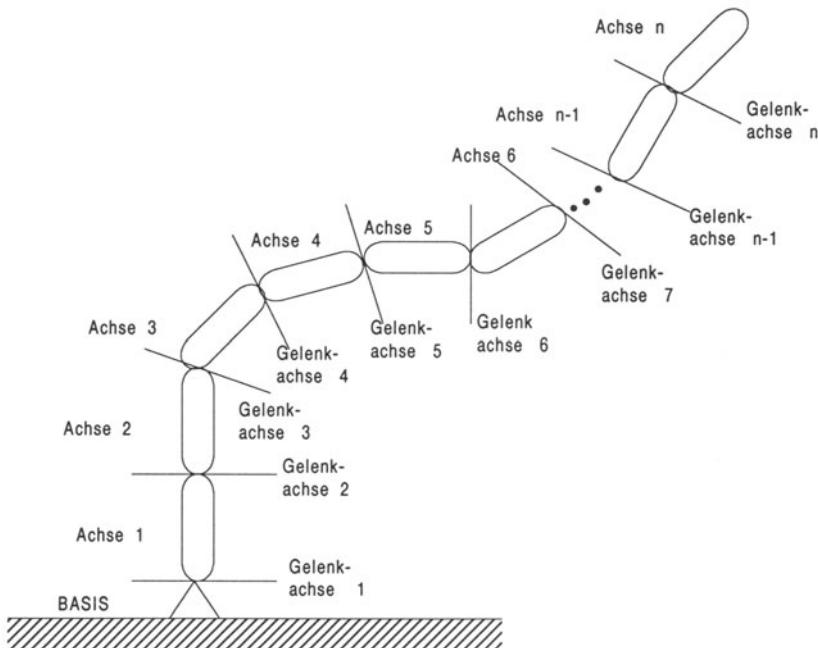


## A Kinematische Beschreibung von Industrierobotern

### A.1 Einführung

Wie in Kapitel 3.1 erwähnt, besteht ein Industrieroboter aus mehreren Achsen, die gelenkig mit einander verbunden sind. Diese Achsen werden im mechanischen Sinn als starre Körper idealisiert, und die Gelenke stellen die Verbindungen zwischen diesen starren Körpern dar. Die Art der Gelenke beeinflußt den Getriebefreiheitsgrad des Industrieroboters (vgl. Kapitel 3.2). Die Stellung des Roboters im Raum ergibt sich durch die jeweilige Stellung der Gelenke; bei rotatorischen Achsen somit durch die Winkelstellung des jeweiligen Gelenkes und bei translatorischen Achsen durch die Ausfahrlänge der jeweiligen Achse. Wichtig ist in diesem Zusammenhang, daß die Art des Gelenkes natürlich eine große Rolle hinsichtlich des Getriebefreiheitsgrades spielt. Der Einfachheit halber sollen hier nur Gelenke behandelt werden, die für die nächst folgende Achse einen zusätzlichen Freiheitsgrad mehr zulassen.



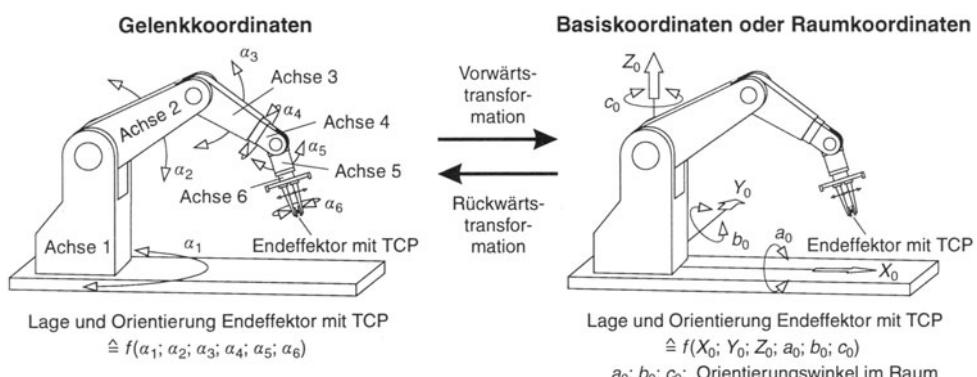
**Bild A-1** Aufbau einer offenen kinematischen Kette (nur rotatorische Achsgelenke)

Den kinematischen Aufbau eines Roboters idealisiert man nach VDI-2861 als sog. offene kinematische Kette. Die Lage der aufeinander folgenden Achsen bzw. Segmente ist einmal durch die geometrische Form der Segmente und zum anderen durch die Stellung der Achsgelenke gegeben. Weiterhin sieht man sehr deutlich, daß sich die Lage aller folgenden Achsen (von  $n-k$  bis  $n$ ) im Raum ändert, wenn die Winkelstellung der Achse  $n-k$  verändert wird.

Ein Industrieroboter besteht aus einer solchen Folge von Achsen bzw. Segmenten (Verbindungselementen, engl.: links), die durch Achsgelenke (engl.: joints) verbunden sind. In Kapitel 7.2.2 wurde bereits darauf hingewiesen, daß es zur Beschreibung der Bewegungen eines Roboters mehrere Koordinatensysteme gibt. Man benutzt einmal das sog. Raumkoordinatensystem – meist ein kartesisches –, und zum anderen verwendet man das Gelenkkoordinatensystem, das die Stellungen der einzelnen Gelenkkachsen angibt. Das Raumkoordinatensystem stellt also in diesem Fall das Basiskoordinatensystem dar, an dem das Segment 1 mit der Gelenkkurve 1 angeflanscht ist.

Heutige Robotersteuerungen bieten dem Bediener fast immer die Möglichkeit, die Bewegungen bzw. die Punkte im Raum wahlweise mit Hilfe des Basis-/Raumkoordinaten- oder mit Hilfe des Gelenkkoordinatensystems zu beschreiben. Der Stellbefehl für die einzelnen Achsen wird aber von der Steuerung immer als Wert für die Gelenkvariable (also bei rotatorischen Achsen ein Winkelwert und bei translatorischen Achsen eine Strecke) an die Regelung weitergegeben. Somit muß die Steuerung immer eine Koordinatentransformation durchführen, wenn der Bediener im Basis-/Raumkoordinatensystem arbeitet, um auf die Gelenkkoordinaten zu gelangen. Diesen Vorgang nennt man *Rückwärtstransformation* bzw. *inverse kinematische Transformation*.

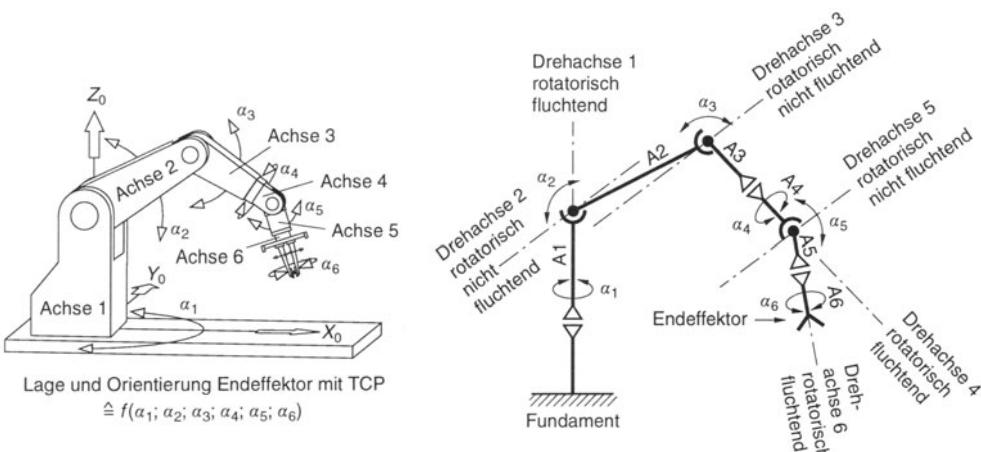
Die Steuerung muß aber dem Bediener Raumpunkte im Basis-/Raumkoordinatensystem anzeigen. Dies bedeutet, daß bei gegebenen Achsstellungen, d.h. bei bekannten Achswinkeln bei rotatorischen Achsen bzw. bei bekannten Strecken bei translatorischen Achsen, aus diesen Werten die Lage (d.h. Position und Orientierung) des letzten Segmentes – des Endeffektors – in Raum-/Basiskoordinaten anzugeben ist. Diesen Vorgang nennt man *Vorwärtstransformation* bzw. *kinematische Transformation*.



**Bild A-2** Vorwärts- und Rückwärtstransformation

Um Position und Orientierung des letzten Segmentes einer solchen offenen kinematischen Kette im ortsfesten Basiskoordinatensystem beschreiben zu können, ist es notwendig, eine mathematische Beziehung zwischen diesem Basiskoordinatensystem und dem Körperkoordinatensystem des  $n$ -ten Segmentes und allen Zwischensegmenten zu kennen. Es sind somit bei einem  $n$ -achsigen Roboter immer mathematische Beziehungen zwischen  $n+1$  Koordinatensystemen ( $n$  Achsen + 1 Basiskoordinatensystem) herzuleiten.

Hierzu benutzt man ein Verfahren, das ursprünglich von *Denavit* und *Hartenberg* zur Beschreibung von mechanischen Starrkörperketten verwendet und von *Paul* erweitert wurde.



**Bild A-3** Kinematik und Koordinatensysteme eines sechsachsigen Knickarmroboters

Kapitel A.2 gibt einige mathematische Grundlagen, die notwendig sind, um dieses Lösungsverfahren zu beschreiben.

## A.2 Mathematische Grundlagen

Vorab eine Anmerkung zur verwendeten Kurzschreibweise der Vektoren  $w$ :

Der „Hochindex“ gibt an, in welchem Koordinatensystem der Vektor gemessen wird.

Der „Tiefindex“ gibt an, welchen „Zielpunkt“ der Vektor hat.

Das allgemeine mathematische Grundproblem zur Beschreibung der Kinematik der Roboterachsen liegt darin, daß eine Lage (d.h. Position und Orientierung) in verschiedenen Koordinatensystemen beschrieben werden muß.

*Beispiel 1*

Es seien zwei Koordinatensysteme in der Ebene gegeben. Der Ursprungsvektor  ${}^0w_{0,1}$  des Koordinatensystems 1 (weiterhin als KS 1 abgekürzt) gemessen im KS 0 sei bekannt:  ${}^0w_{0,1}^T = (5;4)$ . Weiterhin sei die Lage des Punktes P im KS 1 bekannt:  ${}^1w_p^T = (2;4)$ . Gesucht ist die Beschreibung des Punktes P im KS 0, also der Vektor  ${}^0w_p$ .

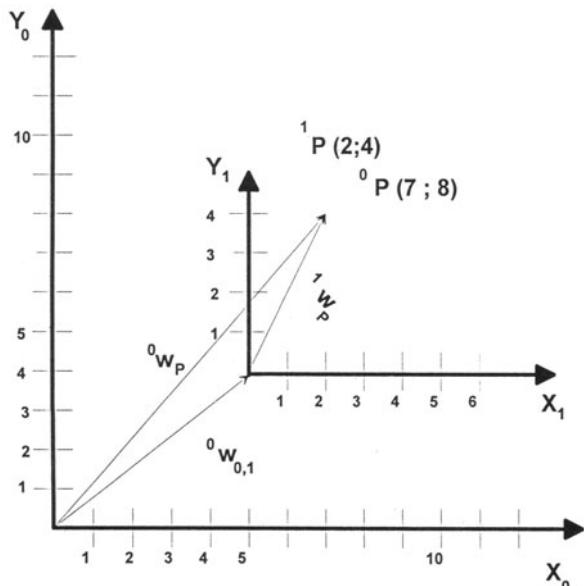
Es scheint sich folgende Beziehung durch einfache Vektoraddition zu ergeben:

$${}^0w_p = {}^0w_{0,1} + {}^1w_p \quad (1)$$

Berechnet man nach dieser Gleichung den Vektor  ${}^0w_p$ , so erhält man als Ergebnis:

$${}^0w_p^T = (7;8)$$

Vergleicht man dies mit Bild A-4, so ist dieses Ergebnis offensichtlich richtig.



**Bild A-4**  
Lage der Koordinatensysteme in  
Beispiel 1

Komplizierter wird es, wenn das KS 1 nicht mehr die gleiche Orientierung wie das KS 0 hat.

*Beispiel 2*

Es seien zwei Koordinatensysteme in der Ebene gegeben, die unterschiedliche Orientierung haben. Der Ursprungsvektor  ${}^0w_{0,1}$  des KS 1 gemessen im KS 0 sei bekannt:  ${}^0w_{0,1}^T = (5;4)$ . Weiterhin sei die Lage des Punktes P im KS 1 bekannt:  ${}^1w_p^T = (2;4)$ . Gesucht ist die Beschreibung des Punktes P im KS 0, also der Vektor  ${}^0w_p$ . Das KS 1 sei um  $45^\circ$  nach links um seinen Ursprung gedreht.

Berechnet man auch hier wieder nach Gl. (1) den Vektor  ${}^0w_p$ , so erhält man:

$${}^0w_p^T = (7;8)$$

Vergleicht man dieses Ergebnis mit Bild A-5, so stellt sich heraus, daß es falsch ist.

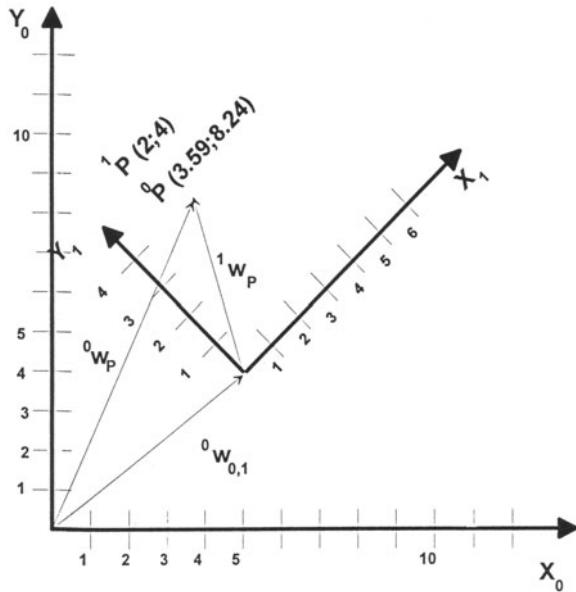


Bild A-5

Lage der Koordinatensysteme in Beispiel 2

Die Ursache liegt darin, daß zwei Vektoren addiert werden, die in unterschiedlichen KS liegen. Bevor man eine solche Addition durchführen kann, muß einer der Vektoren in das KS des anderen transformiert werden. Diese Basistransformation kann entfallen, wenn beide KS die gleiche Orientierung haben, also nur translatorisch verschoben sind.

Die allgemeine mathematische Beziehung für eine solche Koordinatentransformation, bei der sich auch die Orientierung der Koordinatensysteme ändert, lautet:

$${}^0 w_p = {}^0 w_{0,1} + {}^1 K * {}^1 w_p \quad (2)$$

Hierbei ist die Matrix  ${}^1 K$  die Matrix der Basiseinheitsvektoren des KS 1 beschrieben im KS 0, drückt also die Rotation des KS 1 beschrieben im KS 0 aus.

$${}^0 K = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix}$$

Das Beschreiben dieser Rotation kann unter Umständen sehr aufwendig sein, da die KS ja im Raum gedreht sein können. Allgemeine Rotationsbeschreibungsvarianten sind:

- Richtungskosinus
- Euler-Winkel
- Achs-Drehwinkel

Auf nähere Erläuterungen zu diesen Varianten sei hier verzichtet, da sie zur Beschreibung der Roboterkinematiken im Folgenden nicht benötigt werden.

Für Beispiel 1 ergibt sich die Rotationsmatrix zu:

$${}^0_1 K = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Anmerkung: Da es sich nur um ein ebenes Problem handelt, reduziert sich die 3\*3-Matrix zu einer 2\*2-Matrix

Man sieht, daß es sich hierbei um eine Einheitsmatrix handelt. Dies drückt aus, daß keine Rotation des KS 1 bzgl. des KS 0 stattgefunden hat. Berechnet man für Beispiel 1 nach Gl. (2) den gesuchten Vektor, erhält man das erwartete Ergebnis:

$${}^0 w_p^T = (7; 8)$$

Für das Beispiel 2 ergibt sich die Rotationsmatrix zu:

$${}^0_1 K = \begin{bmatrix} \frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} \\ \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} \end{bmatrix}$$

Wendet man nun Gl. (2) auf Beispiel 2 an, so erhält man in Übereinstimmung mit Bild A-5 als Ergebnis

$${}^0 w_p^T = (5 - \sqrt{2}; 4 + 3\sqrt{2}) \approx (3,59; 8,24).$$

Erweitert man das Problem dergestalt, daß man mehr als ein KS hinter das Basiskoordinatensystem schaltet, so verallgemeinert sich Gl. (2) zu:

$${}^0 w_p = {}^0 w_{0,1} + {}^1_1 K * {}^1 w_p \quad (3)$$

mit

$${}^1 w_p = {}^1 w_{0,2} + {}^2_2 K * {}^2 w_p$$

$${}^2 w_p = {}^2 w_{0,3} + {}^3_3 K * {}^3 w_p$$

...

und

$${}^n K = {}^0_1 K * {}^1_2 K * {}^2_3 K * {}^3_4 K * \dots * {}^{n-2}_{n-1} K * {}^{n-1}_n K$$

bzw.

$${}^0_n K = \prod_{k=0}^{n-1} {}^k_{k+1} K$$

(4)

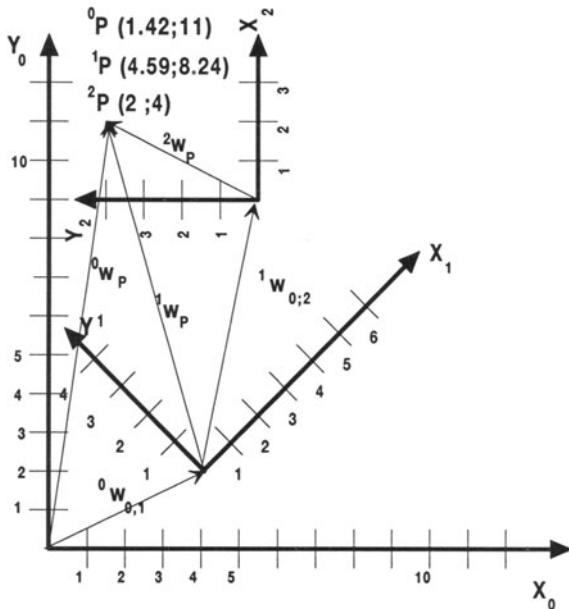
Hierbei beschreibt die Rotationsmatrix  ${}^0_n K$  die Gesamtrotation zwischen dem Basiskoordinatensystem und dem letzten, dem  $n$ -ten Koordinatensystem.

Diese Art der Darstellung der Koordinatentransformation führt zu Problemen, wenn man mehrere Koordinatensysteme hintereinander schaltet. Dann wird diese Art der Darstellung sehr umständlich, da sie mit erheblichem Rechenaufwand verbunden ist.

### Beispiel 3

Es seien drei KS in der Ebene gegeben, die unterschiedliche Orientierung haben. Der Ursprungsvektor  ${}^0 w_{0,1}$  des KS 1 gemessen im KS 0 sei bekannt:  ${}^0 w_{0,1}^T = (4; 2)$ . Auch die Lage des Ursprungvektors sei gegeben:  ${}^1 w_{0,2}^T = (6; 4)$ . Weiterhin sei die Lage des

Punktes P im KS 2 bekannt:  ${}^2 w_p^T = (2; 4)$ . Gesucht ist die Beschreibung des Punktes P im KS 0, also der Vektor  ${}^0 w_p$ . Das KS 1 sei um  $45^\circ$  nach links um seinen Ursprung und das KS 2 nochmals um  $45^\circ$  nach links um seinen Ursprung gedreht.



**Bild A-6**  
Lage der Koordinatensysteme in Beispiel 3

Nach den Gl. (3) und (4) ergibt sich:

$$\begin{aligned} {}^0 w_p &= {}^0 w_{0,1} + {}^1 K * {}^1 w_p \\ {}^0 w_p &= {}^0 w_{0,1} + {}^1 K * ({}^1 w_{0,2} + {}^2 K * {}^2 w_p) \\ {}^0 K &= {}^1 K * {}^2 K \end{aligned}$$

Für Beispiel 3 ergeben sich folgende Ergebnisse:

$$\begin{aligned} {}^0 K &= {}^1 K = \begin{bmatrix} \frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} \\ \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} \end{bmatrix} \\ {}^0 K &= {}^1 K * {}^2 K = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \end{aligned}$$

Man sieht an der Gesamtrotationsmatrix, daß eine Rotation um  $90^\circ$  durchgeführt wurde.

Für die Vektoren ergeben sich:

$$\begin{aligned} {}^0 w_{0,2}^T &= (4 + \sqrt{2}; 2 + 5\sqrt{2}) \approx (5,41; 9,07) \\ {}^1 w_p^T &= (6 - \sqrt{2}; 4 + 3\sqrt{2}) \approx (4,59; 8,24) \\ {}^0 w_p^T &= (\sqrt{2}; 4 + 5\sqrt{2}) \approx (1,41; 11,07) \end{aligned}$$

Diese Art der Berechnung ist sehr umständlich, vor allem, wenn noch mehr Koordinatensysteme hintereinander geschaltet werden.

An den drei Beispielen sieht man, daß sich jede Koordinatentransformation prinzipiell aus einem rotatorischen und einem translatorischen Teil zusammensetzt. Durch die Verwendung von *homogenen Koordinaten* werden die Berechnungen vereinfacht. Auf nähere Erläuterungen zu den homogenen Koordinaten sei hier verzichtet. Man erhält folgende Transformationsgesetzmäßigkeit:

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} A_1 & B_1 & C_1 & T_x \\ A_2 & B_2 & C_2 & T_y \\ A_3 & B_3 & C_3 & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} \quad (5)$$

Hierbei ist die obere 3\*3-Matrix der Rotationsanteil, und der vierte Spaltenvektor ergibt den Translationsanteil.

Es ergeben sich folgende allgemeine Transformationsgesetzmäßigkeiten für das Transformieren eines Ortsvektors  ${}^n w_p$ , der im  $n$ -ten Koordinatensystem gegeben ist und der als Ortsvektor  ${}^0 w_p$  im Basiskoordinatensystem gesucht ist.

$${}^0 w_p = {}^n T * {}^n w_p \quad (6)$$

und

$$\begin{aligned} {}^n T &= {}^1 T * {}^2 T * {}^3 T * \dots * {}^{n-2} T * {}^{n-1} T \\ \text{bzw. } {}^n T &= \prod_{k=0}^{n-1} {}^{k+1} T \end{aligned} \quad (7)$$

Wendet man die Gl. (6) und (7) auf die drei Beispiele an, so erhält man folgende Gleichungen und Lösungen:

*Beispiel 1*

$${}^0 w_p = {}^1 T * {}^n w_p$$

$$\text{mit } {}^1 T = \begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{bmatrix} \text{ ergibt sich:}$$

$${}^0 w_p = \begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{bmatrix} * \begin{pmatrix} 2 \\ 4 \\ 2 \end{pmatrix}$$

$${}^0 w_p = \begin{pmatrix} 7 \\ 8 \\ 1 \end{pmatrix} \text{ stimmt mit vorheriger Lösung überein.}$$

*Beispiel 2*

$${}^0w_p = {}_1^0T * {}^n w_p$$

mit  ${}^0_1T = \begin{bmatrix} \frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} & 5 \\ \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} & 4 \\ 0 & 0 & 1 \end{bmatrix}$  ergibt sich:

$${}^0w_p = \begin{bmatrix} \frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} & 5 \\ \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} & 4 \\ 0 & 0 & 1 \end{bmatrix} * \begin{pmatrix} 2 \\ 4 \\ 2 \end{pmatrix}$$

$${}^0w_p = \begin{pmatrix} 5-\sqrt{2} \\ 4+3\sqrt{2} \\ 1 \end{pmatrix} \text{ stimmt mit vorheriger Lösung überein.}$$

*Beispiel 3*

$${}^0w_p = {}_2^0T * {}^n w_p$$

mit  ${}^0_1T = \begin{bmatrix} \frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} & 4 \\ \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} & 2 \\ 0 & 0 & 1 \end{bmatrix}$  und  ${}_1^2T = \begin{bmatrix} \frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} & 6 \\ \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} & 4 \\ 0 & 0 & 1 \end{bmatrix}$  ergibt sich:

$${}^0_2T = \begin{bmatrix} 0 & -1 & 4+\sqrt{2} \\ 1 & 0 & 2+5\sqrt{2} \\ 0 & 0 & 1 \end{bmatrix}.$$

Daraus ergibt sich:

$${}^0w_p = \begin{bmatrix} 0 & -1 & 4+\sqrt{2} \\ 1 & 0 & 2+5\sqrt{2} \\ 0 & 0 & 1 \end{bmatrix} * \begin{pmatrix} 2 \\ 4 \\ 1 \end{pmatrix}$$

$${}^0w_p = \begin{pmatrix} \sqrt{2} \\ 4+5\sqrt{2} \\ 1 \end{pmatrix} \approx \begin{pmatrix} 1,42 \\ 11,07 \\ 1 \end{pmatrix} \text{ stimmt mit vorheriger Lösung überein.}$$

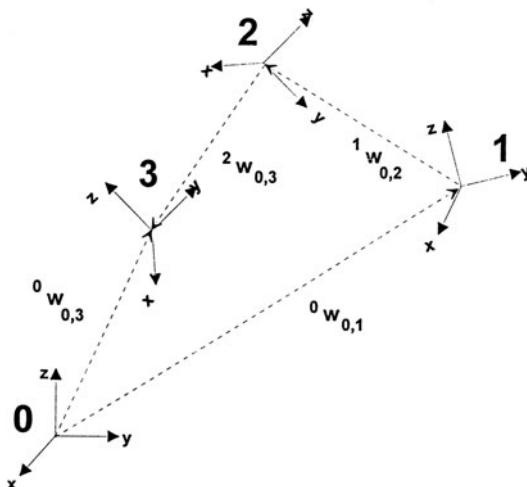
Man sieht bei allen Beispielen, daß sich die Berechnungen vereinfachen und vor allen Dingen wesentlich leichter als Algorithmus programmieren lassen.

