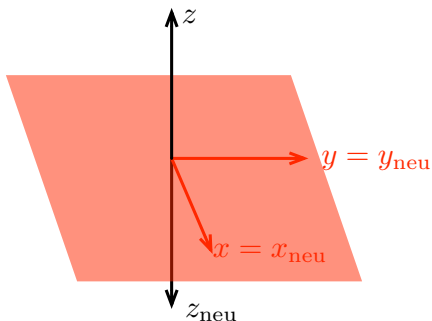
Im \mathbb{R}^2 haben Spiegelungen entlang der x - bzw. y -Achse die Matrixdarstellung

$$\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

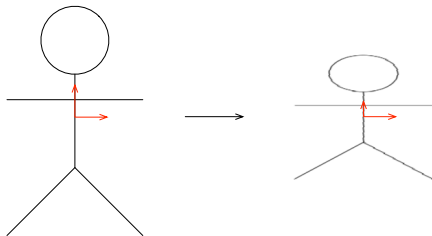
Im \mathbb{R}^3 haben die Spiegelungen entlang x -, y - bzw. z -Achse die Darstellungen

$$\begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}.$$

Die Spiegelungen entlang einer Achse entspricht der Spiegelung an der Ebene senkrecht zu dieser Achse.



Bei einer **Streckung** werden ein oder mehrere Koordinatenachsen mit einem Streckungsfaktor multipliziert.



Die Matrix

$$\begin{pmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{pmatrix}$$

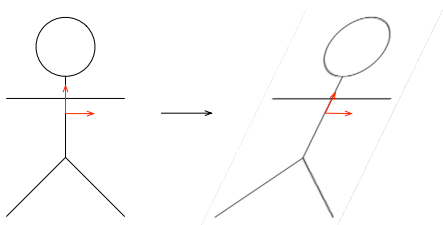
etwa staucht die Figur um den Faktor $\frac{1}{2}$ in der x -Richtung zusammen, aber erhält die Breite in y -Richtung.

Die Matrix

$$\begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{pmatrix}$$

mit $s_x, s_y, s_z > 0$ beschreibt eine Abbildung, die die drei Achsen jeweils um den Faktor s_x, s_y bzw. s_z streckt.

Eine **Scherung** ändert die Winkel zwischen den Koordinatenachsen.



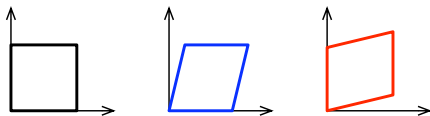
Eine Scherung entlang der x -Achse wird durch

$$\begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix},$$

wobei der Parameter t beschreibt, wie stark „geschert“ wird.
Entsprechend stellt

$$\begin{pmatrix} 1 & 0 \\ t & 1 \end{pmatrix}$$

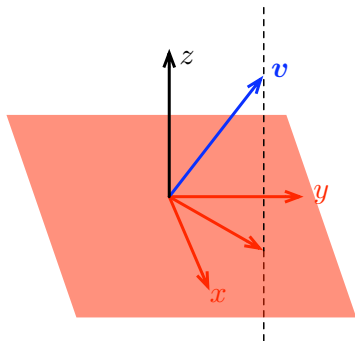
eine Scherung entlang der y -Achse dar.



Eine Scherung im \mathbb{R}^3 parallel zur xy -Ebene wird dargestellt durch

$$\begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix}.$$

Im Gegensatz zu den bisher betrachteten Transformationen sind **Projektionen** *nicht invertierbar*.



Hier wird \mathbf{v} in die **xy-Ebene** projiziert. Alle Punkte auf der gestrichelten Linie werden auf den gleichen Punkt projiziert.

Eine Projektion in die xy -Ebene lässt die x - und y -Koordinaten fest und setzt die z -Koordinate = 0:

$$\begin{pmatrix} x \\ y \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Entsprechend muss eine Projektion auf eine Achse zwei Koordinaten = 0 setzen, z.B. stellt

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

die Projektion auf die x -Achse dar.

