Fakultät I
Elektro- und Informationstechnik

Fachhochschule
Hannover

Industrieroboter

Kinematik und Programmierung

Vorlesungsumdruck

WS 2011/12

Vers. 2017

Prof. Dr.-Ing. E. Kunze

INHALTSVERZEICHNIS

1 Grundlagen

Der Begriff „Roboter“ geht auf das Theaterstück „Rossum’s universelle Roboter“ des Tschechen Karel Capek aus dem Jahre 1921 zurück, in dem künstliche Menschen im Mittelpunkt stehen. Heute bezeichnet man universell einsetzbare und programmierbare Bewegungsautomaten als Roboter. Als Industrie-Roboter (IR) werden solche Roboter bezeichnet, die zur Automatisierung von Handhabungs-, Montage- und Bearbeitungsaufgaben eingesetzt werden. Dabei ist die Wirtschaftlichkeit der wichtigste Einsatzgrund, aber auch die Qualitätsverbesserung und die Entlastung des Menschen von gesundheitsschädlichen Aufgaben wie z. B. Schweiß- und Lackieraufgaben oder das Tragen von schweren Lasten ist von Bedeutung. Was man sich unter einem Industrie-Roboter vorzustellen hat, erläutert die folgende Definition:

Industrie-Roboter:

> Industrie-Roboter sind universell einsetzbare Bewegungsautomaten mit mehreren Achsen, deren Bewegung hinsichtlich Bewegungsfolge und -wegen bzw. -winkeln frei programmierbar (d. h. ohne mechanische Eingriffe veränderbar) und gegebenenfalls sensorgeführt ist. Sie sind mit Greifer, Werkzeugen oder anderen Fertigungsmitteln ausrüstbar und können Handhabungs- und/oder Fertigungsaufgaben ausführen.

Industrie-Roboter gehören zu den Handhabungsgeräten. Unter Handhabung versteht man die Lagerung, die Positionierung und den Transport von Material im Bereich von Arbeitsplätzen oder Fertigungseinrichtungen. Handhabungsgeräte sind jene Maschinen, die die genannten Vorgänge ausführen.

Handhaben (VDI-Richtlinie 2860):

> Handhaben ist das Schaffen, das definierte Verändern oder vorübergehende Aufrechterhalten einer vorgegebenen räumlichen Anordnung von geometrisch bestimmten Körpern.

Die Bewegungsabläufe von IR werden durch kinetische und kinematische Gesetze beschrieben, die folgendermaßen definiert sind:

Kinematik

> Die Kinematik ist die Lehre von der geometrischen und analytischen Beschreibung der Bewegungszustände von Punkten und Körpern. Kräfte und Momente als Ursachen werden nicht berücksichtigt.

Kinetik

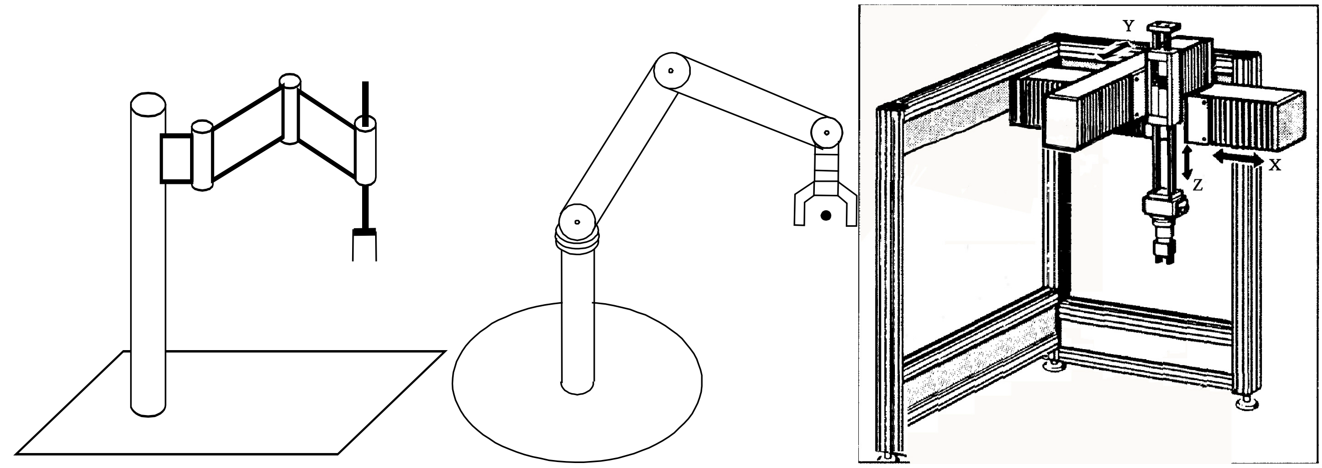
> Die Kinetik untersucht die Bewegung von Massepunkten und Körpern sowie von Systemen daraus als Folge der auf sie wirkenden Kräfte und Momente unter Berücksichtigung der Gesetze der Kinematik

1.1 Aufbau eines Industrie-Roboters

Ein Industrieroboter besteht aus einem kinematischen Arm mit Sensoren und Antrieben sowie aus einer programmierbaren Steuerung. Der kinematische Arm besteht aus mehreren Gliedern, die durch Gelenke miteinander verbunden sind. Diese sind Dreh- oder Schubgelenke mit jeweils einem Freiheitsgrad. Die Glieder bilden eine offene kinematische Kette. Das erste Glied ist mit der Umgebung verbunden, das letzte trägt am Ende einen Effektor (Greifer, Werkzeug, Fertigungsmittel) [1].