Noch einfacher wird das Problem der Koordinatentransformation, wenn kein Ortsvektor im letzten Koordinatensystem gegeben ist, sondern lediglich der Ursprung des letzten Koordinatensystems nach Position und Orientierung gefragt ist. Dies sei am nächsten Beispiel verdeutlicht.

#### Beispiel 4

Gegeben seien ein Basiskoordinatensystem 0 und drei dahinter geschaltete KS 1 bis KS 3. Es sei jeweils die Translationsmatrix  ${}_{i+1}^iT$  bekannt, also  ${}_1^0T, {}_2^1T, {}_3^2T$ . Die KS sollen alle unterschiedliche Orientierungen haben. Gesucht sind Lage und Orientierung von KS 3 in Koordinaten von KS 0.

Bild A-7 erläutert dieses Beispiel.



**Bild A-7**  
Lage der Koordinatensysteme in Beispiel 4

Um Lage und Orientierung von KS 3 im Basiskoordinatensystem 0 ausdrücken zu können, braucht man lediglich die Matrix  ${}_3^0T$  nach Gl. (7) zu berechnen und erhält:

$${}_3^0T = {}_1^0T * {}_2^1T * {}_3^2T$$

Auf ein Zahlenbeispiel sei hier verzichtet. Betrachtet man in Beispiel 3 die  ${}_2^0T$ -Matrix,

$${}_2^0T = \begin{bmatrix} 0 & -1 & 4 + \sqrt{2} \\ 1 & 0 & 2 + 5\sqrt{2} \\ 0 & 0 & 1 \end{bmatrix},$$

so sieht man, daß sich im dritten Spaltenvektor die Gesamttranslation des Ursprungs des KS 2, der Vektor

$${}^0w_{0,2}^T = (4 + \sqrt{2}; 2 + 5\sqrt{2}) \approx (5,41; 9,07)$$

in homogenen Koordinaten abbildet. Ferner drückt die obere 2\*2-Matrix die Gesamtrotation des KS 2 beschrieben im KS 0 aus.

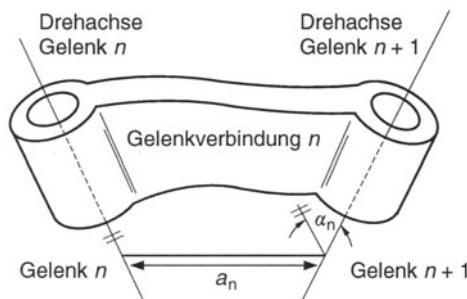
### ZUSAMMENFASSUNG

Mit Hilfe dieses Verfahrens ist man also in der Lage, die einzelnen Koordinatensysteme mathematisch so zu verknüpfen, daß man jedes einzelne in das Basiskoordinatensystem transformieren kann.

*Paul* hat, aufbauend auf den Überlegungen von *Denavit und Hartenberg*, ein Verfahren entwickelt, das diese Art der Koordinatentransformation mit homogenen Koordinaten benutzt. Ferner hat *Paul* Regeln aufgestellt, mit deren Hilfe es relativ einfach möglich ist, den Rotations- und den Translationsanteil in der Gesamttransformationmatrix zu beschreiben. Diese Vorgehensweise wird im nächsten Kapitel erläutert.

## A.3 Roboterachsenbeschreibung nach *Denavit und Hartenberg*

Die Vorgehensweise sei hier an einer allgemeinen Roboterachse dargestellt.



**Bild A-8**  
Aufbau einer allgemeinen Roboterachse

Eine solche Roboterachse verbindet im allgemeinen Fall zwei Achsgelenke (rotatorische bzw. lineare) miteinander. In Bild A-8 sind dies zwei nicht fluchtende Drehgelenke (Gelenk  $n$  und Gelenk  $n+1$ ). Die jeweiligen Drehachsen, um die die Achse drehen kann, sind entsprechend eingezeichnet.

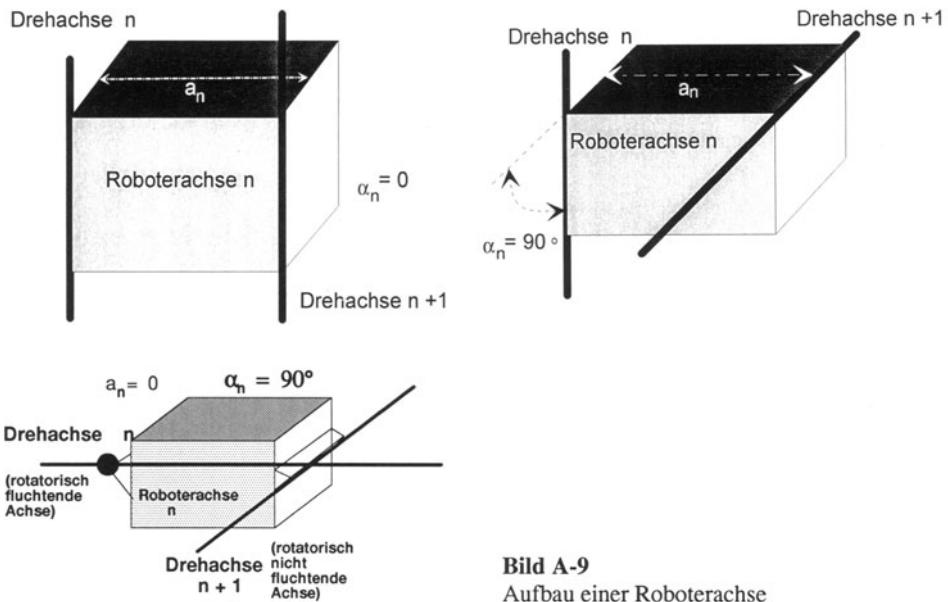
Für die Berechnungen spielt lediglich die Lage der beiden Gelenke zueinander eine Rolle. Diese Lage wird durch zwei Größen beschrieben:

- *Abstand*  $a_n$ 
  - im allgemeinen liegen die beiden Drehachsen  $n$  und  $n+1$  windschief im Raum. Dadurch ist der Abstand dieser Geraden über die Länge der gemeinsamen Normalen bestimmt
  - bei parallelen Geraden ist der Abstand in jedem Punkt gleich
  - bei sich schneidenden Geraden ist der Abstand Null
- *Winkel*  $\alpha_n$  zwischen den beiden Gelenkachsen, wenn eine Gelenkachse entlang der gemeinsamen Normalen verschoben wird, bis sie die andere schneidet

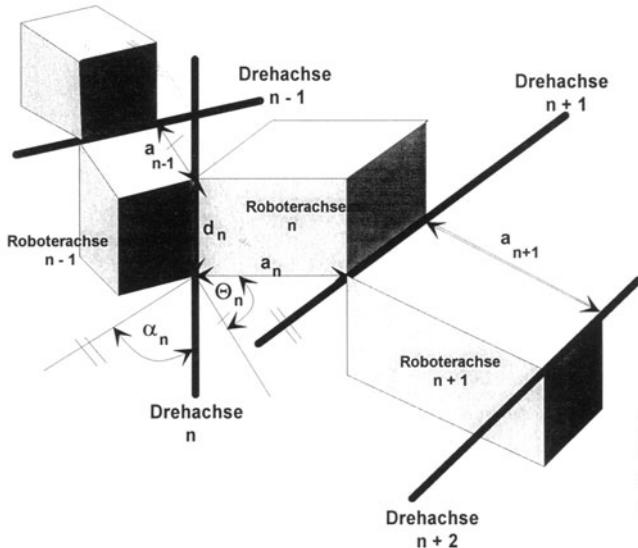
Man bezeichnet die beiden Größen  $a_n$  und  $\alpha_n$  auch als *Länge* und *Verdrillung* der Roboterachse.

Bei einfacheren Kinematiken – und nur solche seien hier weiterhin betrachtet – sind die beiden Drehachsen parallel zueinander bzw. um  $90^\circ$  gedreht oder schneiden sich.

In Bild A-9 sind alle Roboterachsen als rechtwinklige Quader idealisiert.



Ein Industrieroboter besteht nach der gültigen Definition aus mindestens drei Achsen. Will man nun die Lage zweier Roboterachsen zueinander beschreiben, benötigt man zwei weitere Parameter.



**Bild A-10**  
Lage zweier Roboterachsen (die Roboterachsen sind als rechtwinklige Quader idealisiert.)

Die Lage zweier Roboterachsen zueinander läßt sich durch die folgenden Größen beschreiben:

- Der Abstand der beiden gemeinsamen Normalen  $a_n$  und  $a_{n-1}$  auf der Drehachse  $n$  wird als  $d_n$  bezeichnet.
- Der Winkel, den diese beiden gemeinsamen Normalen bilden, erhält die Bezeichnung  $\theta_n$ .

Durch die vier Größen  $a_n$ ,  $\alpha_n$ ,  $d_n$  und  $\theta_n$ , die im Folgenden als DH-Parameter bezeichnet werden, kann die Lage zweier Roboterachsen zueinander eindeutig angegeben werden.

#### ZUSAMMENFASSUNG *DENAVIT-HARTENBERG-PARAMETER*

- Die Länge der gemeinsamen Normalen zwischen den beiden Gelenkachsen wird durch die Größe  $a_n$  beschrieben.
- Der Winkel  $\alpha_n$  ist als Winkel zwischen den beiden Gelenkachsen  $n$  und  $n+1$  definiert, wenn eine Gelenkachse entlang der gemeinsamen Normalen bis zum Schnittpunkt verschoben wird.
- Der Abstand  $d_n$  ist definiert als die Strecke zwischen den beiden gemeinsamen Normalen  $a_n$  und  $a_{n-1}$ , welche auf der Drehachse  $n$  liegt.
- Die Größe  $\theta_n$  beschreibt den zwischen den gemeinsamen Normalen  $a_n$  und  $a_{n-1}$  eingeschlossenen Winkel.

## A.4 Beschreibung der Roboterkinematik nach Paul

Wie schon erwähnt, muß bei einer kinematischen Kette die Lage (d.h. Position und Orientierung) eines jeden Kettengliedes bekannt sein, da Veränderungen eines Kettengliedes (linear oder rotatorisch) bekanntlich auch die Lage aller nachfolgenden Kettenglieder verändert. Gesucht ist ein Verfahren, das die einzelnen Roboterachsen in Koordinaten der direkten Vorgängerachse beschreiben kann. Die Vorgehensweise von *Denavit und Hartenberg* gibt keine Beschränkung zur Lage der Koordinatensysteme zueinander an, da die Ermittlung der DH-Parameter nicht an Koordinatensysteme gebunden ist. *Paul* hat eine handhabbare Form entwickelt, mit der es möglich ist, die Lage von Koordinatensystemen in Roboterachsen mit Hilfe der DH-Parameter zu ermitteln. Die allgemeine Vorgehensweise läßt sich überblickartig in drei Unterpunkte gliedern:

- jede Roboterachse erhält ein eigenes, körperfestes Koordinatensystem
- jede Roboterachse wird im Koordinatensystem des direkten Vorgängers beschrieben
- es werden homogene Koordinaten für die Beschreibung benutzt

Somit ist es möglich, die Lage der Roboterachsen durch Transformationsmatrizen zu beschreiben. Diese Transformationsmatrizen sind nur noch von den geometrischen Abmessungen der einzelnen Roboterachsen und den Drehgelenkstellungen abhängig.

Es ergibt sich nach Paul folgendes Schema zur Ermittlung des Koordinatensystems  $n$ , das die Roboterachse  $n$  (mit rotatorischen Drehgelenken) beschreibt:

- Der Schnittpunkt der Drehachse  $n+1$  mit der gemeinsamen Normalen  $a_n$  der Drehachsen  $n$  und  $n+1$  bildet den Ursprung des Koordinatensystems der Roboterachse  $n$ , wenn die beiden Drehachsen windschief sind.

Gibt es keine gemeinsame Normale – dies ist der Fall, wenn sich die beiden Drehachsen  $n$  und  $n+1$  schneiden –, so ist der Schnittpunkt der beiden Drehachsen  $n$  und  $n+1$  der Koordinatenursprung des Koordinatensystems der Roboterachse  $n$ .

Gibt es unendlich viele gemeinsame Normalen – dies ist der Fall, wenn beide Drehachsen  $n$  und  $n+1$  parallel sind –, so verschiebt man den Ursprung des Koordinatensystems der Roboterachse  $n$  so auf der Drehgelenkachse  $n+1$ , daß der DH-Parameter  $d_n$  zu Null wird.

- Die  $z$ -Achse des Koordinatensystems der Roboterachse  $n$  – als  $z_n$  bezeichnet – zeigt in Richtung der Drehachse des Drehgelenkes  $n+1$ .
  - Die Richtung der verlängerten gemeinsamen Normalen von Drehgelenk  $n$  zu Drehgelenk  $n+1$  ergibt die Richtung der  $x$ -Achse des Koordinatensystems der Roboterachse  $n$  – als  $x_n$  bezeichnet.
- Schneiden sich die beiden Drehachsen  $n$  und  $n+1$ , dann wird die Richtung der  $x$ -Achse durch das Kreuzprodukt der beiden Drehachsen  $n$  und  $n+1$  festgelegt.
- Durch die  $y_n$ -Achse wird das Koordinatensystem der Roboterachse  $n$  zu einem Rechtssystem.

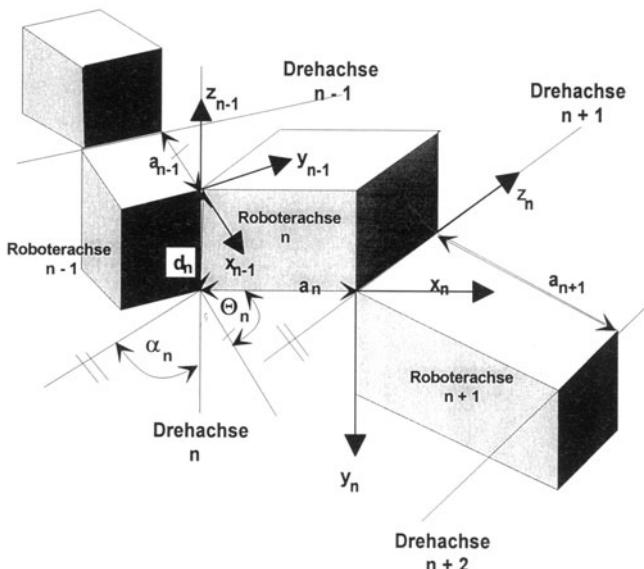


Bild A-11  
DH-Parameter und Koordinatensysteme nach Paul

Der weitaus größte Teil der im Moment auf dem Markt befindlichen Industrieroboter zeichnet sich durch die Spezialfälle

- parallele Gelenkachsen oder
- sich schneidende Gelenkachsen

aus. Die Regeln von Paul vereinfachen sich dadurch wie folgt:

- Für parallele Drehgelenke  $n$  und  $n+1$  gilt:
  - Der Koordinatenursprung des Koordinatensystems der Roboterachse  $n$  wird auf der Drehgelenkachse  $n+1$  so verschoben, daß der DH-Parameter  $d_n$  zur folgenden Roboterachse zu Null wird.
  - Die Drehachse des Drehgelenkes  $n+1$  wird zur  $z_n$ -Achse.
  - Die Richtung der gemeinsamen Normalen von Drehgelenk  $n$  und  $n+1$  ergibt die Richtung der  $x_n$ -Achse.
- Schneiden sich zwei aufeinander folgende Gelenkachsen  $n$  und  $n+1$ , so ergibt sich:
  - Der Schnittpunkt der beiden Drehgelenkachsen wird zum Ursprung des Koordinatensystems der Roboterachse  $n$ .
  - Die Gelenkachse des Drehgelenks  $n+1$  ergibt die  $z_n$ -Achse.
  - Das Kreuzprodukt der Drehachsen  $n$  und  $n+1$  ergibt die Richtung der  $x_n$ -Achse.
  - Die  $y_n$ -Achse ergänzt das Koordinatensystem zu einem mathematischen Rechtssystem.

Zur vollständigen Beschreibung fehlt noch das Basiskoordinatensystem, welches das Bezugssystem darstellt. Die Lage dieses Koordinatensystems muß auch den Regeln von Paul entsprechen, und somit ergibt sich:

- Der Ursprung des Basiskoordinatensystems wird so verschoben, daß der Parameter  $d_0$  zu Null wird.
- Die Gelenkachse 1 gibt die Richtung der  $z_0$ -Achse an.
- Die  $x_0$ -Achse paßt sich den nachfolgenden  $x$ -Achsen an.

Da zur Festlegung des Koordinatensystems der jeweiligen Roboterachse zwei Drehgelenke notwendig sind, ist das Koordinatensystem der letzten Achse nicht eindeutig definiert. Meist wird der Ursprung in die Mitte des Anschlußflansches für den Endeffektor gelegt, da sich dadurch die mathematischen Berechnungen für den Endeffektor vereinfachen.

Man sollte alle  $x$ -Achsen der Roboterachsenkoordinatensysteme in die gleiche Richtung zeigen lassen, da sich dadurch  $\theta_n = 0$  für alle Achsen ergibt. Man nennt diese Stellung auch Nullstellung, wobei sie nicht mit der Nullstellung des Wegmeßsystems der einzelnen Achsen zu verwechseln ist.

#### ZUSAMMENFASSUNG BESCHREIBUNG NACH PAUL

- Jede Roboterachse erhält ein eigenes, körperfestes Koordinatensystem.
- Jede Roboterachse wird im Koordinatensystem des direkten Vorgängers beschrieben.
- Für die Beschreibung werden homogene Koordinaten benutzt.
- Als Bezugsbasis wird ein System 0 vereinbart, auch Basiskoordinatensystem genannt, das keine Bewegungsmöglichkeit besitzt.
- Für parallele Drehgelenke  $n$  und  $n+1$  gilt:
  - Der Koordinatenursprung des Koordinatensystems der Roboterachse  $n$  wird auf der Drehgelenkachse  $n+1$  so verschoben, daß der DH-Parameter  $d_n$  zur folgenden Roboterachse zu Null wird.
  - Die Drehachse des Drehgelenkes  $n+1$  wird zur  $z_n$ -Achse.
  - Die Richtung der gemeinsamen Normalen von Drehgelenk  $n$  und  $n+1$  ergibt die Richtung der  $x_n$ -Achse.

- Schneiden sich zwei aufeinander folgende Gelenkkachsen  $n$  und  $n+1$ , so ergibt sich:
  - Der Schnittpunkt der beiden Drehgelenkkachsen wird zum Ursprung des Koordinatensystems der Roboterachse  $n$ .
  - Die Gelenkkurve des Drehgelenks  $n+1$  ergibt die  $z_n$ -Achse.
  - Das Kreuzprodukt der Drehachsen  $n$  und  $n+1$  ergibt die Richtung der  $x_n$ -Achse.
  - Die  $y_n$ -Achse ergänzt das Koordinatensystem zu einem mathematischen Rechtssystem.

Man kann nun zeigen, daß sich der Übergang vom Koordinatensystem  $n-1$  zum Koordinatensystem  $n$  durch folgende Transformationen beschreiben läßt:

- Rotation um die Achse  $z_{n-1}$  mit dem Winkel  $\theta_n$ ,  $\text{ROT}(z_{n-1}; \theta_n)$
- Translation in Richtung der  $z_{n-1}$ -Achse mit der Länge  $d_n$ ,  $\text{TRANS}(z_{n-1}; d_n)$
- Translation in Richtung der  $x_n$ -Achse mit der Länge  $a_n$ ,  $\text{TRANS}(x_n; a_n)$
- Rotation um die Achse  $x_n$  mit dem Winkel  $\alpha_n$ ,  $\text{ROT}(x_n; \alpha_n)$

Es ergibt sich eine Transformationsmatrix, die die Lage von Koordinatensystem  $n$  im Koordinatensystem  $n-1$  in homogenen Koordinaten beschreibt:

$${}^{n-1}_n T = \text{ROT}(z_{n-1}; \theta_n) * \text{TRANS}(z_{n-1}; d_n) * \text{TRANS}(x_n; a_n) * \text{ROT}(x_n; \alpha_n)$$

$${}^{n-1}_n T = \begin{bmatrix} \cos \theta_n & -\sin \theta_n & \cos \alpha_n & \sin \theta_n \sin \alpha_n & a_n \cos \theta_n \\ \sin \theta_n & \cos \theta_n & \cos \alpha_n & -\cos \theta_n \sin \alpha_n & a_n \sin \theta_n \\ 0 & 0 & \sin \alpha_n & \cos \alpha_n & d_n \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

Wie schon im letzten Kapitel erwähnt, beschreibt der obere Teil, die  $3 \times 3$ -Matrix, den Rotationsanteil, während der vierte Spaltenvektor den Translationsanteil der Transformation beschreibt.

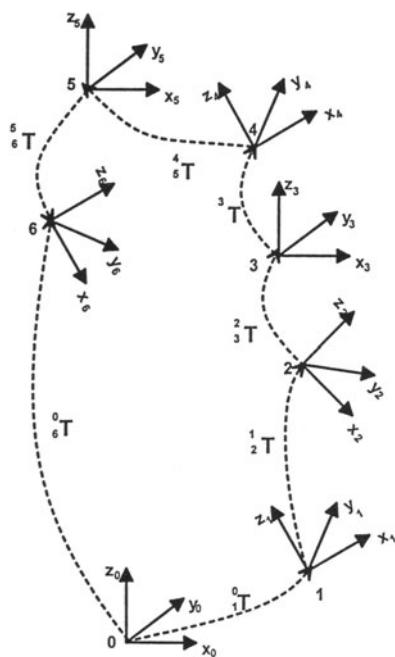
Durch konsequente Anwendung dieser Beschreibung auf die Roboterachsen eines Industrieroboters lassen sich so die Beziehungen zwischen beliebigen Achsen herleiten. Es ergibt sich allgemein für einen Roboter mit  $k$  Achsen:

$$\begin{aligned} {}^0_k T &= {}^0_1 T * {}^1_2 T * {}^2_3 T * \dots * {}^{k-1}_k T * {}^{k-1}_k T \\ \text{bzw. } {}^0_k T &= \prod_{i=0}^{k-1} {}^i_{i+1} T \end{aligned} \quad \text{vgl. Gl. (7)}$$

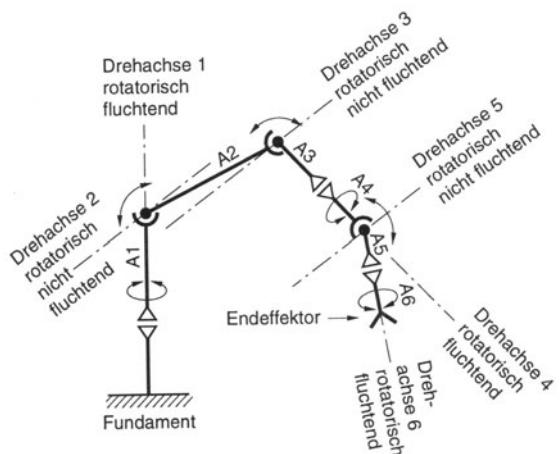
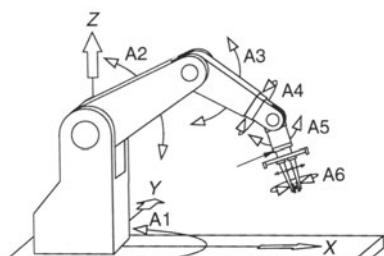
Somit ist es gelungen, mit Hilfe der Methode von *Paul* Transformationsmatrizen herzuleiten, die es ermöglichen, jedes Teilstück des Roboters (d.h. jede Achse) ins Basiskoordinatensystem zu transferieren. Es braucht also lediglich die Gesamttransformationsmatrix  ${}^0_k T$  berechnet zu werden, und man kann die Lage und die Orientierung des Endeffektors bestimmen.

