



64-544

Grundlagen der Signalverarbeitung und Robotik

[http://tams.informatik.uni-hamburg.de/
lectures/2011ss/vorlesung/GdSR](http://tams.informatik.uni-hamburg.de/lectures/2011ss/vorlesung/GdSR)

Jianwei Zhang



Universität Hamburg
Fakultät für Mathematik, Informatik und Naturwissenschaften
Fachbereich Informatik
Technische Aspekte Multimodaler Systeme

Sommersemester 2011



Gliederung

1. Einführung
2. Grundlagen der Robotik
3. Grundlagen der Sensorik
4. Scandaten verarbeiten
5. Rekursive Zustandsschätzung
6. Fuzzy-Logik
7. Steuerungsarchitekturen





64-544 Vorlesung

Vorlesung	Do. 10:15 -11:45 Uhr
Raum	OC 24 B
Web	http://kogs-www.informatik.uni-hamburg.de/~neumann/Signalverarbeitung-SoSe-2011/
Name	Prof. Dr. Jianwei Zhang Bernd Schütz
Büro	F-308
E-mail	zhang@informatik.uni-hamburg.de
Sekretariat	Tatjana Tetsis
Büro	F-311
Telefon	(040) 42883-2430
E-mail	tetsis@informatik.uni-hamburg.de



64-545 Übungen

Übungen Mi. 10:15 - 11:45 Uhr, Do. 12:30 - 14:00 Uhr

Raum

Web

<http://kogs-www.informatik.uni-hamburg.de/~neumann/Signalverarbeitung-SoSe-2011/Uebungen/>

Name

Manfred Grove (Do.)
 Christian Bähnisch (Mi.)

Büro

F-332

Telefon

(040) 42883-2511

E-mail

grove@informatik.uni-hamburg.de



Organisatorisches

- ▶ Übung
 - ▶ Ausgabe der Aufgabenblätter:
Wie im ersten Teil der Veranstaltung.
 - ▶ Scheinkriterien:
Wie im ersten Teil der Veranstaltung.
 - ▶ Laborbesichtigungen, Vorstellung einzelner Roboterplattformen
Mi. 01.06., 10:45-11:45 Uhr, Vogt-Kölln-Str. 30, Haus F, Raum F325



Terminübersicht, Teil 2 – Robotik

26.05.11 VL

01.06.11 Laborführung TAMS

02.06.11 VL, Übung Christi Himmelfahrt

08.06.11 Übung

09.06.11 VL, Übung

15.06.11 Übung Pfingstferien

16.06.11 VL, Übung Pfingstferien

22.06.11 Übung

23.06.11 VL, Übung

29.06.11 Übung

30.06.11 VL, Übung

06.07.11 Übung

07.07.11 VL, Übung

13.07.11 Übung

14.07.11 VL, Übung



Gliederung

1. Einführung

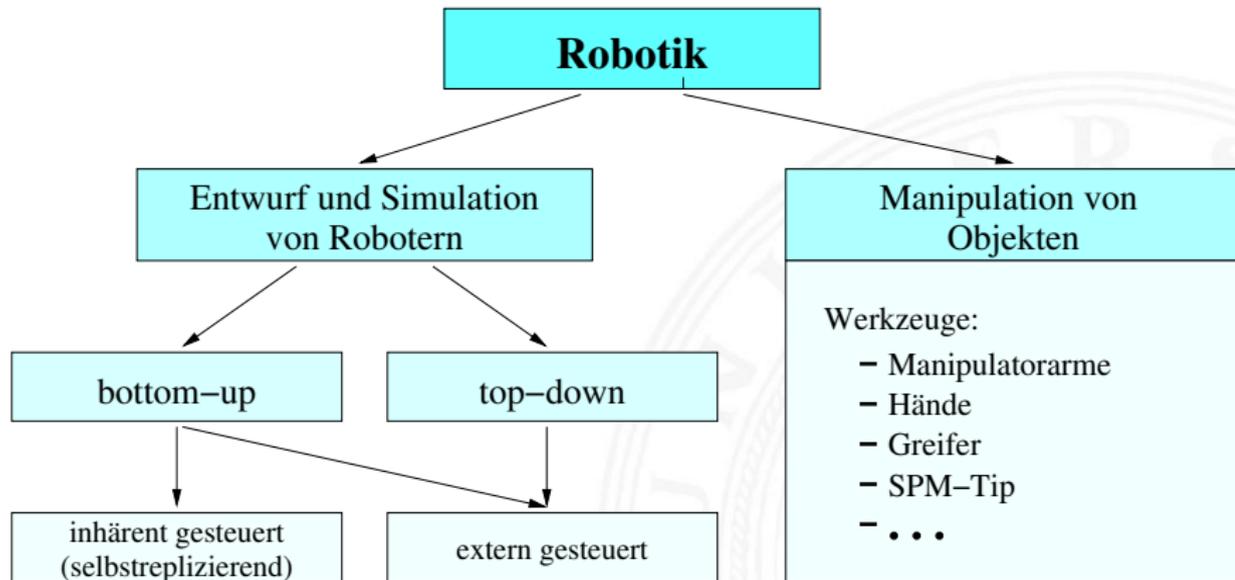
Motivation

2. Grundlagen der Robotik
3. Grundlagen der Sensorik
4. Scandaten verarbeiten
5. Rekursive Zustandsschätzung
6. Fuzzy-Logik
7. Steuerungsarchitekturen



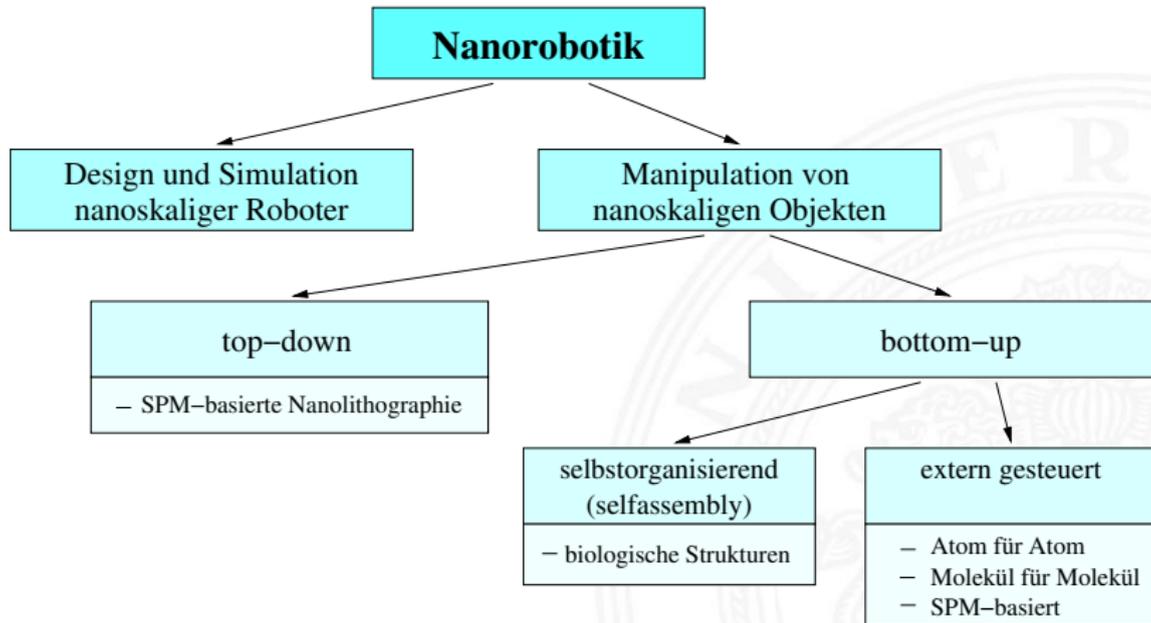


Nanorobotik und Makrorobotik





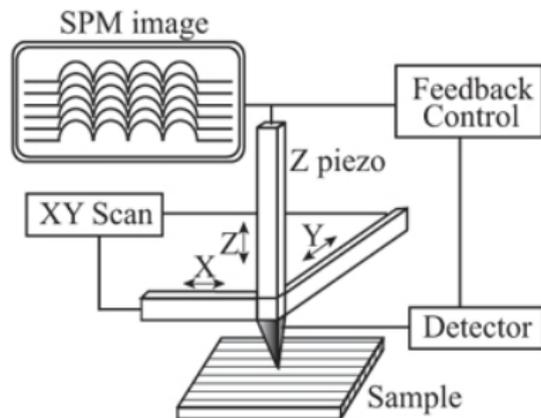
Nanorobotik und Makrorobotik (cont.)



