Vorlesung: Einführung in die Robotik

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

Universität Hamburg
Fachbereich Informatik
AB Technische Aspekte Multimodaler Systeme

11. November 2004

Inverse Kinematik von Manipulatoren

Die Problematik:

Robotermanipulatoren werden meistens im *Gelenkwinkelraum* gesteuert.

Zu handhabende Objekte werden aber meistens im Weltkoordinatensystem dargestellt.

Um einen bestimmten Tool-Frame T bezüglich des Welt-Frames zu erreichen, werden die Gelenkwinkel $\theta(t) = (\theta_1(t), \theta_2(t), ..., \theta_n(t))^T$ in folgenden zwei Schritten berechnet:

1. Berechne $T_6 = Z^{-1}BGE^{-1}$;
2. Berechne $\theta_1, \theta_2, ..., \theta_n$ aus T_6.

$\Rightarrow$: inverse Kinematik, sogar wichtiger als direkte Kinematik

Inhaltsverzeichnis

Lösbarkeit am Beispiel PUMA 560

$$T_6 = T'T'' = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

wobei

$$n_x = C_1[C_{23}(C_4C_5C_6 - S_4S_6) - S_{23}S_5C_6] - S_1(S_4C_5C_6 + C_4S_6) \tag{1}$$
$$n_y = S_1[C_{23}(C_4C_5C_6 - S_4S_6 - S_{23}S_5S_6] + C_1(S_4C_5C_6 + C_4S_6) \tag{2}$$
$$n_z = -S_{23}[C_4C_5C_6 - S_4S_6] - C_{23}S_5C_6 \tag{3}$$
$$S_x = ... \tag{4}$$
$$S_y = ... \tag{5}$$
$$S_z = ... \tag{6}$$
$$a_x = ... \tag{7}$$
$$a_y = ... \tag{8}$$
$$a_z = ... \tag{9}$$
$$p_x = C_1[d_6(C_{23}C_4S_5 + S_{23}C_5) + S_{23}d_4 + a_3C_{23} + a_2C_2] - S_1(d_6S_4S_5 + d_2) \tag{10}$$
$$p_y = S_1[d_6(C_{23}C_4S_5 + S_{23}C_5) + S_{23}d_4 + s_3C_{23} + a_2C_2] + C_1(d_6S_4S_5 + d_2) \tag{11}$$
$$p_z = d_6(C_{23}C_5 - S_{23}C_4S_5) + C_{23}d_4 - a_3S_{23} - a_2S_2 \tag{12}$$

Anmerkungen

- Nicht lineare Gleichungen
- Existenz der Lösungen:
  Arbeitsraum (“workspace”): das Volumen des Raums wo der End-Effektor des Manipulators erreichen kann.
  - “dextrous workspace”
  - “reachable workspace”
- Mehrere Gelenkstellungen, die zur gleichen Effektorstellung führen
  Z.B. für PUMA 560:
  - Mehrdeutigkeiten der Lösung $\theta_1, \theta_2, \theta_3$ zu gegebenen $\mathbf{p}$.
  - Für jede Lösung $\theta_4, \theta_5, \theta_6$, eine alternative Lösung ist:

$$\theta_4' = \theta_4 + 180^\circ$$
$$\theta_5' = -\theta_5$$
$$\theta_6' = \theta_6 + 180^\circ$$

- Lösungsstrategien: geschlossene Lösungen vs. numerische Lösungen

Lösungsmethoden

Geschlossene Form:

- algebraische Lösung
  +: korrekte Lösungen aus Gleichungen
  -: geometrisch nicht anschaulich
- geometrische Lösung
  +: Fallunterscheidung der Roboter-Konfigurationen
  -: Robotertypen-spezifisch

Numerische Form:

- iterative Verfahren
  +: Verfahren übertragbar
  -: rechenintensiv, nicht garantierte Konvergenz bei Sondernfällen

"Die inverse Kinematik aller Systeme mit 6 translatorischen und rotatorischen Freiheitsgraden in einer einfachen seriellen Kette ist numerisch lösbar."

Analytische Lösbarkeit eines Manipulators

Unter bestimmten Voraussetzungen an die Armgeometrie (ausreichende Bedingungen) existieren die geschlossenen Lösungen:

- Entweder: 3 aufeinanderfolgende Achsen schneiden sich in einem Punkt
- oder: 3 aufeinanderfolgende Achsen sind parallel

Es ist wichtig, einen Manipulator so zu entwerfen, daß eine Lösung in der geschlossenen Form existiert.

So sind fast alle Industrieroboter entworfen.

Beispiel PUMA:

Die Achsen 4, 5 und 6 schneiden sich.

