

Einführung in die Robotik

Jianwei Zhang
zhang@informatik.uni-hamburg.de

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

03. May 2011

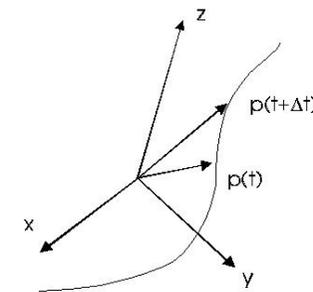
Gliederung

- Allgemeine Informationen
- Einführung
- Koordinaten eines Manipulators
- Kinematik-Gleichungen
- Inverse Kinematik von Manipulatoren
- Differentielle Bewegungen mit homogenen Transformationen**
 - Differentielle Translation und Rotation
 - Differentielle homogene Transformation
 - Differentielle Rotationen um x-, y-, z-Achse
- Jacobi-Matrix eines Manipulators
- Aufgabenbeschreibung
- Robotergrammierung auf drei Ebenen
- Trajektoriegenerierung

Gliederung (cont.)

- Trajektoriegenerierung
- 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

Differentielle Bewegungen



Es gilt

$$\begin{aligned} \Delta \mathbf{p}(t) &= \mathbf{p}(t + \Delta t) - \mathbf{p}(t) \\ &= (T(t + \Delta t) - T(t))\mathbf{p}_0 = (\Delta T(t))\mathbf{p}_0 \end{aligned}$$

Differentielle Bewegungen

wobei T eine 4×4 homogene Matrix von $x_0y_0z_0$ zu xyz ist und \mathbf{p}_0 bezüglich $x_0y_0z_0$ ist.

somit gilt:

$$\begin{aligned} \dot{\mathbf{p}}(t) &= \lim_{\Delta t \rightarrow 0} \frac{\Delta \mathbf{p}(t)}{\Delta t} = \frac{dT(t)}{dt} \mathbf{p}_0 \\ &= \left(\frac{dT(t)}{dt} T^{-1}(t) \right) T(t) \mathbf{p}_0 = \left(\frac{dT(t)}{dt} T^{-1}(t) \right) \mathbf{p} \end{aligned}$$

Ableitung einer homogenen Transformation - I

Gegeben sei eine Transformation T :

$$T = \begin{bmatrix} t_{1,1} & t_{1,2} & t_{1,3} & t_{1,4} \\ t_{2,1} & t_{2,2} & t_{2,3} & t_{2,4} \\ t_{3,1} & t_{3,2} & t_{3,3} & t_{3,4} \\ t_{4,1} & t_{4,2} & t_{4,3} & t_{4,4} \end{bmatrix}$$

wobei jedes Element eine Funktion von einer bestimmten Variable x ist:

$$dT = \begin{bmatrix} \frac{\partial t_{1,1}}{\partial x} & \frac{\partial t_{1,2}}{\partial x} & \frac{\partial t_{1,3}}{\partial x} & \frac{\partial t_{1,4}}{\partial x} \\ \frac{\partial t_{2,1}}{\partial x} & \frac{\partial t_{2,2}}{\partial x} & \frac{\partial t_{2,3}}{\partial x} & \frac{\partial t_{2,4}}{\partial x} \\ \frac{\partial t_{3,1}}{\partial x} & \frac{\partial t_{3,2}}{\partial x} & \frac{\partial t_{3,3}}{\partial x} & \frac{\partial t_{3,4}}{\partial x} \\ \frac{\partial t_{4,1}}{\partial x} & \frac{\partial t_{4,2}}{\partial x} & \frac{\partial t_{4,3}}{\partial x} & \frac{\partial t_{4,4}}{\partial x} \end{bmatrix} dx$$

Ableitung einer homogenen Transformation - I

Beispiel:

$$A = \begin{bmatrix} \cos \theta & -\sin \theta \cos \alpha & \sin \theta \sin \alpha & a \cos \theta \\ \sin \theta & \cos \theta \cos \alpha & -\cos \theta \sin \alpha & a \sin \theta \\ 0 & \sin \alpha & \cos \alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$dA = ?$

Differentielle Translation und Rotation - I

Fall 1: Die differentielle Translation und Rotation seien bezüglich des Basis-Koordinatenframes.

Sei eine homogene Transformation gegeben:

$$T + dT = \text{Trans}_{dx,dy,dz} \text{Rot}_{k,d\theta} T$$

wobei

$\text{Trans}_{dx,dy,dz}$: eine Transformation, die eine Translation von dx, dy, dz in Basis-Koordinaten darstellt.