Exkurs: Scanning Probe Microscopes

SPM Family

Scanning Probe Microscope



SPM Family

- STM (1982)
(Scanning Tunneling Microscope)
- SCM (1985)
(Scanning Capacitance Microscope)
- AFM (1986)
(Atomic Force Microscope)
- MFM (1987)
(Magnetic Force Microscope)

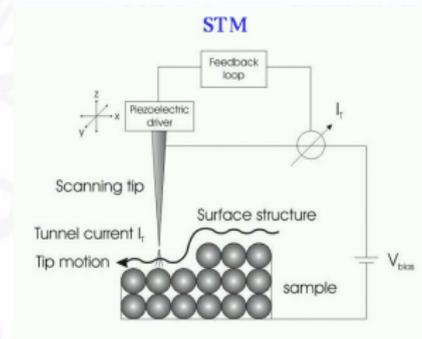
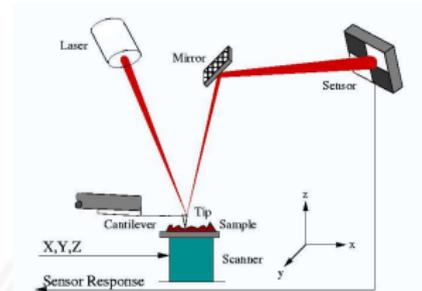
Quelle: quanta.iis.u-tokyo.ac.jp

Exkurs: Scanning Probe Microscopes (cont.)

- ▶ AFM
(Atomic Force
Microscope)

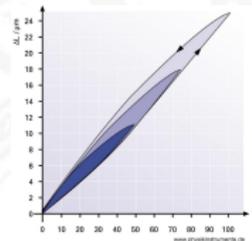


- ▶ STM
(Scanning
Tunneling
Microscope)



SPM als Roboter

- ▶ 3 DOF-Robot
 - ▶ Bewegung in x, y, z
 - ▶ x, y normalerweise open-loop
 - ▶ z normalerweise closed-loop
- ▶ Piezos als Aktuatoren
 - ▶ Hysterese
 - ▶ Kriechen (creep)
- ▶ Sensorik
 - ▶ Fusion von Sensordaten
(z. B. AFM: laterale Kräfte & Topographie)
- ▶ Temperaturdrift





Manipulation

- ▶ Makrorobotik
 - ▶ physikalische Prozesse und Mechanik weitestgehend verstanden
 - ▶ leistungsfähige Modelle vorhanden
 - ▶ viele Möglichkeiten der Automation
- ▶ Nanorobotik
 - ▶ physikalische u. chemische Prozesse in aktiver Forschung
 - ▶ Umgebung hat starken Einfluss auf das Experiment
 - ▶ nicht beschränkt auf mechanische Interaktion
(Temperatur, Licht, Elektrostatik, PH-Wert, . . .)



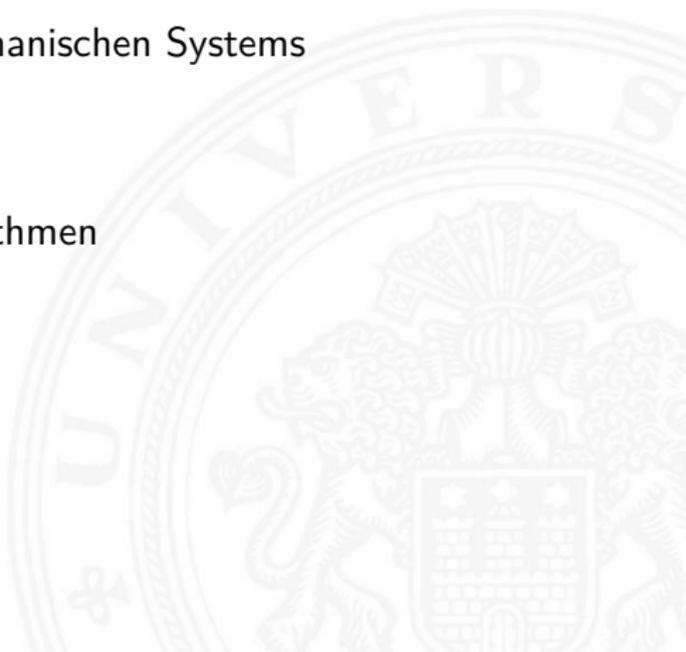
Nanomanipulation & Automation

- ▶ viele Unwägbarkeiten
 - ▶ fast alle Bereiche unter aktueller Forschung
- ⇒ möglichst auf Bewährtes zurückgreifen!
- ▶ möglichst viel aus Makrorobotik adaptieren
- aber:** wegen grundlegend anderer physikalischer u. chemischer Effekte im nanoskaligen Bereich werden neue Konzepte und Verfahren entwickelt werden müssen.
- ⇒ Kennenlernen von Konzepten aus der Makrorobotik, die auch in der Nanorobotik eingesetzt werden können.



Wie sind Robotersysteme aufgebaut?

- ▶ Methodik
- ▶ Entwurf und Aufbau des mechanischen Systems
- ▶ Kinematik und Dynamik
- ▶ Modellierung und Simulation
- ▶ Controller-Design und -Algorithmen
- ▶ Softwarearchitektur
- ▶ Systemintegration
- ▶ ...



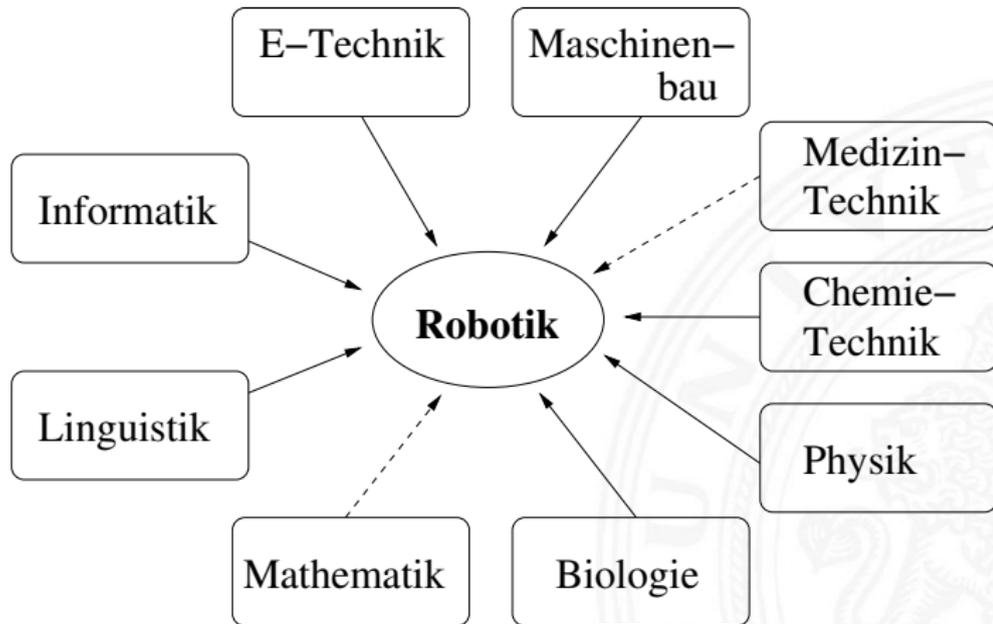


Welche (Teil-) Aufgaben werden vom System durchgeführt?