Für einen sechsachsigen Roboter ergibt sich somit obige Gleichung zu:

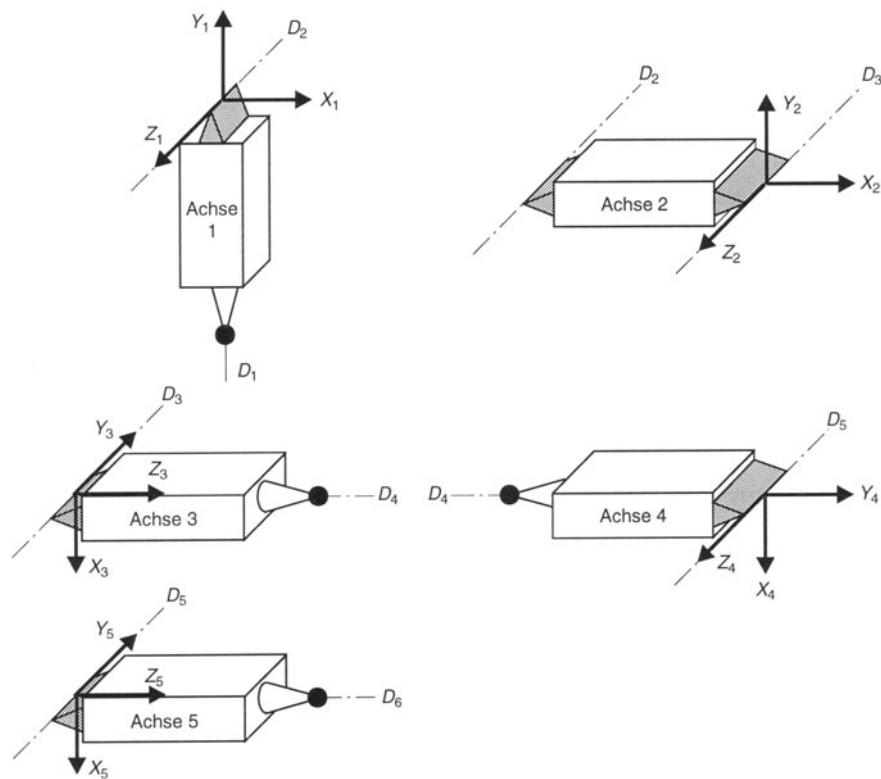
$${}^0_6 T = {}^0_1 T * {}^1_2 T * {}^2_3 T * {}^3_4 T * {}^4_5 T * {}^5_6 T$$



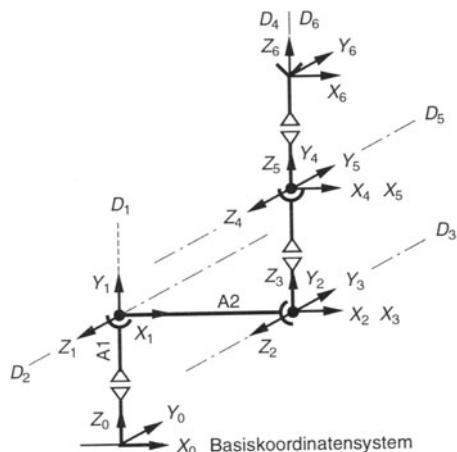
**Bild A-12**  
Transformationsmatrizen eines sechsachsigen Industrieroboters



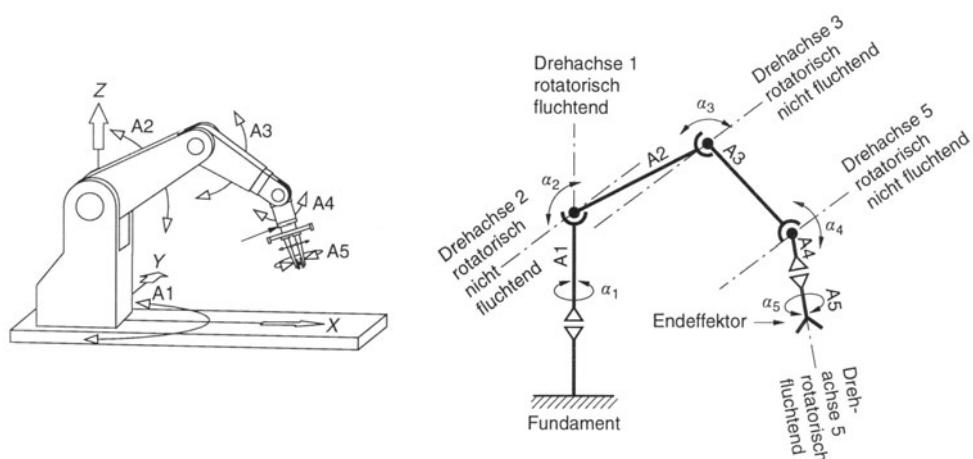
**Bild A-13** Realer sechsachsiger Knickarmroboter und kinematisches Ersatzschaltbild



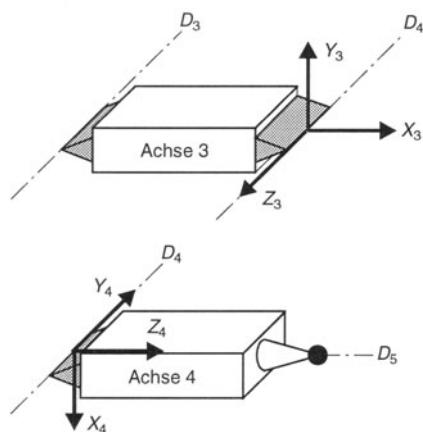
**Bild A-14** Körperfeste Koordinatensysteme der Achsen 1 bis 5 eines sechsachsigen Knickarmroboters nach Paul (die Achsen sind vereinfacht dargestellt)



**Bild A-15**  
Nullstellung der Koordinatensysteme eines sechsachsigen Knickarmroboters nach Paul



**Bild A-16** Realer fünfachsiger Knickarmroboter und kinematisches Ersatzschaltbild



**Bild A-17**

Körperfeste Koordinatensysteme der Achsen 3 und 4 eines fünfachsigen Knickarmroboters nach Paul (die Lage der körperfesten Koordinatensysteme der Achsen 1 und 2 ist wie in Bild A-14)

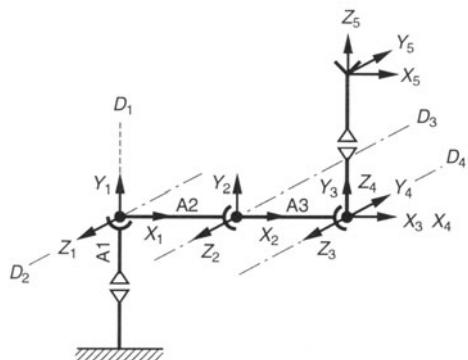
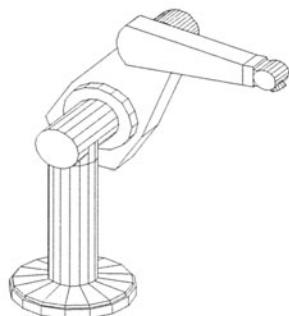
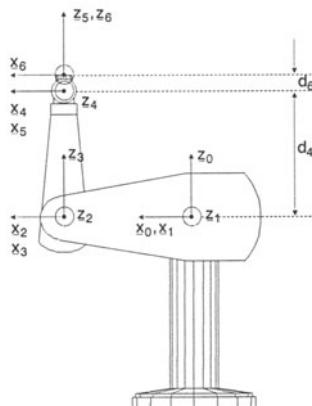


Bild A-18

Nullstellung der Koordinatensysteme eines fünfachsigen Knickarmroboters nach Paul



Graphisches Modell des  
Puma 560



i	$\Theta_i/^\circ$	$d_i/\text{mm}$	$a_i/\text{mm}$	$\alpha_i/^\circ$
1	$\Theta_1$	0.00	0.00	-90.00
2	$\Theta_2$	149.18	432.00	0.00
3	$\Theta_3$	0.00	0.00	90.00
4	$\Theta_4$	0.00	0.00	-90.00
5	$\Theta_5$	0.00	0.00	90.00
6	$\Theta_6$	0.00	0.00	0.00

DH-Parameter

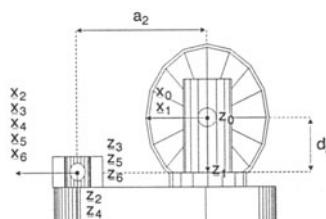
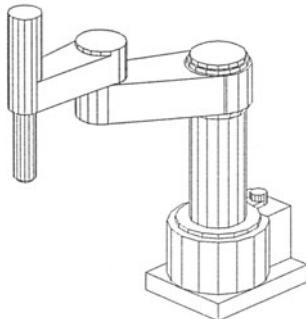
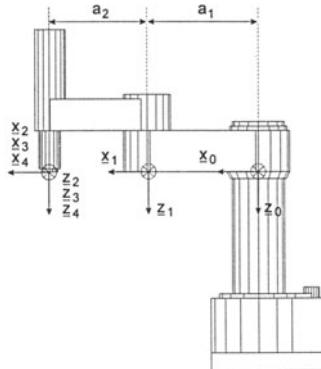


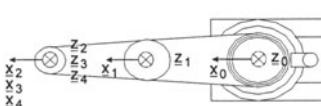
Bild A-19

Lage der Koordinatensysteme und DH-Parameter am PUMA 560 (Schwinn)

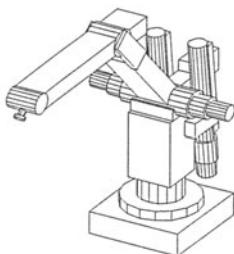
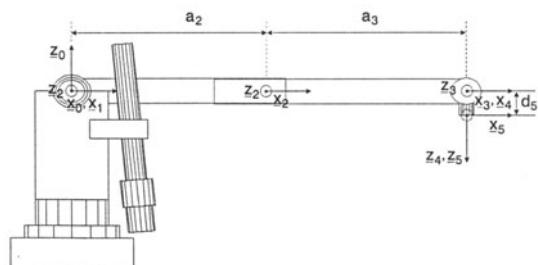
Graphisches Modell des  
AdeptOne

i	$\Theta_i/^\circ$	$d_i/\text{mm}$	$a_i/\text{mm}$	$a_i/^\circ$
1	$\Theta_1$	0.00	425.00	0.00
2	$\Theta_2$	0.00	375.00	0.00
3	0.00	$d_3$	0.00	0.00
4	$\Theta_4$	0.00	0.00	0.00

DH-Parameter

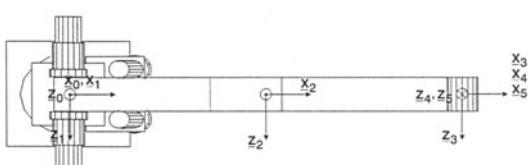
**Bild A-20**

Lage der Koordinatensysteme und DH-Parameter am ADEPTONE (Schwinn)

Graphisches Modell des  
Asea IRB L 6/2

i	$\Theta_i/^\circ$	$d_i/\text{mm}$	$a_i/\text{mm}$	$a_i/^\circ$
1	$\Theta_1$	0.00	0.00	90.00
2	$\Theta_2$	0.00	690.00	0.00
3	$\Theta_3$	0.00	670.00	0.00
4	$\Theta_4$	0.00	0.00	90.00
5	$\Theta_5$	95.00	0.00	0.00

DH-Parameter

**Bild A-21** Lage der Koordinatensysteme und DH-Parameter am ASEA IRB L 6/2 (Schwinn)

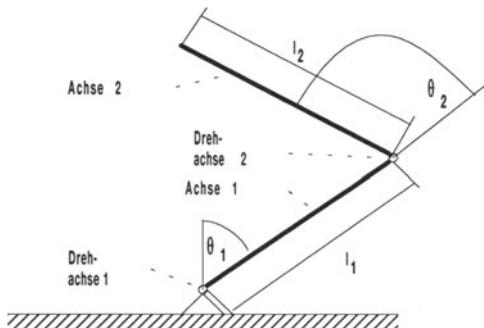
## A.5 Vorwärts- und Rückwärtstransformation beim Zweiarmanipulator

Hier soll am einfachen Beispiel des planaren (d.h. ebenen) Zweiarmanipulators die Vorgehensweise mit Hilfe der DH-Parameter zur Berechnung der Transformationsmatrizen aufgezeigt werden.

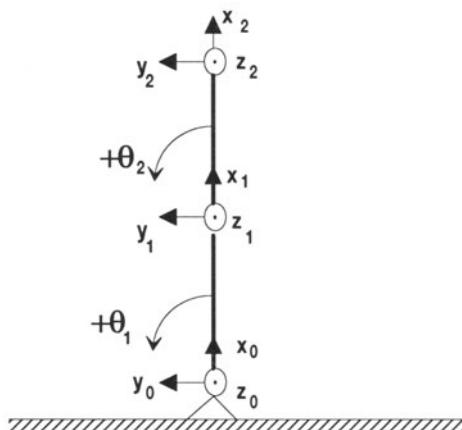
- Vorwärtstransformation: aus gegebenen Drehwinkeln der Drehachsen Lage und Orientierung des Endeffektors im xy-Basiskoordinatensystem berechnen; und
- Rückwärtstransformation: aus gegebener Lage und Orientierung des Endeffektors im xy-Basiskoordinatensystem die Drehwinkel der Drehachsen berechnen.

Beide Berechnung sind eine der wichtigsten Aufgaben einer jeden Robotersteuerung, die neben dem Gelenkkoordinatensystem noch andere Koordinatensysteme benutzt.

Nach der strengen Industrieroboterdefinition handelt es sich hier natürlich nicht um einen Roboter. Die prinzipielle Vorgehensweise bei der Vorwärt- und der Rückwärtstransformation kann aber relativ einfach aufgezeigt werden. Ferner erhält man Einblick in die prinzipiellen mathematischen Probleme, die bei der Berechnung entstehen und kann die Auswirkungen – hier speziell die Mehrdeutigkeiten – sehr schön demonstrieren.



**Bild A-22**  
Planarer Zweiarmanipulator



### ARBEITSBEREICHE

Achse 1:  $\pm 180^\circ$

Achse 2:  $\pm 180^\circ$

$i$	$\theta_i$	$d_i$	$a_i$	$\alpha_i$
1	$\theta_1$	0	$l_1$	0
2	$\theta_2$	0	$l_2$	0

**Bild A-23** Koordinatensysteme, DH-Parameter und Arbeitsbereich am Zweiarmanipulator

Nach Gl. (8) ergeben sich folgende Transformationsbeschreibungen der körperfesten Koordinatensysteme:

$${}^0_1 T = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & l_1 \cos \theta_1 \\ \sin \theta_1 & \cos \theta_1 & 0 & l_1 \sin \theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1_2 T = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & l_2 \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & l_2 \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Nach Gl. (7) ergibt sich die resultierende Transformationsmatrix durch Multiplikation der beiden Transformationsmatrizen und Anwendung der Additionstheoreme zu:

$${}^0_2 T = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) & 0 & l_2 \cos(\theta_1 + \theta_2) + l_1 \cos \theta_1 \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 & l_2 \sin(\theta_1 + \theta_2) + l_1 \sin \theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8a)$$

### A.5.1 Vorwärtstransformation

Wie bereits erwähnt, sollen hier aus den Gelenkvariablen  $\theta_1$  und  $\theta_2$  Position und Orientierung des Endeffektors berechnet werden. Es sind somit die Gelenkkoordinaten  $\theta_1$  und  $\theta_2$  bekannt. Diese sollen in das kartesische Basiskoordinatensystem 0 umgerechnet werden. (beachten Sie die Zählrichtung von  $\theta_i$  laut Bild A-23)

#### Beispiel 1

$\theta_1 = -90^\circ$  und  $\theta_2 = 90^\circ$  seien gegeben. Gesucht sind Lage und Orientierung des KS 2 beschrieben im KS 0. Gesucht ist also die Transformationsmatrix  ${}^0_2 T$ , in der die Gesamtrotation und die Gesamttranslation des KS 2 beschrieben in Basiskoordinaten abzulesen sind.

Vorab zur Verdeutlichung die Lage des Zweiarmanipulators in der Ebene für die beiden Winkelwerte.

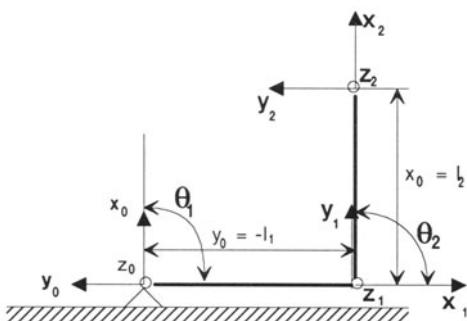


Bild A-24

Zweiarmanipulator für  $\theta_1 = -90^\circ$  und  $\theta_2 = 90^\circ$

Die resultierende Transformationsmatrix für diese beiden Winkelwerte ergibt sich nach Gl. (8a) zu:

$${}^0_2 T = {}^0_1 T * {}^1_2 T = \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & -l_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Man sieht hier sehr schön, daß die obere 3\*3-Matrix eine Einheitsmatrix ist, und kann somit interpretieren, daß keine Rotation des KS 2 stattfand. Den translatorischen Anteil kann man am vierten Spaltenvektor ablesen, und die Koordinaten des Ursprungs des KS 2 gemessen im Basiskoordinatensystem 0 ergeben sich zu:

$$x_0 = l_2 \text{ und } y_0 = -l_1.$$

Berechnet man zur Kontrolle auch noch die Transformationsmatrix

$${}^0_1 T = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & -l_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

so zeigt sich, daß der Ursprung des Koordinatensystems der Achse 1 bei

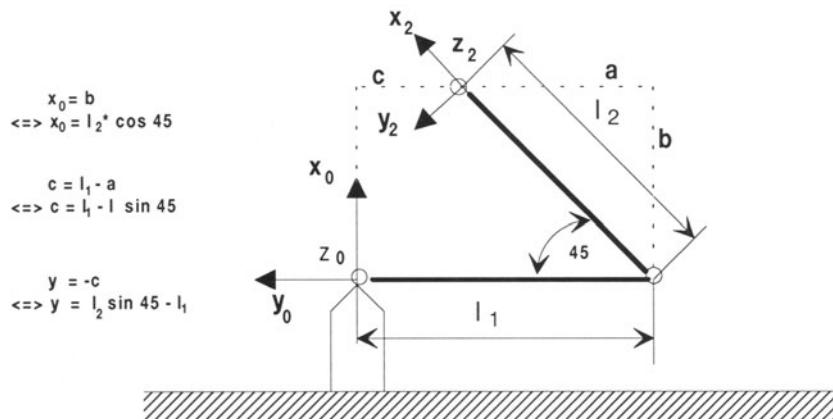
$$x_0 = 0 \text{ und } y_0 = -l_1$$

liegt; das Koordinatensystem 1 ist um  $-90^\circ$  zur Ausgangslage gedreht (vgl. Bild A-24).

In einem weiteren Beispiel soll gezeigt werden, daß auch Rotationen des KS 2 in der Transformationsmatrix abgebildet werden.

### Beispiel 2

$\theta_1 = -90^\circ$  und  $\theta_2 = 135^\circ$  seien gegeben. Gesucht sind Lage und Orientierung des KS 2 beschrieben im KS 0. Gesucht ist also die Transformationsmatrix  ${}^0_2 T$ , in der Gesamtrotation und Gesamttranslation des KS 2 beschrieben in Basiskoordinaten abzulesen sind.



**Bild A-25** Zweiarmsmanipulator für  $\theta_1 = -90^\circ$  und  $\theta_2 = 135^\circ$

Unter Anwendung der Gesetzmäßigkeit  $\sin 45^\circ = \cos 45^\circ = \frac{1}{2}\sqrt{2}$  ergibt sich die Transformationsmatrix für diese beiden Winkelwerte zu:

$${}^0_2T = \begin{bmatrix} \frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} & 0 & l_2\cos 45^\circ \\ \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} & 0 & -l_1 + l_2\sin 45^\circ \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Die obere  $2 \times 2$ -Matrix verdeutlicht die  $45^\circ$ -Rotation des KS 2 nach links ausgedrückt im KS 0. Der dritte Spaltenvektor gibt die Lage des Ursprungs gemessen im KS 0 an.

#### ZUSAMMENFASSUNG

Man sieht mit Hilfe der beiden Beispiele, daß durch Berechnung der Gesamttransformationsmatrix  ${}^0_nT$  sowohl die Gesamtrotation als auch die Gesamttranslation abgebildet wird. Die Berechnung dieser Matrix läßt sich sehr gut in eine Programmiersprache übertragen, was einen großen Vorteil gegenüber anderen Methoden darstellt.

### A.5.2 Rückwärtstransformation

Bei diesem Rechenweg muß aus der Angabe von Position und Orientierung des KS 2 im Basiskoordinatensystem 0 auf die Gelenkwinkel  $\theta_i$  geschlossen werden.

#### Beispiel 3

*Der Einfachheit halber werde Beispiel 2 „rückwärts“ gerechnet. Es ist somit die Gesamttransformationsmatrix mit folgenden Werten gegeben:*

$${}^0_2T = \begin{bmatrix} \frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} & 0 & l_2\cos 45^\circ \\ \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} & 0 & -l_1 + l_2\sin 45^\circ \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

*Mit Hilfe dieser Matrix müssen dann die Winkelwerte  $\theta_i$  berechnet werden.*

Zur Lösung des Problems betrachtet man sich die allgemeine Transformationsmatrix  ${}^0_2T$

$${}^0_2T = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) & 0 & l_2\cos(\theta_1 + \theta_2) + l_1\cos\theta_1 \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 & l_2\sin(\theta_1 + \theta_2) + l_1\sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (8a)$$

Man sieht, daß man die Gesamttransformationsmatrix vereinfacht schreiben kann als

$${}^0_2T = \begin{bmatrix} a & -b & 0 & t_x \\ b & a & 0 & t_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Setzt man beide Matrizen gleich, erhält man folgende Gleichungen:

$$a = \cos(\theta_1 + \theta_2) \quad (9)$$

$$b = \sin(\theta_1 + \theta_2) \quad (10)$$

$$t_x = l_2 \cos(\theta_1 + \theta_2) + l_1 \cos \theta_1 \quad (11)$$

$$t_y = l_2 \sin(\theta_1 + \theta_2) + l_1 \sin \theta_1 \quad (12)$$

Die Werte von  $a$ ,  $b$ ,  $t_x$ ,  $t_y$ ,  $l_1$  und  $l_2$  sind vorgegeben, gesucht sind die beiden Winkelwerte für die Gelenkstellungen. Setzt man Gl. (10) in Gl. (12) ein, erhält man:

$$\begin{aligned} t_y &= l_2 b + l_1 \sin \theta_1 \\ \Rightarrow \sin \theta_1 &= \frac{t_y - l_2 b}{l_1} \\ \Rightarrow \theta_1 &= \arcsin \frac{t_y - l_2 b}{l_1} \end{aligned} \quad (13)$$

Mit Gl. (9) ergibt sich:

$$\begin{aligned} a &= \cos(\theta_1 + \theta_2) \\ \arccos a &= \theta_1 + \theta_2 \\ \theta_1 &= \arccos a - \theta_2 \end{aligned} \quad (14)$$

Setzt man die gegebenen Zahlenwerte für  $a$ ,  $b$ ,  $t_x$ ,  $t_y$ ,  $l_1$  und  $l_2$  ein und berechnet das Ergebnis „mit einem Taschenrechner“, so erhält man als Ergebnis

$$\theta_1 = -90^\circ \text{ und } \theta_2 = 135^\circ.$$

Damit scheint auf den ersten Blick das gewünschte Ergebnis erzielt und die Rückwärtstransformation gelöst. Dem ist jedoch nicht so.

Nimmt man an, daß beide Achsen des Zweiarmanipulators links oder rechts herum drehen können, d.h. negative und positive Winkelwerte annehmen können, so können diese beiden bislang errechneten Winkel nicht die einzige Lösung sein. Das „Problem“ liegt beim Taschenrechner und an der Eindeutigkeit der Cosinus- und Sinus-Funktion. Die Arcus-Cosinus-Funktion z.B. bildet „im Taschenrechner“ nur Winkelwerte im Bereich  $0^\circ \leq \alpha \leq 180^\circ$  ab. Es existiert aber für  $\theta_1$  nicht nur die Lösung  $\theta_1 = -90^\circ$  sondern auch  $\theta_1 = -270^\circ$ . Ähnlich bildet die Arcus-Sinus-Funktion „im Taschenrechner“ nur für Winkelwerte im Bereich  $-90^\circ \leq \alpha \leq 90^\circ$  ab. Auch hier existiert neben der Lösung  $\theta_2 = 135^\circ$  noch die Lösung  $\theta_2 = -225^\circ$ . Dies führt somit zu insgesamt vier Lösungsmöglichkeiten der Achsstellungen bei der Rückwärtstransformation des Zweiarmanipulators:

$$L1 = (-90^\circ ; 135^\circ)$$

$$L2 = (-90^\circ ; -225^\circ)$$

$$L3 = (270^\circ ; 135^\circ)$$

$$L4 = (270^\circ ; -225^\circ)$$

Aus diesen vier Lösungsmöglichkeiten muß die Steuerung nun diejenige berechnen, die vom Roboter überhaupt angefahren werden kann; d.h. diejenige Stellung, die innerhalb des vorgegebenen Arbeitsbereiches liegt. In diesem Beispiel ist dies nur L1.

**ZUSAMMENFASSUNG**

- Es zeigt sich, daß bei der Rückwärtstransformation immer Mehrdeutigkeiten auftreten, da mit transzendenten Funktionen gearbeitet werden muß und diese nicht eineindeutig sind. Dies führt dazu, daß die Steuerung aus einer Vielzahl von Lösungsmöglichkeiten die richtige auswählen muß. Die Anzahl der Lösungsmöglichkeiten und somit auch der Rechenaufwand steigen mit der Anzahl der Achsen.
- Ferner ist es bei mehrachsigen Robotern nicht mehr so einfach wie beim Zweiarmanipulator möglich, durch einfaches Gleichsetzen von Matrizen die Lösungen herzuleiten. Dazu sind wesentlich kompliziertere Lösungsalgorithmen notwendig, auf die hier nicht eingegangen werden kann. Es sei auf die Literatur (z.B. SCHWINN, Grundlagen der Roboterkinematik) verwiesen.

## B Bahnberechnungen

### B.1 Grundlagen

Das prinzipielle Vorgehen bei der Roboterprogrammierung ist immer gleich. Der Anwender teilt die relevanten Punkte, die zur Durchführung der Handhabungsaufgabe benötigt werden. Die Steuerung speichert diese Punkte ab, meistens in Gelenkkoordinaten. Dann wird im eigentlichen Roboterprogramm vom Programmierer festgelegt, in welcher Reihenfolge die Punkte abzufahren sind. Ferner wird bestimmt, wie die Bahn des TCP; d.h. das Bewegungsmuster der Achsen, zwischen den Punkten aussieht. Hierzu stehen prinzipiell folgende Möglichkeiten zur Verfügung:

- PTP-Steuerung (asynchron und synchron)
- Multipunkt-Steuerung
- Bahnsteuerung (Gerade, Kreis etc.)

Wie erwähnt, braucht die Steuerung dazu einen sog. Interpolator, der die Aufgabe hat, den Stellweg (Winkel bei rotatorischen Achsen und Wege bei linearen Achsen) und die Achsgeschwindigkeit für jede Achse für die auszuführende Bewegung zu berechnen. Art und Umfang der Berechnungen unterscheiden sich je nach Steuerungsart, die verwirklicht werden soll.