Bild 1.1 zeigt die drei Grundtypen der Industrieroboter, den Horizontalknickarm-Roboter, auch SCARA[1] genannt, den Vertikalknickarm-Roboter und den Portalroboter, dessen Bewegungsachsen sich in x-, y- und z-Richtung bewegen.

Die SCARAs sind mit 4 Freiheitsgraden ausgestattet, die Vertikalknickarm-Roboter mit bis zu 6 Freiheitsgraden. Das Portal besitzt 3 Freiheitsgrade und die daran angekoppelbare Hand bis zu 3 weitere Freiheitsgrade. Neben den Grundtypen sind auch noch Mischtypen im Einsatz, die Linear- und Drehbewegungen im Positionierteil kombinieren.



Horizontalknickarm-Roboter (SCARA) Vertikalknickarm-Roboter Portalroboter

Bild 1.1: Typen der Industrieroboter

Die kinematische Konstruktion wird durch Antriebe, Steuerung, und Meßsysteme zu einer funktionsfähigen Einheit ergänzt. Das Zusammenwirken dieser Funktionsgruppen verdeutlicht Bild 1.2. Zum Roboter im engeren Sinn gehören der kinematische Aufbau, die Antriebe, die internen Sensoren und die Steuerung. Der Effektor und die externen Sensoren sind vom Anwendungsfall abhängig.

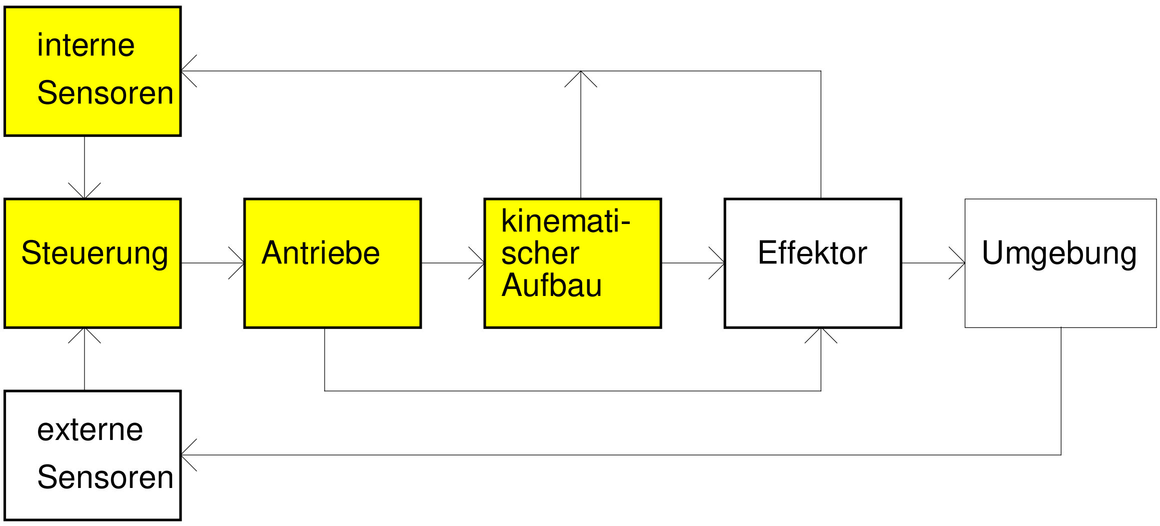


Bild 1.2: Funktionsgruppen eines Industrie-Roboters

[1] *Selective Compliance Assembly Robot Arm*

Die internen Sensoren bestimmen ganz wesentlich die Positioniergenauigkeit des Roboters. Der kinematische Aufbau hat die Aufgabe, den Effektor im Arbeitsraum zu positionieren und zu orientieren, d. h. ihm eine bestimmte Stellung zu geben.

1.2 Definitionen und Begriffe

Nachfolgend werden einige Begriffe erläutert, die zur Beschreibung und Anwendung von Robotern von Bedeutung sind:

Position

Der Ort des Ursprungs eines körpereigenen Koordinatensystems in einem Bezugssystem heißt die *Position* des Körpers.

Der Ort des Ursprungs des Effektor-Koordinatensystems in dem Punkt P eines Bezugssystems heißt die *Position* des Effektors.

Orientierung Stellung (Lage)

Die *Stellung* umfaßt die Position und die Orientierung in einem Punkt P. Sie wird auch gelegentlich als *Lage* bezeichnet.

Stellung (Lage)

Die *Stellung* umfaßt die Position und die Orientierung in einem Punkt P. Sie wird auch gelegentlich als *Lage* bezeichnet.