$\text{Rot}_{k,d\theta}$: eine Transformation, die eine differentielle Drehung $d\theta$ um einen Vektor \mathbf{k} in Basis-Koordinaten darstellt.

dT kann wie folgt berechnet werden:

$$dT = (\text{Trans}_{dx,dy,dz} \text{Rot}_{k,d\theta} - I) T \quad (21)$$

Differentielle Translation und Rotation - II

Fall 2: Die differentielle Translation und Rotation seien bezüglich des gegebenen Koordinatenframes T :

$$T + dT = T \text{Trans}_{dx,dy,dz} \text{Rot}_{k,d\theta} \text{ wobei}$$

$\text{Trans}_{dx,dy,dz}$: eine Transformation, die eine Translation von dz, dy, dz bezüglich des Koordinatenframes T darstellt.

$\text{Rot}_{k,d\theta}$: eine Transformation, die eine differentielle Drehung $d\theta$ um einen Vektor \mathbf{k} im Koordinatenframe T darstellt.
 dT kann wie folgt berechnet werden:

$$dT = T(\text{Trans}_{dx,dy,dz} \text{Rot}_{k,d\theta} - I) \quad (22)$$

Differentielle homogene Transformation - I

Man definiert:

$$\Delta = \text{Trans}_{dx,dy,dz} \text{Rot}_{k,d\theta} - I$$

Dann wird (21) zu:

$$dT = \Delta \cdot T$$

(22) wird zu:

$$dT = T \cdot \mathbf{T} \Delta$$

Differentielle homogene Transformation - II

Die Transformation der Translation um \mathbf{d} :

$$\text{Trans}_{\mathbf{d}} = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

wobei ein differentielles Vektor \mathbf{d} die differentielle Veränderung $d_x \vec{i} + d_y \vec{j} + d_z \vec{k}$ ($\vec{i}, \vec{j}, \vec{k}$ sind drei Einheitsvektoren auf x, y, z) darstellt.

Die Transformation der Rotation mit θ um den Vektor $\mathbf{k} = k_x \vec{i} + k_y \vec{j} + k_z \vec{k}$ ist:

Differentielle homogene Transformation - III

$$\text{Rot}_{\mathbf{k},\theta} = \begin{bmatrix} k_x k_x \text{vers}\theta + \cos\theta & k_y k_x \text{vers}\theta - k_z \sin\theta & k_z k_x \text{vers}\theta + k_y \sin\theta & 0 \\ k_x k_y \text{vers}\theta + k_z \sin\theta & k_y k_y \text{vers}\theta + \cos\theta & k_z k_y \text{vers}\theta - k_x \cos\theta & 0 \\ k_x k_z \text{vers}\theta - k_y \sin\theta & k_y k_z \text{vers}\theta + k_z \sin\theta & k_z k_z \text{vers}\theta + \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (23)$$

wobei $\text{vers}\theta \equiv 1 - \cos\theta$.

(Ableitung s. auch Paul 81, 1.12 "General Rotation Transformation")

Differentielle homogene Transformation - IV

Es gilt:

$$\lim_{\theta \rightarrow 0} \sin \theta \rightarrow d\theta$$

$$\lim_{\theta \rightarrow 0} \cos \theta \rightarrow 1$$

$$\lim_{\theta \rightarrow 0} \text{vers} \theta \rightarrow 0$$

(23) wird dann zu:

$$\text{Rot}_{\mathbf{k}, \theta} = \begin{bmatrix} 1 & -k_z d\theta & k_y d\theta & 0 \\ k_z d\theta & 1 & -k_x d\theta & 0 \\ -k_y d\theta & k_x d\theta & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (24)$$

Differentielle homogene Transformation - V

$$\Delta = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & -k_z d\theta & k_y d\theta & 0 \\ k_z d\theta & 1 & -k_x d\theta & 0 \\ -k_y d\theta & k_x d\theta & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -k_z d\theta & k_y d\theta & d_x \\ k_z d\theta & 0 & -k_x d\theta & d_y \\ -k_y d\theta & k_x d\theta & 0 & d_z \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Differentielle Rotationen um x-, y-, z-Achse - I

Wenn die Rotationen jeweils um x-, y- und z-Achse sind, sind die Rotation-Transformationen wie folgt dargestellt:

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

$$R_{z,\phi} = \begin{bmatrix} C\phi & -S\phi & 0 & 0 \\ S\phi & C\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Differentielle Rotationen um x-, y-, z-Achse - II

Wenn die differentielle Veränderung betrachtet sei, gilt es dann:
 $\sin \theta \rightarrow \delta \theta$ und $\cos \theta \rightarrow 1$.