- ▶ Lokalisation
- ▶ Pfadplanung
- ▶ Bewegung
- ▶ Manipulation
- ▶ Kontrolle von Kraft, Geschwindigkeit. . .
- ▶ Sensordatenerfassung und -verarbeitung
- ▶ Fusion und Interpretation der Daten
- ▶ Mensch-Maschine Schnittstelle
- ▶ Interaktionsmöglichkeiten
- ▶ . . .



Interdisziplinäres Forschungsgebiet





Interdisziplinäres Forschungsgebiet (cont.)

Robotik umfasst sehr viele Teilgebiete selbst aus der Informatik

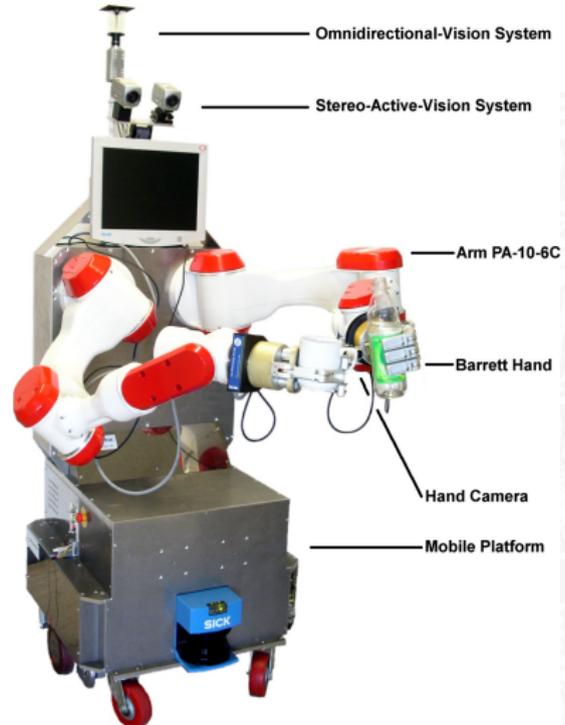
- ▶ Architektur- und Systementwurf
- ▶ Künstliche Intelligenz
- ▶ Bildverarbeitung
- ▶ Sprachverarbeitung
- ▶ Neuronale Netze
- ▶ Regelungstechnik
- ▶ Mechatronik
- ▶ ...



Ein Roboter aus der Makrowelt

TASER – TAMS Service Robot

Forschungsplattform
für Servicerobotik



TAMS Service Robot

Hardwareplattform / Aktoren

- ▶ Neobotix MP-L655
 - ▶ Differentialantrieb
 - ▶ 6 Stunden Akkukapazität
- ▶ 2 × Mitsubishi PA10-6C Arm
 - ▶ 6 Achsen
 - ▶ 10 Kg Tragkraft
- ▶ Barrett 3-Finger Hand: BH8-262
- ▶ Schwenk/Neige Kamerakopf
- ▶ Lautsprecher



TAMS Service Robot (cont.)

Sensoren

- ▶ Kamerasysteme
 - ▶ Stereo Kameras
Sony DFW-VL500 (FireWire, VGA)
 - ▶ Omnidirektionales Sichtsystem
DFW-SX900 (FireWire, 1280 × 960)
 - ▶ 2 × Handkamera
- ▶ Laserentfernungsmesser
 - ▶ 2 × Sick LMS200
 - ▶ Ethernet Anbindung
Embedded-Prozessor
- ▶ Gyroskop in der Plattform



TAMS Service Robot (cont.)

- ▶ Kraft-/Moment Sensoren
 - ▶ Gelenke der Arme
 - ▶ Finger der Hände
- ▶ Positionssensoren
 - ▶ Plattform
 - ▶ Arm
 - ▶ Hand und Finger
 - ▶ Schwenk-/Neige-Einheit

Rechner

- ▶ Standard PC-Hardware
- ▶ Betriebssystem: Linux



TAMS Plattformen

► sehr unterschiedliche Plattformen

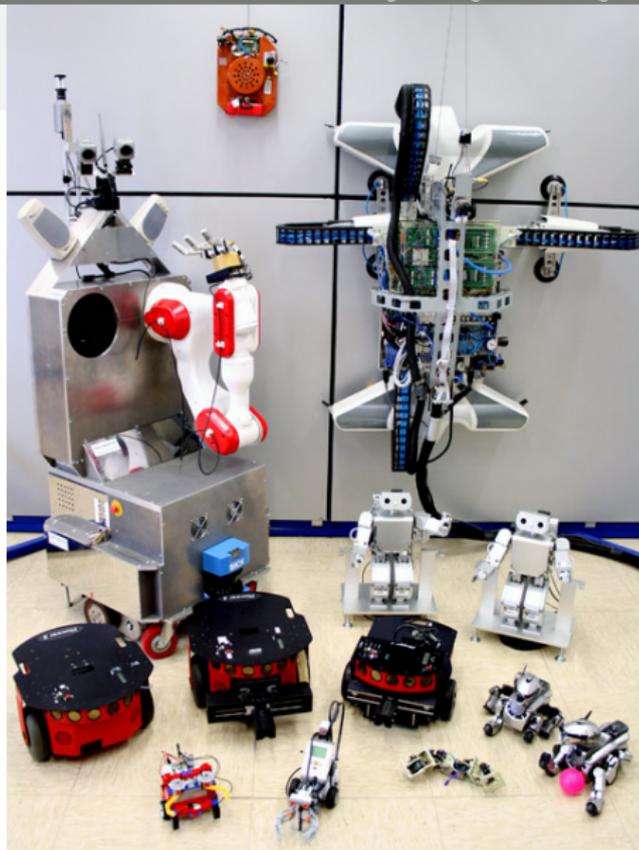
- Service-Roboter
- Humanoide Roboter
- Fensterputzer (pneumatisch)
- Snake-like
- Wall-climbing Module
- Pioneer
- Aibo
- „Edutainment“

⇒ Systemarchitekturen

⇒ Sensoren, Aktoren

⇒ mechanischer Aufbau

⇒ ...





Robotik macht Spass!

- ▶ Roboter bewegen sich - Rechner nicht
- ▶ Interdisziplinarität:
 - ▶ Soft- und Hardwaretechnik
 - ▶ Sensortechnik
 - ▶ Mechatronik
 - ▶ Regelungstechnik
 - ▶ Multimedia, ...
- ▶ Ein Traum der Menschheit:
"Computer sind das bis heute genialste Produkt menschlicher Faulheit."
Computer \Leftrightarrow Roboter



Gliederung

1. Einführung
2. Grundlagen der Robotik
 - Grundbegriffe
 - Roboterklassifikation
 - Koordinatensysteme
 - Homogene Transformationen
 - Verknüpfung der Drehmatrizen
 - Koordinaten-Frames
 - Zusammenfassung der homogenen Transformationen
 - Roboterkinematik
3. Grundlagen der Sensorik
4. Scandaten verarbeiten
5. Rekursive Zustandsschätzung



Gliederung (cont.)