Die Stellung eines völlig frei im Raum beweglichen Effektors ist durch drei Translationen (x, y, z) und drei Rotationen (α, β, γ) eindeutig bestimmt. Die Stellung kann daher durch den Vektor

$$r = \{x, y, z, \alpha, \beta, \gamma\}^T$$

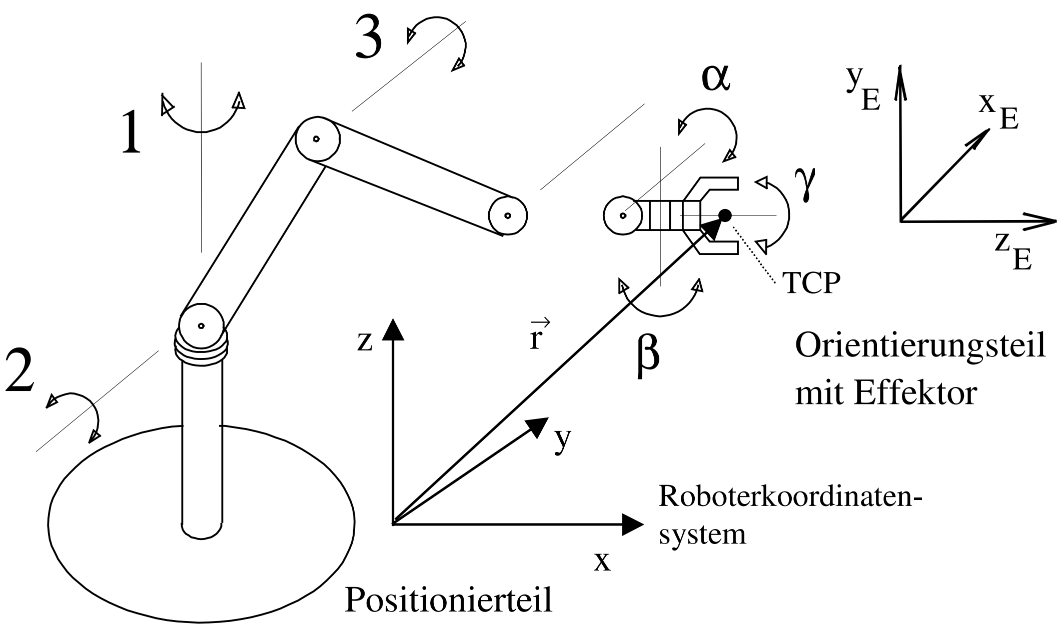


Bild 1.3: Industrie-Roboter mit Positionier- und Orientierungsteil

beschrieben werden. Die Zusammenhänge zeigt Bild 1.3.

Die Winkel zwischen den Achsen eines körpereigenen Koordinatensystems und des Bezugssystems bedeuten die Orientierung des Körpers.