⇒

$$R_{x,\delta_x} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -\delta_x & 0 \\ 0 & \delta_x & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad R_{y,\delta_y} = \begin{bmatrix} 1 & 0 & \delta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\delta_y & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_{z,\phi} = \begin{bmatrix} 1 & -\delta_z & 0 & 0 \\ \delta_z & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Differentielle Rotationen um x-, y-, z-Achse - III

Werden die Terme mit 2. Ordnung vernachlässigt, dann bekommen wir:

$$R_{z,\delta_x} R_{y,\delta_y} R_{x,\delta_z} = \begin{bmatrix} 1 & -\delta_z & \delta_y & 0 \\ \delta_z & 1 & -\delta_x & 0 \\ -\delta_y & \delta_x & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (25)$$

Man vergleicht (24) und (25):

$$k_x d\theta = \delta_x$$

$$k_y d\theta = \delta_y$$

$$k_z d\theta = \delta_z$$

Differentielle Rotationen um x-, y-, z-Achse - IV

(5) kann wie folgt umgeschrieben werden:

$$\Delta = \begin{bmatrix} 0 & -\delta_z & \delta_y & d_x \\ \delta_z & 0 & -\delta_x & d_y \\ -\delta_y & \delta_x & 0 & d_z \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Δ kann als Bestand von zwei Vektoren \mathbf{d} und δ betrachtet werden.

Gliederung

- Allgemeine Informationen
- Einführung
- Koordinaten eines Manipulators
- Kinematik-Gleichungen
- Inverse Kinematik von Manipulatoren
- Differentielle Bewegungen mit homogenen Transformationen
- Jacobi-Matrix eines Manipulators**
- Singuläre Konfigurationen
- Aufgabenbeschreibung
- Roboterprogrammierung auf drei Ebenen
- Trajektoriegenerierung
- Trajektoriegenerierung
- Einführung in RCCL

Gliederung (cont.)

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

Jacobi-Matrix eines Manipulators

Eine Jacobi-Matrix ist eine multidimensionale Darstellung von Ableitungen.

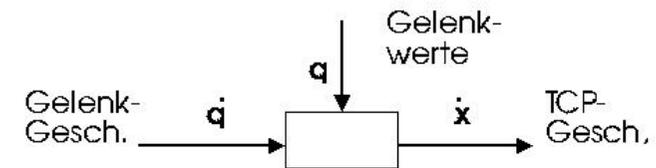
Die Jacobi-Matrix eines Manipulators verbindet die Gelenkgeschwindigkeiten mit Kartesischen Geschwindigkeiten des TCP.
Sei ein Manipulator mit 6 Freiheitsgraden betrachte:

$$T_6 = A_1 A_2 A_3 A_4 A_5 A_6$$

dann gilt mit der Jacobi-Matrix:

$$\begin{bmatrix} T_6 d_x \\ T_6 d_y \\ T_6 d_z \\ T_6 \delta_x \\ T_6 \delta_y \\ T_6 \delta_z \end{bmatrix} = J_{6 \times 6} \begin{bmatrix} dq_1 \\ dq_2 \\ dq_3 \\ dq_4 \\ dq_5 \\ dq_6 \end{bmatrix}$$

Singuläre Konfigurationen - I



Fragestellung: ist die Jacobi-Matrix invertierbar?

Wenn ja, dann gilt:

$$\dot{q} = J^{-1}(q)\dot{x}$$

⇒ um den Effektor eines Roboters im Kartesischen Arbeitsraum mit einer bestimmten Geschwindigkeit zu bewegen.

Singuläre Konfigurationen - II

Bei den meisten Manipulatoren existieren Werte von q wo die Jacobi-Matrix *singulär* wird ($\det J = 0$, J nicht mehr invertierbar).
Solche Konfigurationen werden als *Singuläritäten* des Manipulators bezeichnet.

Zwei Haupttypen von Singularitäten:

- ▶ Arbeitsraum-Grenze-Singuläritäten
- ▶ Arbeitsraum-Interne-Singuläritäten

Beispiel 1: MHI PA 10-6C

Beispiel 2: 2-Gelenkarm

Gliederung

- Allgemeine Informationen
- Einführung
- Koordinaten eines Manipulators
- Kinematik-Gleichungen
- Inverse Kinematik von Manipulatoren
- Differentielle Bewegungen mit homogenen Transformationen
- Jacobi-Matrix eines Manipulators
- Aufgabenbeschreibung**
- Robotergrammierung auf drei Ebenen
- Trajektoriegenerierung
- Trajektoriegenerierung
- Einführung in RCCL
- Dynamik

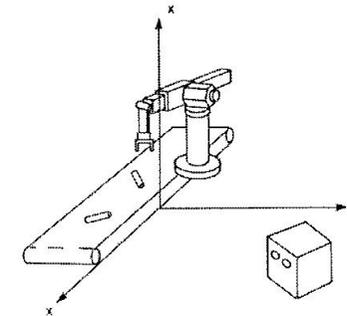
Gliederung (cont.)