6. Fuzzy-Logik

7. Steuerungsarchitekturen

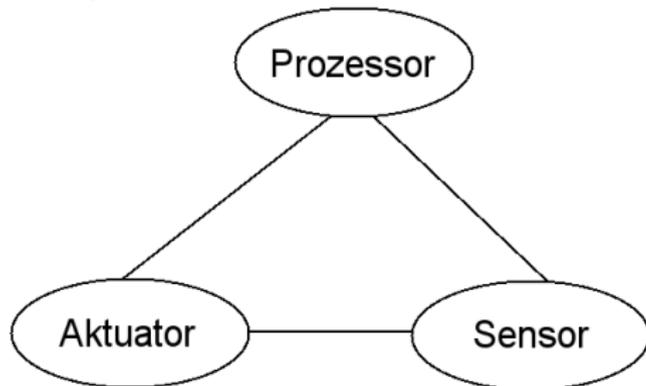




Einführung

Grundbegriffe

Komponenten eines Roboters



Robotik: intelligente Verbindung von Rechnern, Sensorik und Aktuatoren.



Definition von Industrierobotern

Ein Roboter ist laut RIA (*Robot Institute of America*):

...a reprogrammable and multifunctional manipulator, devised for the transport of materials, parts, tools or specialized systems, with varied and programmed movements, with the aim of carrying out varied tasks.

Die JARA (Japan Robot Association) z. B. gibt folgende Merkmale vor:

- ▶ Manual Manipulator
- ▶ fixed Sequence Robot
- ▶ variable Sequence Robot
- ▶ Playback Robot
- ▶ Numerical Control Robot
- ▶ Intelligent Robot



Hintergrund einiger Termini

- “**Robot**” wurde populär mit einem Theaterstück von Karel Capek 1923, war ein tüchtiger Diener.
- “**Robotics**” wurde erfunden von Isaac Asimov 1942.
- “**Autonomous**”: (wörtlich) (gr.) “nach eigenen Gesetzen lebend”
 (*Auto*: Selbst; *nomos*: Gesetz)
- “**Personal Robot**”: ein kleines mobiles Robotersystem mit einfachen Fähigkeiten vom Sichtsystem, Sprechen, Bewegung, usw. (ab 1980).
- “**Service Robot**”: ein mobiles Handhabungssystem mit Sensoren für anspruchsvolle Operationen in Service-Bereichen (ab 1989).



Freiheitsgrade eines Roboters

Degrees of Freedom (DOF):

Die Anzahl der unabhängigen Bewegungen f im Bezugskordinatensystem. Der Freiheitsgrad (DOF) wird von der Anzahl der unabhängigen Variablen des Steuerungssystems bestimmt.

- ▶ Auf einer Ebene: 2 Translationen, 1 Rotation; Freiheitsgrad: $f=3$
- ▶ In einem Raum: 3 Translationen, 3 Rotationen; Freiheitsgrad: $f=6$

Bewegungsfreiheitsgrad

Anzahl der Gelenke F eines Manipulators, die unabhängig gesteuert werden.

- ▶ Freiheitsgrad eines Rotationsgelenks:
- ▶ Freiheitsgrad eines Translationsgelenks:
- ▶ Bewegungsfreiheitsgrad:

$$F_R \leq 3$$

$$F_T = 1$$

$$F = \sum_{i=1}^n (F_{R_i} + F_{T_i})$$



Freiheitsgrade: Beispiele

- ▶ Auf dem Boden bewegliches Fahrzeug:
 - ▶ Translation auf der Bodenfläche (x, y -Koordinate)
 - ▶ Drehung um die senkrecht zur Bodenfläche stehende z -Achse
 Freiheitsgrad $f = 3$
- ▶ Festkörper im dreidimensionalen Raum:
 - ▶ Ort im Raum (x, y, z -Koordinate, Translationen in drei Richtungen)
 - ▶ Rotation um die Achsen des kartesischen Koordinatensystems
 Freiheitsgrad $f = 6$
- ▶ Roboterarm mit 8 Drehachsen:
 - ▶ Bewegungsfreiheitsgrad F ist 8
 Freiheitsgrad $f = 6$
- ▶ menschliche Hand
 - ▶ Bewegungsfreiheitsgrad F ist 22
 Freiheitsgrad $f = 6$



Roboterklassifikation

nach Antriebsprinzip

- ▶ elektrisch
- ▶ hydraulisch
- ▶ pneumatisch
- ▶ biologisch





Roboterklassifikation

nach Arbeitsbereich

▶ stationär

- ▶ Arme mit 2 DOF
- ▶ Arme mit 3 DOF
- ▶ ...
- ▶ Arme mit 6 DOF
- ▶ Redundante Arme (> 6 DOF)
- ▶ Multifinger-Hand

▶ mobil

- ▶ *Automated Guided Vehicles*
- ▶ Portalroboter
- ▶ mobile Plattform
- ▶ Laufmaschinen und fliegende Roboter
- ▶ Anthropomorphe Roboter (Humanoids)



Roboterklassifikation

nach Art der Gelenke

- ▶ translatorisch (“linear joint”, “translational”, “cartesian”, “prismatic”)
- ▶ rotatorisch
- ▶ Kombinationen



Roboterklassifikation

nach Roboterkoordinaten-Systemen

- ▶ Kartesische
- ▶ Zylindrische
- ▶ Kugelförmige





Roboterklassifikation

nach Anwendung

- ▶ Objekt-Manipulation
- ▶ Objekt-Bearbeitung
- ▶ Transport
- ▶ Montage
- ▶ Qualitätsprüfung
- ▶ Einsätze in nicht zugänglichen Bereichen
- ▶ Land- und Forstwirtschaft
- ▶ Unterwasser
- ▶ Bauwesen
- ▶ Service-Roboter in Medizin, Hausarbeit, ...



Roboterklassifikation

nach Intelligenz

- ▶ manuelle Steuerung
- ▶ programmierbar für wiederholte Bewegungen
- ▶ mit Wahrnehmungs- und Reaktionsfähigkeit
- ▶ lernfähig auf Aufgabenebene



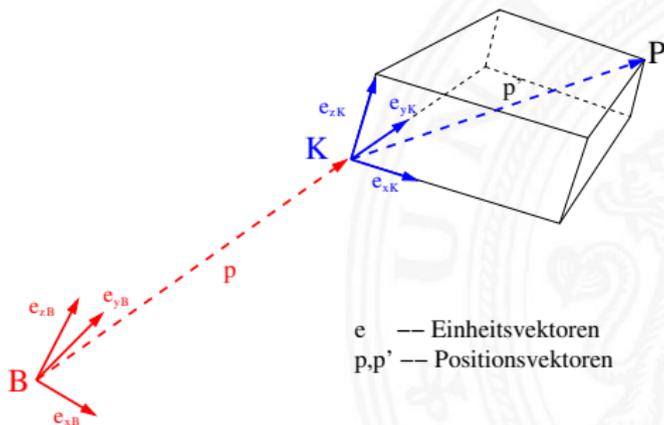
Literatur

Wichtige Sekundärliteratur:

- ▶ **K. S. Fu, R. C. Gonzales and C. S. G. Lee**, *Robotics: Control, Sensing, Vision and Intelligence*, McGraw-Hill, 1987
- ▶ **R. P. Paul**, *Robot Manipulators: Mathematics, Programming and Control*, MIT Press, 1981
- ▶ **J. J. Craig**. *Introduction to Robotics*, Addison-Wesley, 1989.

Koordinatensysteme

Die Lage von Gegenständen, also ihre **Position** und **Orientierung** im euklidischen Raum, lässt sich beschreiben durch Angabe eines kartesischen Koordinatensystems (KS) hier K und der Angabe der Transformation zwischen diesem und dem Ursprungs-Koordinatensystem hier B.