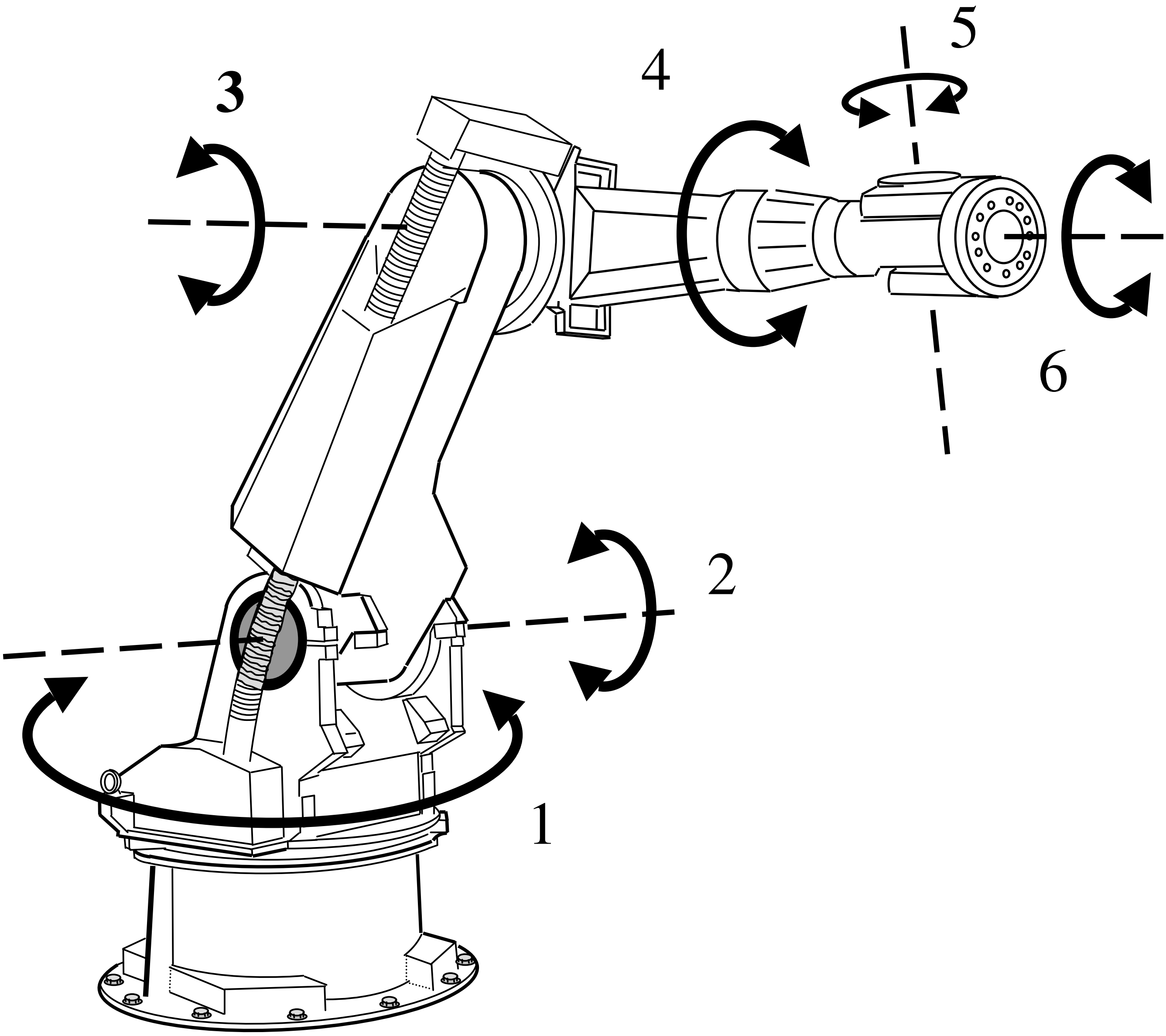


Bild 1.4: Beispiel für einen 6-achsigen Industrie-Roboter

Als *Hauptachsen* bezeichnet man jene 3 Achsen, die für die räumliche Positionierung (x, y, z) erforderlich sind (Positions-Teilkette). Die Orientierung des Effektors (α, β, γ) wird über die *Handachsen* eingestellt (Orientierungs-Teilkette). Zum Anfahren einer Stellung des Effektors müssen also 3 Wegkoordinaten und 3 Winkelkoordinaten unabhängig voneinander eingestellt werden.

Separierbarkeit

IR mit trennbaren Haupt- und Handachsen bezeichnet man als *separierbar*.

Freiheitsgrad

Die Zahl der unabhängig voneinander einstellbaren Koordinaten bezeichnet man als *Freiheitsgrade* des IR.

Das System von Bild 1.3 hat die 6 Freiheitsgrade der Achsen 1, 2, 3 sowie der Drehwinkel α, β und γ. Eine praktische Ausführung davon zeigt Bild 1.4 in Form eines Vertikalknickarm-Roboters mit 6 Achsen. Dieser ist nicht separierbar, da die Handachsen 4 bis 6 integraler Bestandteil der Konstruktion sind.

Die Aufgabe des kinematischen Arms ist es, den Effektor in die gewünschte Stellung zu bringen. Zur Beschreibung der Stellung legt man in den wirksamen Punkt des Effektors, dem TCP (Tool Centre Point), ein Koordinatensystem, dessen Ursprung in die gewünschte Position und dessen Drehlage in die gewünschte Orientierung gebracht wird. Bild 1.5 zeigt einen solchen Effektor in Form eines Greifers. Zwischen den Greiferbacken liegt der TCP und in diesem der Ursprung des Effektorkoordinatensystems mit den Achsen x_E , y_E , z_E. Der Ursprung dieses Koordinatensystems entspricht der Position des Greifers und die Ausrichtung seiner Koordinatenachsen im Bezugssystem x, y, z seiner Orientierung.

Das Effektor-Koordinatensystem wird bei Greifern so angelegt, daß die z-Achse nach vorn gerichtet ist und die y-Achse parallel zur Bewegungsrichtung der Greifbacken verläuft. Die x-Achse ergibt sich dann im Sinne eines Rechtssystems.

Eulerwinkel

Die Orientierung des Effektors wird gelegentlich durch drei nach Euler definierte Winkel φ, ϑ und ψ angegeben [3, 5].

Die Definition geht aus Bild 1.6 hervor. Vom Koordinatensystem S_0 mit dem Index 0 ausgehend wird zuerst um die z_0-Achse mit Winkel φ in das System S_1 gedreht. Danach erfolgt die Drehung um die y_1-Achse mit Winkel ϑ ins System S_2. Die nochmalige Drehung um die z_2-Achse mit Winkel ψ führt schließlich zu S_3.

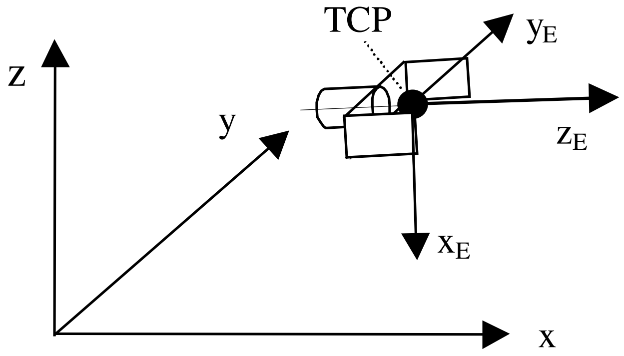


Bild 1.5: Effektor-Koordinatensystem

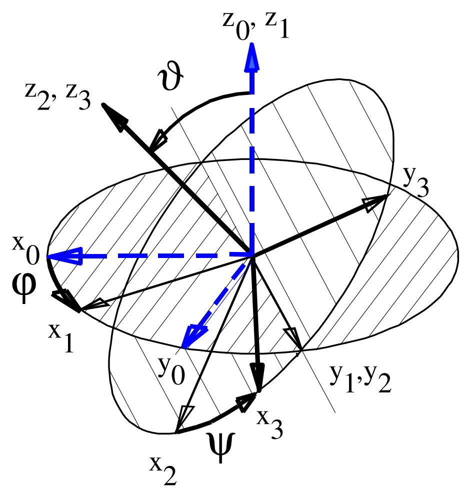
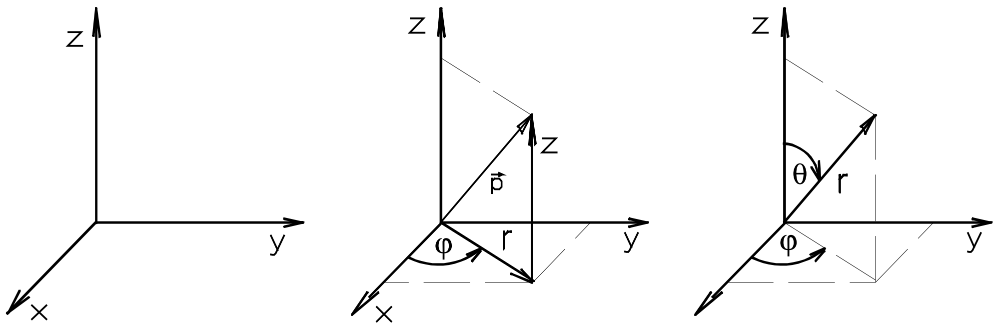