Iwasawa-Zerlegung

Jede invertierbare Matrix $A \in GL_2(\mathbb{R})$ lässt sich darstellen als Verknüpfung einer **Rotation**, einer **Spiegelung**, einer **Streckung** und einer **Scherung**:

$$A = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} \pm 1 & 0 \\ 0 & \pm 1 \end{pmatrix} \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix}.$$

Ein vergleichbares Resultat gilt für $A \in GL_n(\mathbb{K})$ für beliebiges n und $\mathbb{K} = \mathbb{R}$ oder $\mathbb{K} = \mathbb{C}$.

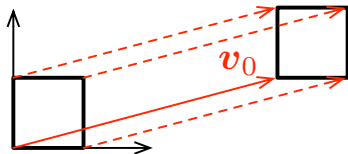
Verschiebungen

Im Gegensatz zu den bisher betrachteten Transformationen sind **Verschiebungen** *nicht linear*.

Eine Verschiebung um einen festen Vektor \mathbf{v}_0 ist durch die Vorschrift

$$\mathbf{x} \mapsto \mathbf{x} + \mathbf{v}_0$$

gegeben.



Trick: Homogene Koordinaten

Es ist dennoch möglich, Verschiebungen durch Matrizen darzustellen.

Dazu bettet man den \mathbb{R}^3 in den \mathbb{R}^4 ein:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \mapsto \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}.$$

Diese Darstellung bezeichnet man als **homogene Koordinaten**.

Trick: Homogene Koordinaten

Jetzt wird die Verschiebung um den konstanten Vektor

$$\mathbf{v} = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}$$

dargestellt durch die Matrixmultiplikation

$$\begin{pmatrix} x + v_x \\ y + v_y \\ z + v_z \\ \mathbf{1} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & v_x \\ 0 & 1 & 0 & v_y \\ 0 & 0 & 1 & v_z \\ 0 & 0 & 0 & \mathbf{1} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ \mathbf{1} \end{pmatrix}.$$

Trick: Homogene Koordinaten

Ein Element $\mathbf{v} \in \mathbb{R}^3$ kann sowohl als **Punkt im Raum** als auch als **Richtungsvektor** („Pfeil“) interpretiert werden.

In homogenen Koordinaten lassen sich diese beiden Konzepte unterscheiden:

$$\begin{pmatrix} v_x \\ v_y \\ v_z \\ 1 \end{pmatrix}$$

bezeichnet einen Punkt im Raum, aber

$$\begin{pmatrix} v_x \\ v_y \\ v_z \\ 0 \end{pmatrix}$$

bezeichnet einen Richtungsvektor.

Affine Transformationen

Sei $A \in \mathbb{R}^{3 \times 3}$ eine Matrix, die eine lineare geometrische Transformation beschreibt und sei $\mathbf{v} \in \mathbb{R}^3$.

Eine **affine Transformation**

$$\mathbf{x} \mapsto A\mathbf{x} + \mathbf{v}$$

ist die Verknüpfung einer linearen Transformation mit einer Verschiebung.

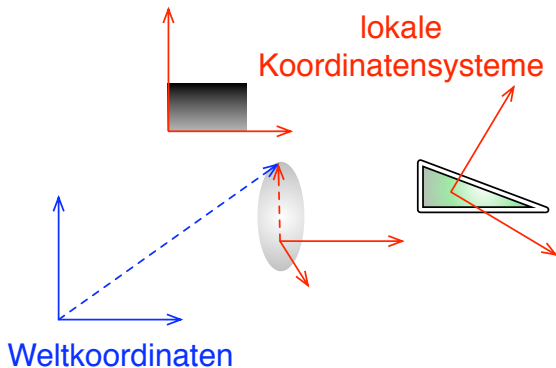
Jede affine Transformation lässt sich in homogenen Koordinaten durch eine Matrix der Form

$$\begin{pmatrix} & & & v_x \\ & A & & v_y \\ & & & v_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

darstellen.

- **In der Theorie:**
Ein Vektorraum hat den 0-Vektor als Koordinatenursprung.
- **In der Realität:**
Im Anschauungsraum kann jeder Punkt als Koordinatenursprung gewählt werden.
- Im letzteren Fall spricht man von **affinen Räumen**.
- **In der Anwendung:**
Man legt einen festen Ursprung und drei feste Achsen als **Weltkoordinatensystem** fest und ordnet zusätzlich jedem geometrischen Objekt ein **lokales Koordinatensystem** zu.