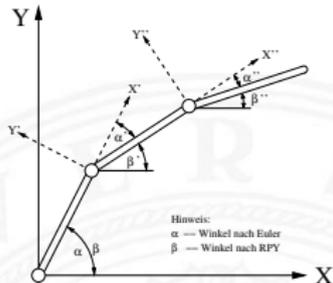
Beschreibung von Position und Orientierung eines Objektes

Position (Objekt-Koordinaten):

- ▶ gegeben durch $\mathbf{p} = [p_x, p_y, p_z]^T \in \mathcal{R}^3$

Orientierung:

- ▶ Euler-Winkel ϕ, θ, ψ
 - ▶ Drehungen werden nacheinander um die Achsen der neuen Koordinatensysteme durchgeführt; z. B. $ZX'Z''$
 - ▶ Dreh- oder Rotationsmatrix $R \in \mathcal{R}^{3 \times 3}$:
 - ▶ redundant; 9 Parameter bei $f=3$
- ▶ Roll-Pitch-Yaw
 - ▶ auf Objektkoordinatensystem bezogen (in Luft- u. Seefahrt üblich)
 - ▶ Rotationen erfolgen auf die unveränderten Achsen (X Rollen, Y Nicken, Z Gieren)



$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

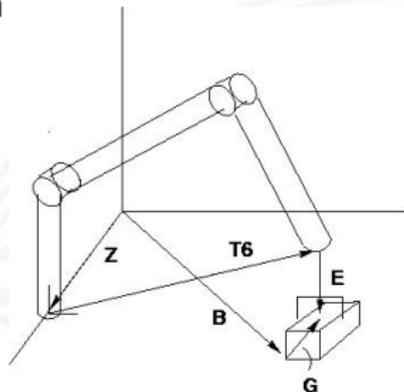
Koordinaten-Transformation

► Überführung von Koordinatensystemen:

Frame: ein Bezugskordinatensystem

Typische Frames:

- Roboterbasis
- Endeffektor
- Tisch (Welt)
- Objekt
- Kamera
- Bildschirm
- ...



Frame-Transformationen überführen einen Frame in einen anderen.



Koordinaten-Transformation (cont.)

- ▶ Matrizenmultiplikation mit einer 3×3 -Matrix kann Rotation, Skalierung oder Scherung beschreiben
 - ▶ **Aber:** Verschiebung (Translation) erfordert Vektoraddition
- ▶ Dies kann umgangen werden, indem eine weitere Spalte zur Matrix hinzugefügt wird.
 - ▶ **Aber:** Die Matrix ist dann nicht mehr invertierbar
- ▶ Übergang nach \mathcal{R}^4 (4×4 -Matrix), um sowohl Rotation, Translation, Scherung, Projektion, lokale und komplette Skalierung zu beschreiben
- ▶ Verwendung homogener Koordinaten



Homogene Koordinaten

- ▶ *Homogene Koordinaten* sind bekannt aus der Computergrafik, um Probleme in der Matrixberechnung zu umgehen
- ▶ Die Punkte eines n -dimensionalen Raums werden in einem $n + 1$ dimensionalen Raum dargestellt
- ▶ Aus $p = (x, y, z) \in \mathcal{R}^3$ wird $p' = (hx, hy, hz, h) \in \mathcal{R}^4$, wobei $h \neq 0 \in \mathcal{R}$
 Beispiel:
 $(2, 5, 4) \in \mathcal{R}^3 \rightarrow \dots, (1, 2.5, 2, 0.5), \dots, (2, 5, 4, 1), \dots, (4, 10, 8, 2), \dots \in \mathcal{R}^4$
- ▶ In der Robotik ist $h = 1$, was einer direkten Projektion zwischen n - und $(n + 1)$ -dimensionalem Raum entspricht
- ▶ h ist eine Art Skalierung



Homogene Transformationen

$$H = \left[\begin{array}{c|c} \begin{array}{l} \textit{Rotation} \\ \textit{Scherung} \\ \textit{lokale Skalierung} \end{array} & \begin{array}{l} \textit{Translation} \\ \\ \\ \end{array} \\ \hline \begin{array}{l} \\ \\ \textit{Projektion} \end{array} & \begin{array}{l} \\ \\ \textit{Skalierung} \end{array} \end{array} \right]$$

$$= \left[\begin{array}{c|c} \begin{array}{c} 3 \\ \times \\ 3 \end{array} & \begin{array}{c} 3 \\ \times \\ 1 \end{array} \\ \hline \begin{array}{c} 1 \times 3 \end{array} & \begin{array}{c} 1 \times 1 \end{array} \end{array} \right]$$



Homogene Transformationen (cont.)

- ▶ In der Robotik ist man nur an Rotation und Translation interessiert Daher ergibt sich:

$$H = \left[\begin{array}{ccc|c} \textit{Rotation} & & & \textit{Translation} \\ \hline & & & \\ & & & \\ & & & \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

- ▶ somit aus \vec{p} und R : $H = \begin{bmatrix} R & \vec{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} \in \mathcal{R}^{4 \times 4}$
- ▶ Verkettung mehrerer H über Matrixmultiplikation
- ▶ nicht kommutativ, d.h. $A \cdot B \neq B \cdot A$

Translation

Verschieben um Vektor $[p_x, p_y, p_z]^T$:

$$T_{(p_x, p_y, p_z)} = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} \vec{a}'^T &= T_{(p_x, p_y, p_z)} \cdot \vec{a}^T \\ &= (a_x + p_x, a_y + p_y, a_z + p_z, 1)^T \end{aligned}$$



Rotation um x -Achse

Drehung um die x -Achse um den Winkel ψ :

$$R_{x(\psi)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\psi) & -\sin(\psi) & 0 \\ 0 & \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} \vec{a}'^T &= R_{x(\psi)} \cdot \vec{a}^T \\ &= \begin{pmatrix} a_x \\ a_y \cdot \cos(\psi) - a_z \cdot \sin(\psi) \\ a_y \cdot \sin(\psi) + a_z \cdot \cos(\psi) \\ 1 \end{pmatrix} \end{aligned}$$



Exkurs: Rotation in der euklidischen Ebene \mathcal{R}^2

- Drehung eines Vektors \vec{p}_1 in \mathcal{R}^2 auf dem Einheitskreis gegen den Uhrzeigersinn um einen festen Ursprung um den Winkel θ .

$$\vec{p}_1 = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \quad \begin{matrix} x_1 = \cos(\alpha) \\ y_1 = \sin(\alpha) \end{matrix} \quad \vec{p}_2 = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \quad \vec{p}_2 = R(\vec{p}_1, \theta)$$

$$x_2 = \cos(\alpha + \theta) = \cos \alpha \cdot \cos \theta - \sin \alpha \cdot \sin \theta = x_1 \cos \theta - y_1 \sin \theta$$

$$y_2 = \sin(\alpha + \theta) = \sin \alpha \cdot \cos \theta + \cos \alpha \cdot \sin \theta = y_1 \cos \theta + x_1 \sin \theta$$

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$$

- also:

$$\vec{p}_2 = R_{(\theta)} \vec{p}_1 \quad \text{mit} \quad R_{(\theta)} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$$



Rotation um y -Achse

Drehung um die y -Achse um den Winkel θ :

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Rotation um z-Achse

Drehung um die z-Achse um den Winkel ϕ :

$$R_z(\phi) = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 & 0 \\ \sin(\phi) & \cos(\phi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

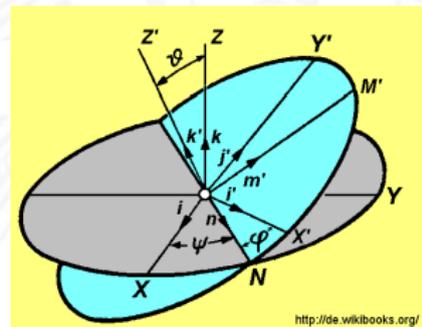


Folge von Drehungen

Drehungen beziehen sich auf die neuen, transformierten Achsen:
 Sequentielle Rechtsmultiplikationen der Transformationsmatrizen
 nach der Reihenfolge der Drehungen.

