Vorlesung: Einführung in die Robotik

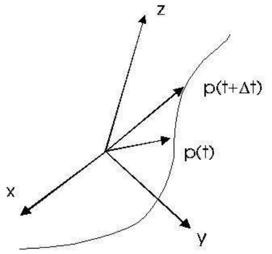
Prof. J. Zhang
zhang@informatik.uni-hamburg.de

Universität Hamburg
Fachbereich Informatik
AB Technische Aspekte Multimodaler Systeme

18. November 2004

Inhaltsverzeichnis

Differentielle Bewegungen



Es gilt

$$\Delta \mathbf{p}(t) = \mathbf{p}(t+\Delta t) - \mathbf{p}(t)$$
$$= (T(t+\Delta t) - T(t))\mathbf{p_0} = (\Delta T(t))\mathbf{p_0}$$

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:

$$\dot{\mathbf{p}}(t) = \lim_{\Delta t \to 0} \frac{\Delta \mathbf{p}(\mathbf{t})}{\Delta t} = \frac{dT(t)}{dt}\mathbf{p_0}$$
$$= (\frac{dT(t)}{dt} T^{-1}(t)) T(t) \mathbf{p_0} = (\frac{dT(t)}{dt} T^{-1}(t))\mathbf{p}$$

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:

$$T + dT = Trans_{dx,dy,dz} Rot_{k,d\theta} T$$

wobei

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

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

dT kann wie folgt berechnet werden:

$$dT = (Trans_{dx,dy,dz} Rot_{k,d\theta} - I)T \tag{1}$$

Differentielle Translation und Rotation - II

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

$$T + dT = T Trans_{dx,dy,dz} Rot_{k,d\theta} \text{ wobei}$$

$\boxed{Trans_{dx,dy,dz}:}$

eine Transformation, die eine Translation von dz, dy, dz bezüglich des Koordinatenframes T darstellt.

$\boxed{Rot_{k,d\theta}:}$

eine Transformation, die eine differentielle Drehung $d\theta$ um einen Vektor $\mathbf{k}$ im Koordinatenframe T darstellt.

dT kann wie folgt berechnet werden:

$$dT = T(Trans_{dx,dy,dz} Rot_{k,d\theta} - I) \tag{2}$$

Differentielle homogene Transformation - I

Man definiert:

$$\boldsymbol{\Delta} = Trans_{dx,dy,dz} Rot_{k,d\theta} - I$$

Dann wird (1) zu:

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

(2) wird zu:

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

Differentielle homogene Transformation - II

Die Transformation der Translation um $\mathbf{d}$:

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

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

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

Differentielle homogene Transformation - III

$$Rot_{\mathbf{k},\theta} = \begin{bmatrix} k_x k_x vers\theta + \cos\theta & k_y k_x vers\theta - k_z \sin\theta & k_z k_x vers\theta + k_y \sin\theta & 0 \\ k_x k_y vers\theta + k_z \sin\theta & k_y k_y vers\theta + \cos\theta & k_z k_y vers\theta - k_x \cos\theta & 0 \\ k_x k_z vers\theta - k_y \sin\theta & k_y k_z vers\theta + k_z \sin\theta & k_z k_z vers\theta + \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

wobei $vers\theta \equiv 1 - \cos\theta$.

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

Differentielle homogene Transformation - IV

Es gilt:

$$\begin{aligned} \lim_{\theta \to 0} \sin\theta &\rightarrow d\theta \\ \lim_{\theta \to 0} \cos\theta &\rightarrow 1 \\ \lim_{\theta \to 0} vers\theta &\rightarrow 0 \end{aligned}$$

(3) wird dann zu:

$$Rot_{\mathbf{k},\theta} = \begin{bmatrix} 1 & -k_z d\theta & k_y d\theta & 0 \\ k_z \theta & 0 & -k_x d\theta & 0 \\ -k_y d\theta & k_x d\theta & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

Differentielle homogene Transformation - V

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

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

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:

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

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

Wenn die differentielle Veränderung betrachtet sei, gilt es dann:

$sin\theta \rightarrow d\theta$ und $cos\theta \rightarrow 1$.

$\Rightarrow$

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

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

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

$$R_{z,\delta_x} R_{y,\delta_y} R_{x,\delta_z} = \begin{bmatrix} 1 & \delta_z & \delta_y & 0 \\ \delta_z & 1 & -\delta_x & 0 \\ -\delta_y & \delta_x & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

Man vergleicht (4) und (5):

$$k_x d\theta = \delta_x$$

$$k_y d\theta = \delta_y$$

$$k_z d\theta = \delta_z$$

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

(154) 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.

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.