Weltkoordinaten und lokale Koordinaten

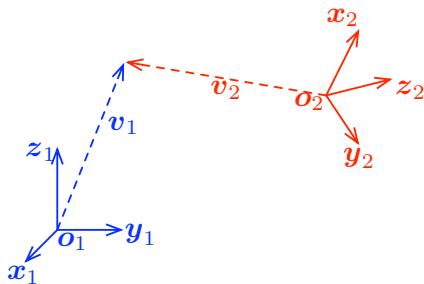


Ein Koordinatensystem $KS(\mathbf{x}, \mathbf{y}, \mathbf{z}; \mathbf{o})$ im \mathbb{R}^3 ist gegeben durch

- den Ursprung \mathbf{o}
- und drei Basisvektoren \mathbf{x} , \mathbf{y} , \mathbf{z} , die die Koordinatenachsen aufspannen.

Koordinatenwechsel

Sind zwei Koordinatensysteme $KS_1(x_1, y_1, z_1; o_1)$ und $KS_2(x_2, y_2, z_2; o_2)$ gegeben, so hat jeder Punkt im Raum eine Koordinatendarstellung v_1 bzgl. KS_1 und eine Darstellung v_2 bzgl. KS_2 .



Wie kann man die Darstellung bzgl. KS_1 umrechnen in die Darstellung bzgl. KS_2 (und umgekehrt)?

- Die Darstellung bzgl. KS_1 erhält man, indem man so tut, als sei \mathbf{o}_1 der 0-Vektor und dann \mathbf{v}_1 als Linearkombination der Basis $\mathbf{x}_1, \mathbf{y}_1, \mathbf{z}_1$ bestimmt.
- Wenn KS_1 und KS_2 im gleichen Ursprungspunkt liegen (also $\mathbf{o}_1 = \mathbf{o}_2$), so erhält man den Wechsel vom Koordinatensystem KS_1 zu KS_2 durch einfachen Basiswechsel von der Basis $\mathbf{x}_1, \mathbf{y}_1, \mathbf{z}_1$ zur Basis $\mathbf{x}_2, \mathbf{y}_2, \mathbf{z}_2$.

Genauer:

$$\mathbf{v}_2 = A_{21}\mathbf{v}_1,$$

wobei A_{21} die Übergangsmatrix dieses Basiswechsels ist.

- Ist aber $\mathbf{o}_1 \neq \mathbf{o}_2$, so muss noch die Verschiebung der beiden Ursprungspunkte berücksichtigt werden.

Für $\mathbf{o}_1 \neq \mathbf{o}_2$ kann man sich das Vorgehen wie folgt überlegen:

- 1 Bestimme durch Basiswechsel A_{21} den Koordinatenwechsel zum Koordinatensystem $KS'_2(\mathbf{x}_2, \mathbf{y}_2, \mathbf{z}_2; \mathbf{o}_1)$ mit den Achsen von KS_2 , aber dem Ursprung von KS_1 :

$$\mathbf{v}'_2 = A_{21}\mathbf{v}_1.$$

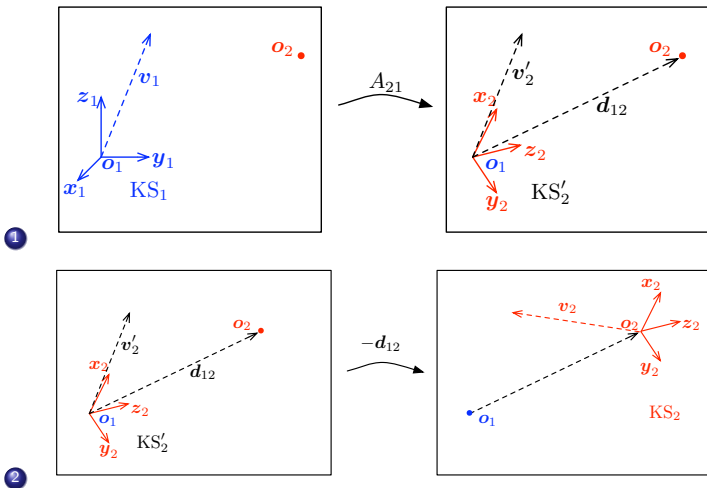
- 2 KS_2 und KS'_2 unterscheiden sich durch eine Parallelverschiebung um den Vektor \mathbf{d}_{12} mit $\mathbf{o}_2 = \mathbf{o}_1 + \mathbf{d}_{12}$.
- 3 Also erhält man den Koordinatenwechsel von KS'_2 zu KS_2 , indem man den Vektor \mathbf{d}_{12} in das Koordinatensystem KS'_2 umrechnet, $\mathbf{d}'_{12} = A_{21}\mathbf{d}_{12}$, und von \mathbf{v}'_2 abzieht:

$$\mathbf{v}_2 = \mathbf{v}'_2 - \mathbf{d}'_{12}.$$

- 4 Den Koordinatenwechsel von KS_1 zu KS_2 erhält man durch Zusammensetzen der Koordinatenwechsel von KS_1 zu KS'_2 und dem Wechsel von KS'_2 zu KS_2 :

$$\mathbf{v}_2 = A_{21}(\mathbf{v}_1 - \mathbf{d}_{12}).$$

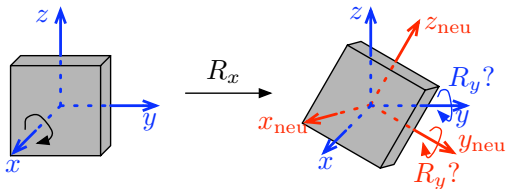
Koordinatenwechsel



Verkettete Transformationen: Beispiel

Ein Würfel, dessen Mittelpunkt im Ursprung liegt, soll zuerst um die x -Achse gedreht werden, dann um die y -Achse.

- Die jeweiligen Rotationsmatrizen seien R_x und R_y .
- Dies lässt zwei Interpretationen zu:



- Nach der ersten Transformation R_x stellt sich die Frage, bzgl. welcher y -Achse die zweite Transformation durchgeführt wird: Bzgl. der y -Achse des Weltkoordinatensystems oder der y_{neu} -Achse des lokalen Koordinatensystems des Würfels?

- 1 Falls die zweite Rotation auf die y -Achse des **Weltkoordinatensystems** bezogen ist:
 - Die Rotation wird durch die Matrix R_y dargestellt.
 - Da Matrizen *von links nach rechts* auf Vektoren durch Multiplikation operieren, beschreibt

$$R_y R_x$$

die verkettete Transformation, bei der zuerst um die x -Achse, dann um die y -Achse des **Weltkoordinatensystems** gedreht wird.

Verkettete Transformationen: Beispiel

- ② Falls die zweite Rotation auf die y_{neu} -Achse des am Würfel „befestigten“ **lokalen Koordinatensystems** bezogen ist:
- Im **lokalen Koordinatensystem** wird die Drehung um die y_{neu} -Achse durch die Matrix R_y ausgedrückt.
 - *Problem:* Wie drückt man die Rotation um die y_{neu} -Achse in **Weltkoordinaten** aus?
 - *Idee:* Transformiere durch R_x^{-1} das **lokale Koordinatensystem** zurück aufs **Weltkoordinatensystem**, drehe dort mit R_y um die y -Achse, und transformiere durch R_x zurück aufs **lokale Koordinatensystem**.
 - Also:

$$R_x R_y R_x^{-1}$$

beschreibt die Drehung um die y_{neu} -Achse des **lokalen Koordinatensystems** (aber ausgedrückt im **Weltkoordinatensystem**).

- Verkettung der beiden Rotationen um x - und y_{neu} -Achse:

$$R_{y_{\text{neu}}} R_x = (R_x R_y R_x^{-1}) R_x = R_x R_y,$$

also gerade die *umgekehrte Reihenfolge* wie im Fall 1.

Verkettete Transformationen

Allgemein:

Seien A_1, \dots, A_k Transformationsmatrizen, die in der Reihenfolge 1 bis k angewendet werden sollen.