Bei der PTP-Steuerung interessieren nur Anfangs- und Endpunkt der Bewegung. Den Anwender interessiert es nicht, auf welchem Weg die Roboterachsen das Ziel erreichen. Der Interpolator braucht also keine mathematische Bahn zwischen den Anfangs- und den Endpunkt zu legen. Er kann direkt die Differenz der Gelenkkoordinaten der einzelnen Achsen, die zwischen Anfangs- und Endpunkt liegen, berechnen und mit diesen Werten dann den Bewegungsablauf berechnen. Je nachdem, ob die Achsen zur gleichen Zeit zum Stillstand kommen oder nicht, unterscheidet man zwischen synchroner und asynchroner PTP-Steuerung. Bei synchroner PTP-Steuerung ist neben den zu verfahrenden Gelenkwinkeln auch noch ein Geschwindigkeitsprofil für jede Achse zu berechnen. Bei der asynchronen PTP-Steuerung entfällt dies, da jede Achse versucht, sich mit maximaler Achsgeschwindigkeit zu bewegen; die Achse, die den kürzesten Weg zurückzulegen hat, beendet ihre Bewegung zuerst.

Bei der Multipunkt-Steuerung – die auch als Quasi-Bahnsteuerung bezeichnet wird – ist der Anwender daran interessiert, daß die Bewegung zwischen Anfangs- und Endpunkt auf einer bestimmten Bahn erfolgt. Man unterscheidet zwei Arten der Multipunkt-Steuerung (vgl. Kapitel 7.3.3):

- Beim Abfahren einer beliebigen Raumkurve muß der Bediener alle Zwischenpunkte gelehrt haben. Die Bewegung zwischen den einzelnen Punkten ist eine PTP-Bewegung und wird genauso berechnet wie diese. Diese Art der Bewegung stellt somit mehr eine softwaremäßige Erleichterung für den Bediener dar, damit er sich Programmieraufwand sparen kann; bei 100 Zwischenpunkten, die alle angefahren werden müssen, ist die Zeitsparnis beim Programmieren erheblich.

- Beim Linearisieren einer Bahn muß vom Interpolator eine Gerade im Raum berechnet werden, die zwischen Anfangs- und Endpunkt liegt. Auf dieser Geraden werden dann Zwischenpunkte berechnet – die Anzahl wird entweder vom Bediener vorgegeben oder ist im Interpolator implementiert. Diese berechneten Zwischenpunkte werden dann wieder in PTP-Fahrt angefahren, wobei auch hier wieder das „PTP-Berechnungs-Verfahren“ angewendet wird.

Ganz anders verhält es sich bei der Bahnsteuerung. Hier muß sich der Roboter auf einer genau definierten mathematischen Kurve – meistens Gerade oder Kreis – zwischen Anfangs- und Endpunkt bewegen. Die einzelnen Achsen müssen also in Lage und Geschwindigkeit so abgestimmt werden, daß diese Bahn abgefahren werden kann. Dies ist mit erheblichem Rechenaufwand verbunden.

Der Rechenaufwand hängt ferner auch noch von der Anzahl der bewegten Roboterachsen ab. Wie in Anhang A aufgezeigt, muß bei Robotern mit mehr als drei Achsen die Transformationsmatrix bestimmt werden. Dies und die Lösungsermittlung für die Vorwärts- und Rückwärtstransformation führen zu aufwendigen Berechnungen.

Bei fast allen Roboterherstellern wird als Basiskoordinatensystem ein feststehendes kartesisches Koordinatensystem verwendet. Dies bedeutet, daß sämtliche Gelenkkoordinaten immer in kartesische Koordinaten umgerechnet werden müssen. Das kartesische Grundsystem hat den Vorteil, daß die Bedienbarkeit gesteigert wird, da man davon ausgehen kann, daß dieses Koordinatensystem vielen bekannt ist.

## B.2 Industrieroboter mit maximal drei Achsen

Auch hier lassen sich prinzipiell die drei oben genannten Steuerungsarten (PTP etc.) verwirklichen.

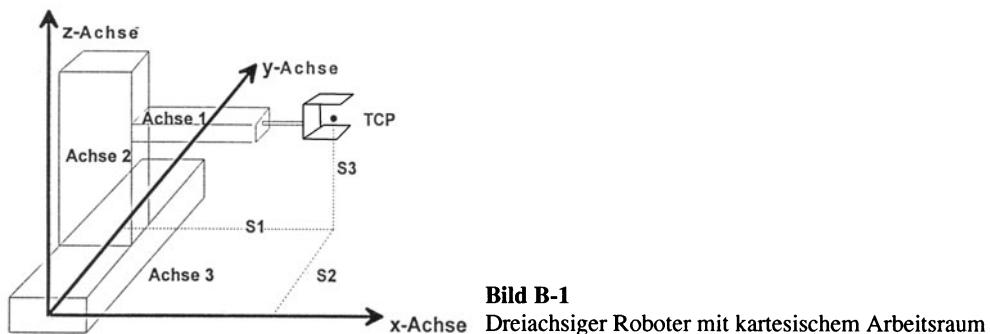
Bei den dreiachsigem Industrierobotern braucht man nicht so aufwendige Lösungsstrategien zu ermitteln, da sie mit ihren drei Achsen immer eine feste Orientierung haben und lediglich die Bewegung des TCP im Raum interessiert. Dadurch bekommt man einfache Transformationsgesetzmäßigkeiten. Den kinematischen Aufbau dieser Roboter gestaltet man so, daß sie einem der drei Koordinatensysteme

- kartesisches Koordinatensystem,
- Zylinderkoordinatensystem oder
- Kugelkoordinatensystem

entsprechen. Man hat allerdings auch hier bei der Rückwärtstransformation von Zylinder- bzw. Kugelkoordinaten in kartesische Koordinaten mit Eindeutigkeitsproblemen zu kämpfen (vgl. Kapitel 7.2.1). Allerdings ist ein Interpolator für solche Anwendungen wesentlich einfacher zu entwickeln als für mehrachsige Industrieroboter.

### B.2.1 Dreiachsiger Roboter mit kartesischem Arbeitsraum

Dieser Roboter soll aus drei rechtwinklig im Raum zu einander angeordneten Schubgelenken (d.h. Linearachsen) bestehen.



Die Stellung des Endeffektors ist gegeben durch die drei Achspositionen (Achse 1, Achse 2, Achse 3), also durch  $(s_1, s_2, s_3)$ . Die Steuerung braucht lediglich die eingegebenen Werte  $s_1, s_2$  und  $s_3$  – die Verfahrwege der Linearachsen – an die Lageregelung weiterzugeben und kann damit den Endeffektor zum gewünschten Zielpunkt bewegen.

Die Umrechnung der Punkte im Raum ins kartesische Koordinatensystem, das hier als Basiskoordinatensystem eingezeichnet ist, ist trivial. Es gilt:

$$P(x, y, z) \text{ mit}$$

$$x = s_1$$

$$y = s_2$$

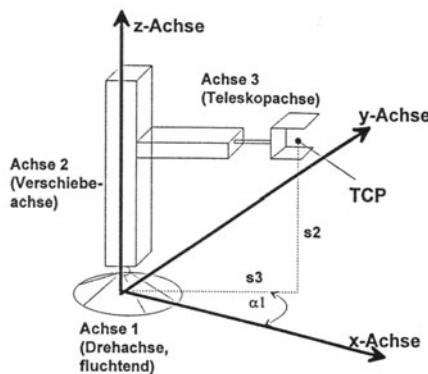
$$z = s_3$$

Bei dieser Anordnung der Roboterachsen sind die Gelenkkoordinaten gleich den Koordinaten im xyz-Basiskoordinatensystem; das Gelenkkoordinatensystem entspricht somit dem Basiskoordinatensystem (auch Weltkoordinatensystem genannt). Von daher treten auch bei der Koordinatentransformation keine Probleme auf.

### B.2.2 Dreiachsiger Roboter mit zylindrischem Arbeitsraum

Hierbei handelt es sich um einen Roboter mit einem Drehgelenk (Achse 1) und zwei translatorischen Achsen (Achse 2 und Achse 3). Die drei Achsen sind folgendermaßen angeordnet:

Achse 1 dreht um die in der Skizze eingezeichnete z-Achse, während Achse 2 Achse 3 in z-Richtung auf- bzw. abwärts bewegen kann. Achse 3 selbst kann als Schubachse linear ein- bzw. ausfahren.



**Bild B-2**  
Dreiachsiger Roboter mit zylindrischem Arbeitsraum

Auch hier gilt, wie schon in Kapitel 7.2.2 erwähnt: Wenn man die Lage des TCP im Raum angeben will, hat man zwei Möglichkeiten:

- Angabe der jeweiligen Achsstellung (Gelenkkoordinaten)  
Die Koordinatenbeschreibung des TCP in Bezug auf das Gelenkkoordinatensystem ergibt sich zu

$$P(\text{Achse } 1, \text{Achse } 2, \text{Achse } 3) \text{ mit}$$

$$\text{Achse } 1 = \alpha_1$$

$$\text{Achse } 2 = s_2$$

$$\text{Achse } 3 = s_3.$$

Bei dieser Angabe treten keine Probleme mit Mehrdeutigkeiten auf.

- Angabe mit dem Bezugssystem Basiskoordinaten  $xyz$   
Die Koordinatenangabe des TCP erfolgt bezogen auf das Basiskoordinatensystem  $xyz$ :

$$P(x,y,z)$$

Will man von Achskoordinaten in das Basissystem  $xyz$  umrechnen, so ergeben sich dabei keine Probleme, und die Transformationsgesetze für den Punkt  $P(x,y,z)$  lauten:

$$x = s_3 \cos \alpha_1$$

$$y = s_3 \sin \alpha_1$$

$$z = s_2$$

Will man hingegen vom Basissystem  $xyz$  in Achskoordinaten umrechnen, so ergeben sich die beschriebenen Transformationsprobleme. Die Transformationsgleichungen lauten:

$$\alpha_1 = \arctan y/x$$

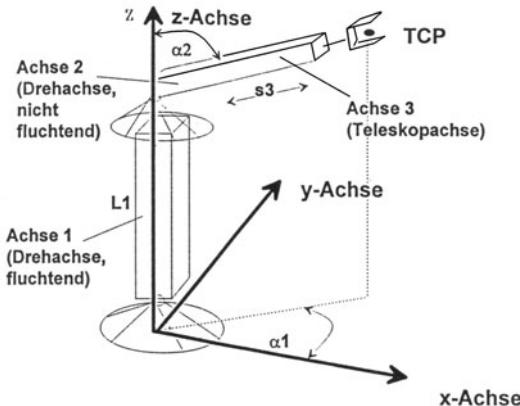
$$s_2 = z$$

$$s_3 = \sqrt{x^2 + y^2}$$

### B.2.3 Dreiachsiger Roboter mit kugelförmigem Arbeitsraum

Dieser Roboter besteht aus zwei rotatorischen Achsen (Achse 1 und Achse 2) und einer translatorischen Achse (Achse 3).

Achse 1 dreht um die in der untenstehenden Abbildung eingezeichnete  $z$ -Achse. Achse 3 sitzt mit einem weiteren Drehgelenk an Achse 1. Dieses Drehgelenk stellt Achse 2 dar. Das Drehgelenk der Achse 2 dreht nicht wie Achse 1 um die  $z$ -Achse, sondern senkrecht dazu. Achse 3 ist eine Schubachse und kann somit linear ein- und ausfahren.



**Bild B-3**  
Dreiachsiger Roboter mit kugelförmigem  
Arbeitsraum

Auch hier ist die Stellung des Endeffektors eindeutig durch die Stellung der Achspositionen (Achse 1, Achse 2, Achse 3) gegeben. Dabei ist die Lage des TCP im Raum durch die Angabe

$P$  (Achse 1, Achse 2, Achse 3) mit

$$\text{Achse } 1 = \alpha_1$$

$$\text{Achse } 2 = \alpha_2$$

$$\text{Achse } 3 = s_3.$$

eindeutig beschrieben. Auch hier ergeben sich keine Mehrdeutigkeiten, was die Stellung des Roboters angeht. Die Steuerung braucht lediglich die eingegebenen Werte  $\alpha_1$ ,  $\alpha_2$  und  $s_3$  an die Lageregelung weiterzugeben und kann damit den Endeffektor zum gewünschten Zielpunkt bewegen.

Ähnlich wie bei dem dreiachsigen Roboter mit zylindrischem Arbeitsraum hat man, neben den Gelenkkordinaten, eine weitere Möglichkeit, die Lage des TCP im Raum anzugeben: sie kann auch im Basiskoordinatensystem  $xyz$  beschrieben werden:

$P$  ( $x,y,z$ ) mit

$$x = s_3 \sin \alpha_2 \cos \alpha_1$$

$$y = s_3 \sin \alpha_2 \sin \alpha_1$$

$$z = s_3 \cos \alpha_2 + L_1$$

( $L_1$  = Abstand zwischen Drehgelenk Achse 1 und Drehgelenk Achse 2)

Diese Transformationsgesetze gelten, wenn man von gegebenen Achskoordinaten in Basiskoordinaten umrechnen will. Hierbei ergeben sich keine Probleme mit der Eindeutigkeit der Winkelfunktionen.

Anders verhält es sich beim Umrechnen von Basiskoordinaten in Gelenkkoordinaten. Hier hat man wieder die Probleme der Eindeutigkeit. Es gelten die Transformationsgleichungen:

$$\alpha_1 = \arctan y/x$$

$$\alpha_2 = \arccos ((z - L_1) / s_3)$$

$$s_3 = \sqrt{x^2 + y^2} + (z - L_1)^2$$

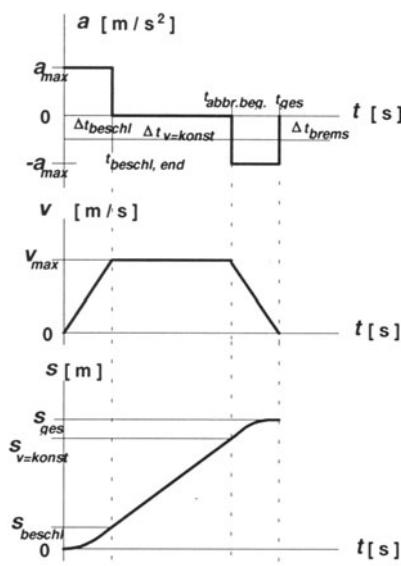
## B.3 Punktsteuerungen für mehr als drei Achsen

### B.3.1 Interpolation für eine einzelne Achse

Das prinzipielle Vorgehen bei der Berechnung der Bahnpараметер soll zunächst der Einfachheit halber an einer einzelnen Achse dargestellt werden. Es sollen ferner folgende grundlegende Bedingungen gelten:

- für die Beschleunigungs- und Bremsphase der Achse gilt  
 $a = \text{konst}$
- der Betrag der Beschleunigung soll gleich dem Betrag der Verzögerung sein  
 $a_{\text{beschl}} = -a_{\text{brems}}$

Mit Hilfe der kinematischen Grundgleichung für die gleichförmig beschleunigte Bewegung ( $a = \text{konst}$ ) und für die gleichförmige Bewegung ( $a = 0$  und  $v = \text{konst}$ ) ergeben sich die nachfolgende Beziehungen.



**Bild B-4**

$a$ - $t$ -,  $v$ - $t$ - und  $s$ - $t$ -Diagramm für eine Bewegung einer Roboterachse mit Erreichen der maximalen Achsgeschwindigkeit  $v_{\text{max}}$

Aus Bild B-4 ergeben sich anschaulich die folgenden Beziehungen für die Zeiten:

$$t_0 = t_{beschl,beginn} \quad (1)$$

$$t_{beschl,ende} = t_{vkonst,beginn} \quad (2)$$

$$t_{vkonst,ende} = t_{brems,beginn} \quad (3)$$

$$t_{beschl,ende} = t_{ges} \quad (4)$$

Somit ergibt sich

$$\begin{aligned} \Delta t_{beschl} &= t_{beschl,ende} - t_{beschl,beginn} \\ &= t_{beschl,ende} - t_0 \end{aligned} \quad (5)$$

(Zeit, in der aus der Ruhe bis  $v_{max}$  beschleunigt wird; d.h.  $a > 0$ )

$$\Delta t_{vkonst} = t_{vkonst,ende} - t_{vkonst,beginn} \quad (6)$$

(Zeit, in der mit  $v_{max} = \text{konst}$  gefahren wird; d.h.  $a = 0$ )

$$\begin{aligned} \Delta t_{brems} &= t_{brems,ende} - t_{brems,beginn} \\ &= t_{ges} - t_{brems,beginn} \end{aligned} \quad (7)$$

(Zeit, in der bis zum Stillstand verzögert wird; d.h.  $a < 0$ )

Ferner gilt

$$\Delta t_{brems} = \Delta t_{beschl} \quad (8)$$

(Bremszeit = Beschleunigungszeit, da  $a_{beschl} = -a_{brems}$ )

Somit ergibt sich für die Gesamtzeit  $t_{ges}$  des Vorganges folgende Gleichung:

$$\begin{aligned} t_{ges} &= \Delta t_{beschl} + \Delta t_{vkonst} + \Delta t_{brems} \\ t_{ges} &= 2 \Delta t_{beschl} + \Delta t_{vkonst} \end{aligned} \quad (9)$$

Für die Beschleunigung ergibt sich:

$$a(t) = a_{max} \quad \text{für } t_0 = t_{beschl,beginn} \leq t \leq t_{beschl,ende} \quad (10)$$

$$a(t) = 0 \quad \text{für } t_{vkonst,beginn} < t \leq t_{vkonst,ende} \quad (11)$$

$$a(t) = -a_{max} \quad \text{für } t_{brems,beginn} < t \leq t_{brems,ende} = t_{ges} \quad (12)$$

Für die Geschwindigkeit ergibt sich:

$$v(t) = a_{max} t \quad \text{für } t_0 = t_{beschl,beginn} \leq t \leq t_{beschl,ende} \quad (13)$$

$$v(t) = v_{max} = a_{max} \Delta t_{beschl} \quad \text{für } t_{vkonst,beginn} < t \leq t_{vkonst,ende} \quad (14)$$

$$\begin{aligned} v(t) &= a_{max} (\Delta t_{beschl} + t_{brems,beginn}) - a_{max} t \\ &\quad \text{für } t_{brems,beginn} < t \leq t_{brems,ende} = t_{ges} \end{aligned} \quad (15)$$

Für den zurückgelegten Weg ergibt sich:

$$s(t) = \frac{1}{2} a_{max} t^2 \quad \text{für } t_{beschl,beginn} \leq t \leq t_{beschl,ende} \quad (16)$$

$$s(t) = a_{max} \Delta t_{beschl} t - \frac{1}{2} a_{max} \Delta t_{beschl}^2 \quad \text{für } t_{vkonst,beginn} < t \leq t_{vkonst,ende} \quad (17)$$

$$s(t) = a_{max} \left( -\frac{1}{2} (\Delta t_{beschl}^2 + t_{brems,beginn}^2) + (\Delta t_{beschl} + t_{brems,beginn}) t - \frac{1}{2} t^2 \right) \quad (18)$$

für  $t_{brems,beginn} < t \leq t_{brems,ende} = t_{ges}$

Herleitung von Gl. (15):

$$\begin{aligned} v(t) &= v_{max} - a_{max} (t - t_{brems,beginn}) \\ &= a_{max} \Delta t_{beschl} - a_{max} (t - t_{brems,beginn}) \\ &= a_{max} \Delta t_{beschl} - a_{max} t + a_{max} t_{brems,beginn} \\ &= a_{max} (\Delta t_{beschl} + t_{brems,beginn}) - a_{max} t \end{aligned}$$

Herleitung von Gl. (17):

$$\begin{aligned} s(t) &= \frac{1}{2} a_{max} \Delta t_{beschl}^2 + v_{max} (t - \Delta t_{beschl}) \\ &= \frac{1}{2} a_{max} \Delta t_{beschl}^2 + a_{max} \Delta t_{beschl} (t - \Delta t_{beschl}) \\ &= \frac{1}{2} a_{max} \Delta t_{beschl}^2 + a_{max} \Delta t_{beschl} t - a_{max} \Delta t_{beschl}^2 \\ &= a_{max} \Delta t_{beschl} t - \frac{1}{2} a_{max} \Delta t_{beschl}^2 \end{aligned}$$

Herleitung von Gl. (18):

Für den bis im dritten Zeitintervall zurückgelegten Weg ergibt sich:

$$s(t) = s_{beschl} + s_{vkonst} + s_{brems} \quad (I)$$

$$\text{mit } s_{beschl} = \frac{1}{2} a_{max} \Delta t_{beschl}^2 \quad (II)$$

$$s_{vkonst} = v_{max} \Delta t_{vkonst} = a_{max} \Delta t_{beschl} \Delta t_{vkonst} \quad (III)$$

$$s_{brems} = v_0 t + \frac{1}{2} a t^2 \text{ (allgemein)}$$

$$\text{hier: } t \equiv t - t_{brems,beginn}$$

$$v_0 = v_{max} = a_{max} \Delta t_{beschl}$$

$$a = -a_{max}$$

ergibt

$$s_{brems} = a_{max} \Delta t_{beschl} (t - t_{brems,beginn}) - \frac{1}{2} a_{max} (t - t_{brems,beginn})^2 \quad (IV)$$

Setzt man nun (II), (III) und (IV) in (I) ein ergibt sich:

$$\begin{aligned} s(t) &= \frac{1}{2} a_{max} \Delta t_{beschl}^2 + a_{max} \Delta t_{beschl} \Delta t_{vkonst} \\ &\quad + a_{max} \Delta t_{beschl} (t - t_{brems,beginn}) - \frac{1}{2} a_{max} (t - t_{brems,beginn})^2 \\ &= \frac{1}{2} a_{max} \Delta t_{beschl}^2 + a_{max} \Delta t_{beschl} \Delta t_{vkonst} + a_{max} \Delta t_{beschl} t \\ &\quad - a_{max} \Delta t_{beschl} t_{brems,beginn} - \frac{1}{2} a_{max} (t^2 - 2 t t_{brems,beginn} + t_{brems,beginn}^2) \\ &= \frac{1}{2} a_{max} \Delta t_{beschl}^2 + a_{max} \Delta t_{beschl} \Delta t_{vkonst} + a_{max} \Delta t_{beschl} t \\ &\quad - a_{max} \Delta t_{beschl} t_{brems,beginn} - a_{max} \left( \frac{1}{2} t^2 - t t_{brems,beginn} + \frac{1}{2} t_{brems,beginn}^2 \right) \end{aligned}$$

$$= a_{max} \left( \frac{1}{2} \Delta t_{beschl}^2 + \Delta t_{beschl} \Delta t_{vkonst} + \Delta t_{beschl} t \right. \\ \left. - \Delta t_{beschl} t_{brems,beginn} - \frac{1}{2} t^2 + t t_{brems,beginn} - \frac{1}{2} t_{brems,beginn}^2 \right)$$

mit  $t_{brems,beginn} = \Delta t_{beschl} + \Delta t_{vkonst}$  ergibt sich:

$$s(t) = a_{max} \left( \frac{1}{2} \Delta t_{beschl}^2 + \Delta t_{beschl} \Delta t_{vkonst} + t_{beschl} t \right. \\ \left. - \Delta t_{beschl} (\Delta t_{beschl} + \Delta t_{vkonst}) - \frac{1}{2} t^2 + t t_{brems,beginn} - \frac{1}{2} t_{brems,beginn}^2 \right) \\ = a_{max} \left( \frac{1}{2} \Delta t_{beschl}^2 + \Delta t_{beschl} \Delta t_{vkonst} + t_{beschl} t \right. \\ \left. - \Delta t_{beschl}^2 - \Delta t_{beschl} \Delta t_{vkonst} - \frac{1}{2} t^2 + t t_{brems,beginn} - \frac{1}{2} t_{brems,beginn}^2 \right) \\ = a_{max} \left( \frac{1}{2} \Delta t_{beschl}^2 - \Delta t_{beschl}^2 - \frac{1}{2} t_{brems,beginn}^2 + \Delta t_{beschl} t \right. \\ \left. + t t_{brems,beginn} - \frac{1}{2} t^2 \right) \\ s(t) = a_{max} \left( -\frac{1}{2} (\Delta t_{beschl}^2 + t_{brems,beginn}^2) + t (\Delta t_{beschl} + t_{brems,beginn}) - \frac{1}{2} t^2 \right) \quad (18)$$

Im allgemeinen Berechnungsfall muß man davon ausgehen, daß von der Herstellerseite die Werte für  $v_{max}$  und  $a_{max}$  bekannt sind. Die Aufgabenstellung wird also folgendermaßen lauten:

Die Roboterachse steht auf dem Startwert  $s(t_0)$  und soll auf den Endwert  $s(t_{ges})$  gefahren werden (Bsp.: Drehung einer rotatorischen Achse vom Startwinkel  $0^\circ$  bis zum Endwinkel  $200^\circ$ ). Aufgrund dieser Angaben sind die entsprechenden Werte zu ermitteln.