Beispiel: Rotation um $Z X' Z''$:

1. Eine Drehung ψ um die z -Achse $R_{z,\psi}$
2. Eine Drehung θ um die neue x -Achse $R_{x',\theta}$
3. Eine Drehung ϕ um die neue z -Achse $R_{z',\phi}$



<http://de.wikibooks.org/>

Anmerkung:

Die Folge der Drehungen lässt sich auch durch Drehungen um die raumfesten (ungedrehten) Achsen, allerdings in umgekehrter Reihenfolge, verwirklichen: 1. Drehung um die Z -Achse um ϕ , 2. Drehung um die X -Achse um θ und Drehung um die Z -Achse um ψ .

Verknüpfung der Drehmatrizen

$$\begin{aligned}
 R_{\psi,\theta,\phi} &= R_{z,\psi} R_{x',\theta} R_{z'',\phi} \\
 &= \begin{bmatrix} C\psi & -S\psi & 0 & 0 \\ S\psi & C\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\theta & -S\theta & 0 \\ 0 & S\theta & C\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\phi & -S\phi & 0 & 0 \\ S\phi & C\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} C\psi C\phi - S\psi C\theta S\phi & -C\psi S\phi - S\psi C\theta C\phi & S\psi S\theta & 0 \\ S\psi C\phi + C\psi C\theta S\phi & -S\psi S\phi + C\psi C\theta C\phi & -C\psi S\theta & 0 \\ S\theta S\phi & S\theta C\phi & C\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

Bemerkung: Matrizenmultiplikation ist nicht kommutativ:

$$AB \neq BA$$

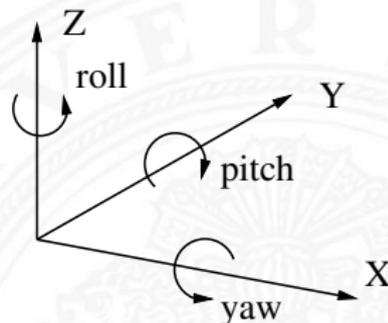


Folge von Drehungen (cont.)

Drehungen beziehen sich auf die Hauptachsen des Ursprungskoordinatensystems: Sequentielle Linksmultiplikationen der Transformationsmatrizen nach Reihenfolge der Drehungen.

Beispiel: Rotation um $X Y Z$ (RPY):

1. Eine Drehung ψ um die x -Achse $R_{x,\psi}$ (yaw)
2. Eine Drehung θ um die y -Achse $R_{y,\theta}$ (pitch)
3. Eine Drehung ϕ um die z -Achse $R_{z,\phi}$ (roll)



Bezogen auf die jeweils mitrotierten Achsen lässt sich die Folge auch durch sequentielle Rechtsmultiplikationen beschreiben mit:

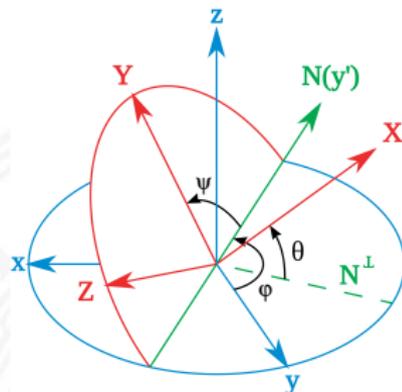
1. Drehung um die z -Achse um ϕ ($R_{z,\phi}$)
2. Drehung um die transformierte y -Achse um θ ($R_{y',\theta}$)
3. Drehung um die abermals transformierte x -Achse um ψ ($R_{x'',\psi}$)



Folge von Drehungen, Beispiel

Rotationsfolge 1 (RPY):

1. Drehung um die x -Achse mit $\psi = 20^\circ$
2. Drehung um die y -Achse mit $\theta = 30^\circ$
3. Drehung um die z -Achse mit $\phi = 50^\circ$



Rotationsfolge 2 (Z Y' X''):

1. Drehung um die z -Achse mit $\phi = 50^\circ$
2. Drehung um die mitgedrehte y -Achse (y') mit $\theta = 30^\circ$
3. Drehung um die abermals mitgedrehte x -Achse (x'') mit $\psi = 20^\circ$



Verknüpfung der Drehmatrizen

$$\begin{aligned}
 R_{\phi, \theta, \psi} &= R_{z, \phi} R_{y, \theta} R_{x, \psi} \\
 &= \begin{bmatrix} C\phi & -S\phi & 0 & 0 \\ S\phi & C\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\theta & 0 & S\theta & 0 \\ 0 & 1 & 0 & 0 \\ -S\theta & 0 & C\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\psi & -S\psi & 0 \\ 0 & S\psi & C\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} C\phi C\theta & C\phi S\theta S\psi - S\phi C\psi & C\phi S\theta C\psi + S\phi S\psi & 0 \\ S\phi C\theta & S\phi S\theta S\psi + C\phi C\psi & S\phi S\theta C\psi - C\phi S\psi & 0 \\ -S\theta & C\theta S\psi & C\theta C\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

Beispiel: Homogene Koordinaten

Punkte im Raum können beschrieben werden durch:

- ▶ Positionsvektoren
- ▶ Rotationsmatrix

$$\begin{aligned} {}^0p &= {}^0d_1 + {}^0p_1 \\ &= {}^0d_1 + {}^0R_1 {}^1p_1 \end{aligned}$$

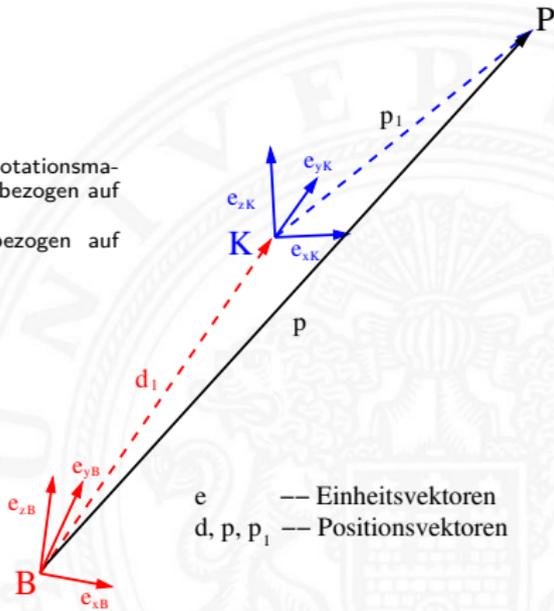
Hinweis:

0R_1 bedeutet: Rotationsmatrix für Frame 1 bezogen auf Frame 0.

0p : Vektor p bezogen auf Frame 0.

In homogenen Koordinaten:

$$\begin{aligned} {}^0p_H &= \begin{bmatrix} {}^0R_1 & {}^0d_1 \\ 0 & 1 \end{bmatrix} \begin{pmatrix} {}^1p_1 \\ 1 \end{pmatrix} \\ &= {}^0H_1 {}^1p_H \end{aligned}$$





Koordinaten-Frames

Sie werden über die Elemente der homogenen Transformation als vier Vektoren dargestellt.

$$H = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$



Inverse Transformationen

Die Inverse einer Drehmatrix ist einfach ihre Transponierte:

$$R^{-1} = R^T \text{ und } RR^T = I$$

wobei I die Identitätsmatrix ist.