Bild 1.6: Eulersche Winkel

Wesentlich ist, daß die Drehlage durch aufeinanderfolgende Rotationen um die z-Achse, danach um die gedrehte y-Achse und schließlich noch einmal um die gedrehte z-Achse gefunden wird. Die so definierten Eulerwinkel stellen eine, aber nicht die einzige Möglichkeit zur Beschreibung einer Orientierung dar.

1.3 Koordinatensysteme

Die Beschreibung der Positionen und Bewegungen der Industrie-Roboter erfolgt in Kartesischen Koordinaten x, y, z, da diese vom Anwender am leichtesten erfaßt werden können. Unabhängig davon kann der Bewegungsraum des Gerätes in Kugel- oder Zylinderkoordinaten, je nach seiner Konstruktion, besser zu beschreiben sein, die in Bild 1.7 dargestellt sind.



a) Kartesische Koordinaten b) Zylinderkoordinaten c) Kugelkoordinaten

Bild 1.7: Koordinatensysteme

Zylinder-Koordinaten:
$x = r \cos\varphi$
$y = r \sin\varphi$
$z = z$

Kugel-Koordinaten:
$x = r \sin\theta \cos\varphi$
$y = r \sin\theta \sin\varphi$
$z = r \cos\theta$

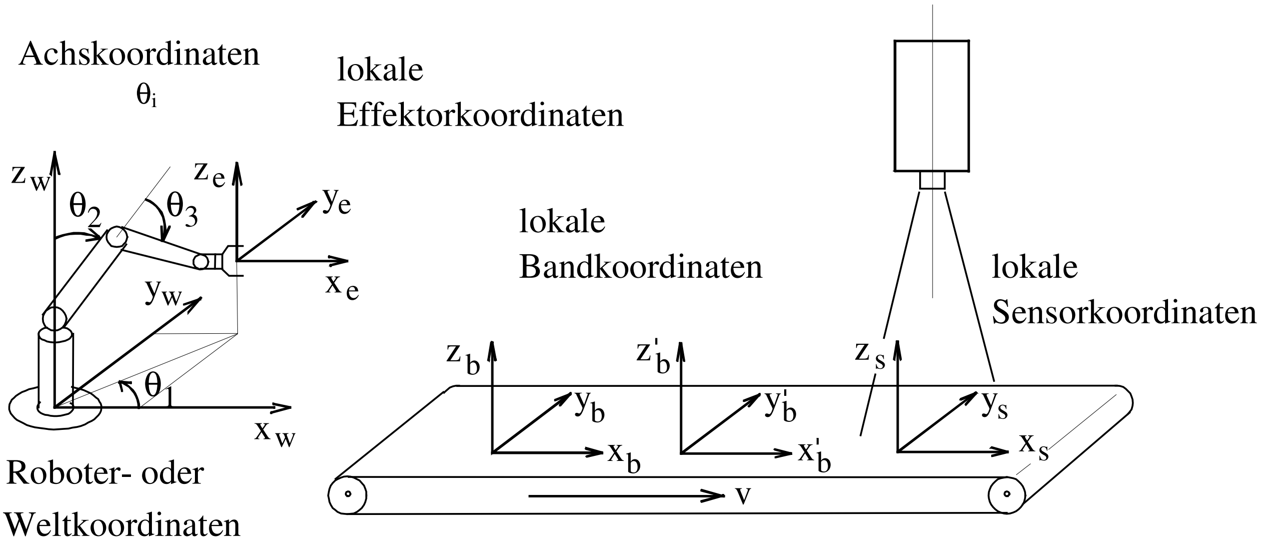


Bild 1.8: Koordinatensystemen bei einem Industrieroboter in einer Arbeitszelle

Das Zusammenwirken der Komponenten von IR-Systemen wird mit Hilfe von Koordinatensystemen beschrieben. Die Position des Effektors wird stets im kartesischen Roboterkoordinatensystem (Basiskoordinatensystem) angegeben. Dieses liegt im Weltkoordinatensystem der Arbeitszelle oder stimmt mit diesem überein. Daneben können lokale Koordinatensysteme vorkommen, z. B. Sensor- und Bandkoordinaten, wie Bild 1.8 zeigt.

Weltkoordinatensystem

Raumfestes, kartesisches Bezugssystem der Roboterumgebung, bzw. der Arbeitszelle.

Roboterkoordinatensystem (Basiskoordinaten)

Raumfestes, kartesisches Koordinatensystem, in dem der Roboter die Stellung des Effektors angibt. Kann mit dem Weltkoordinatensystem zusammenfallen.

Lokales Koordinatensystem

Koordinatensystem der Umgebung, das nicht mit den absoluten Koordinatensystemen übereinstimmt und auch beweglich sein kann, z. B. Sensor- oder Werkstückkoordinatensystem oder ein Gelenkkoordinatensystem.

Achskoordinaten q

Achskoordinaten geben die Bewegung der Roboterachsen in Form von Winkeln (Rotationsachsen) oder Ausfahrlängen (Translationsachsen) an. Die Werte beziehen sich auf eine definierte Nullstellung der Roboterachsen.