Relativ einfach lassen sich die Beschleunigungszeiten  $\Delta t_{beschl} = \Delta t_{brems} -$  Gl. (8) – ermitteln. Nach Gl. (14) gilt

$$v_{max} = a_{max} \Delta t_{beschl} \Leftrightarrow \Delta t_{beschl} = \frac{v_{max}}{a_{max}} \quad (19)$$

Schwieriger wird es für die Bestimmung der Gesamtzeit des Vorganges  $t_{ges}$ . Hierzu ist folgende Herleitung notwendig:

Nach Gl. (18) gilt

$$s(t) = \frac{1}{2} a_{max} \Delta t_{beschl}^2 + v_{max} \Delta t_{vkonst} + v_{max} (t - t_{brems,beginn}) \\ - \frac{1}{2} a_{max} (t - t_{brems,beginn})^2 \quad \text{für } t_{brems,beginn} < t \leq t_{brems,end} = t_{ges}$$

für  $t = t_{ges}$  ergibt sich somit nach den Gl. (I), (II), (III) und (IV)

$$\begin{aligned} s(t_{ges}) &= \frac{1}{2} a_{max} \Delta t_{beschl}^2 + v_{max} \Delta t_{vkonst} + v_{max} (t_{ges} - t_{brems,beginn}) \\ &\quad - \frac{1}{2} a_{max} (t_{ges} - t_{brems,beginn})^2 \end{aligned}$$

Diese Gleichung vereinfacht sich mit Gl. (7) und Gl. (8) zu

$$\begin{aligned} s(t_{ges}) &= \frac{1}{2} a_{max} \Delta t_{beschl}^2 + v_{max} \Delta t_{vkonst} + v_{max} \Delta t_{beschl} - \frac{1}{2} a_{max} \Delta t_{beschl}^2 \\ &= v_{max} \Delta t_{vkonst} + v_{max} \Delta t_{beschl} \\ &= v_{max} (\Delta t_{vkonst} + \Delta t_{beschl}) \end{aligned}$$

Aus dieser Gleichung lässt sich die Unbekannte  $\Delta t_{vkonst}$  ermitteln:

$$\Delta t_{vkonst} = \frac{s(t_{ges})}{v_{max}} - \Delta t_{beschl} \quad (20)$$

Mit Gl. (19) lässt sich dann auch Gl. (9) lösen:

$$\begin{aligned} t_{ges} &= 2 \frac{v_{max}}{a_{max}} + \frac{s(t_{ges})}{v_{max}} - \Delta t_{beschl} \\ t_{ges} &= 2 \frac{v_{max}}{a_{max}} + \frac{s(t_{ges})}{v_{max}} - \frac{v_{max}}{a_{max}} \\ t_{ges} &= \frac{v_{max}}{a_{max}} + \frac{s(t_{ges})}{v_{max}} \end{aligned} \quad (20a)$$

Dieses einfache Interpolationsverfahren soll zunächst an einem Beispiel erläutert werden.

#### Beispiel 1

Eine rotatorische Roboterachse soll vom Wert  $s(t_0) = 0^\circ$  auf den Endwert  $s(t_{ges}) = 200^\circ$  gebracht werden. Die Herstellerangaben für die maximale Achsgeschwindigkeit  $v_{max}$  und die maximale Achsbeschleunigung  $a_{max}$  sind:

$$\begin{aligned} v_{max} &= 100^\circ \text{ s}^{-1} \text{ und} \\ a_{max} &= 200^\circ \text{ s}^{-2} \end{aligned}$$

Nach Gl. (19) ergeben sich für die Beschleunigungs- und die Bremszeit folgende Werte:

$$\Delta t_{beschl} = \Delta t_{brems} = \frac{v_{max}}{a_{max}} = \frac{100^\circ \text{ s}^{-1}}{200^\circ \text{ s}^{-2}} \Rightarrow \Delta t_{beschl} = \Delta t_{brems} = 0,5 \text{ s}$$

Mit Hilfe von Gl. (20) kann dann  $\Delta t_{vkonst}$  ermittelt werden:

$$\Delta t_{vkonst} = \frac{s(t_{ges})}{v_{max}} - \Delta t_{beschl} = \frac{200^\circ}{100^\circ \text{ s}^{-1}} - 0,5 \text{ s} \Rightarrow \Delta t_{vkonst} = 1,5 \text{ s}$$

Somit ergibt sich die Gesamtzeit des Bewegungsvorgangs nach Gl. (9) zu:

$$t_{ges} = 2\Delta t_{beschl} + \Delta t_{vkonst} = 2 \cdot 0,5 \text{ s} + 1,5 \text{ s} \Rightarrow t_{ges} = 2,5 \text{ s}$$

Damit ist der gesamte Bewegungsvorgang berechenbar.

Man hat mit diesen Gleichungen leider noch nicht alle Möglichkeiten abgedeckt, die eintreten können. Dies sei an einem einfachen Beispiel demonstriert:

**Beispiel 2**

Eine rotatorische Roboterachse soll vom Wert  $s(t_0) = 0^\circ$  auf den Endwert  $s(t_{\text{ges}}) = 200^\circ$  gebracht werden. Die Herstellerangaben für die maximale Achsgeschwindigkeit  $v_{\text{max}}$  und die maximale Achsbeschleunigung  $a_{\text{max}}$  sind:

$$\begin{aligned}v_{\text{max}} &= 200^\circ \text{ s}^{-1} \text{ und} \\a_{\text{max}} &= 50^\circ \text{ s}^{-2}\end{aligned}$$

Nach Gl. (19) ergibt sich für die Beschleunigungs- und die Bremszeit folgender Wert:

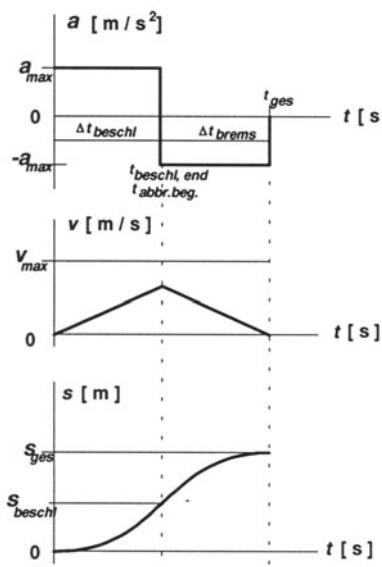
$$\Delta t_{\text{beschl}} = \Delta t_{\text{brems}} = \frac{v_{\text{max}}}{a_{\text{max}}} = \frac{200^\circ \text{ s}^{-1}}{50^\circ \text{ s}^{-2}} \Rightarrow \Delta t_{\text{beschl}} = \Delta t_{\text{brems}} = 4 \text{ s}$$

Mit Hilfe von Gl. (20) kann dann  $\Delta t_{v\text{kons}}$  ermittelt werden:

$$\Delta t_{v\text{kons}} = \frac{s(t_{\text{ges}})}{v_{\text{max}}} - \Delta t_{\text{beschl}} = \frac{200^\circ}{200^\circ \text{ s}^{-1}} - 4 \text{ s} \Rightarrow \Delta t_{v\text{kons}} = -3 \text{ s}$$

Eine negative Zeit, in der mit konstanter Geschwindigkeit  $v_{\text{max}}$  gefahren wird, gibt physikalisch keinen Sinn. Die Interpretation dieser Variante liegt darin, daß der Gesamtweg für diesen Vorgang so kurz ist, daß die Maximalgeschwindigkeit  $v_{\text{max}}$  nicht erreicht werden kann. Vielmehr muß die Bewegung der Achse direkt vom Beschleunigen ( $a > 0$ ) ins Verzögern ( $a < 0$ ) übergehen. Diesen Vorgang verdeutlicht Bild B-5.

Tritt dieser Fall ein, so wird die maximale Geschwindigkeit der Achse zu keinem Zeitpunkt erreicht. Vielmehr geht die Achse direkt vom Beschleunigungszustand in den Bremszustand über. Damit müssen auch die Bewegungsgleichungen geändert werden. Es ergeben sich somit nur noch zwei Bewegungszustände, sprich Zeitintervalle, für die die entsprechenden Bewegungsgleichungen zu finden sind.

**Bild B-5**

$a$ - $t$ -,  $v$ - $t$ - und  $s$ - $t$ -Diagramm für die Bewegungen einer Roboterachse, bei der die maximale Achsgeschwindigkeit  $v_{\text{max}}$  nicht erreicht wird

Vorab eine Zusammenfassung der Zeiten

$$t_0 = t_{beschl,beginn} \quad (21)$$

$$t_{beschl,ende} = t_{brems,beginn} \quad (22)$$

$$t_{beschl,ende} = t_{ges} \quad (23)$$

Somit ergibt sich

$$\begin{aligned} \Delta t_{beschl} &= t_{beschl,ende} - t_{beschl,beginn} \\ &= t_{beschl,ende} - t_0 \\ &= t_{brems,beginn} - t_0 \end{aligned} \quad (24)$$

(Zeit, in der aus der Ruhe beschleunigt wird; d.h.  $a > 0$ )

Ferner gilt

$$\Delta t_{brems} = \Delta t_{beschl} \quad (25)$$

(Bremszeit = Beschleunigungszeit, da  $a_{beschl} = -a_{brems}$ )

Somit ergibt sich für die Gesamtzeit  $t_{ges}$  des Vorgangs folgende Gleichung:

$$\begin{aligned} t_{ges} &= \Delta t_{beschl} + \Delta t_{brems} \\ t_{ges} &= 2 \Delta t_{beschl} \\ t_{ges} &= 2 \Delta t_{brems} \end{aligned} \quad (26)$$

Für die Beschleunigung ergibt sich:

$$a(t) = a_{max} \quad \text{für } t_0 = t_{beschl,beginn} \leq t \leq t_{beschl,ende} \quad (27)$$

$$a(t) = -a_{max} \quad \text{für } t_{brems,beginn} < t \leq t_{brems,ende} = t_{ges} \quad (28)$$

Für die Geschwindigkeit ergibt sich:

$$v(t) = a_{max} t \quad \text{für } t_0 = t_{beschl,beginn} \leq t \leq t_{beschl,ende} \quad (29)$$

$$v(t) = 2a_{max} t_{beschl,ende} - a_{max} t \quad \text{für } t_{brems,beginn} < t \leq t_{brems,ende} = t_{ges} \quad (30)$$

Für die Wege ergibt sich:

$$s(t) = \frac{1}{2} a_{max} t^2 \quad \text{für } t_0 = t_{beschl,beginn} \leq t \leq t_{beschl,ende} \quad (31)$$

$$s(t) = a_{max} \left( -t_{beschl,ende}^2 + 2t t_{beschl,ende} - \frac{1}{2} t^2 \right) \quad \text{für } t_0 = t_{beschl,beginn} \leq t \leq t_{beschl,ende} \quad (32)$$

Herleitung von Gl. (30):

$$\text{allgemein} \quad v(t) = v_0 + a t$$

$$\text{hier gilt} \quad v_0 = a_{max} t_{beschl,ende} \quad \text{und} \quad a = -a_{max}$$

$$\text{somit} \quad v(t) = a_{max} t_{beschl,ende} - a_{max} (t - t_{brems,beginn})$$

$$\begin{aligned} \text{mit Gl. (22)} \quad v(t) &= a_{max} t_{beschl,ende} - a_{max} (t - t_{beschl,ende}) \\ &= a_{max} t_{beschl,ende} - a_{max} t + a_{max} t_{beschl,ende} \\ &= 2a_{max} t_{beschl,ende} - a_{max} t \end{aligned}$$

Herleitung von Gl. (32):

$$\begin{aligned}
 & \text{allgemein} \quad s(t) = s_0 + v_0 t + \frac{1}{2} a t^2 \\
 & \text{mit} \quad t \equiv t - t_{\text{brems beginn}} = t - t_{\text{beschl ende}} \\
 & \text{und} \quad s(0) = \frac{1}{2} a_{\max} t_{\text{beschl ende}}^2 \\
 & \quad v(0) = a_{\max} t_{\text{beschl ende}} \\
 & \quad a = -a_{\max} \\
 & \text{somit } s(t) = \frac{1}{2} a_{\max} t_{\text{beschl ende}}^2 + a_{\max} t_{\text{beschl ende}} (t - t_{\text{beschl ende}}) \\
 & \quad - \frac{1}{2} a_{\max} (t - t_{\text{beschl ende}})^2 \\
 & = \frac{1}{2} a_{\max} t_{\text{beschl ende}}^2 + a_{\max} (t t_{\text{beschl ende}} - t_{\text{beschl ende}}^2) \\
 & \quad - \frac{1}{2} a_{\max} (t^2 - 2t t_{\text{beschl ende}} + t_{\text{beschl ende}}^2) \\
 & = a_{\max} \left( \frac{1}{2} t_{\text{beschl ende}}^2 + t t_{\text{beschl ende}} - t_{\text{beschl ende}}^2 \right. \\
 & \quad \left. - \frac{1}{2} t^2 + t t_{\text{beschl ende}} - \frac{1}{2} t_{\text{beschl ende}}^2 \right) \\
 s(t) & = a_{\max} \left( -t_{\text{beschl ende}}^2 + 2t t_{\text{beschl ende}} - \frac{1}{2} t^2 \right)
 \end{aligned}$$

Zur Berechnung des Umschaltpunktes setzt man wiederum den bekannten Wert für  $s(t_{\text{ges}})$  in Gl. (32) ein und erhält dann

$$t_{\text{beschl}} = \sqrt{\frac{s(t_{\text{ges}})}{a_{\max}}} \quad (33)$$

Eine andere Möglichkeit, Gl. (33) herzuleiten, ist der Ansatz, daß nach  $t_{\text{beschl}}$  der halbe Weg zurückgelegt sein muß. Somit ergibt sich Gl. (31) zu:

$$\begin{aligned}
 s(t_{\text{beschl}}) &= \frac{1}{2} a_{\max} t_{\text{beschl}}^2 = \frac{1}{2} s(t_{\text{ges}}) \\
 \Rightarrow s(t_{\text{ges}}) &= a_{\max} t_{\text{beschl}}^2 \\
 \Rightarrow t_{\text{beschl}} &= \sqrt{\frac{s(t_{\text{ges}})}{a_{\max}}}
 \end{aligned}$$

Für das oben erwähnte Beispiel ergibt sich dann folgende Lösung:

$$t_{\text{beschl}} = \sqrt{\frac{s(t_{\text{ges}})}{a_{\max}}} = \sqrt{\frac{100^\circ}{50^\circ \text{s}^{-1}}} = \sqrt{2} \text{ s}$$

### B.3.2 Kopplung mehrerer Achsen

Für einen Industrieroboter, der ja per Definition aus mindestens drei frei programmierbaren Achsen besteht, muß das oben beschriebene Verfahren für eine Achse entsprechend erweitert werden. Die prinzipiellen Bahngleichungen aus dem vorherigen Kapitel verändern sich nicht.

Zunächst sei auch hier der Fall betrachtet, daß die Maximalgeschwindigkeit  $v_{max}$  erreicht wird. Der Index  $i$  bezeichnet die Nummer der jeweiligen Roboterachse. Somit ergibt sich

- für die Achsbeschleunigungen

$$a_i(t) = a_{max,i} \quad \text{für } t_{0,i} = t_{beschl,beginn,i} \leq t \leq t_{beschl,ende,i} \quad (34)$$

$$a_i(t) = 0 \quad \text{für } t_{vkonst,beginn,i} < t \leq t_{vkonst,ende,i} \quad (35)$$

$$a_i(t) = -a_{max,i} \quad \text{für } t_{brems,beginn,i} < t \leq t_{brems,ende,i} = t_{ges,i} \quad (36)$$

- für die Geschwindigkeiten

$$v_i(t) = a_{max,i} t \quad \text{für } t_{0,i} = t_{beschl,beginn,i} \leq t \leq t_{beschl,ende,i} \quad (37)$$

$$v_i(t) = v_{max,i} = a_{max,i} \Delta t_{beschl,i} \quad \text{für } t_{vkonst,beginn,i} < t \leq t_{vkonst,ende,i} \quad (38)$$

$$v_i(t) = a_{max,i} (\Delta t_{beschl,i} + t_{brems,beginn,i}) - a_{max,i} t \quad \text{für } t_{brems,beginn,i} < t \leq t_{brems,ende,i} = t_{ges,i} \quad (39)$$

- für den zurückgelegten Weg

$$s_i(t) = \frac{1}{2} a_{max,i} t^2 \quad \text{für } t_{0,i} = t_{beschl,beginn,i} \leq t \leq t_{beschl,ende,i} \quad (40)$$

$$s_i(t) = a_{max,i} \Delta t_{beschl,i} t - \frac{1}{2} a_{max,i} \Delta t_{beschl,i}^2 \quad \text{für } t_{vkonst,beginn,i} < t \leq t_{vkonst,ende,i} \quad (41)$$

$$s_i(t) = a_{max,i} \left( -\frac{1}{2} (\Delta t_{beschl,i}^2 + t_{brems,beginn,i}^2) + t (\Delta t_{beschl,i} + t_{brems,beginn,i}) - \frac{1}{2} t^2 \right) \quad \text{für } t_{brems,beginn,i} < t \leq t_{brems,ende,i} = t_{ges,i} \quad (42)$$

Hierbei tritt allerdings folgendes Problem auf: Wenn eine Bewegung des Roboters von einem Punkt im Raum zu einem anderen Punkt erfolgen soll, so müssen sich eventuell alle gekoppelten Achsen bewegen. Sie werden aber nur in Ausnahmefällen den gleichen Weg zurückzulegen haben.

Mit den obigen Gleichungen [Gl. (34) bis Gl. (42)] lassen sich für jede der  $i$  Achsen Bahnfunktionen berechnen, die von allen anderen Achsen unabhängig sind. Werden diese unabhängigen Funktionen an die Steuerung – und von da aus die Lageregelung der jeweiligen Achse – weitergegeben, so spricht man von einer *unsynchronisierten PTP-Bewegung* – auch *asynchrone PTP-Bewegung* genannt. Das Bewegungsende der einzelnen Achsen fällt hierbei im allgemeinen nicht zusammen.

Will man die Bewegung der einzelnen Achsen synchronisieren – *synchrone PTP-Bewegung* –, so müssen die Bewegungsgleichungen der einzelnen Achsen angeglichen werden. Man ermittelt die Achse, deren Bewegungsvorgang am längsten dauert, und gleicht alle anderen daran an.

$$t_{ges,max} = \max(t_{ges,i})$$

Auch hier ergibt sich die Beschleunigungszeit der Achse  $i$  nach Gl. (19) zu:

$$\Delta t_{besch,i} = \frac{a_{max,i}}{v_{max,i}}$$

Ferner lässt sich nach Gl. (20a) die Gesamtzeit für die Bewegung ausrechnen nach:

$$t_{ges,i} = v_{max,i}/a_{max,i} + s(t_{ges,i})/v_{max,i}$$

Zur Synchronisation wird nun  $t_{ges,i}$  der einzelnen Achse gleich dem Maximum  $t_{ges,max}$  gesetzt. Es ergibt sich:

$$t_{ges,max} = v_{max,i}/a_{max,i} + s(t_{ges,i})/v_{max,i} \quad (43)$$

Wie man sieht, hat man zur Anpassung der jeweiligen Achse zwei Möglichkeiten: Man kann in Gl. (43)  $v_{max,i}$  oder  $a_{max,i}$  variieren, um die Zeitanpassung durchzuführen. Dadurch ist der Index *max* nicht mehr gerechtfertigt und soll durch *red* ersetzt werden.

Für die Reduzierung der Achsbeschleunigung ergibt sich folgende Gleichung:

$$t_{ges,max} = \frac{v_{max,i}}{a_{red,i}} + \frac{s(t_{ges})}{v_{max,i}} \quad (44)$$

Für die Reduzierung der Maximalgeschwindigkeit erhält man:

$$t_{ges,max} = \frac{v_{red,i}}{a_{max,i}} + \frac{s(t_{ges})}{v_{red,i}} \quad (45)$$

Löst man Gl. (44) nach  $a_{red,i}$  auf, erhält man:

$$a_{red,i} = \frac{v_{max,i}}{t_{ges,max} - \frac{s(t_{ges,i})}{v_{max,i}}} \quad (46)$$

Löst man Gl. (45) nach  $v_{red,i}$  auf, erhält man die quadratische Gleichung:

$$v_{red,i} = \frac{1}{2} a_{max,i} t_{ges,max} \pm \sqrt{\frac{1}{4} a_{max,i}^2 t_{ges,max} - s(t_{ges,i}) a_{max,i}} \quad (47)$$

Es ergeben sich zwei Lösungsmöglichkeiten, von der nur eine physikalisch sinnvoll ist. Die reduzierte Geschwindigkeit  $v_{red,i}$  muß ja kleiner sein als die Maximalgeschwindigkeit  $v_{max,i}$  der jeweiligen Achse. Da der Wurzelterm größer Null ist, muß das negative Vorzeichen gewählt werden. Man erhält:

$$v_{red,i} = \frac{1}{2} a_{max,i} t_{ges,max} - \sqrt{\frac{1}{4} a_{max,i}^2 t_{ges,max} - s(t_{ges,i}) a_{max,i}} \quad (48)$$

Ähnlich wie bei der Interpolation für eine Achse, kann es auch hier vorkommen, daß eine bzw. mehrere Achsen nicht die Maximalgeschwindigkeit in der zur Verfügung stehenden Zeit erreichen. Man erhält aus den Gl. (27) bis (32):

- für die Beschleunigung

$$a_i(t) = a_{max,i} \quad \text{für } t_{0,i} = t_{beschl,beginn,i} \leq t \leq t_{beschl,ende,i} \quad (49)$$

$$a_i(t) = -a_{max,i} \quad \text{für } t_{brems,beginn,i} < t \leq t_{brems,ende,i} = t_{ges,i} \quad (50)$$

- für die Geschwindigkeiten

$$v_i(t) = a_{max,i} t \quad \text{für } t_{0,i} = t_{beschl,beginn,i} \leq t \leq t_{beschl,ende,i} \quad (51)$$

$$v_i(t) = 2a_{max,i} t_{beschl,ende,i} - a_{max,i} t \quad \text{für } t_{brems,beginn,i} < t \leq t_{brems,ende,i} = t_{ges,i} \quad (52)$$

- für die zurückgelegten Wege

$$s_i(t) = \frac{1}{2} a_{max,i} t^2 \quad \text{für } t_{0,i} = t_{beschl,beginn,i} \leq t \leq t_{beschl,ende,i} \quad (53)$$

$$s_i(t) = a_{max,i} \left( -t_{beschl,ende,i}^2 + 2t_{beschl,ende,i} t - \frac{1}{2} t^2 \right) \quad \text{für } t_{brems,beginn,i} < t \leq t_{brems,ende,i} = t_{ges,i} \quad (54)$$

Zur Berechnung des Umschaltpunktes setzt man wiederum den bekannten Wert für  $s(t_{ges,i})$  in Gl. (54) ein und erhält dann

$$\Delta t_{beschl,i} = \sqrt{\frac{s(t_{ges,i})}{a_{max,i}}} \quad (55)$$

Eine andere Möglichkeit, Gl. (55) herzuleiten, ist der Ansatz, daß nach  $\Delta t_{beschl,i}$  der halbe Weg zurückgelegt sein muß. Somit ergibt sich wieder Gl. (55).

Damit ergibt sich die Gesamtzeit für den Vorgang zu:

$$t_{ges,i} = 2 \cdot \sqrt{\frac{s(t_{ges,i})}{a_{max,i}}} \quad (56)$$

Bei der Synchronisation mehrerer Achsen kann in Gl. (56) der Wert  $t_{ges,max}$  eingesetzt und die Gleichung nach  $a_{red,i}$  aufgelöst werden. Dadurch erhält man:

$$\begin{aligned} t_{ges,max} &= 2 \cdot \sqrt{\frac{s(t_{ges,i})}{a_{red,i}}} \\ \Leftrightarrow a_{red,i} &= \frac{4 s(t_{ges,i})}{t_{ges,max}} \end{aligned} \quad (57)$$

Die andere Möglichkeit ist, daß man durch die Reduzierung der Maximalgeschwindigkeit  $v_{max,i}$  auf  $v_{red,i}$  wieder in ein Bewegungsmuster mit zwischenzeitlicher konstanter Geschwindigkeit  $v_{red,i}$  kommt. Auf diese Herleitung sei hier verzichtet.

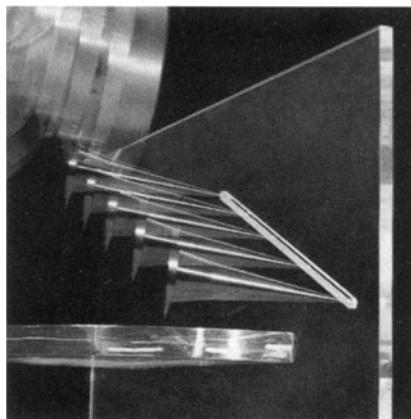
Abschließend sei zu diesem Themengebiet bemerkt, daß es noch andere Verfahren wie z.B. Polynominterpolation zwischen zwei Punkten bzw. Interpolation mit kubischen Splines gibt, um die Bahnberechnung durchzuführen. Auf die Herleitungen sei hier mit Hinweis auf die weiterführende Literatur (z.B. Schwinn: Grundlagen der Roboterkinematik) verzichtet.

## B.4 Bahnsteuerungen für mehr als drei Achsen

Viele Anwendungsgebiete in der Fertigungstechnik, in der Industrieroboter eingesetzt werden, wie

- Montage,
- Entgraten,
- Spritzlackieren
- Schweißen,
- Brennschneiden etc.

erfordern die Möglichkeit, den Endeffektor auf genau definierten Bahnen zu bewegen

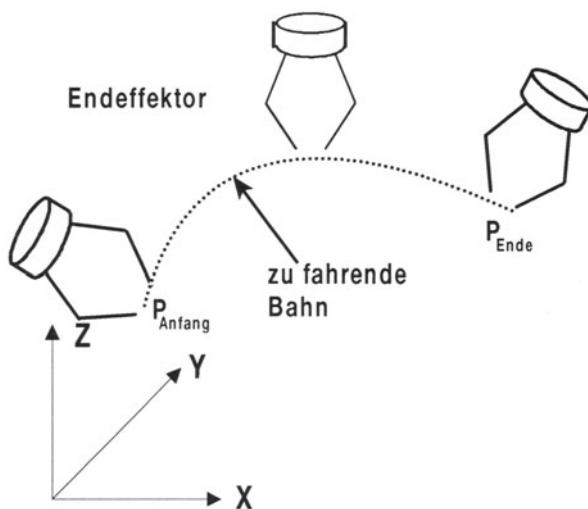


**Bild B-6**

Zusammenspiel mehrerer Bewegungssachsen eines IR zum Erzeugen einer Geraden (KUKA)

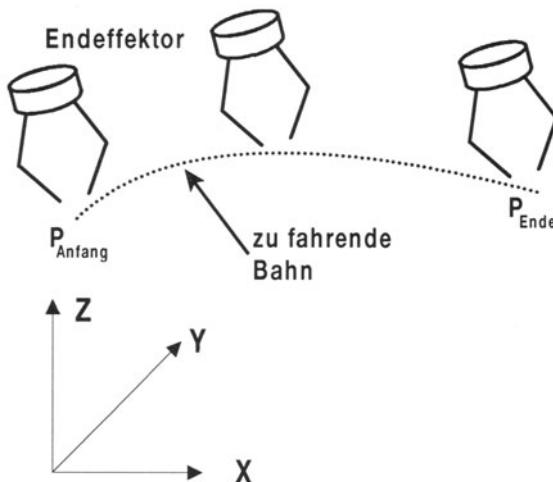
Wie in Bild B-6 deutlich zu sehen ist, erfordert die Bewegung auf einer genau definierten Bahn im Raum, daß im allgemeinen Fall alle Achsen des Roboters entsprechend verschieden geführt werden müssen.