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

Aufgabenbeschreibung

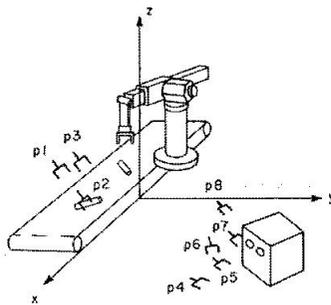
Montage Beispiel

(Paul81, Chapter 5)



Zuerst wurden eine Reihenfolge von Positionen des Manipulator-Endeffektors (pn) spezifiziert.

Aufgabenbeschreibung-Montage Beispiel-(Forts.)



Dann kann die gesamte Aufgabe mit einigen primitiven Roboteroperationen bezüglich dieser nummerierten Positionen beschrieben werden:

Aufgabenbeschreibung-Montage Beispiel-(Forts.)

- | | | |
|-----------|----|------------------------------------|
| ▶ MOVE | p1 | Fahre zum Bolzen |
| ▶ MOVE | p2 | Stelle den Greifer über den Bolzen |
| ▶ GRASP | | Greife den Bolzen |
| ▶ MOVE | p3 | Aufhebe den Bolzen senkrecht |
| ▶ MOVE | p4 | Fahre zum Loch an einem Winkel |
| ▶ MOVE | p5 | Stopp beim Kontakt mit dem Loch |
| ▶ MOVE | p6 | Richte den Bolzen aus |
| ▶ MOVE | p7 | Füge den Bolzen ins Loch ein |
| ▶ RELEASE | | Lasse den Bolzen los |
| ▶ MOVE | p8 | Fahre vom Bolzen weg |

$$\text{MOVE } p_n = \text{MOVE } Z \ T_6 \ E$$

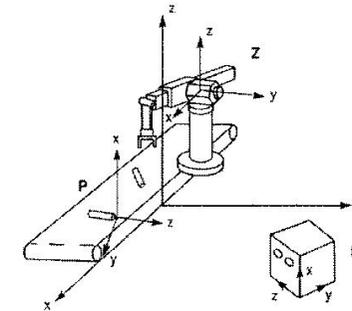
Aufgabenbeschreibung-Montage Beispiel-(Forts.)

wobei die folgenden Repräsentationen gelten:

- ▶ Z : die Basis-Position des Manipulators bezüglich des Welt-Koordinaten-systems;
- ▶ T_6 : das Manipulator-Ende bezüglich seines Basis-Koordinatensystems;
- ▶ E : ein Werkzeug oder ein Endeffektor am Manipulator-Ende.

Aufgabenbeschreibung-Montage Beispiel-(Forts.)

Weiterhin werden noch die folgenden Transformationen eingeführt:



Aufgabenbeschreibung-Montage Beispiel-(Forts.)

- ▶ P : die Position des Bolzens im Basis-Koordinatensystem
- ▶ H : die Position des Würfels mit zwei Löchern
- ▶ ${}^H HR_i$: die Position des betrachteten i -ten Lochs im Würfel bezüglich des H -Koordinaten-systems
- ▶ ${}^P PG$: die Greifposition auf dem Bolzen bezüglich des P -Koordinatensystems
- ▶ ${}^P PA$: die Annäherungsposition des Greifers zum Bolzen
- ▶ ${}^P PD$: die Verlasseposition des Greifers zum Bolzen
- ▶ ${}^{HR} PHA$: die Annäherungsposition des Greifers zum i -ten Loch
- ▶ ${}^{HR} PCH$: die Kontaktposition des Bolzens auf dem Loch
- ▶ ${}^{HR} PAL$: die Einfügeposition des Bolzens
- ▶ ${}^{HR} PN$: die Endposition nach dem Einfügen des Bolzens in das Loch

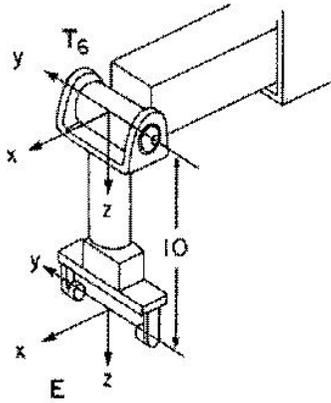
Aufgabenbeschreibung-Montage Beispiel-(Forts.)