- Dann beschreibt

$$A_k \cdots A_2 A_1$$

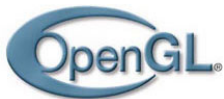
die verkettete Transformation im **Weltkoordinatensystem**,

- und

$$A_1 \cdots A_{k-1} A_k$$

die verkettete Transformation bzgl. der Achsen des **lokalen Koordinatensystems**, das am Würfel befestigt ist.

- Falls Verschiebungen auftreten, verwende homogene Koordinaten.

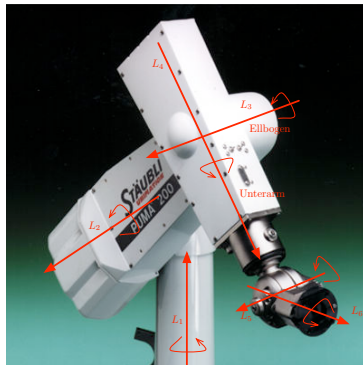


OpenGL (Open Graphics Library) ist ein Industriestandard für 3D-Computergraphik.

- OpenGL-Routinen sind in jeder Graphikkarte implementiert.
- Objekte werden aus *geometrischen Primitiven* aufgebaut (Punkte, Linien, Polygone).
- OpenGL ist ein Zustandsautomat.
Dementsprechend muss ein OpenGL-Programm als Folge von Zustandsänderungen eines Automaten entworfen werden.

- Geometrische Transformationen werden in OpenGL durch Matrizen in homogenen Koordinaten dargestellt.
- Matrizen werden aneinandermultipliziert, in umgekehrter Reihenfolge der Ausführung: Operationen beziehen sich auf Achsen des transformierten Objekts.
- Verständnis von lokalen und globalen Koordinatensystemen ist unerlässlich!

Die einzelnen Bauteile eines Roboters sind verbunden durch Schubgelenke oder Drehgelenke.



Im Bild der Puma 200 Roboterarm mit sechs Drehgelenken. Am Ende des Armes wird üblicherweise ein Werkzeug befestigt.

- Zur Steuerung des Roboters werden über kleine Motoren die Winkel an den Gelenkachsen geändert.
- Die Lage des Armes (insbesondere des Werkzeugs am Ende) ist durch die Winkel eindeutig festgelegt.
- Umgekehrt können mehrere Winkeleinstellungen zur gleichen Positionierung des Werkzeugs führen („Mehrdeutigkeiten“).

Klassische Probleme der Robotersteuerung

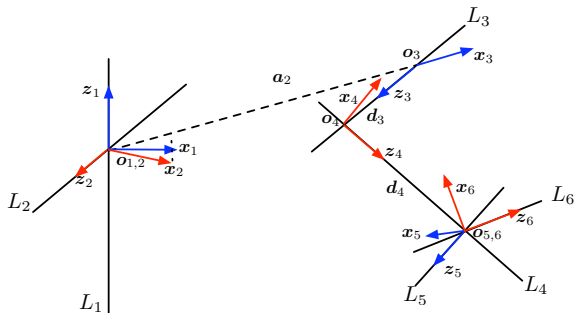
- 1 Das Problem der **direkten Kinematik** besteht darin, bei gegebenen Gelenkwinkeln die Position des Roboterarmes (bzw. des Werkzeugs) zu bestimmen.
- 2 Das Problem der **inversen Kinematik** besteht darin, zu einer vorgeschriebenen Position des Werkzeugs diejenigen Gelenkwinkel zu bestimmen, die den Roboterarm in diese Position bringen.

Wie geht man die Lösung dieser Probleme an?

- Lege ein Weltkoordinatensystem fest und versee jedes Gelenk mit einem lokalen Koordinatensystem.
- Die **Denavit-Hartenberg-Konfiguration** (DH) ist eine Konvention, die vorgibt, wie die lokalen Koordinatensysteme in jeder Achse zu wählen sind.

Denavit-Hartenberg-Konfiguration

Nochmal der Puma 200 - etwas abstrakter dargestellt.



Die Koordinatensysteme sind durch die DH-Konfiguration festgelegt.