Der in der Anwendung am häufigsten vorkommende Fall bei der Bahnsteuerung ist, daß zwischen zwei oder mehreren geteachten Punkten irgendeine – mathematisch definierte – Bahn gefahren werden muß, d.h. der TCP soll sich genau auf dieser Bahn bewegen. Dabei spielt die Orientierung des Endeffektors eine untergeordnete Rolle. Die Orientierung wird von der Anfangsorientierung kontinuierlich in die Endorientierung überführt. Diese beiden Orientierungen sind durch die geteachten Punkte gegeben.

**Bild B-7**

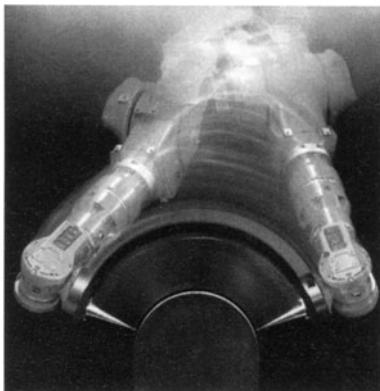
Kontinuierliche Überführung der Orientierung des Endeffektors von einer Anfangs- in eine Endorientierung

Es gibt aber Anwendungen, die verlangen, daß die Orientierung bezüglich der Bahn konstant gehalten werden muß (z.B. beim Schweißen).

**Bild B-8**

Konstante Orientierung des Endeffektors bezüglich der Bahn

Ferner gibt es auch Roboterbewegungen, bei denen die Werkzeugposition bezüglich des TCP konstant bleibt ( $x$ -,  $y$ - und  $z$ -Koordinaten) und nur die Orientierungswinkel verändert werden. Man kann dabei eine sog. Nickbewegung ausführen oder zusätzlich noch eine sog. Gierbewegung und eine sog. Rollbewegung. Diese Bewegungskombination nennt man Quirlen. Dabei rotiert die Mittelachse des Endeffektors auf einer Kegelmantellinie, während der Endeffektor sich zusätzlich um die eigene Achse dreht.



**Bild B-9**  
Nickbewegung (KUKA)



**Bild B-10**  
Quirlen (KUKA)

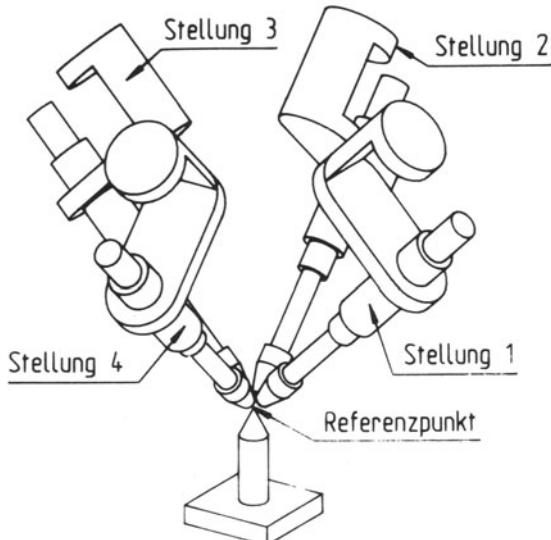
Dieses Quirlen nutzen manche Roboterhersteller zum einfachen Vermessen eines Werkzeuges, das seinen Arbeitspunkt nicht im TCP des Flansches hat. Man kann damit auf eine relativ einfache Art und Weise die Werkzeugkorrekturdaten ermitteln.

In den Werksunterlagen der Firma REIS ist dazu folgendes zu finden:

„Die Spezialfunktion Hand-/Werkzeugvermessung erlaubt, die Daten von unbekannten Werkzeugen (Handlänge, Werkzeuglänge und Werkzeugwinkel) automatisch in der Robotersteuerung zu vermessen. Hierzu steht eine komfortable Bedienoberfläche zur Verfügung. Verändert sich die Werkzeugspitze während des Prozesses, können durch Anfahren eines Referenzpunktes in 4 verschiedenen Stellungen die vorhandenen Programme ohne weitere Korrektur genutzt werden. Stillstandzeiten durch Programmänderungen entfallen.“

Es zeigt sich, daß prinzipiell zwei Anforderungen an eine Bahnsteuerung gestellt werden:

- Die Bahnverfolgung muß auf einer mathematisch beschreibbaren Kurve im Raum mit einem definierten Punkt des Endeffektors (meist der TCP) möglich sein.
- Die Orientierung des Endeffektors muß bezüglich dieser Bahn beeinflußbar sein.



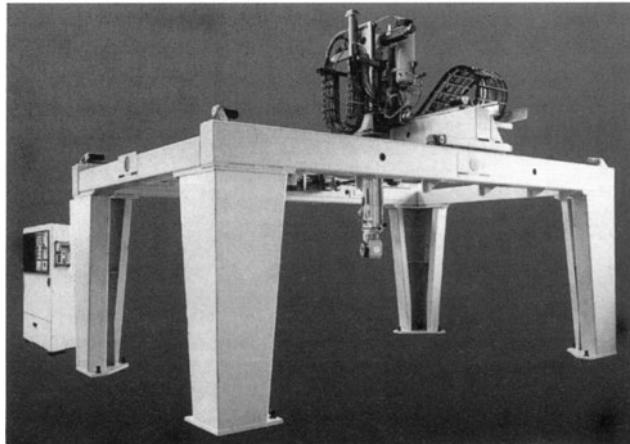
**Bild B-11**

Automatische TCP Vermessung mit einem Schweißbrenner (REIS)

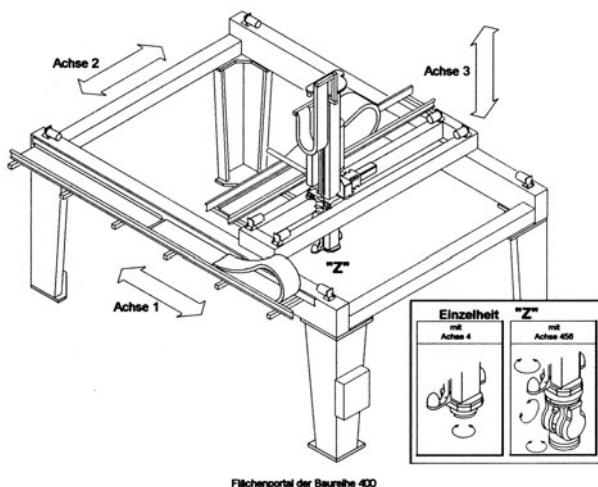
Zusammenfassend kann man sagen, daß die mathematischen Herleitungen zu diesem Thema sehr komplex und umfangreich sind und somit den Rahmen dieses Buches sprengen würden. Ferner bringt die Bahnberechnung gerade bei der Bahnsteuerung informatio-nstechnische Probleme, da die Berechnungen dermaßen umfangreich sind und in sehr kurzen Zeitintervallen wiederholt werden müssen, daß man Interpolationstakte hat, die zeitkritisch bezogen auf den Prozessor sind. Dies hat Auswirkungen auf die Taktzeiten für die Lageregelung und somit auf das Fahrverhalten des Roboters. Deswegen werden heute auch Mehrprozessorsteuerungen in Industrierobotern eingesetzt, die sich die „Arbeit“ teilen. Auch auf diese Problematik kann nur hingewiesen werden.

## Anhang C Beispiele ausgeführter Roboter

### C.1 Portalroboter der Firma DÜRR



**Bild C-1**  
Portalroboter der Firma DÜRR



**Bild C-2**  
Schemazeichnung (DÜRR)

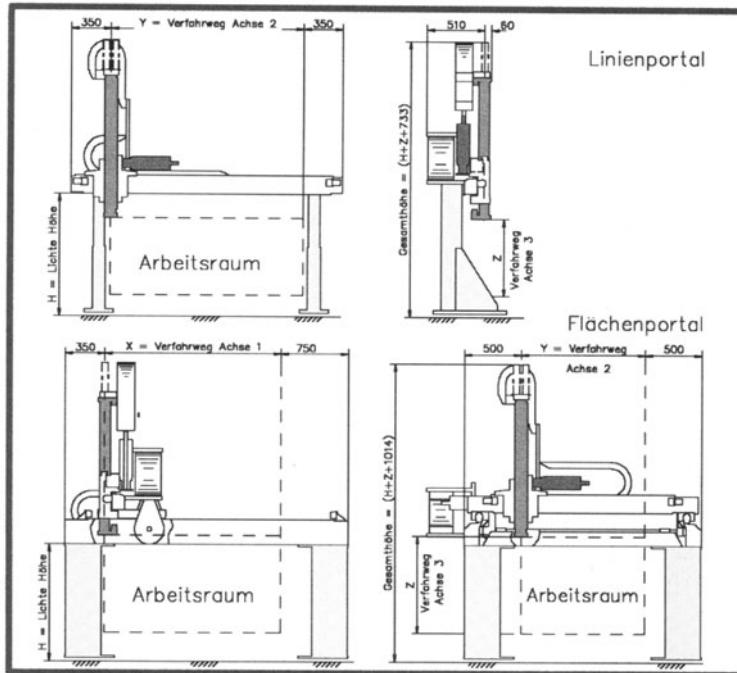
**Tabelle C-1** Technische Daten der Portalroboter Baureihe 025 (DÜRR)

Tabelle: Technische Daten

Portalroboter - Baureihe		025	Handachsen (optional)	für Baureihe 025														
Aufbau		Portal, kartesisch																
Ausführung	Linienportal	Flächenportal																
Typ	L 025	A 025																
Nutzlast an Flansch Achse 3 (bei Verfahrtsweg Achse 3 =)	40 ... 25 kg (400 ... 1000 mm)																	
Verfahrtsweg	<table> <tr> <td>Achse 1</td><td>-</td><td>1 ... 10 m</td><td></td><td></td></tr> <tr> <td>Achse 2</td><td>1 ... 10 m</td><td>0,8 ... 2,0 m</td><td></td><td></td></tr> <tr> <td>Achse 3</td><td></td><td>0,4 ... 1,0 m</td><td></td><td></td></tr> </table>	Achse 1	-	1 ... 10 m			Achse 2	1 ... 10 m	0,8 ... 2,0 m			Achse 3		0,4 ... 1,0 m				
Achse 1	-	1 ... 10 m																
Achse 2	1 ... 10 m	0,8 ... 2,0 m																
Achse 3		0,4 ... 1,0 m																
Geschwindigkeit	<table> <tr> <td>Achse 1</td><td>-</td><td>2,0 m/s</td><td></td><td></td></tr> <tr> <td>Achse 2</td><td></td><td>2,0 m/s</td><td></td><td></td></tr> <tr> <td>Achse 3</td><td></td><td>1,25 m/s</td><td></td><td></td></tr> </table>	Achse 1	-	2,0 m/s			Achse 2		2,0 m/s			Achse 3		1,25 m/s				
Achse 1	-	2,0 m/s																
Achse 2		2,0 m/s																
Achse 3		1,25 m/s																
Beschleunigung	<table> <tr> <td>Achse 1</td><td>-</td><td>3,1 m/s<sup>2</sup></td><td></td><td></td></tr> <tr> <td>Achse 2</td><td></td><td>4,0 m/s<sup>2</sup></td><td></td><td></td></tr> <tr> <td>Achse 3</td><td></td><td>7,5 m/s<sup>2</sup></td><td></td><td></td></tr> </table>	Achse 1	-	3,1 m/s <sup>2</sup>			Achse 2		4,0 m/s <sup>2</sup>			Achse 3		7,5 m/s <sup>2</sup>				
Achse 1	-	3,1 m/s <sup>2</sup>																
Achse 2		4,0 m/s <sup>2</sup>																
Achse 3		7,5 m/s <sup>2</sup>																
Wiederholgenauigkeit		$\pm 0,15 \text{ mm}$																
Motorentyp		AC bürstenlos																
Wegmeßsystem		Absolutwertgeber																

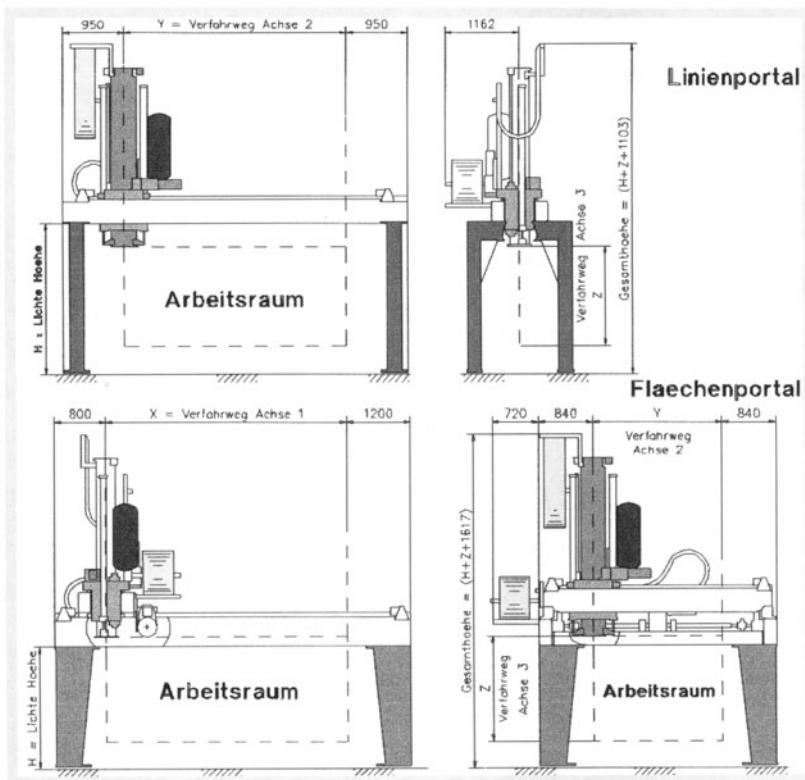
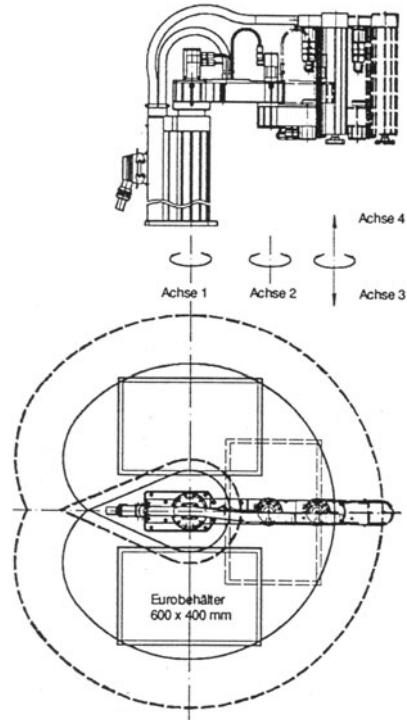
**Tabelle C-2** Technische Daten der Portalroboter Baureihe 400 (DÜRR)

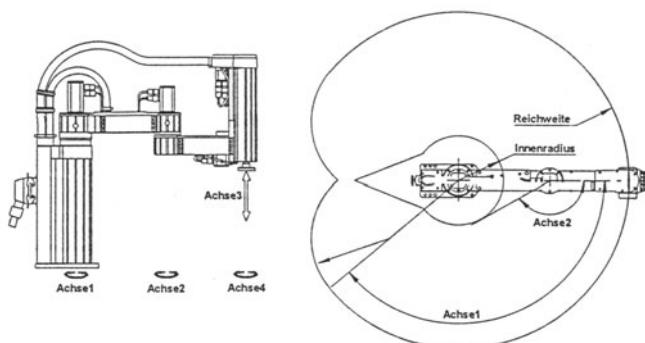
Tabelle: Technische Daten

Portalroboter - Baureihe	400	Handachsen (optional)	für Baureihe 400
Aufbau	Portal, kartesisch		
Ausführung	Linienportal Flächenportal		
Typ	L 400 A 400		
Nutzlast an Flansch Achse 3 (bei Verfahrtsweg Achse 3 =)	540 ... 400 kg (630 ... 2000 mm)		
Verfahrtsweg	Achse 1 - 2 ... 20 m Achse 2 2 ... 20 m 1,6 ... 4,0 m Achse 3 0,63 ... 2,0 m		
Geschwindigkeit	Achse 1 - 1,5 m/s Achse 2 1,5 m/s Achse 3 1,2 m/s	Mögliche Ausführungen	Achse 4 360° (120°/s) Achse 4,5,6 360° (110°/s) Achse 6 220° (110°/s)
Beschleunigung	Achse 1 - 1,67 m/s² Achse 2 2,38 m/s² Achse 3 2,5 m/s²	Drehwinkel Achse 4 Beschleunigung Achse 5 Achse 6	240°/s² 220°/s² 220°/s²
Wiederholgenauigkeit	± 0,5 mm	Nutzlast	Achse 4 300 kg Achse 6 120 kg
Motorentyp	AC bürstenlos		
Wegmeßsystem	Absolutwertgeber		

## C.2 Scara-Roboter der Firma BOSCH



**Bild C-3**  
BOSCH-Turboscara Roboterfamilie



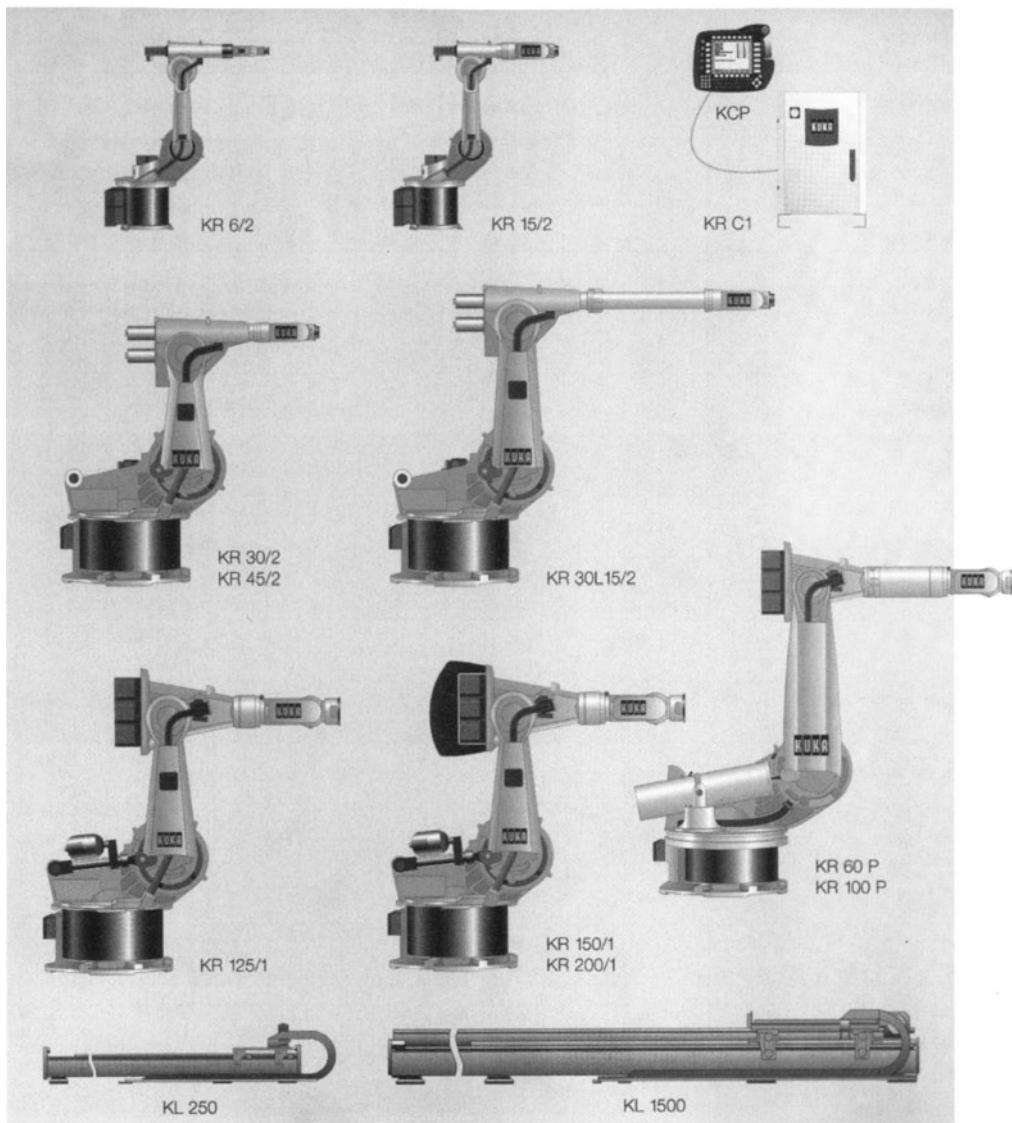
**Bild C-4** BOSCH-Turboscara Universalgerät

**Tabelle C-3** Technische Daten Universalgerät Turboscara (BOSCH)

	Standard SR60	abweichende Daten der Optionen:		Standard SR80	abweichende Daten der Optionen:	
		SR60HT			SR80HT	
<b>Arbeitsbereich</b>						
max. Arbeitsradius	mm	600		800		
Schwenkwinkel Achse 1	Grad	± 140		± 140		
Schwenkwinkel Achse 2	Grad	± 150		± 150		
Hub Achse 3 Standard	mm	200		200		
Option	mm	300		300		
Option	mm	450		450		
Drehwinkel Achse 4	Grad	360		360		
<b>Traglast</b>						
Nennlast	kg	2		2		
Maximallast	kg	5*		5*		
Nenenträgheitsmoment A4	kgcm²	150	500	150	500	
max. Trägheitsmoment A4	kgcm²	250	1000	250	1000	
<b>max. Geschwindigkeit</b>						
Übergang A1/2	mm/s	4400		5100		
Achse 1	Grad/s	265		220		
Achse 2	Grad/s	350		325		
Achse 3	mm/s	700		700		
Achse 4	Grad/s	1100	720	1100	720	
<b>Beschleunigung</b>						
bei Nennlast						
Achse 1	Grad/s²	720		620		
Achse 2	Grad/s²	1400		1400		
Achse 3	mm/s²	10000		10000		
Achse 4	Grad/s²	15000	3000	15000	3000	
<b>Reaktionskräfte</b>						
Horizontal Fx,Fy	N	60 / 150		60 / 150		
Vertikal Fz	N	150 / 250		150 / 250		
Moment Mz	Nm	2 / 4	4 / 8	2 / 4	4 / 8	
<b>Genauigkeit</b>						
Positionstreubreite	mm	± 0,025		± 0,025		
* Achse 4	Grad	± 0,1		± 0,1		
Auflösung	mm	0,01		0,01		
<b>Motorleistung</b>						
Achse 1	W	950		950		
Achse 2	W	150		450		
Achse 3	W	150		150		
Achse 4	W	60	150	60	150	
<b>Masse</b>	kg	55	58	57	58	
<b>Schutzzart</b>	DIN 40050	IP54		IP54		
<b>Temperaturbereich</b>	* C	+10 bis +40		+10 bis +40		
<b>Anwendersignale</b>						
• elektrisch: verdrahtete Eingänge verdrahtete Ausgänge	V DC	5		5		
Spannungsbereich	A	8		8		
Strombelastbarkeit		24		24		
- pneumatisch: Luftschlauch LW2		0,5		0,5		
maximaler Druck	bar	2		2		
		8		8		

\*: höhere Traglast mit reduzierten Geschwindigkeiten/Beschleunigungen auf Anfrage bei unseren Vertriebspartnern