Die Inverse von (1) ist:

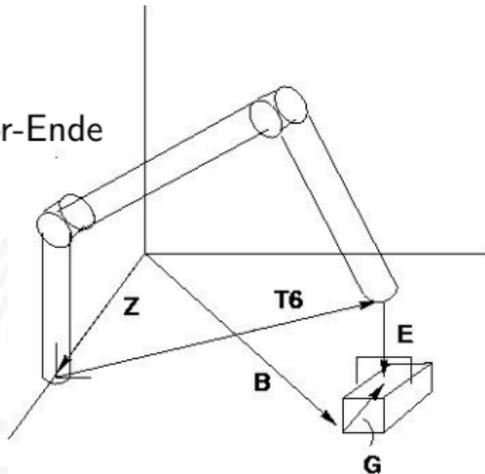
$$H^{-1} = \begin{bmatrix} r_{11} & r_{21} & r_{31} & -\mathbf{p}^T \cdot \mathbf{r}_1 \\ r_{12} & r_{22} & r_{32} & -\mathbf{p}^T \cdot \mathbf{r}_2 \\ r_{13} & r_{23} & r_{33} & -\mathbf{p}^T \cdot \mathbf{r}_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

wobei \mathbf{r}_1 , \mathbf{r}_2 , \mathbf{r}_3 und \mathbf{p} die vier Spaltenvektoren von (1) sind und \cdot das Skalarprodukt von Vektoren darstellt.

Relativtransformationen

Man hat die folgenden Transformationen:

- ▶ Z : Welt \rightarrow Manipulator-Basis
- ▶ T_6 : Manipulator-Basis \rightarrow Manipulator-Ende
- ▶ E : Manipulator-Ende \rightarrow Endeffektor
- ▶ B : Welt \rightarrow Objekt
- ▶ G : Objekt \rightarrow Endeffektor

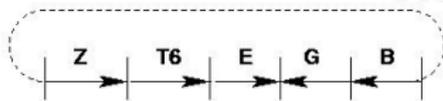




Gleichung der Transformation

Es gibt zwei Beschreibungen der Position des Endeffektors, eine in Bezug auf das Objekt und die andere auf den Manipulator. Sie beschreiben die gleiche Sache:

$$ZT_6E = BG$$



Um die Manipulator-Transformation zu finden:

$$T_6 = Z^{-1}BGE^{-1}$$

Um die Position des Objekts zu bestimmen:

$$B = ZT_6EG^{-1}$$

Dies wird auch als kinematische Kette bezeichnet.



Zusammenfassung der homogenen Transformationen

- ▶ Eine homogene Transformation beschreibt die Position und Orientierung eines Koordinaten-Frames im Raum.
- ▶ Wenn der Koordinaten-Frame bezüglich eines Festkörpers definiert wird, ist die Position und Orientierung des Festkörpers auch eindeutig spezifiziert.
- ▶ Die Beschreibung eines Objektes A kann über eine homogene Transformation bezüglich des Objektes B abgeleitet werden. Umgekehrt geht es auch mit der inversen Transformation.



Zusammenfassung der homogenen Transformationen

- ▶ Mehrere Translationen und Rotationen können zusammengefasst werden. Es gilt:
 - ▶ Wenn die Rotationen / Translationen bezüglich des aktuellen neu definierten (oder veränderten) Koordinatensystems durchgeführt werden, müssen die entsprechenden neu hinzukommenden Transformationsmatrizen über eine Rechtsmultiplikation verknüpft werden.
 - ▶ Wenn sie alle bezüglich des festen Referenz-Koordinatensystems durchgeführt werden, müssen die zusätzlichen Transformationsmatrizen durch eine Linksmultiplikation verknüpft werden.
- ▶ Eine homogene Transformation kann in eine Rotation und in eine Translation zerlegt werden.



Roboterkinematik

Unter Kinematik eines Roboters versteht man die Transformationsvorschrift, die den Zusammenhang zwischen den Gelenkkordinaten eines Roboters \mathbf{q} und den Umweltkoordinaten des Endeffektors \mathbf{x} beschreibt. Sie wird nur durch die Geometrie des Roboters bestimmt.

- ▶ Basisframe;
- ▶ Bezug jedes Frames zu einander;
- ▶ Bildung einer rekursiven Kette;
- ▶ Ziel: Absolute Bestimmung der Position des Endeffektors (Tool Center Point, TCP) im kartesischen Koordinatensystem;



Kinematik-Gleichungen

- ▶ Manipulator: eine Reihe von Gliedern über Gelenke verbundenen
- ▶ Jedem Glied einen eigenen Koordinaten-Frame
- ▶ Homogene Matrix A beschreibt relative Translation und Rotation zwischen zwei aufeinander folgenden Gelenken.
- ▶ Beispiel: Manipulator mit n Gelenken:
 - A_1 : Position und Orientierung des ersten Gliedes
 - A_2 : Position und Orientierung des 2. Gliedes bezüglich Glied 1
 - A_3 : Position und Orientierung des 3. Gliedes bezüglich Glied 2
 - A_n : Position und Orientierung Glied n bezüglich Glied $(n-1)$;

Das folgende Produkt wird definiert als:

$$T_6 = A_1 A_2 \dots A_{n-1} A_n$$



Denavit Hartenberg Konvention

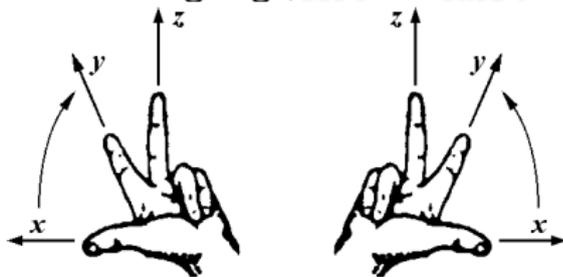
- ▶ Ziel: Berechnung von $T_6 = \prod_{i=1}^n A_i$
 - ▶ T_6 definiert, wie die n Gelenkwinkel zusammenzufassen sind, um 6 kartesische Freiheitsgrade zu beschreiben.
- ▶ Festlegung eines Koordinatensystems pro Segment $i = 1..n$
- ▶ Ermittlung von 4 Parametern pro Segment $i = 1..n$
- ▶ Erstellung der Transformation A_i pro Segment $i = 1..n$
- ▶ Ausgangspunkt: Das Koordinatensystem Σ_0 ist das ortsfeste Ausgangskordinatensystem in der Basis des Manipulators



Denavit Hartenberg Konvention (cont.)

Für die Festlegung des Koordinatensystems gilt:

- ▶ Die z_i -Achse wird entlang der Bewegungsachse des $(i+1)$ -ten Gelenks gelegt.
- ▶ Die x_i -Achse ist senkrecht zur z_{i-1} -Achse und zeigt von ihr weg.
- ▶ Die y_i -Achse wird so festgelegt, dass ein rechtshändiges Koordinatensystem entsteht.

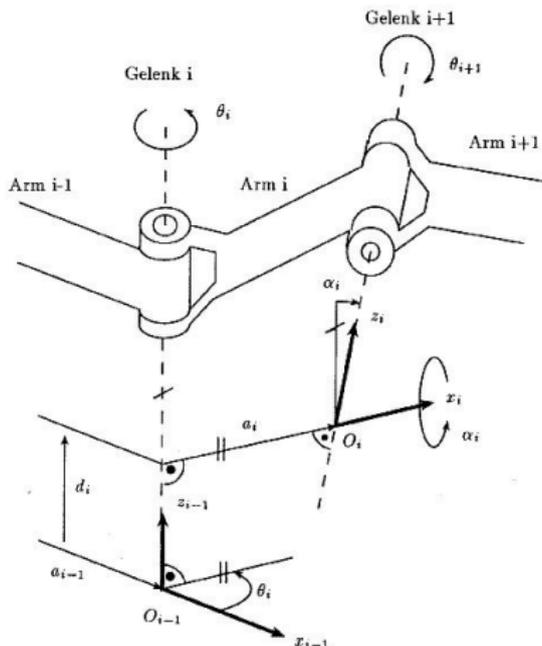


Linkshändiges Koordinatensystem

Rechtshändiges Koordinatensystem

www.wikipedia.org

Denavit Hartenberg Konvention (cont.)