Gelenk	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	l_1	0	θ_2
3	0	l_2	0	θ_3

$$^0T_3 = \begin{bmatrix} C_{123} & -S_{123} & 0 & l_1C_1 + l_2C_{12} \\ S_{123} & C_{123} & 0 & l_1S_1 + l_2S_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Die algebraische Lösung des Beispiel 1 - I

Spezifikation des End-Effektors: (x, y, ϕ). Für einen solchen Vektor gilt es dann:

$$^0T_3 = \begin{bmatrix} C_\phi & -S_\phi & 0 & x \\ S_\phi & C_\phi & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Wir bekommen vier Gleichungen:

$$C_\phi = C_{123} \tag{13}$$
$$S_\phi = S_{123} \tag{14}$$
$$x = l_1C_1 + l_2C_{12} \tag{15}$$
$$y = l_1S_1 + l_2S_{12} \tag{16}$$

Die algebraische Lösung des Beispiel 1 - II

(Ableitung)

Eine spezielle Funktion $atan2$ wird definiert:

$$\theta = atan2(y,x) = \begin{cases} 0 & \text{für } \; x=0, y=0 \\ \pi/2 & \text{für } \; x=0, y>0 \\ 3*\pi/2 & \text{für } \; x=0, y<0 \\ atan(y,x) & \text{für +x and +y} \\ 2\pi - atan(y,x) & \text{für +x und -y} \\ \pi - atan(y,x) & \text{für -x und +y} \\ \pi + atan(y,x) & \text{fr̈ -x und -y} \end{cases}$$

Die Lösungen:

$$\theta_2 = atan2(S_2, C_2)$$

wobei $S_2 = \pm\sqrt{1-C_2^2}$, und $C_2 = \frac{x^2+y^2-l_1^2-l_2^2}{2l_1l_2}$.

$$\theta_1 = atan2(y,x) - atan2(k_2, k_1)$$

wobei $k_1 = l_1 + l_2C_2$, und $k_2 = l_2S_2$.

θ_3 kann aus der folgenden Gleichung berechnet werden:

$$\theta_1 + \theta_2 + \theta_3 = atan2(S_\phi, C_\phi) = \phi$$

Die geometrische Lösung des Beispiel 1 - I

Berechne θ_2 über "Gesetz des Cosinus":

$$x^2 + y^2 = l_1^2 + l_2^2 - 2l_1l_2\cos(180+\theta_2)$$

Die Lösung:

$$\theta_2 = \pm cos^{-1}\frac{x^2+y^2-l_1^2-l_2^2}{2l_1l_2}$$

$$\theta_1 = \beta \pm \psi$$

wobei gilt:

$$\beta = atan2(y,x), \quad \cos\psi = \frac{x^2+y^2-l_1^2-l_2^2}{2l_1\sqrt{x^2+y^2}}$$

Für $\theta_1, \theta_2, \theta_3$ gilt:

$$\theta_1 + \theta_2 + \theta_3 = \phi$$

Algebraische Lösung mit Hilfe v. Konvertierung in Poly

Um transzendentale Gleichungen in polynomische Gleichungen zu konvertieren, können die folgenden Substitutionen verwendet werden:

$$u = tan\frac{\theta}{2}$$

$$\cos\theta = \frac{1-u^2}{1+u^2}$$

$$\sin\theta = \frac{2u}{1+u^2}$$

Beispiel:

Wir haben eine transzendentale Gleichung:

$$a\cos\theta + b\sin\theta = c$$

Nach der Konvertierung:

$$a(1-u^2)+2bu=c(1+u^2)$$

Die Lösung von u:

$$u=\frac{b\pm\sqrt{b^2-a^2-c^2}}{a+c}$$

Dann:

$$\theta=2tan^{-1}(\frac{b\pm\sqrt{b^2-a^2-c^2}}{a+c})$$

Algebraische Lösung des PUMA 560 - I

Berechnung von $\theta_1,\theta_2,\theta_3$:
Die ersten drei Gelenkwinkel $\theta_1,\theta_2,\theta_3$ bestimmen die Position des Manipulator-Endpunktes $(p_x,p_y,p_z)^T$ (im Fall $d_6=0$).

$$p_x=C_1[S_{23}d_4+a_3C_{23}+a_2C_2]-S_1d_2 \qquad (17)$$
$$p_y=S_1[S_{23}d_4+a_3C_{23}+a_2C_2]+C_1d_2 \qquad (18)$$
$$p_z=C_{23}d_4-a_3S_{23}-a_2S_2 \qquad (19)$$

Daraus ergibt sich:

$$\theta_1=tan^{-1}(\frac{\mp p_y\sqrt{p_x^2+p_y^2-d_2^2}-p_xd_2}{\mp p_x\sqrt{p_x^2+p_y^2-d_2^2}+p_yd_2})$$