Achszählerkoordinaten q_Z

Sie geben die Bewegung der Roboterachsen in Form von Winkelkodiererwerten (absolut oder inkremental) wieder. Diese Angaben beziehen sich ebenfalls auf eine definierte Nullstellung.

1.4 Steuerung und Programmierung

Die Leistungsfähigkeit eines IR hängt in starkem Maße von seiner Steuerung ab. Sie bestimmt auch, wie der Roboter zu programmieren ist. Im wesentlichen lassen sich 3 Arten von Steuerungen unterscheiden.

Play-Back Steuerung

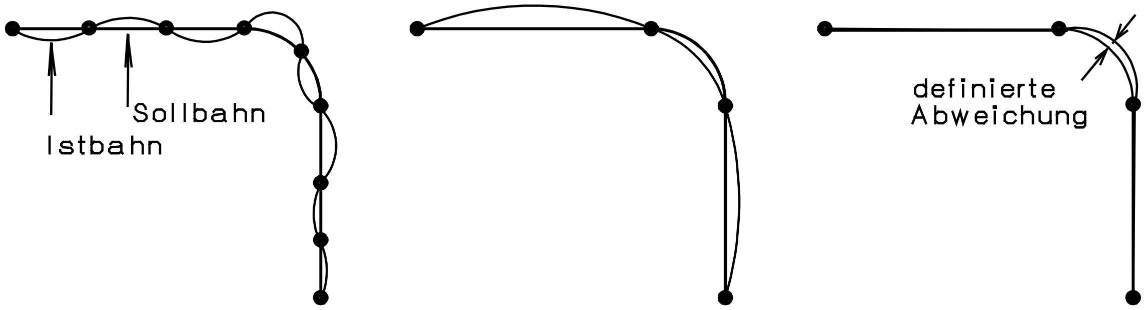
Zur Programmierung wird der IR bei abgeschalteten Motoren von Hand oder mit einer Hilfsvorrichtung geführt. Dabei werden in regelmäßigen Zeitabständen die Achspositionen abgespeichert. Die gespeicherten Achspositionen werden beim Abfahren des Programmes (play back = zurückspielen) im gleichen oder geänderten Zeittakt wieder angefahren.

Bei diesem Verfahren ist die Bahngeschwindigkeit nur wenig beeinflußbar und die Korrektur von Bahndaten aufwendig. Der Bahnverlauf zwischen den gespeicherten Punkten kann von der gewünschten Sollbahn erheblich abweichen. Die Play-Back-Steuerung wird heute kaum noch verwendet.

Punktsteuerung

Nur die Bahnpunkte werden programmiert, deren Position genau angefahren werden soll. Die Steuerung interpoliert die Achsbewegungen zwischen den Bahnpunkten. Daher ist der Bahnverlauf zwischen den programmierten Punkten nicht vorhersehbar. Der genaue Verlauf zwischen den Punkten ist aber bei den in Frage kommenden Einsatzfällen nicht wichtig. Interessant ist nur der Zielpunkt.

Bei modernen Systemen erfolgt die Fahrt von Punkt zu Punkt so, daß alle Achsen die Bewegung gleichzeitig beginnen und am Zielpunkt auch gleichzeitig beenden. Die schnelleren Achsen richten sich dabei nach den langsameren. Diese Fahrweise wird als Synchrone Punkt-zu-Punkt-Steuerung bezeichnet (Synchron PTP).



Play-Back-Steuerung Punktsteuerung Bahnsteuerung

Bild 1.9: Bahnverlauf bei verschiedenen Steuerungsarten

Die Programmierung erfolgt durch Abspeichern definierter Bahnpunkte, die durch Handführung oder mittels eines Handbediengerätes durch die eigenen Antriebe angefahren werden. Man spricht dann auch von teach in, d. h. Belehren des Systems durch zeigen (to teach = lehren). Die Eingabe der Bahnpunkte kann aber auch auf einem externen Programmiergerät (meist ein PC) mit Hilfe einer Roboter-Programmiersprache erfolgen (textuelle Off-Line-Programmierung). Bei dieser Steuerungsart ist die Korrektur einzelner Bahnpunkte einfach. Die Verfahrgeschwindigkeit zwischen den Bahnpunkten kann beliebig beeinflußt werden.

Bahnsteuerung

Die Bahn wird mit Hilfe einiger weniger Bahnpunkte und Orientierungswinkel in einem raumfesten Koordinatensystem vorgegeben. Die Steuerung ermittelt die Bahn durch Interpolation im raumfesten Koordinatensystemen. Lineare, zirkulare und parabolische Interpolationen sind möglich. Die Sollwerte für die Achsantriebe werden über eine Transformation der raumfesten Koordinaten in Achskoordinaten ermittelt. Die Bewegung wird mit programmierbarer Geschwindigkeit entlang definierter Bahnen zwischen den eingegebenen Punkten ausgeführt. Die Programmierung erfolgt wie bei der Punktsteuerung. Bei fortschrittlicheren Systemen kann die Bahn auf CAP-Systemen[2] graphisch programmiert werden (graphische Off-Line-Programmierung).