### C.3 Knickarmroboter der Firma KUKA



**Bild C-5** Knickarmroboter (KUKA)

**Tabelle C-4** Technische Daten Knickarmroboter (KUKA)

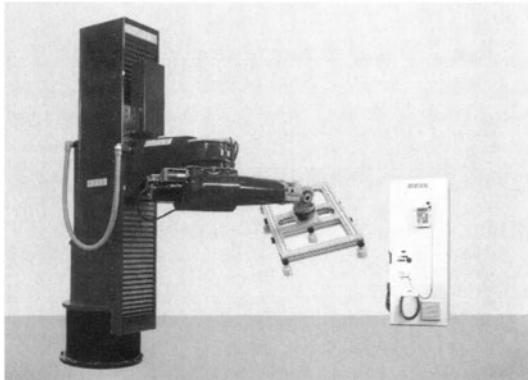
Typen/ Type/ Références	Grunddaten/Principal data/Caractéristiques de base						
	Traglast <sup>1</sup> Payload <sup>1</sup> Charge utile <sup>1</sup>	Zusatzzast Supplementary load Charge supplémentaire	Max. Reichweite <sup>2</sup> Max. reach <sup>2</sup> Portée maxi. <sup>2</sup>	Anzahl der Achsen Number of axes Nombre d'axes	Wiederholgenauigkeit/ Repeatability Précision de répétabilité	Gewicht (Roboter), ca. Weight (robot), approx. Poids (robot), env.	Einbaulage Mounting position Implantation
KR 6/2	6 kg	10 kg	1570	6	<±0,1 mm	205 kg	Variabel/ Variable
KR 15/2	15 kg	10 kg	1570	6	<±0,1 mm	222 kg	Variabel/ Variable
KR 30/2	30 kg	35 kg	2041	6	<±0,15 mm	867 kg	Variabel/ Variable
KR 30L 15/2	15 kg	35 kg	3086	6	<±0,15 mm	930 kg	Boden, Decke/ Floor, ceiling/ Au sol, au plafond
KR 45/2	45 kg	35 kg	2041	6	<±0,15 mm	880 kg	Variabel/ Variable
KR 125/1	125 kg 100 kg 90 kg	120 kg 120 kg 120 kg	2410 2610 2810	6	<±0,2 mm	975 kg 990 kg 995 kg	Boden, Decke/ Floor, ceiling/ Au sol, au plafond
KR 150/1	150 kg 150 kg 120 kg	95 kg 80 kg 80 kg	2410 2610 2810	6	<±0,2 mm	1120 kg 1135 kg 1140 kg	Boden, Decke/ Floor, ceiling/ Au sol, au plafond
KR 200/1	200 kg	80 kg	2410	6	<±0,2 mm	1120 kg	Boden, Decke/ Floor, ceiling/ Au sol, au plafond
KR 125 W/1	125 kg	120 kg	2410	6	<±0,2 mm	1590 kg	Wand/ Wall/ Mur
KR 60 P	60 kg	30 kg	3500	6	<±0,5 mm	1540 kg	Boden, Decke/ Floor, ceiling/ Au sol, au plafond
KR 100 P	100 kg	50 kg	3500	6	<±0,5 mm	1545 kg	Boden, Decke/ Floor, ceiling/ Au sol, au plafond
KR 125 S/1	125 kg	80 kg	2550	6	<±0,3 mm	1450 kg	Boden/ Floor/ Au sol
KR 125 K	125 kg	125 kg	3100	6	<±0,3 mm	1670 kg	Boden/ Floor/ Au sol
KR 250	250 kg	Modul-Baukasten aus Lineareinheiten Modular system of linear units Solutions modulaires d'unités linéaires			<±0,2 mm	Hubabhängig/ Dependent on travel/ Suivant longueurs de course	Boden, Decke/ Floor, ceiling/ Au sol, au plafond
KL 1500	1500 kg						

Roboter-Sonderbauformen auf Anfrage / Special robot designs available upon request / Types de robots spéciaux disponibles :

KR/KL-Typen mit Robotersteuerung KR C1 / KR/KL models with robot controller KR C1 / Types KR/KL avec commande de robot

<sup>1)</sup> Gültig für Standardausführung / Valid for standard version / Valable pour l'exécution standard<sup>2)</sup> Angaben in mm, bezogen auf Schnittpunkt Achsen 4/5 / Specifications in mm, referred to intersection of axes 4 and 5 / Spécifications en mm, par rapport

## C.4 Industrieroboter der Firma REIS



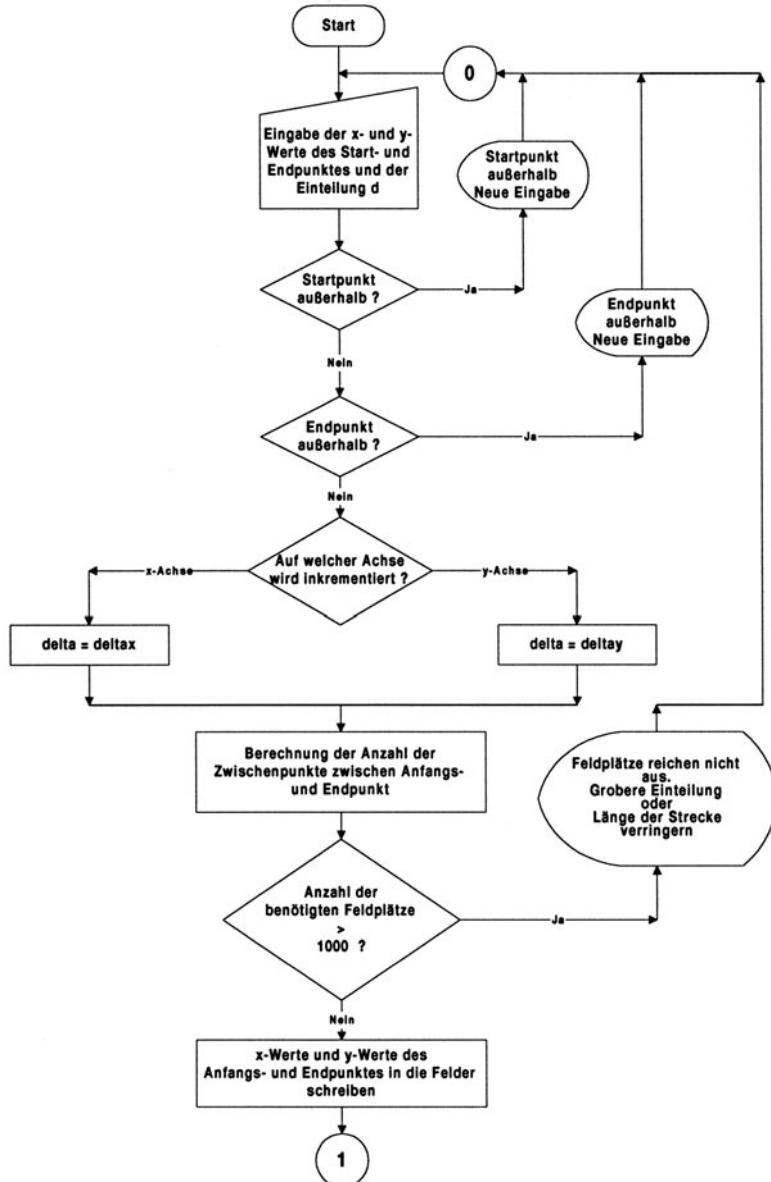
**Bild C-6** Roboter RH130 (Reis)

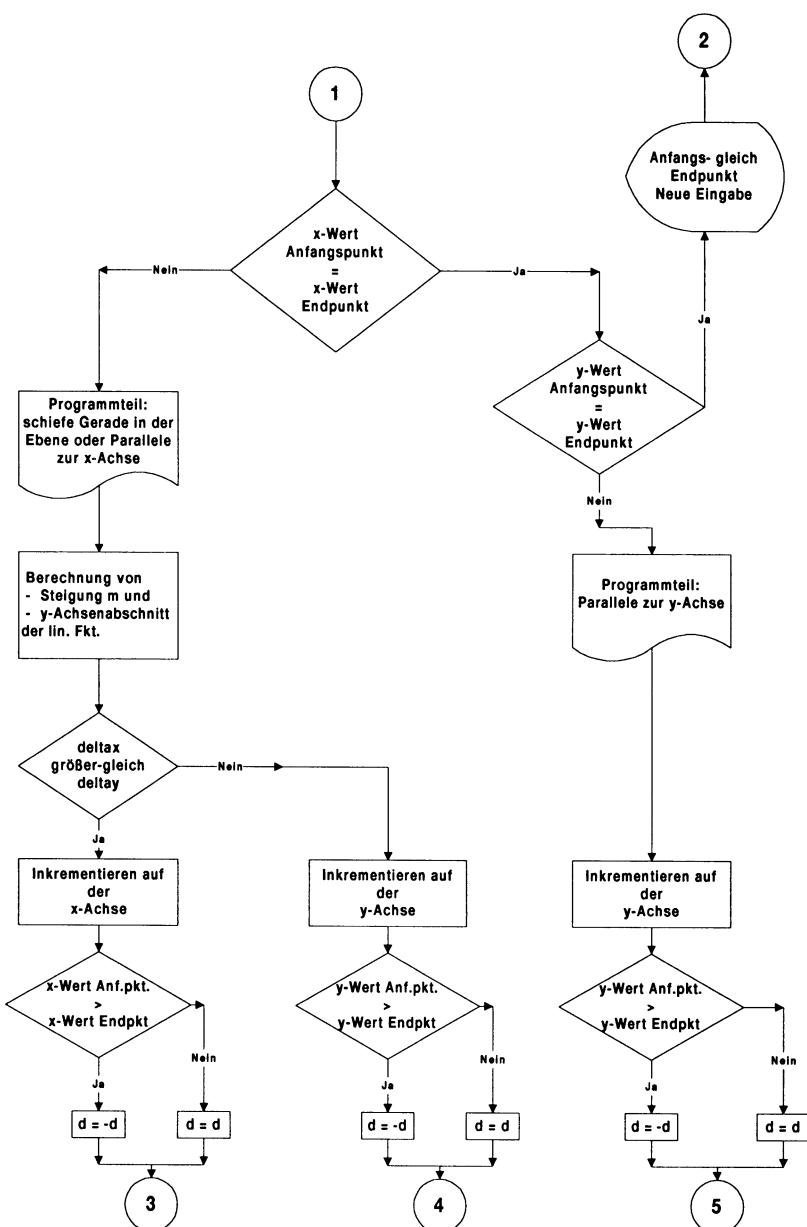
**Tabelle C-5** Technische Daten RH130 (Reis)

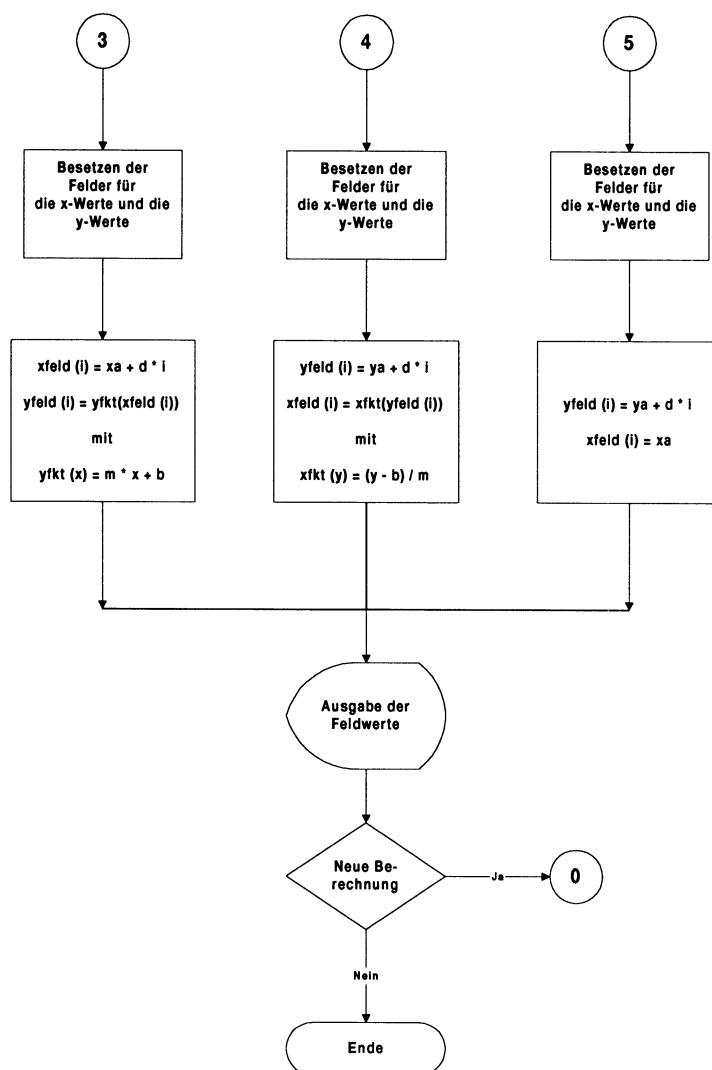
<p><b>VORTEILE:</b></p> <ul style="list-style-type: none"> <li>- FEM-optimierte Konstruktion, modular aufgebaut</li> <li>- Hohe Steifigkeit</li> <li>- AC-Servomotoren</li> <li>- Alle Achsen gebremst</li> <li>- Servicefreundlich, Bauteilaustausch in Minuten</li> <li>- Absolutes Resolver-Wegmeßsystem</li> <li>- Gekappte Antriebe</li> </ul>		<p><b>REIS-ROBOTstar IV:</b></p> <p>Wichtige Steuerungsfunktionen:</p> <ul style="list-style-type: none"> <li>- Automatische Programmumpassung</li> <li>- Fehlertolerantes Wegmeßsystem</li> <li>- Programmierbares Regelverhalten</li> <li>- Mehrachsen-Transformation</li> <li>- Werkstückinterpolation</li> <li>- Lichtbogenpendelsensor</li> <li>- Automatische Hand-Werkzeugvermessung</li> <li>- Prozeßdatenerfassung</li> <li>- Online-Parameteroptimierung</li> <li>- Online-Positionskorrektur</li> <li>- Online-Geschwindigkeitskorrektur</li> <li>und viele weitere wichtige Funktionen</li> </ul> <p><b>Hardwarekonfiguration:</b></p> <ul style="list-style-type: none"> <li>- Modulares Multiprozessorsystem mit VME-Bus</li> <li>- Rechenleistung erweiterbar</li> </ul> <p>Ansteuerbare Achsen: 6 bis max. 18</p> <p>Speicherkapazität: 128 KB bis 2 MB/ 3500 bis 8000 Raumpunkte</p> <p>Speicher batteriegepuffert</p> <p><b>Schnittstellen:</b></p> <ul style="list-style-type: none"> <li>Binäre Ein-/Ausgänge: 16 bis max. 128</li> <li>Analog-Ein-/Ausgänge: 4 bis 8 erweiterbar</li> <li>Serialle Schnittstellen: RS 232/RS 422</li> <li>Sensorschnittstellen: für Laserscanner, Bildverarbeitungssysteme, Analogsensoren und Suchsensoren</li> </ul> <p><b>Archivierung:</b></p> <ul style="list-style-type: none"> <li>3,5" Diskette, DNC-Rechner, Offline-PC</li> <li>Druckeranschluß: Option</li> </ul> <p><b>Bewegungsarten:</b></p> <ul style="list-style-type: none"> <li>PTP (Punkt zu Punkt). Synchrone Achseninterpolation</li> <li>CP-Lineal-Zirkular: Geraden- und Kreisinterpolation mit wählbaren Arten zur kontinuierlichen Nachstellung der Werkzeugorientierung</li> <li>CP-Betrieb mit Zusatzachsentransformation bis zu 18 Achsen</li> </ul> <p><b>Programmierung:</b> Teach-In-Koordinatensysteme: ansteuerbar über Verfahrtasten oder Joystick; Achskoordinaten, kartesische World-Werkzeug- und Drehtischkoordinaten</p> <p><b>Programmierverfahren:</b> Teach-In am PHG, Numerische Eingabe, Offline-Programmierung am PC wahlweise unter Verwendung von CAD-Daten, grafische Offline-Programmierung und Simulation mit ROBCAD.</p>
Geschwindigkeiten		RH130
Achse 1	90°/s	
Achse 2	800 mm/s	
Achse 3	90°/s	
Achse 4	140°/s	
Achse 5	130°/s	
Achse 6	215°/s	
Maximallast		130 kg
Arbeitsraum in mm	A B C D E F	600 1500 (2000*/2500*) 3565 (4065*/4565*) 5194 5700 2350
* Option		
Bewegungsmöglichkeiten	A1 A2 A3 A4 A5 A6	330° 1500 mm (2000*/2500*) 280° 360° 240° 720°
* Option		
Wiederholgenauigkeit		± 0,2 mm
Elektrischer Anschlußwert		11 kVA
Gewicht ohne Schaltschrank		2820 kg

## D Beispielprogramm Geradeninterpolation

### D.1 Programmablaufplan







## D.2 FORTRAN-Programm

```
c      FORTRAN-Programm zur Geradeninterpolation
c
c      von Jörg Bartenschlager
c      19.05.1996
c
c          common /block1/rm,b
c          dimension xfeld(1000)
c          dimension yfeld(1000)
c
c          do 11, i = 1, 1000
c          xfeld(i) = 0.
11      yfeld(i) = 0.
c
c      Sprungmarke für Programmbeginn
c      nach Fehlern oder Programmende
*****  

c
1      continue
c
c      Eingabe des Start- und Endpunktes
c      und des Inkrementes d
*****  

call leer
write(*,*)'Eingabe des x-Wertes des
      Startpunktes'
read(*,*)xa
write(*,*)'Eingabe des y-Wertes des
      Startpunktes'
read(*,*)ya
write(*,*)'Eingabe des x-Wertes des
      Endpunktes'
read(*,*)xe
write(*,*)'Eingabe des y-Wertes des
      Endpunktes'
read(*,*)ye
write(*,*)'Eingabe des Inkrementes d'
read(*,*)d
```

```

c Überprüfung, ob Start- und Endpunkt im
richtigen Intervall zwischen -100 und 100
für die x- und die y-Achse
c ****
c
if(xa.le.100.or.xa.ge.-100.or.ya.le.100.or.ya.ge.-
100) goto 2
    call leer
    write(*,*)"Ihr eingegebener Startpunkt liegt außer-
halb"
    write(*,*)"Bitte neue Eingabe"
    call weiter
    goto 1
2 continue
if(xe.le.100.or.xe.ge.-100.or.ye.le.100.or.ye.ge.-
100) goto 3
    call leer
    write(*,*)"Ihr eingegebener Endpunkt liegt außer-
halb"
    write(*,*)"Bitte neue Eingabe"
    call weiter
    goto 1
3 continue
c
c Überprüfung, ob auf der x-Achse oder der y-Achse
gerechnet wird
c d.h. Berechnung von deltax und deltay
c ****
deltax = abs (xe - xa)
deltay = abs (ye - ya)
if (deltax.ge.deltay) then
    delta = deltax
else
    delta = deltay
end if
c
c Berechnung der Anzahl der Zwischenpunkte
zwischen Endpunkt Pe und Anfangspunkt Pa
c ****
rnum1 = (delta / d) - 1.
rnum2 = aint (rnum1)
if (rnum1.gt.rnum2) then
    inum = rnum2 + 1
else
    inum = rnum1
end if

```

```
c      Überprüfung, ob die Feldanzahl von
c      1000 ausreicht
c      ****
c
c      if(inum.lt.999 ) goto 200
c      call leer
c      write(*,*)'Die Feldanzahl, die benötigt wird, ist'
c      write(*,*)'größer als die im Programm vorgesehene'
c      write(*,*)'von 1000'
c      write(*,*)'Deshalb neue Eingabe'
c      call weiter
c      goto 1
c
c      Ausgabe der Anzahl der Stützpunkte
c      zwischen Anfangs- und Endpunkt
c      ****
c
c      200   continue
c      call leer
c      write(*,*)'Die Anzahl der Stützpunkte, die'
c      write(*,*)'zwischen dem Anfangs- und Endpunkt lie-
gen, '
c      write(*,*)'beträgt :',inum
c      call weiter
c
c      x- und y-Werte des Anfangs- und des Endpunktes
c      in die jeweiligen Felder schreiben
c      ****
c
c      xfeld(0) = xa
c      yfeld(0) = ya
c      xfeld(inum+1) = xe
c      yfeld(inum+1) = ye
c
c      Abfrage ob a) schiefe Gerade oder b) Parallele
c      zur y-Achse
c
c      ****
c
c      if(xa.eq.xe) goto 4
c
```

```

C      ****
C      PROGRAMMTEIL:   a)    SCHIEFE GERADE IN DER EBENE
C                      ODER PARALLEL ZUR X-ACHSE
C      ****
C
C      Berechnung der Steigung m und des y-
Achsenabschnittes
C
***** ****
C
C      rm = (ye - ya) / (xe - xa)
C      b = ya - xa * rm
C
C      Überprüfen, ob deltax größer als deltay
C      ****
C
C      if (deltay.le.deltax) goto 22
C      if (ya.gt.ye) then
C          d = -1 * d
C          else
C              d = d
C          end if
C
C      Auffüllen der Felder für die x- und die y-Werte
C      für deltay größer als deltax
C      ****
C
C      izaehl = 1
21      if (izaehl.gt.inum) goto 23
C          yfeld(izaehl) = ya + d *izaehl
C          xfeld(izaehl) = xfkt(yfeld(izaehl))
C          izaehl = izaehl + 1
C          goto 21
C      Auffüllen der Felder für die x- und die y-Werte
C      für deltax größer als deltay
C      ****
C
C      22      if (xa.gt.xe) then
C          d = -1 * d
C          else
C              d = d
C          end if
C          izaehl = 1
6       if (izaehl.gt.inum) goto 23
C          xfeld(izaehl) = xa + d * izaehl
C          yfeld(izaehl) = yfkt(xfeld(izaehl))
C          izaehl = izaehl + 1
C          goto 6
4       continue

```

```
c      Überprüfung, Anfangs- gleich Endpunkt
c      ****
c
c      if (ya.ne.ye) goto 100
c      write(*,*)'Anfangspunkt gleich Endpunkt'
c      write(*,*)'Neue Eingabe'
c      goto 1
c
c      ****
c      PROGRAMMTEIL:    b)  PARALLELE ZUR Y-ACHSE
c      ****
c
100     continue
      if (ya.gt.ye) then
      d = -1 * d
      else
      d = d
      end if
      izaehl = 1
30      if (izaehl.gt.inum) goto 23
      yfeld(izaehl) = ya + d * izaehl
      xfeld(izaehl) = xa
      izaehl = izaehl + 1
      goto 30
c
c      Sprungmarke zum Zusammenführen der Programmteile
c      ****
c
23     continue
c
c      Ausgabe der Feldwerte
c      ****
c
      do 10, i = 0,inum + 1
      write(*,*)i,'   x = ',xfeld(i),i,'   y = ',yfeld(i)
10      continue
c
c      Neue Berechnung oder Programmende ?
c      ****
c
      call leer
      write(*,*)'Wollen Sie eine neue '
      write(*,*)'Berechnung oder wollen Sie'
      write(*,*)'das Programm beenden?'
      write(*,*)
      write(*,*)'Neue Berechnung --> 1 eingeben'
      write(*,*)'Programmende      --> 2 eingeben'
      read(*,*)iende
      if(iende.eq.1) goto 1
```

```
C*****  
C  
C      ENDE DES HAUPTPROGRAMMES  
C  
C*****  
      end  
C=====  
C  
C      UNTERPROGRAMME  
C  
C=====  
C  
C      Subroutine Leerzeilen  
C      -----  
C  
      subroutine leer  
      write(*,*)  
      write(*,*)  
      write(*,*)  
      end  
C  
C      Subroutine Weiterschalten  
C      -----  
C  
      subroutine weiter  
1      continue  
      write(*,*)"Weiter ----> 1"  
      read(*,*) i  
      if(i.ne.1) goto 1  
      end  
C  
C      Function zur linearen Funktion:  y = m * x + b  
C      -----  
C  
      function yfkt(x)  
      common /block1/rm,b  
      yfkt = rm * x + b  
      end  
C  
C      Function zur Umkehrfunktion der lin. Fkt. x=(y-b)/m  
C      -----  
C  
      function xfkt (y)  
      common /block1/rm,b  
      xfkt = (y -b) / rm  
      end
```

# Literaturverzeichnis

- ABB Industrietechnik AG*: Firmenprospekte. Lampertheim
- ADIRO*: Grundlagen der Robotertechnik. Esslingen o.J.
- Altenhein, A.*: Einsatzplanung. In: Warnecke, H.-J.; Schraft, R. D.: Industrieroboter. Handbuch für Industrie und Wissenschaft. Heidelberg u.a. 1990
- Arbeitsgemeinschaft der Eisen- und Metall-Berufsgenossenschaft*, Fachausschuß Eisen und Metall II, Mainz, Sicherer Betrieb von Industrierobotern. Mainz o.Z.
- Bartelt, R.; Burkhardt, G.*: Mit Vollgas in die Kurve. Neuartiges Bedien- und Programmierkonzept. Roboter-Zeitschrift für Automation 1994(3)
- Baumer electric GmbH*: Firmenprospekte. Friedberg
- Baumüller GmbH*: Firmenprospekte. Nürnberg
- Bautz*: Firmenprospekte. Weiterstadt
- Beuthner, A.*: Kontrollierte Mechanik. Greifersysteme für jeden Anwendungsfall. Robotertechnik, Sonderpublikation 1993, Verlag moderne Industrie Landsberg
- Blume, Ch.; Jacob, W.*: Programmiersprachen für Industrieroboter, Würzburg 1983
- Bosch (Hrsg.)*: Flexible Automation. Produkt-Datenblatt, Druckschrift-Nr. 3842394271 IA, Stuttgart
- Bosch (Hrsg.)*: Handbuch Flexible Automation. Stuttgart o.J.
- Breuer, S.; Scharf, S.; Thorausch, M.*: COSIMIR, Softwarepaket zur zellenorientierten Simulation und Programmierung von Industrierobotern. FESTO Didactic KG, Esslingen 1993
- Brodbeck, B.*: Untersuchung des Arbeitsverhaltens programmierbarer Handhabungsgeräte. Mainz 1979
- Brodbeck, B.; Schiele, G.*: Prüfstand für Industrieroboter. Forschungsbericht der ARGE Handhabungssysteme. o.O. 1990
- Bundesanstalt für Arbeitsschutz*: Arbeitsschutz und Roboter. Dortmund 1993
- Capek, K.*: Rossum's Universal Robots. Berlin o.J.
- Desoyer, K.; Kopacek, P.; Troch, I.*: Industrieroboter und Handhabungsgeräte. Aufbau, Dynamik, Steuerung, Regelung und Einsatz. München 1985
- Dreher, H.; Krüll, G.*: Alles im Griff. Planung und Auswahl von Greiferwechselsystemen. Roboter-Zeitschrift für Automation, Sonderpublikation 1993
- DÜRR GmbH*, Produktbereich Automation + Fördertechnik: Firmenprospekte
- EPSON Deutschland GmbH*: Firmenprospekte
- EUROBTEC*: PSI-Handbuch. Programmiersystem für Industrieroboter. Fürth 1991
- Europäische Normen*:
- |                               |   |
|-------------------------------|---|
| EN 292-1, 1991                | Sicherheit von Maschinen. Grundbegriffe, allgemeine Gestaltungselemente. Teil 1: Grundsätzliche Terminologie, Methodik. o.O. 1991 |
| EN (DIN) 775, ISO 10218, 1993 | Industrieroboter Sicherheit. Berlin 1993  |
- ExpertTeam Sim Tec*: FACTOR, zeitdynamische Simulation für die effiziente Produktion und Logistik. Firmenprospekte und Seminarunterlagen. Düsseldorf 1994
- Fachausschuß „Eisen und Metall“ II*: Prüfgrundsatz Industrieroboter. Mainz 1988
- Fachgemeinschaft Montage-Handhaben-Industrieroboter* im Verband Deutscher Maschinen- und Anlagenbau e.V. (VDMA): Leitfaden Handhaben-Montage-Industrieroboter. Mitgliederverzeichnis. Frankfurt am Main 1994
- Fachgemeinschaft Montage-Handhabung-Industrieroboter (MHI)*: Portrait der Branche 1993/94. Frankfurt am Main April 1994
- FANUC Robotics Deutschland GmbH*: Firmenprospekte
- FESTO Didaktik KG*: Grundlagen der Robotertechnik (BP 70). Stuttgart o.J.
- Fezer*: Vakuum-Transport-Technik. Esslingen o.J.
- Fibro*: Firmenprospekt. Fibromanta – Handhabungstechnik Greifer. Hassmersheim o.J.