Parameter zur Beschreibung von zwei beliebigen Gelenken

Zwei Parameter zur Bestimmung der Struktur des Gelenkes i :

- ▶ a_i : die kürzeste Entfernung zwischen der z_{i-1} -Achse und der z_i -Achse
- ▶ α_i : der Drehwinkel um die x_i -Achse, der die z_{i-1} -Achse auf die z_i -Achse ausrichtet

Zwei Parameter zur Bestimmung der relativen Distanz und Winkel der benachbarten Gelenke:

- ▶ d_i : die Entfernung vom Ursprung O_{i-1} des $(i-1)$ -ten Koordinatensystems bis zum Schnittpunkt der z_{i-1} -Achse mit der x_i -Achse.
- ▶ θ_i : der Gelenkwinkel um die z_{i-1} -Achse von der x_{i-1} -Achse zur Projektion der x_i -Achse in die x_{i-1}, y_{i-1} -Ebene.



Einige Sonderfälle

Achtung: Die Denavit-Hartenberg Notation ist nicht eindeutig!

- ▶ parallele Achsen ($d_i=0$)
- ▶ sich schneidende Achsen
(KS_i liegt im Schnittpunkt, $a_i = 0$)
- ▶ Lage von KS_n nicht eindeutig, da kein Gelenk $n + 1$ existiert; x_n muss aber trotzdem auf einer Normalen zu z_{n-1} liegen; übliche Wahl: z_n zeigt "aus Hand heraus"



Vorgehen bei gegebener Struktur

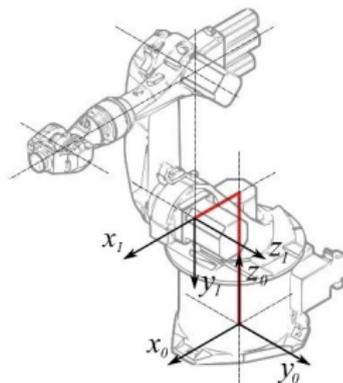
- ▶ Achsen bestimmen und von 1 bis n durchnummerieren
- ▶ KS_{i-1} auf Dreh- bzw. Schubachse i positionieren, z_{i-1} zeigt in Achsrichtung i
- ▶ Normale zw. den Achsen bestimmen; x_i festlegen (in Richtung der Normalen)
- ▶ y_i bestimmen (Rechtssystem)
- ▶ Denavit-Hartenberg Parameter ablesen
- ▶ Gesamttransformation berechnen

DV-Transformation zwischen zwei Gelenken

Beispiel: Überführung eines Koordinatensystems T_0 in ein Koordinatensystem T_1

- ▶ Gelenkwinkel: Rotation θ_1 um die z_0 -Achse, damit die x_0 -Achse parallel zu der x_1 -Achse liegt

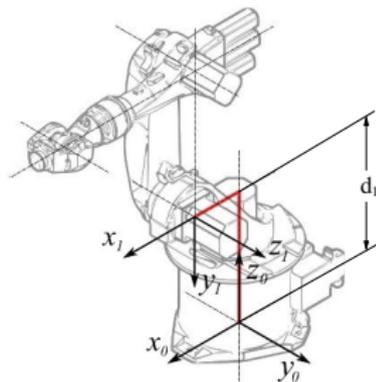
$$R_z(\theta_1) = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



DV-Transformation zwischen zwei Gelenken (cont.)

- ▶ Gelenkabstand: Translation d_1 entlang der z_0 -Achse bis zu dem Punkt, wo sich z_0 und x_1 schneiden

$$T(0, 0, d_1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

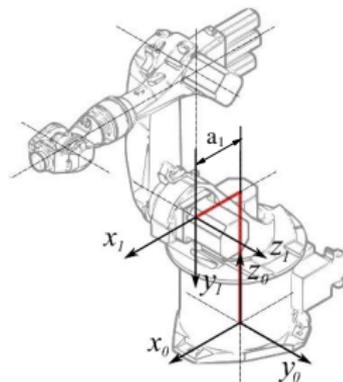




DV-Transformation zwischen zwei Gelenken (cont.)

- ▶ Armlänge: Translation a_1 entlang der x_1 -Achse, um die Ursprünge der Koordinatensysteme in Deckung zu bringen

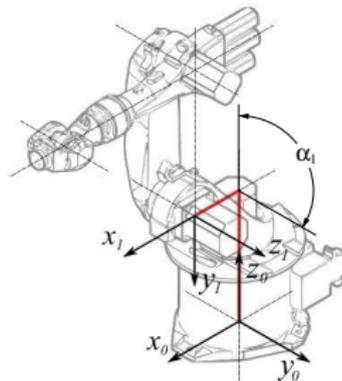
$$T(a_1, 0, 0) = \begin{bmatrix} 1 & 0 & 0 & a_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



DV-Transformation zwischen zwei Gelenken (cont.)

- ▶ Verwindung: Rotation α_1 um die x_1 -Achse, um die z_0 -Achse in die z_1 -Achse zu überführen

$$R_{x(\alpha_1)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_1) & -\sin(\alpha_1) & 0 \\ 0 & \sin(\alpha_1) & \cos(\alpha_1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$





DV-Transformation zwischen zwei Gelenken (cont.)

- ▶ Gesamttransformation von T_0 nach T_1 :

$${}^0T_1 = R_z(\theta_1) \cdot T(0, 0, d_1) \cdot T(a_1, 0, 0) \cdot R_x(\alpha_1)$$

$$= \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 \cos\alpha_1 & \sin\theta_1 \sin\alpha_1 & a_1 \cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 \cos\alpha_1 & -\cos\theta_1 \sin\alpha_1 & a_1 \sin\theta_1 \\ 0 & \sin\alpha_1 & \cos\alpha_1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- ▶ Denavit-Hartenberg-Parameter: $\theta_n, d_n, a_n, \alpha_n$
 - ▶ a_n, α_n – konstant, konstruktionsbedingt
 - ▶ θ_n, d_n – variabel
 - ▶ Rotationsgelenk: θ_n variabel, d_n fix
 - ▶ Translationsgelenk, d_n variabel, θ_n fix



Position

T_6 definiert, wie die n Gelenkwinkel zu 12 nichtlinearen Formeln zusammenzufassen sind, um 6 kartesische Freiheitsgrade zu beschreiben.

- ▶ **Vorwärtskinematik K** definiert als:
 - ▶ $K : \vec{\theta} \in \mathcal{R}^n \rightarrow \vec{x} \in \mathcal{R}^6$
 - ▶ Gelenkwinkel \rightarrow Position + Orientierung
- ▶ **Inverse Kinematik K^{-1}** definiert als:
 - ▶ $K^{-1} : \vec{x} \in \mathcal{R}^6 \rightarrow \vec{\theta} \in \mathcal{R}^n$
 - ▶ Position + Orientierung \rightarrow Gelenkwinkel
 - ▶ nichttrivial, weil K i.A. nicht eindeutig invertierbar



Bahnplanung

Da T_6 nur die Ziel**position** beschreibt, ist explizite Generierung einer Trajektorie nötig, je nach *constraints* unterschiedlich für:

- ▶ Gelenkwinkelraum
- ▶ kartesischen Raum

Interpolation durch:

- ▶ stückweise Geraden
- ▶ stückweise Polynome
- ▶ B-Splines
- ▶ ...

Diese Aspekte werden wegen der beschränkten Zeit nicht weiter vertieft. Siehe hierzu relevante Veranstaltungen unter:
<http://tams-www.informatik.uni-hamburg.de/lectures/>