Denavit-Hartenberg-Matrix

Werden die Koordinatensysteme gemäß der DH-Konfiguration festgelegt, so wird die Transformation vom $i + 1$ -ten zum i -ten Koordinatensystem in homogenen Koordinaten durch die **Denavit-Hartenberg-Matrix** dargestellt:

$$A_i = \begin{pmatrix} \cos(\vartheta_i) & -\sin(\vartheta_i) \cos(\alpha_i) & \sin(\vartheta_i) \sin(\alpha_i) & a_i \cos(\vartheta_i) \\ \sin(\vartheta_i) & \cos(\vartheta_i) \cos(\alpha_i) & -\cos(\vartheta_i) \sin(\alpha_i) & a_i \sin(\vartheta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Hierbei ist ϑ_i der Gelenkwinkel im i -ten Gelenk.

Für sechs Gelenke ergibt sich die Transformation vom Weltkoordinatensystem zum lokalen Koordinatensystem des Werkzeugs durch

$$A = A_1 A_2 A_3 A_4 A_5 A_6.$$

In dieser Reihenfolge beziehen sich die einzelnen Transformationen immer auf die Achsen der lokalen Koordinatensysteme in den Gelenken.

Gegeben: Gelenkwinkel $\vartheta_1, \dots, \vartheta_6$.

Gesucht: Position und Orientierung des Werkzeugs.

Lösung:

- Mit den bekannten Winkeln kann man die DH-Matrizen aufstellen.
- Damit wird einfach die homogene Transformationsmatrix A ausgerechnet:

$$A = \begin{pmatrix} & & & v_1 \\ & R & & v_2 \\ & & & v_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Die Position ist $\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$, und die Orientierung kann man anhand der **Euler-Winkel** aus der Rotation R ablesen.

Gegeben: Positionsvektor \mathbf{v} und Orientierung (gegeben durch eine Rotation R).

Gesucht: Gelenkwinkel $\vartheta_1, \dots, \vartheta_6$, so dass das Werkzeug des Roboterarmes die gegebene Position und Orientierung annimmt.

Lösungsansätze:

- Schrittweises Auflösen nach $\vartheta_6, \dots, \vartheta_1$ anhand der Gleichungen, die durch die Matrizenmultiplikation gegeben sind.
- Ersetze Ausdrücke $\sin(\vartheta_i)$ und $\cos(\vartheta_i)$ durch Variablen s_i, c_i . Dies führt auf polynomiale Gleichungssysteme, die mit Verfahren der Computeralgebra lösbar sind (Gröbner-Basen).
- Linearisiere das Problem durch Rückführen der Transformationen auf die Lie-Algebra $\mathfrak{se}_3(\mathbb{R})$.
- Geometrisch intuitive Lösung mit Clifford-Algebren.
- Heuristiken.

Die folgenden Bereiche vermitteln ein vertieftes Verständnis für geometrische Transformationen:

- Projektive Geometrie.
- Differentialgeometrie.

Für ein tieferes Verständnis der geometrischen Aspekte der Robotik mag man sich mit folgenden Themengebieten beschäftigen:

- Analytische Mechanik.
- Clifford- und Quaternionenalgebra.
- Lie-Gruppen und Lie-Algebren.

ITEC Dillmann



Industrielle Anwendungen der Informatik und Mikrosystemtechnik






- Vorlesung: Robotik 1 - 3
- Vorlesung: Maschinelles Lernen
- Vorlesung: Medizinische Simulationssysteme
- diverse Praktika zur Robotik

IPR Wörn



Institut für Prozessrechentechnik, Automation und Robotik

- Vorlesung: Robotik in der Medizin
- Vorlesung: Steuerungstechnik für Roboter
- Vorlesung: Innovative Konzepte zur Programmierung von Industrierobotern
- Vorlesung: Steuerungstechnik für Werkzeugmaschinen
- Praktikum: Algorithmen und Medizin

-  W. Boehm, H. Prautzsch
Geometric Concepts for Geometric Design (A K Peters)
-  J.D. Foley, A. van Dam, S.K. Feiner, J.F. Hughes
Computer Graphics - Principles and Practice (Addison Wesley)
-  W. Globke
Kinematik des Puma 200
-  J. Kuipers
Quaternions and Rotation Sequences (PUP)
-  J.S. Selig
Geometric Fundamentals of Robotics (Springer)