

# Einführung in die Robotik

**Jianwei Zhang**  
zhang@informatik.uni-hamburg.de

**T | A** Universität Hamburg  
**M | S** Fakultät für Mathematik, Informatik und Naturwissenschaften  
Department Informatik  
Technische Aspekte Multimodaler Systeme

07. April 2009

## Gliederung

### Allgemeine Informationen

#### Einführung

Grundbegriffe

Roboterklassifikation

Koordinatensysteme

Aufgaben

#### Koordinaten eines Manipulators

Warum Koordinaten-Transformation

Homogene Transformation

Verknüpfung der Drehmatrizen

Inverse Transformationen

Gleichung der Transformation

Zusammenfassung der homogenen Transformationen

## Gliederung (cont.)

### Kinematik-Gleichungen

- Denavit-Hartenberg-Konvention
- Parameter zur Beschreibung von zwei beliebigen Gelenken
- Frame-Transformation zwischen zwei Gelenken
- Beispiel mit PUMA 560

### Inverse Kinematik von Manipulatoren

- Analytische Lösbarkeit eines Manipulators
- Beispiel 1: ein planarer dreigelenkiger Manipulator
- Algebraische Lösung des PUMA 560
- Die Lösung für RPY-Winkel
- Geometrische Lösung des PUMA 560
- Eine Programmierumgebung für Roboter unter UNIX: RCCL

### Differentielle Bewegungen mit homogenen Transformationen

## Gliederung (cont.)

- Differentielle Translation und Rotation
- Differentielle homogene Transformation
- Differentielle Rotationen um x-, y-, z-Achse

### Jacobi-Matrix eines Manipulators

- Singuläre Konfigurationen

### Aufgabenbeschreibung

### Roboterprogrammierung auf drei Ebenen

- Off-line Programmierung

### Trajektoriegenerierung

- Generierung von Trajektorien
- Trajektorien im multidimensionalen Raum
- Kubische Polynome zwischen zwei beliebigen Konfigurationen
- Lineare Funktion mit parabolischen Übergängen

## Gliederung (cont.)

Bestimmung der Geschwindigkeiten bei den Zwischenpunkte  
Faktoren für zeitoptimale Bewegungen - Bogenlänge  
Faktoren für zeitoptimale Bewegungen - Krümmung  
Faktoren für zeitoptimale Bewegungen - Bewegungszeit  
Dynamische Constraints aller Gelenke

### Trajektoriengenerierung

Interpolationsverfahren  
Bernstein-Polynome  
B-Splines

### Einführung in RCCL

Umgebungsvariablen  
Compilieren von RCCL Programmen  
Der Simulator

## Gliederung (cont.)

Bewegungstypen in RCCL  
Konfigurationen  
Trajektoriengenerierung in RCCL

### Dynamik

Probleme der Dynamik von Manipulatoren  
Beispiel für einen zweigelenkigen Manipulator  
Lagrange'sche Gleichungen

### Roboterregelung

Klassifikation der Regelung von Roboterarmen  
Gelenkregler des PUMA-Roboters  
Interne Sensorik von Robotern  
Regelungssystem eines Roboters  
Lineare Regelung für Trajektorienverfolgung

## Gliederung (cont.)

Modellbasierte Regelung für Trajektorienverfolgung  
Regelung im Kartesischen Raum

### Programmierung auf Aufgabenebene und Bahnplanung

Grundlage zur Programmierung auf Aufgabenebene

Objekt-Darstellung

Motivation der Bahnplanung

Bewegungsplanung

Konfiguration eines Artefaktes

Planung geometrischer Bahnen

Sichtbarkeitsgraph

Tangentengraph

Voronoi-Diagramm

Heuristische Suche

## Gliederung (cont.)

### Programmierung auf Aufgabenebene und Bahnplanung

Transformation vom Arbeitsraum zum Konfigurationsraum

Berechnung der K-Hindernisse von Polygonen

Berechnung der K-Hindernisse für Stangenkette

Repräsentation des Konfigurationsraums durch

Zerlegungsverfahren

### Programmierung auf Aufgabenebene und Bahnplanung

Potentialfeldmethode

Probabilistische Ansätze

Anwendungsgebiete

Erweiterungen des Basisproblems

### Architekturen sensorbasierter intelligenter Systeme

Architekturen sensorbasierter intelligenter Systeme

## Gliederung (cont.)

Das Perzeption-Aktion-Modell mit Gedächtnis

Das CMAC-Modell

Die Subsumtions-Architektur

Steuerungsarchitektur eines Fisches

Verhaltensfusion

Hierarchie

Eine Architektur für lernende Roboter

Das AuRA-Modell - Arkin '86

Aus- und Rückblick

## Gliederung

Allgemeine Informationen

Einführung

Koordinaten eines Manipulators

Kinematik-Gleichungen

Inverse Kinematik von Manipulatoren

Differentielle Bewegungen mit homogenen Transformationen

Jacobi-Matrix eines Manipulators

Aufgabenbeschreibung

Roboterprogrammierung auf drei Ebenen

Trajektoriegenerierung

Trajektorien-generierung

Einführung in RCCL

Dynamik



## Gliederung (cont.)

Roboterregelung  
Programmierung auf Aufgabenebene und Bahnplanung  
Programmierung auf Aufgabenebene und Bahnplanung  
Programmierung auf Aufgabenebene und Bahnplanung  
Architekturen sensorbasierter intelligenter Systeme  
Aus- und Rückblick