Die Eigenschaften der 3 Steuerungsarten veranschaulicht Bild 1.9. Das Bahnfahren stellt hohe Anforderungen an die arithmetische Verarbeitungsfähigkeit der Steuerung. Da die Bahn in kartesischen Koordinaten vorgegeben wird, müssen aus diesen Angaben über die sogenannte inverse Kinematik die Sollwerte für die Achsregelkreise berechnet werden.
Bei den Teach-in-Programmierweisen spricht man auch von On-Line-Programmierung, während die Programmierung mit Roboterprogrammiersprachen auf einem Entwicklungssystem und mittels CAP- Systemen als Off-Line-Programmierung bezeichnet wird.

[2] CAP = Computer Aided Planing

1.4.1 Typische Aufgaben für Industrie-Roboter

Ihrer Leistungsfähigkeit entsprechend sind die IR für unterschiedliche Aufgabenstellungen geeignet. Typische Aufgaben für die unterschiedlichen Systeme sind:

Play-Back Steuerung

Bewegungen mit geringen Genauigkeitsanforderungen
Oberflächenbeschichten (z. B. Lackieren)
Einlegeaufgaben
Verkettung von Arbeitsmaschinen

Punktsteuerungen

Positionieraufgaben
Punktschweißen
Verkettung von Arbeitsmaschinen
Einlegeaufgaben
Leiterplatten bestücken
palettieren
sortieren

Bahnsteuerung

Bahnfahraufgaben mit hohen Genauigkeitsanforderungen
Bearbeiten von Werkstücken
Kleber auftragen
entgraten
schleifen
konturfräsen
bahnschweißen
Montage
Biegearbeiten

Bearbeitungsaufgaben stellen die anspruchsvollsten Aufgaben für Industrie-Roboter dar. Sie erfordern meistens eine Bahnsteuerung, die noch durch externe Geräte und Sensoren (z. B. Bearbeitungstische und Optische Sensoren) ergänzt wird. Die Sensoren müssen dabei die Möglichkeit haben, den Arbeitsablauf des Roboters auf verschiedenartige Weise zu beeinflussen. Damit ergibt sich für das Gesamtsystem die Funktionsstruktur von Bild 1.10.

Die Ablaufsteuerung entscheidet über die Auswahl des Roboterprogrammes, z. B. wird zwischen den Programmen *Material einlegen*, *Material bearbeiten* und *Werkstück weiterleiten* unterschieden. Der Block Programmablauf bestimmt die anzufahrenden Bahnpunkte. Diese sind in Bewegungssätzen (BWS) abgelegt. Sie enthalten:

Geometriedaten

Position und Orientierung des Effektors in den Programmierten Bahnpunkten

Fahrinformationen

Interpolationsart (linear, zirkular, parabolisch)
Geschwindigkeit und Verhalten in den Bahnpunkten