Die Aufgabe kann nun als eine Reihe von Gleichungen der Transformationen, aus denen die T_6 -Matrizen gelöst werden können, dargestellt werden:

- ▶ p1: $Z T_6 E = P PA$
- ▶ p2: $Z T_6 E = P PG$
- ▶ : Grasp
- ▶ p3: $Z T_6 E = P PD PG$
- ▶ p4: $Z T_6 E = H {}^H HR_i PHA PG$
- ▶ p5: $Z T_6 E = H {}^H HR_i PCH PG$
- ▶ p6: $Z T_6 E = H {}^H HR_i PAL PG$
- ▶ : RELEASE
- ▶ p8: $Z T_6 E = H {}^H HR_i PN PA$

Aufgabenbeschreibung-Montage Beispiel-(Forts.)

Die Tool-Transformation: Dann gilt:



$$T_6 E = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Beim Einsatz von einer Kamera

Es wird noch eine Transformation CAM eingeführt:

- ▶ CAM : die Transformation vom Kamera zum Welt-Koordinatensystem
- ▶ PC : die Transformation vom Bolzen zum Kamera-Koordinatensystem

Dann gilt es für die Position des Bolzens bezüglich des Welt-Koordinatenframes:

$$P = CAM \quad PC$$

Über die Kamera-Welt-Kalibrierung kann CAM wie folgt erhalten werden:

$$CAM = P \quad PC^{-1}$$

Beim Einsatz von einer Kamera-(Forts.)

Objektpositionen bezüglich des Kamera-Frames werden über die folgende Prozedur bekommen:

```
READ(CAMERA, PC);
```

Um den Manipulator zur Position $p1$ zu steuern, soll es gelten:

$$ZT_6 E = P \quad PA$$

Dies bedeutet die folgende Berechnung von T_6 :

$$T_6 = Z^{-1} P \quad PA \quad E^{-1}$$

Analog werden die Positionierungen an $p2 - p8$ realisiert.

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**
 - Off-line Programmierung
- Trajektoriegenerierung
- Trajektoriegenerierung
- Einführung in RCCL

Gliederung (cont.)

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

Roboterprogrammierung auf drei Ebenen

- ▶ "Teaching by Showing"
 - ▶ Führe den Roboter zur Zielposition und Speichere die Trajektorie;
 - ▶ (Optimierung);
 - ▶ "Play Back".
 - ▶ Explizite Roboterprogrammierung
 - ▶ Spezifikation der Roboterposition;
 - ▶ (Manuelle) Umsetzung in explizite Programme mit einer Roboterprogrammiersprache;
- Drei Typen von Roboterprogrammiersprachen:
- ▶ Spezielle Manipulationsprache: z. B. VAL, V+ ...
 - ▶ Roboterbibliothek für eine Standard-Computersprache: z.B. Robot-Basic, Robot-Pascal, RCCL ...
 - ▶ Roboterbibliothek auch als eine allgemeine Programmiersprache: z.B. AML von IBM, KAREL von GMF ...

Roboterprogrammierung auf drei Ebenen-(Forts.)

- ▶ Roboterprogrammierung auf Aufgabenebene
 - ▶ nicht mehr notwendig, dem Roboter das "how-to-do" zu erklären, sondern lediglich "what-to-do" z.B. "HANDY" von MIT AI Lab, verfüge über Planungskomponente;
 - ▶ auch sensorbasierte Bewegungen sind wünschenswert.

Off-line Programmierung - I

- ▶ wird von einem Computer-Graphik-System unterstützt,
- ▶ ermöglicht, Roboterprogramme ohne Zugriff des realen Roboters zu entwickeln.

Vorteile: Effizienz der Programmierung, Sicherheit, Reduzierung der Stilllegungszeit der Produktionslinie.

Die wichtigsten Bestandteile:

- ▶ Benutzerschnittstelle
- ▶ 3D Objektmodellierung
- ▶ Simulation von Kinematik, Dynamik, Multiprozesse, Sensoren
- ▶ Emulation von Bahnplanung
- ▶ Übertragung der Programmiersprache zum Zielsystem
- ▶ Kalibrierung von Roboter-Welt, Roboter-Sensor, Sensor-Welt



Off-line Programmierung - II

Die noch gewünschten Funktionalitäten:

- ▶ Automatische Layoutplanung einer Roboterzelle
- ▶ Kollisionvermeidung und Bahnoptimierung
- ▶ Planung von koordinierten Bewegungen
- ▶ Simulation von Kraftregelung
- ▶ Automatische Schätzung von Fehlern und Toleranzen
- ▶ Automatische Scheduling von mehreren Arbeitszellen