## Allgemeine Informationen (1)

**Vorlesung:** Dienstag 10:15 s.t - 11:45 s.t.  
**Raum:** F334  
**Web:** <http://tams-www.informatik.uni-hamburg.de/lehre/>

**Name:** Prof. Dr. Jianwei Zhang  
**Büro:** F308  
**E-mail:** [zhang@informatik.uni-hamburg.de](mailto:zhang@informatik.uni-hamburg.de)  
**Sprechstunde:** Donnerstag 15:00 - 16:00

**Sekretariat:** Tatjana Tetsis  
**Büro:** F311  
**Tel.:** +49 40 - 42883-2430  
**E-mail:** [tetsis@informatik.uni-hamburg.de](mailto:tetsis@informatik.uni-hamburg.de)

## Allgemeine Informationen (2)

**Übungen:** Dienstag 9:15 s.t - 10:00 s.t.  
**Raum:** F334  
**Web:** <http://tams-www.informatik.uni-hamburg.de/lehre/>

**Name:** Denis Klimentjew  
**Büro:** F330  
**Tel.:** +49 40 - 42883-2508  
**E-mail:** [klimentjew@informatik.uni-hamburg.de](mailto:klimentjew@informatik.uni-hamburg.de)  
**Sprechstunde:** nach Vereinbarung

## Vorkenntnisse

- ▶ Grundlagen der Physik
- ▶ (Grundlagen der Elektrotechnik)
- ▶ lineare Algebra
- ▶ elementare Matrizenalgebra
- ▶ Programmierkenntnisse

# Inhalt

- ▶ mathematische Konzepte (Raumbeschreibung und Koordinaten-Transformationen, Kinematik, Dynamik)
- ▶ Regelungskonzepte (Bewegungsausführung)
- ▶ programmiertechnische Aspekte (RCCL, IGRIP)
- ▶ aufgabenorientierte Bewegungen

# Gliederung

Allgemeine Informationen

Einführung

Grundbegriffe

Roboterklassifikation

Koordinatensysteme

Aufgaben

Koordinaten eines Manipulators

Kinematik-Gleichungen

Inverse Kinematik von Manipulatoren

Differentielle Bewegungen mit homogenen Transformationen

Jacobi-Matrix eines Manipulators

Aufgabenbeschreibung

Roboterprogrammierung auf drei Ebenen



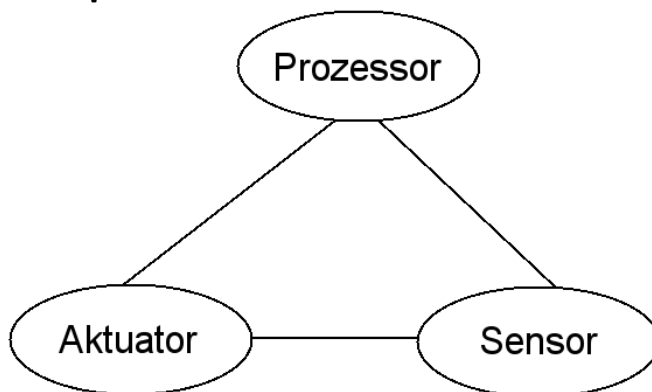
## Gliederung (cont.)

Trajektoriegenerierung  
Trajektoriengenerierung  
Einführung in RCCL  
Dynamik  
Roboterregelung  
Programmierung auf Aufgabenebene und Bahnplanung  
Programmierung auf Aufgabenebene und Bahnplanung  
Programmierung auf Aufgabenebene und Bahnplanung  
Architekturen sensorbasierter intelligenter Systeme  
Aus- und Rückblick