Algebraische Lösung des PUMA 560 - II

$$\theta_3=tan^{-1}(\frac{\mp A_3\sqrt{A_3^2+B_3^2-D_3^2}+B_3D_3}{\mp B_3\sqrt{A_3^2+B_3^2-D_3^2}+A_3D_3})$$

wobei

$$A_3=2a_2a_3$$
$$B_3=2a_2d_4$$
$$D_3=p_x^2+p_y^2+p_z^2-a_2^2-a_3^2-d_2^2-d_4^2$$

und

$$\theta_2=tan^{-1}(\frac{\mp B_2\sqrt{p_x^2+p_y^2-d_2^2}+A_2p_z}{\mp A_2\sqrt{p_x^2+p_y^2-d_2^2}+B_2p_z})$$

wobei

$$A_2=d_4C_3-a_3S_3$$
$$B_2=-a_3C_3-d_4S_3-a_2$$

Die Lösung für RPY-Winkel

(RPY: *Roll, Pitch, Yaw*)

$$T = R_{z,\phi}R_{y,\theta}R_{x,\psi}$$

Wir lösen die folgende Gleichung:

$$R_{z,\phi}^{-1}T = R_{y,\theta}R_{x,\psi}$$

$$\begin{bmatrix} f_{11}(\mathbf{n}) & f_{11}(\mathbf{o}) & f_{11}(\mathbf{a}) & 0 \\ f_{12}(\mathbf{n}) & f_{12}(\mathbf{o}) & f_{12}(\mathbf{a}) & 0 \\ f_{13}(\mathbf{n}) & f_{13}(\mathbf{o}) & f_{13}(\mathbf{a}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C\theta & S\theta S\psi & S\theta C\psi & 0 \\ 0 & C\psi & -S\psi & 0 \\ -S\theta & C\theta S\psi & C\theta C\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

wobei

$$f_{11} = C\phi x + S\phi y$$
$$f_{12} = -S\psi x + C\phi y$$
$$f_{13} = z$$

Die Gleichung mit $f_{12}(\mathbf{n})$ führt:

$$-S\phi n_x + C\phi n_y = 0$$

$\Rightarrow$:

$$\phi = atan2(n_y, n_x)$$

und

$$\phi = \phi + 180^\circ$$

Die Gleichungen mit dem Element 1,3 und 1,1 sind jeweils:

$$-S\theta = n_z$$

und

$$C\theta = C\phi n_x + S\phi n_y$$

$\Rightarrow$:

$$\theta = atan2(-n_z, C\phi n_x + S\phi a_y)$$

Die Gleichungen mit dem Element 2,3 und 2,2 sind jeweils:

$$-S\psi = -S\phi a_x + C\phi a_y$$

$$C\psi = -S\phi o_x + C\phi o_y$$

$\Rightarrow$:

$$\psi = atan2(S\phi a_x - C\phi a_y, -S\phi o_x + C\phi o_y)$$

Geometrische Lösung des PUMA 560 - I

Definition von verschiedenen Arm-Konfigurationen:

Für Schulter:

RIGHT-arm, LEFT-arm

Für Ellbogen:

ABOVE-arm, BELOW-arm

Für Handgelenk:

WRIST-DOWN, WRIST-UP

Darauf basierend werden die folgenden Anzeige-Variablen definiert:

$$ARM = \begin{cases} +1 & \text{RIGHT-arm} \\ -1 & \text{LEFT-arm} \end{cases}$$

$$Elbow = \begin{cases} +1 & \text{ABOVE-arm} \\ -1 & \text{BELOW-arm} \end{cases}$$

$$WRIST = \begin{cases} +1 & \text{WRIST-DOWN} \\ -1 & \text{WRIST-UP} \end{cases}$$

Die gesamte Lösung der inversen Kinematik wird über Analyse solcher Arm-Konfiguration erzielt.

Probleme bei der Entwicklung von Steuerungssoftware

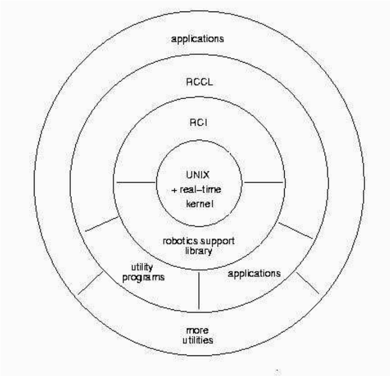
- Bisher war die Entwicklung von Steuerungssoftware an einem bestimmten Robotertyp gekoppelt.
- Jede Maschine hatte damit ihre eigene spezielle Software, die genau die Fähigkeit dieser Baureihe ausnutzte.
- $\Rightarrow$ Erweiterbarkeit und Portierbarkeit war damit eine langwierige und schwierige Aufgabe.