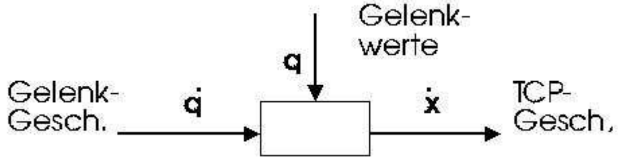
Sei ein Manipulator mit 6 Freiheitsgraden betrachet:

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

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:

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

$\Rightarrow$ um den Effektor eines Roboters im Kartesischen Arbeitsraum mit einer bestimmten Geschwindigkeit zu bewegen.

Singuläre Konfigurationen - II

Bei den meisten Manipulatoren existieren Werte von $\mathbf{q}$ wo die Jacobi-Matrix *singulär* wird

($detJ = 0$, J nicht mehr invertiertbar).

Solche Konfigurationen werden als *Singularitäten* des Manipulators bezeichnet.

Zwei Haupttypen von Singularitäten:

- Arbeitsraum-Grenze-Singularitäten
- Arbeitsraum-Interne-Singularitäten

Beispiel 1:

MHI PA 10-6C

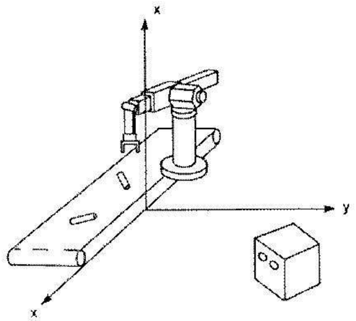
Beispiel 2:

2-Gelenkarm

Aufgabenbeschreibung

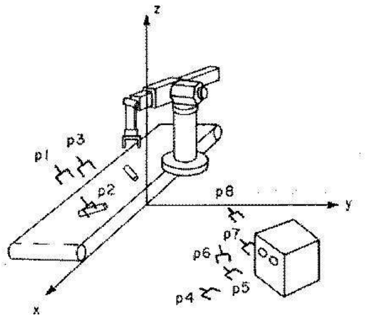
Montage Beispiel

(Paul81, Chapter 5)



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

Aufgabenbeschreibungen - Montage Beispiel - (Forts.)



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

Aufgabenbeschreibungen - 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

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

Aufgabenbeschreibungen - Montage Beispiel - (Forts.)

wobei die folgenden Repräsentationen gelten:

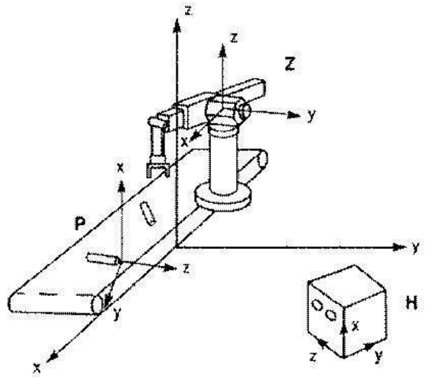
Z: die Basis-Position des Manipulators bezüglich des Welt-Koordinaten--systems;

T_6 : das Manipulator-Ende bezüglich seines Basis-Koordinatensystems;

E: ein Werkzeug oder ein Endeffektor am Manipulator-Ende.

Aufgabenbeschreibungen - Montage Beispiel - (Forts.)

Weiterhin werden noch die folgenden Transformationen eingeführt:



Aufgabenbeschreibungen - 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

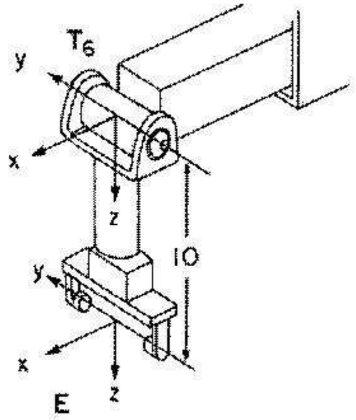
Aufgabenbeschreibungen - Montage Beispiel - (Forts.)

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 HR_i PHA PG
p5: Z T_6 E = H HR_i PCH PG
p6: Z T_6 E = H HR_i PAL PG
RELEASE
p8: Z T_6 E = H HR_i PN PA

Aufgabenbeschreibungen - 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:

$$ZT_6E = P \quad PA$$

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.

Robotergrammierung 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 Manipulationssprache: 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 ...

Robotergrammierung auf drei Ebenen - (Fortsetz.)

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

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 Stillegungszeit der Produktionslinie.

Die wichtigsten Bestandsteile:

- Benutzerschnittstelle
- 3D Objektmodellierung
- Simulation von Kinematik, Dynamik, Multiprocesse, 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