## Einführung

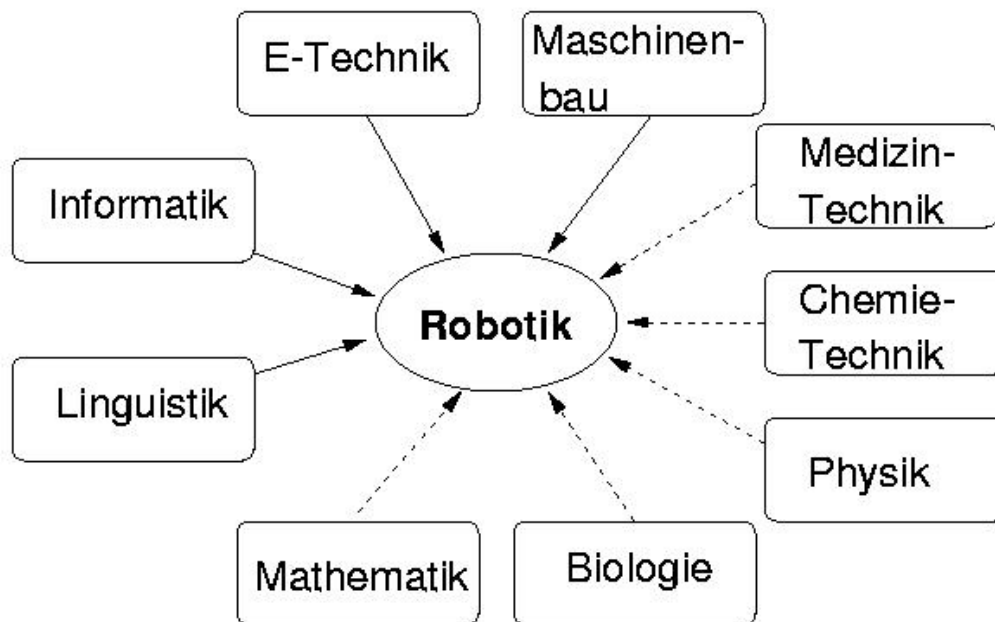
### Grundbegriffe

### Komponenten eines Roboters



*Robotik:* intelligente Verbindung von Rechnern, Sensorik und Aktuatoren.

## Ein interdisziplinäres Gebiet



## 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.*

## Hintergrund einiger Terminie

- “**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 Koordinaten-Ebenen oder Orientierungen auf die sich ein Gelenk oder End-Punkt eines Roboters bewegen kann.

Der DOF wird von der Anzahl der unabhängigen Variablen des Steuerungssystems bestimmt.

- ▶ Auf einer Ebene: translatorische / rotatorische Bewegungen
- ▶ In einem Raum: translatorische / rotatorische Bewegungen - Standort + Orientierung (die maximale Anzahl des DOF eines Festkörpers?)
- ▶ Der DOF eines Manipulators: Anzahl der Gelenke, die unabhängig gesteuert werden.  
Ein “Roboter” sollte mindestens zwei Freiheitsgrade besitzen.

# Roboterklassifikation

nach Antriebsprinzip

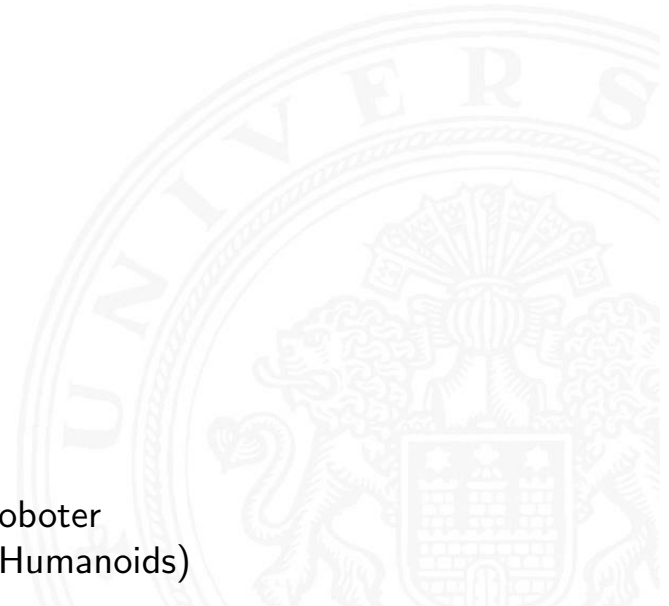
- ▶ elektrisch
- ▶ hydraulisch
- ▶ pneumatisch