Es steht der Bedarf, eine allgemeine Steuerungssoftware zu entwickeln, die folgende Ziele erfüllen soll:

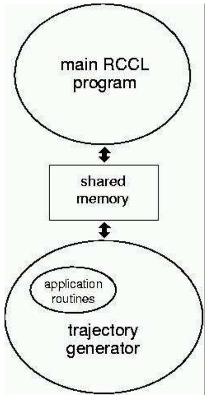
- Steuerungsmöglichkeiten für Low-Level Eigenschaften der Hardware;
- größtmögliche Portierbarkeit auf die verschiedenen Plattformen;
- eine möglichst große Mächtigkeit zur flexiblen und schnellen Erstellung von Anwendungen in derselben Sprache, in der die Steuerungssoftware geschrieben wurde.

Eine Programmierumgebung für Roboter unter UNIX: I

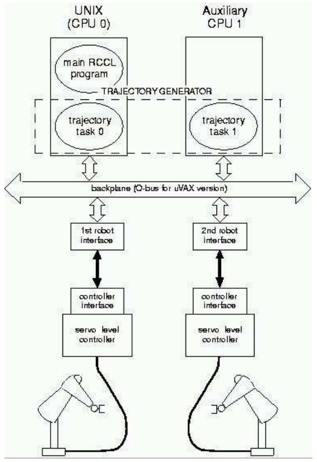
RCCL: *Robot Control C Library*



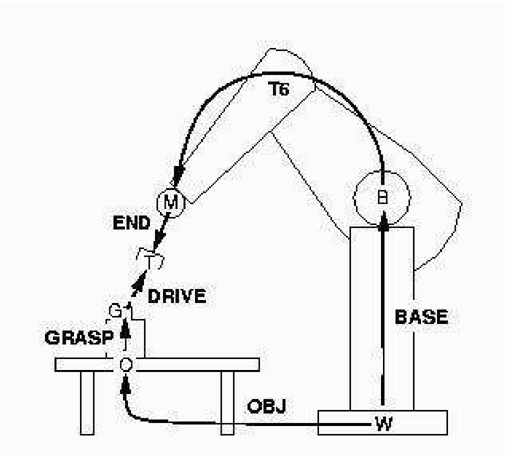
Die Systemarchitektur

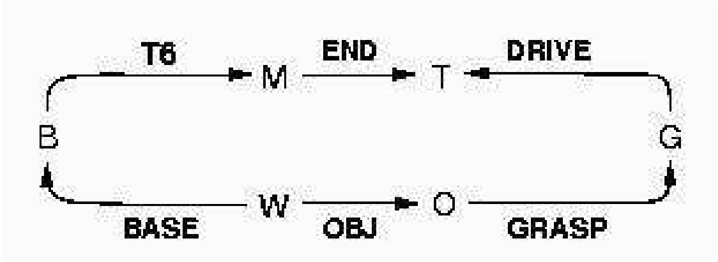


Steuerung von mehreren Robotern



Beschreibung von Bewegungen mit Hilfe der Positionsg





Ein Beispiel - Programm

```
#include <rccl.h>
#include "manex.560.h"

main()
{
        TRSF_PTR p, t;                                          /*#1*/
        POS_PTR pos;                                            /*#2*/
        MANIP *mnp;                                             /*#3*/
        JNTS rcclpark;                                          /*#4*/
        char *robotName;                                        /*#5*/

        rcclSetOptions (RCCL_ERROR_EXIT);                       /*#6*/
        robotName = getDefaultRobot();                          /*#7*/
        if (!getRobotPosition (rcclpark.v, "rcclpark", robotName))
         { printf ("position 'rcclpark' not defined for robot\n");
```

```
        exit(-1);
      }
                                                         /*#8*/
    t = allocTransXyz ("T", UNDEF, -300.0, 0.0, 75.0);
    p = allocTransRot ("P", UNDEF, P_X, P_Y, P_Z, xunit, 180.0);
    pos = makePosition ("pos", T6, EQ, p, t, NULL);      /*#9*/

    mnp = rcclCreate (robotName, 0);                     /*#10*/
    rcclStart();

    movej (mnp, &rcclpark);                              /*#11*/

    setMod (mnp, 'c');                                   /*#12*/
    move (mnp, pos);                                     /*#13*/
    stop (mnp, 1000.0);

    movej (mnp, &rcclpark);                              /*#14*/
```



```
    stop (mnp, 1000.0);

    waitForCompleted (mnp);                              /*#15*/
    rcclRelease (YES);                                   /*#16*/
}
```


Ein Beispiel - Roboterbewegungen