- Frank, A.:* Schneller Zugriff. Instandhaltungsanweisungen auf Videofilm erleichtern Verständlichkeit. Maschinenmarkt 1994(34)
- Gellert, D.; Wahl, G.:* Industrie-Roboter in Industrie und Praxis. München 1985
- Götz, E.:* Automatisierung von Arbeitsmaschinen und Produktionsleittechnik, Teil 1. Technische Hochschule Darmstadt Institut für Produktionstechnik und spanende Werkzeugmaschinen. Ausgabe 1991
- Götz, E.:* Automatisierung von Arbeitsmaschinen und Produktionsleittechnik, Teil 2. Strecken, Messung, Antriebe, Regelungen. Technische Hochschule Darmstadt Institut für Produktionstechnik und spanende Werkzeugmaschinen. Ausgabe 1992
- Grube, G.; Hoppe, M.:* Erkennen, begreifen, handeln. Sensordatenverarbeitung beim Schleifen. Roboter-Zeitschrift für Automation 1993(1)
- Heidenhain, J. (Hrsg.):* Digitale Längen- und Winkelmeßtechnik. Landsberg 1989
- Heidenhain:* Firmenprospekt: Drehgeber. Traunreut 1993
- Heinemann, H.:* Einführung in die Industrieroboter-Technik. Allgemeine Einsatzplanung – von der Idee bis zur Realisierung. Essen 1986
- Hengesbach, K. u.a.:* Fachwissen für Industriemechaniker Fachstufe 2. Köln u.a. 1990
- Hesse, S.:* Atlas der modernen Handhabungstechnik. Darmstadt 1992
- Hübner Elektromaschinen AG:* Firmenprospekt. Berlin
- ifm elektronik GmbH:* Firmenprospekte. Essen
- Interelectric AG:* Firmenprospekte. Sachseln/Schweiz
- IPR, Intelligente Peripherie für Roboter GmbH.:* Firmenprospekt Kollisionsschutz. Schwaigern o.J.
- ISO TR 8373: Manipulating industrial robots – Vocabulary.* o.O. 1993
- Kämpfer, S.:* Roboter: die elektronische Hand des Menschen. 2. Aufl., Düsseldorf 1985
- Kämpfer, S.:* Robotern geht die Arbeit leicht von der Hand. VDI-Nachr. 47/1990
- Kemmer, K.H.:* Arbeitssicherheit beim Betrieb von Industrierobotern. Sonderdruck aus „Moderne Unfallverhütung“, Band 33. Essen 1989
- Kemmer, K.H.:* Arbeitssicherheit beim Einsatz von Industrieroboters. Sonderdruck aus Die BG 9/1984. Bielefeld 1994
- Kief, H. B.:* NC/CNC-Handbuch 93/94. München 1993
- Krause, P. C.; Wasynczuk, O.:* Electromechanical Motion Devices. o.O. o.J.
- KUKA Schweißanlagen und Roboter GmbH:* Firmenprospekt. Augsburg o.J.
- Lehrl, W.:* Bericht über das Gespräch mit Prof. Dnelen (Institut für Werkzeugmaschinen und Fertigungstechnik der TU Berlin) zum Stand der Robotertechnik. In: 5. Tagung des überregionalen Arbeitskreises Automatisierungstechnik. Institut Technik und Bildung, Abt. Berufspädagogik/Elekrotechnik, Universität Berlin. Bremen, Berlin 1991
- Maiß, G. (Verf.):* Bosch Schwenkarmroboter. Turboscaria mit RS 82. Bosch-Veröffentlichung, o.O. o.J.
- Morgan, C.:* Robots – Planning and Implementation. Berlin u.a. 1984
- Naval, M.:* Roboter-Praxis. Aufbau, Funktion und Einsatz von Industrierobotern. Würzburg 1989
- Nicolaisen, P.:* Arbeitsschutz. In: Warnecke, H.-J.; Schraft, R. D.: Industrie Roboter Katalog 1987. Darstellung der auf dem deutschen Markt angebotenen Industrie-Roboter-Systeme. Mainz 1987
- o. V.:* Werkzeuge im inneren Labyrinth. Pendelachse löst aufwendige Konstruktion zum Toleranzausgleich ab. Roboter-Zeitschrift für Automation. 1993(4)
- o.V.:* Kuriositäten und Exoten. Merk- und denkwürdige Roboteranwendungen. Roboter-Zeitschrift für Automation. 1993(4)
- Raab, H. H.:* Handbuch Industrieroboter, 2. Aufl. Braunschweig/Berlin 1986
- REIS GMBH & KG:* Schulungsunterlagen. Obernburg o.J.
- Schanz, G.:* Sensoren. o.O. 1986
- Schmalz:* Firmenprospekt. Glatten o.J.
- Schnell, G.:* Sensoren in der Automatisierungstechnik, 2. Aufl. Vieweg-Verlag, Wiesbaden 1993
- Schraft, R.:* Eiserne Helfer. Serviceroboter – kein Märchen aus künftigen Tagen. Roboter-Zeitschrift für Automation 1993(1)

- Schunk*: Firmenprospekt Greifsysteme. Lauffen 1993
- Schunk*: Spann- und Greiftechnik – Greifsysteme (CD-ROM). Lauffen 1996
- Schweizer, M.*: Roboterbranche mit Rückenwind. Roboter-Zeitschrift für Automation 1993(4)
- Schweizer, M.; Dreher, H.*: Die Qual der Wahl. Auswahlkriterien für Greifersysteme im Montagebereich. Roboter-Zeitschrift für Automation 1993(2)
- Schwinn, W.*: Grundlagen der Roboterkinematik. Schmalbach 1992
- Sensor report*. Zeitschrift für Sensorik – Meßtechnik – Automatisierung
- Slatter, R.*: So genau wie vielseitig. Präzisionsgetriebe in Industrierobotern. Robotertechnik Sonderpublikation 1993, Verlag moderne Industrie Landsberg
- Sommer-Automatic*: Firmenprospekt: Automatisierung aus einer Hand.
- Stegmann*: Firmeninformationen, Datenblatt 88902. Donaueschingen o.J.
- Stegmann*: Firmenprospekt. Donaueschingen o.J.
- Stegmann*: Technische Information. Sychron serielles Interface für absolute Winkelcodierer. Donaueschingen 1991
- T. W.*: Der Traum vom hilfreichen Roboter. Ein Gespräch mit Joseph Engelberger. Roboter-Zeitschrift für Automation 1993(6)
- Törnig, W.*: Numerische Mathematik für Ingenieure und Physiker. Band 2: Eigenwertprobleme und numerische Methoden der Analysis. Berlin, Heidelberg, New York 1979
- VDI-Richtlinien*:
- |                     |  |
|---------------------|--|
| 2221, 1986          | Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte  |
| 2740, Bl. 1E, 07.91 | Mechanische Einrichtungen in der Automatisierungstechnik; Greifer für Handhabungsgeräte und Industrieroboter     |
| 2854, 1991          | Sicherheitstechnische Anforderungen an automatisierte Fertigungssysteme  |
| 2853, 07.87         | Sicherheitstechnische Anforderungen an Bau, Ausrüstung und Betrieb von Industrierobotern                         |
| 2860, 05.90         | Montage und Handhabungstechnik; Handhabungsfunktionen, Handhabungseinrichtungen; Begriffe, Definitionen, Symbole |
| 2861 Bl. 1, 06.88   | Montage- und Handhabungstechnik; Kenngrößen für Industrieroboter; Achsbezeichnungen                              |
| 2861 Bl. 2, 05.88   | Montage- und Handhabungstechnik; Kenngrößen für Industrieroboter; Einsatzspezifische Kenngrößen                  |
| 2861 Bl. 3, 05.88   | Montage- und Handhabungstechnik; Kenngrößen für Industrieroboter; Prüfung der Kenngrößen                         |
- Volmer, J.*: Industrieroboterentwicklung. 2. Auflage, Heidelberg 1985
- Wagner, F.; Pöllath, K.; Schumacher, H.*: Technik und Programmierung von Robotern. Hamburg 1986
- Warnecke, H.-J.; Schraft, R. D.*: Industrie Roboter Katalog 1987. Darstellung der auf dem deutschen Markt angebotenen Industrie-Roboter-Systemen. Mainz 1987
- Warnecke, H.-J.; Schraft, R. D.*: Industrieroboter. Handbuch für Industrie und Wissenschaft. Heidelberg u.a. 1990
- Weck, M.*: Werkzeugmaschinen, Band 1: Maschinenarten, Bauformen und Anwendungen. 3. Aufl. Düsseldorf 1988
- Weck, M.*: Werkzeugmaschinen, Band 3. Automatisierungs- und Steuerungstechnik. 3. Aufl. Düsseldorf 1989
- Zühlke, D.*: Handhabungs-, Transport- und Lagertechnik. Skript 1. Lehrstuhl für Produktionsautomatisierung der Universität Kaiserslautern. o.Z.
- Zühlke, D.*: Handhabungstechnik und Industrieroboter. Skript 2. Lehrstuhl für Produktionsautomatisierung der Universität Kaiserslautern. o.Z.

# Sachwortverzeichnis

## A

Achse  
 Haupt- 22  
 Länge 297  
 letzte 301  
 Neben- 22  
 Verdrillung 297

Akzeptanzbereich 210

Akzeptanzzone 203

Alpha-5/0°-Situation 134

Analysator 209

Anfahrkurve *s. Hysteresekurve*

Ansteuerung

- Block- 62–63
- Rechteck- 62–63
- Sinus- 64
- Vergleich 64–65

Antriebssystem 20

Arbeitsraum 23

- Haupt- 23
- Neben- 23

Aufbau

- kinematischer 288

Ausschwingzeit 26

## B

Backside 133

Bahn

- Linearisierung 141

Bahnabstand 29

Bahnfehler

- dynamischer 117

Bahngeschwindigkeit 26

Bahn-Orientierungsabweichung 29

Bahn-Orientierungsstrebereich 29

Bahnradiusdifferenz 29

Bahnsteuerung 116, 141, 315

- Anforderungen 333

- Quasi- 314

Bahnstrebereich 29

Bahnungenauigkeiten 118

BAPS 142, 147, 227, 232, 237, 239

- Elemente 233

BAPS2 237

Bedämpfungsfahne 160, 161

Bedienfeld 113

Beschleunigung

- resultierende 26

Betauung 199

Betriebsspannung

- gängige 171

Bewegungseinheiten 19

Bewegungseinrichtung

feste Hauptfunktion 10  
 manuell gesteuerte 11  
 programmgesteuerte 11  
 variable Hauptfunktion 10  
 Bewegungsraum 23, 255  
 fester 23  
 variabler 23

## C

CIM 3

Code

- BCD- 79
- Binär- 79
- dekadischer Gray-Excess- 80
- gekappter Gray- 80
- Gray- 76, 79
- Gray-Excess- 80

Codearten 79–81

Codierer

- Multiturn- 78
- Singleturn- 76

Compiler 112

Computer Integrated Manufacturing *s. CIM*

Continious Path *s. CP*

CP 51

## D

Dämpfungsmoment 57

DH-Parameter 299

Drehgeber

- Code- 75–78
- Genaugigkeit 75
- inkrementaler 71

Drehimpulsgeber 58, 84

Drehmoment

- inneres 57
- Nutz- 57

Drehzahl

- Messung 197
- Überwachung 197

Dunkelschaltung 207

## E

Eckenfehler 29

Effektor 85

Einsatzplanung 241

Einschaltkurve 180

Ellbogenstellung 132

Ellbow down 132

Ellbow up 132

Empfangskeule 203

Encoder 58

- Inkremental- 62

Endeffektor 19

- F**
- Fläche  
  aktive 158
- Flansch  
  Aufgaben 87
- FMSsoft 247
- Freiheitsgrad 6
- Frontside 133
- Füllstand  
  Kontrolle 195, 197
- Meßtechnik 197
- Funktion  
  Eineindeutigkeit 124
- Funktionsreserve 204–7
- G**
- Gefahrenbereich 250, 255  
  Sicherheitsabstand 272–73
- Gefahrenraum 23
- Gefahrstelle  
  Mindestabstand 272
- Gegenspannung 57
- Gelenk 287
- Genauigkeit  
  Austausch- 29  
  Programmier- 29  
  Referier- 29  
  Wiederhol- 27
- Geradeninterpolation  
  Beispielprogramm 145
- Geschwindigkeit  
  resultierende 26
- Geschwindigkeitsverstärkung 116
- Gierbewegung 331
- Greifer  
  Adhäsions- 98  
  Austauschbarkeit 88  
  Dauermagnet- 100  
  Doppel- 96  
  Drei-Finger- 98  
  Einfach- 95  
  Einzel- 95  
  Elektromagnet- 100  
  Gummi- 105  
  Gummifinger- 107  
  Hauptaufgaben 86  
  Loch- 98, 106  
  Magnet- 98  
  mechanischer 90  
  Mehrfach- 95  
  Parallel- 101  
  Revolver- 96  
  Röhren- 105  
  Saug- 98  
  Sechspunkt- 102  
  Spreizfinger- 107  
  Stoff- 104  
  Subsysteme 86
- Winkel- 101
- Zapfen- 98
- zusätzliche Funktionen 86
- Zweifach- 95
- Zwei-Finger- 98
- Greifereinrichtung  
  Anpassung 108
- Greifsystem 85  
  mechanisches 88
- Größen  
  statistische 28
- Guckschaltung 176
- H**
- Handhaben  
  automatisches 5  
  Definition nach VDI 5
- Handprogrammiergerät 225
- Hellschaltung 207
- Hintergrundausblendung  
  Triangulationsverfahren 215  
  Winkellichtverfahren 215
- home Impuls 59
- Hysterese 192  
  Schalt- 163, 193
- Hysteresekurve 163
- I**
- Inductosyn 82–83
- Industrial Robot Data 230
- Industrieroboter  
  Definition 18
- international protection *s. IP*
- Interpolation  
  Beispiel 323  
  Geraden- 143–46  
  Kreis- 146–47  
  kubische Spline- 329  
  Linear- 143–46  
  Parabel- 151  
  Polynom- 329  
  sonstige 151  
  Spline- 151
- Interpolator 142, 315
- IP 198, 199
- IRDATA-Richtlinie 230
- IRL 230
- K**
- Kantenlänge  
  Korrekturfaktor 166
- Kapazitätsänderung  
  Größe 188
- Kategorie 0-Stop 258
- Kategorie 1-Stop 258
- Kenngrößen 29  
  Genauigkeits- 29
- Kette  
  kinematische 288

- Kinematik 19  
 Kollisionsschutzsystem 92  
 Kompensationselektrode 198, 199  
 Koordinaten  
     Gelenk- 134–35  
     Greifer- 135–36  
     homogene 294  
     kartesische 119–21  
     Kugel- 129–31  
     Maschinen- 134–35  
     Raum- 131–34  
     Transformation 116, 119  
     Welt- 131–34  
     Zylinder- 121–22  
 Koordinatensystem  
     ebenes kartesisches 121  
     körpereigenes 6  
     linksdrehendes 120  
     rechtshändiges 120  
     Ursprung des Basis- 22  
 Korrekturfaktor 166  
     Reichweite 205  
     Umweltbedingungen 205  
 Kurzschlußschutz 173–75  
     einrastender 174  
     getakteter 175  
     Nachteile 175
- L**  
 Lage 118  
     Festlegung 119  
 Lageregelung 52  
 Laser-Scanner  
     tastender 272  
 Last  
     Effektor- 24  
     Greifer- 24  
     Maximal- 25  
     Nenn- 24  
     Nutz- 24  
     Werkzeug- 24  
 Leitfähigkeit 161  
 Lichtgitter  
     Sicherheits- 272  
 Lichtschrank  
     Sicherheits- 272  
 Lichtvorhang  
     Sicherheits- 272  
 Losbrechmoment 57
- M**  
 Magnet 54  
 Manipulator  
     fernsteuerter 12  
     Manual 18  
 Manual Manipulator 18  
 Maschine  
     Sicherheit 254
- Material  
     Korrekturfaktor 166  
 Materialdicke  
     Korrekturfaktor 167  
 Meridian 129  
 Motografie 118  
 Motor  
     Anschlußspannung 57  
     Schleifringläufer- 54  
 MRL 237
- N**  
 Nickbewegung 331  
 Normmeßplatte 165  
 NOT-AUS 284  
 NOT-AUS-Funktion 258  
 Nutzlast  
     maximale 25  
     zusätzliche 25
- O**  
 Oberfläche  
     Betauung 199  
 Oberschwingungsgehalt 171  
 Optik  
     fokussierte 214  
 Ordnungszustand 7  
 Orientierung 118  
 Orientierungsgrad 6
- P**  
 PAP 127  
 Pendeln 147–49  
 Permeabilität 161  
 Permeabilitätszahl 167  
 Permittivitätszahl  
     Tabelle 189  
 PGH 19, 113  
 PHG2000 237  
 Pick&Place-Geräte 13  
 Planungssoftware 247  
 Playback-Verfahren 223  
 Point To Point *s. PTP*  
 Polarkoordinaten 123–29  
 Polpaarzahl 62  
 Polwinkel 129  
 Polygonzugmethode 143  
 Position 118  
 Positionsgrad 6  
 Produktgestaltung 240  
 Programmablaufplan *s. PAP*  
 Programmierhandgerät *s. PGH*  
 Programmierung  
     Ablauf 231  
     textuelle 229–34  
 Programmenteile 232

- PTP 51, 116, 141, 147  
asynchrone Bewegung 327  
asynchrone Steuerung 314  
synchrone Bewegung 328  
synchrone Steuerung 314
- Q**  
Quirlen 331
- R**  
Radialsymmetrie 129  
Raum  
nicht nutzbarer 23  
Raumkurve  
beliebige 142  
Referenzmarke 71  
abstandscodierte 72  
Regeldifferenz 116  
Regelkreis  
Geschwindigkeits- 83  
Lage- 83  
Positions- 83  
Reichweite 218  
Reichweitenkurve 204  
Reproduzierbarkeit 163, 179  
Resolver 81–82  
Restwelligkeit 171  
Risikoabschätzung  
pr EN 954-1 260  
Risikobewertung 255  
Risikograph 260  
Robot  
Fixed Sequence 18  
intelligent 18  
Numerical control 18  
Playback 18  
Variable Sequence 18  
Robot Control 112  
Roboter  
Begriff 16  
Rollbewegung 331
- S**  
Sauggreifer  
Haltekräfte 99  
Schaltabstand 162  
Arbeits- 163  
erreichbarer 179–80  
Nenn- 162, 193  
Nutz- 163  
Real- 162, 193  
Schalter  
Positions- 197  
Schaltfrequenz  
Ermittlung 169  
Schaltpunkt  
Drift 163  
Wiederholgenauigkeit 163  
Schaltpunktdrift 193
- Schaltzone  
aktive 158  
Schleppabstand 116  
Schnittstelle 22  
mechanische 87
- Schritt  
kleinster verfahrbarer 29
- Schutzeinrichtung  
bereichssichernde 275  
optoelektronische 272
- Schutzmaßnahmen  
Auswahl 254
- Schutztüren  
Überwachung 284, 285, 286
- Sendekugel 203
- Sensor 20, 91  
Applikationsgruppen 153  
dunkelschaltender 220  
Durchlicht- 207  
Eigenschaften 156  
Elementar- 156  
Empfindlichkeit 193  
Hall- 62  
hellschaltender 220  
Leistungsaufnahme 172  
Ring- 170  
Schlitz- 170  
Struktur 155  
Zuverlässigkeit 202
- Servomotor  
Anforderungen 52  
Auswahlkriterien 52
- Sicherheit  
bei Inspektion 281  
bei Wartung 282
- Sicherheitseinrichtung 20
- Sicherheitsfunktion  
Rückstellung 259
- Sicherheitskategorie  
Bestimmung 260–61
- Sicherheitskreise  
geforderte Funktionen 258–59
- Sicherheitsschaltgerät 279
- Sicherheitszuhaltung 285
- Skin-Effekt 167
- Statorfrequenz 62
- Stellbefehl 288
- Steuerung 19  
Bahn- 51  
Multipunkt- 314
- Stillsetzen  
gesteuertes 258  
ungesteuertes 258
- Stillstandsremse 60
- Stoff  
diamagnetischer 167  
paramagnetischer 167

- Stop-Funktion 258  
 Strahlungsspektrum  
   Aufteilung 201  
 Streubreite  
   Orientierungs- 29  
   Positions- 29  
 Streuung  
   Exemplar- 179, 193  
   Hysterese- 179  
 Strobe-Signal 79
- T**  
 Tacho  
   Digital- 84  
 Tachogenerator 59, 62, 83  
 Tachometerdynamo 83  
 Taktzeiten 5  
 Tastbereich 216  
 Tastweite 216, 218  
 TCP 23, 51, 117, 118, 131, 136, 143  
 Teachbox 113  
 Teachen 135, 137, 141, 142, 143  
 Teaching-Box 225  
   Funktionen 225  
 Teach-in-Verfahren 223  
 Teachphase 113  
 Teile  
   sicherheitsbezogene 256  
 Temperaturfehler 29  
 Tool Center Point s. *TCP*  
 Transformation  
   inverse kinematische 288  
   kinematische 288  
   Koordinaten- 291  
   Rückwärts- 288  
   Vorwärts- 288  
 Triangulationsverfahren 215  
 Tripelspiegel 209
- Ü**  
 Überlastfestigkeit 176  
 Überlastungsschutz 173–75  
 Überschleifpunkt 148  
 Überschwingfehler 29, 117  
 Überschwingweite 26  
 ULS-Einheit 92  
 Umkehrspanne 29  
 Umschaltpunkt  
   Berechnung 326  
 Unfall  
   roboterspezifischer 250  
 Unfallverhütung 5
- V**  
 Verpolungssicherheit 175–76  
 Verrundungsfehler 117, 118  
 Verschleifen 147–49  
   Bahn- 147  
   Geschwindigkeits- 148
- W**  
 Wegmeßsystem 20  
 Wegmessung  
   inkrementale 74  
 Werkstückhandhabung 30  
 Werkzeug 22  
 Werkzeughandhabung 30  
 Werkzeugwechsler 109  
 Winkelcodierer  
   Absolute 75–78  
 Winkellichtverfahren 215
- Z**  
 Zählungen 197  
 Zeitkonstante  
   elektrische 52  
   mechanische 52, 54  
 Zustand  
   gefährbringender 255  
 Zykluszeit 26

# **Robotik**

Grundwissen für die  
berufliche Bildung

von Stefan Hesse und Günther Seitz

1996. VIII, 171 S. mit 158 Abb.  
(Viewegs Fachbücher der Technik)  
Kart. DM 32,--  
ISBN 3-528-04951-0

*Aus dem Inhalt:* Funktionen und Funktionsträger - Anwendungen  
der Robotertechnik - Aufbau von Industrierobotern - Komponen-  
ten - Sensorik - Programmierung - Arbeitssicherheit

Den Bedürfnissen der beruflichen Ausbildung in der Metalltech-  
nik angepaßt, werden hier alle wesentlichen Inhalte knapp, aber  
leicht verständlich dargestellt. Im Mittelpunkt stehen Technik und  
Anwendung der Roboter.

Stand 04.02.98. Änderungen vorbehalten.  
Erhältlich im Buchhandel oder beim Verlag.

Abraham-Lincoln-Str. 46  
Postfach 1547  
65005 Wiesbaden  
Fax: (06 11) 78 78-4 00  
<http://www.vieweg.de>



# **Atlas der modernen Handhabungstechnik**

von Stefan Hesse

1993. 458 S. im Ringordner. DM 228,--  
ISBN 3-528-04988-X

*Aus dem Inhalt:* Grundlagen der Handhabungstechnik - Objekte der Handhabungstechnik - Handhabungseinrichtungen - Zuführsysteme - Werkstückbereitstellung und Werkstückträger - Schwingfördertechnik - Industrierobotertechnik - Greifertechnik - Handhabung in der Montage - Verkettung - Sensorik - Steuerungen in der Handhabungstechnik - Arbeitssicherheit in der Handhabungstechnik - Handhabungssysteme

Stand 04.02.98. Änderungen vorbehalten.  
Erhältlich im Buchhandel oder beim Verlag.

Abraham-Lincoln-Str. 46  
Postfach 1547  
65005 Wiesbaden  
Fax: (06 11) 78 78-4 00  
<http://www.vieweg.de>