# Roboterklassifikation

nach Arbeitsbereich

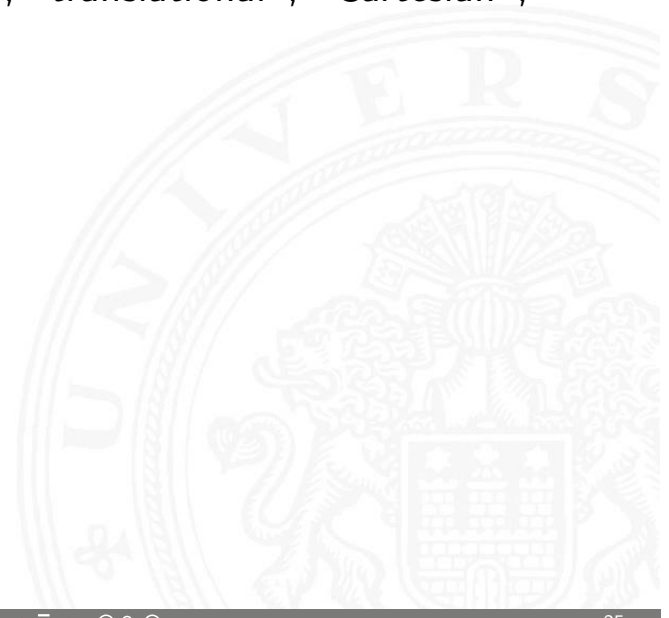
- ▶ stationär
  - ▶ Arme mit 2 DOF
  - ▶ Arme mit 3 DOF
  - ▶ ...
  - ▶ Arme mit 6 DOF
  - ▶ Redundante Arme ( $> 6$  DOF)
  - ▶ Multifinger-Hand
- ▶ mobil
  - ▶ *Automated Guided Vehicles*
  - ▶ Portalroboter
  - ▶ mobile Plattform
  - ▶ Laufmaschinen und fliegende Roboter
  - ▶ Anthropomorphe Roboter (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-Modifikation
- ▶ 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

## 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

## Literatur

Die offiziellen Folien (mit mehr Literaturhinweisen) liegen im TAMS unter "Lehrveranstaltungen"

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).

## Beschreibung von Position und Orientierung

- ▶ **Position:**
  - ▶ gegeben durch  $\vec{p} \in \mathcal{R}^3$
  
- ▶ **Orientierung:**
  - ▶ gegeben durch Projektion  $\vec{n}, \vec{o}, \vec{a} \in \mathcal{R}^3$  der Achsen des KS ins Ursprungssystem
  - ▶ zusammengefasst zu Rotationsmatrix  $R = \begin{bmatrix} \vec{n} & \vec{o} & \vec{a} \end{bmatrix} \in \mathcal{R}^{3 \times 3}$
  - ▶ redundant, da 9 Parameter für 3 Freiheitsgrade
  - ▶ andere Darstellungsformen möglich, z.B. *roll, pitch, yaw* Winkel



## Homogene Transformation

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

## Positionsgleichungen

- ▶ Beschreibung komplexer Situationen über Verkettung mehrerer  $T$  zu einer Positionsgleichung
- ▶ z.B.  $BASE \cdot T_6 \cdot TOOL = OBJ \cdot GRASP$
- ▶ auflösen nach  $T_6 = BASE^{-1} \cdot OBJ \cdot GRASP \cdot TOOL^{-1}$
- ▶ beschreibt Ziel**position**, nicht **Weg** dahin!

## Roboterkinematik

Vielfach besteht nur Interesse an Position und Orientierung des Roboter**greifers**. Ein Roboter ist dann nur ein Objekt wie jedes andere, beschrieben über eine Transformation wie alle anderen.

## Denavit Hartenberg Konvention

- ▶ Definition eines Koordinatensystems pro Segment  $i = 1..n$
- ▶ Definition von 4 Parametern pro Segment  $i = 1..n$
- ▶ Definition einer Transformation  $A_i$  pro Segment  $i = 1..n$
- ▶  $T_6 = \prod_{i=1}^n A_i$

## 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

## Differentielle Bewegung

Die nichtlineare Kinematik  $K$  kann linearisiert werden über die Taylorreihe  $f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!} (x - x_0)^n$ .

- ▶ Die Jacobimatrix  $J$  als Faktor für  $n = 1$  der mehrdimensionalen Taylorreihe ist definiert als:
  - ▶  $J(\vec{\theta}) : \dot{\vec{\theta}} \in \mathcal{R}^n \rightarrow \dot{\vec{x}} \in \mathcal{R}^6$
  - ▶ Gelenkgeschwindigkeit  $\rightarrow$  kartesische Geschwindigkeit
- ▶ Inverse Jacobimatrix  $J^{-1}$  definiert als:
  - ▶  $J^{-1}(\vec{\theta}) : \dot{\vec{x}} \in \mathcal{R}^6 \rightarrow \dot{\vec{\theta}} \in \mathcal{R}^n$
  - ▶ kartesische Geschwindigkeit  $\rightarrow$  Gelenkgeschwindigkeit
  - ▶ nichttrivial, weil  $J$  nicht zwangsweise invertierbar (z.B. nicht quadratisch)

## 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
- ▶ ...

## Aufgaben

- 1.1 Lesen: J. F. Engelberger. *Robotics in Service*, The MIT Press, 1989. (vorhanden im Semesterapparat)
- 1.2 Machen Sie sich mit C-Programmierung vertraut.
- 1.3 Wiederholen Sie Ihre Kenntnisse in der linearen Algebra, insbesondere in der elementaren Matrizenalgebra.