Zusatzinformationen

Wartezeiten, Steuerdaten für Peripheriegeräte

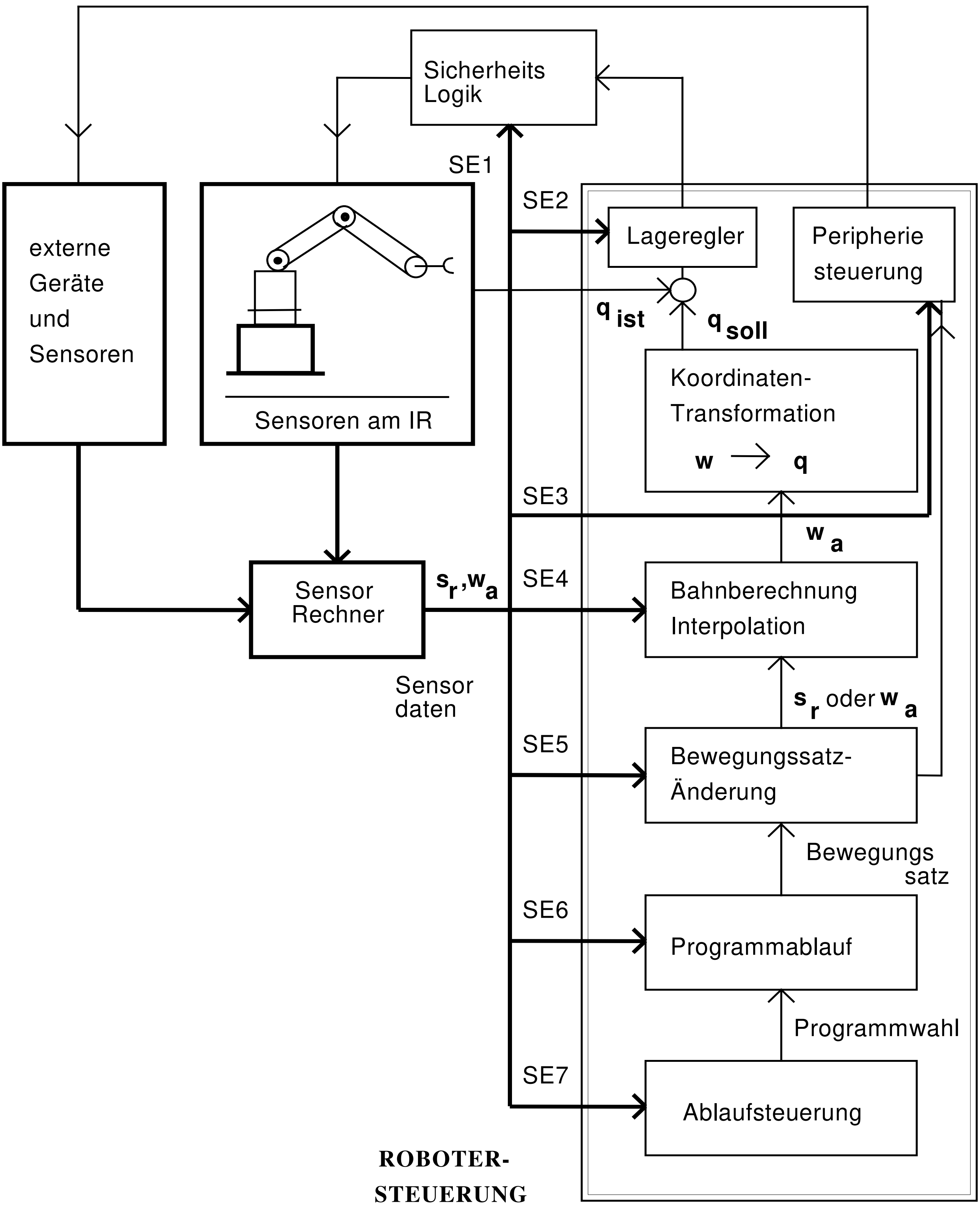


Bild 1.10: Funktionsstruktur

Im Block *Bewegungssatz-Änderungen* können die BWS durch die Sensordaten an die aktuellen Anforderungen angepaßt werden. Z. B. kann die Lage und Geschwindigkeit eines Teiles berücksichtigt werden, das vom Band gegriffen werden soll.

Aus den Geometriedaten wird im Block *Bahnberechnung u. Interpolation* nach dem vorgesehenen Interpolationsverfahren eine dichte Folge von Bahnpunkten w_{ai} in absoluten Kartesischen Koordinaten berechnet.Mit Hife des Blockes *Koordinaten-Transformation* werden daraus die Sollwerte q_{soll} für die Lageregler der Achsen bestimmt.

Die Lageregler steuern die Leistungsverstärker der Achsen und sorgen dafür, daß die Istwerte q_{ist} der Achspositionen mit den gewünschten Werten q_{soll} übereinstimmen. Dabei wird der Regelfehler $\Delta q = q_{soll}-q_{ist}$, der sogenannte Schleppfehler, überwacht. Bei Überschreiten eines Grenzwertes bleibt das Gerät stehen. Desgleichen wird aus q_{ist} fortlaufend die Istbahn berechnet und damit der Bahnabstand überwacht. Auch hier führen unzulässige Abweichen zum Abbruch der Bewegung.

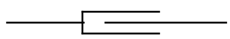
Die Peripherie-Steuerung steuert die peripheren Geräte nach den Vorgaben der Bewegungssätze. Eine externe Sicherheitslogik hat die Aufgabe das System in Gefahrenzuständen abzuschalten (z. B. wenn Endschalter ansprechen oder Personen in den Bewegungsraum des Roboters eindringen).

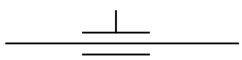
Die externen Sensoren und Geräte greifen, ggf. über einen Sensorrechner, in die Roboter-Steuerung ein. Dafür müssen geeignete Schnittstellen zur Verfügung stehen. Hardwaremäßig erfolgt der Anschluß je nach Bedarf über Digitaleingänge, Serielle Schnittstelle oder IEC-Bus. Softwaremäßig müssen die Sensoren auf unterschiedlichen Ebenen eingreifen können. Dazu dienen die Sensorein-griffsstellen SE2-7. Wenn beispielsweise Geometriedaten übergeben werden sollen, also Positions- und Orientierungsdaten, dann können diese über SE4 oder SE5 in das Steuerprogramm eingreifen. Geometriedaten werden in sensoreigenen Koordinaten übergeben und müssen von der Roboter-Steuerung in die Weltkoordinaten w_a transformiert werden.

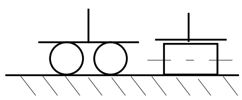
1.5 Symbolische Beschreibung der kinematischen Struktur

Die kinematische Struktur eines Industrie-Roboters kann mit wenigen, einfachen Symbolen anschaulich dargestellt werden. Entsprechende Symbole wurden in der VDI-Richtlinie 2861 festgelegt.

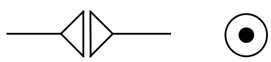
Translatorische Achsen (X, Y, Z, U, V, W)

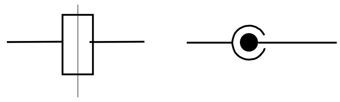
fluchtend 

nicht fluchtend 

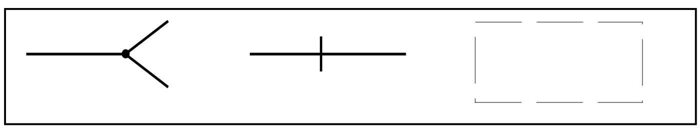
Verfahrachse 

Rotatorische Achsen (A, B, C, D, E, F)

fluchtend 

nicht fluchtend 

Anderes (Q, R, S, T)



Greifer Systemgrenze Abgrenzung der Nebenachsen

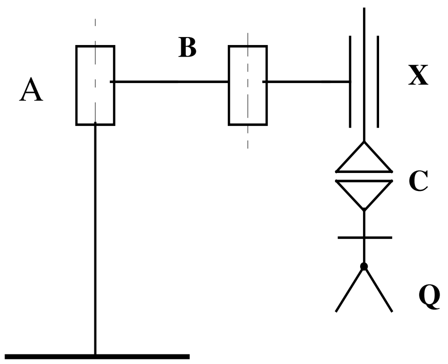


Bild 1.11: SCARA in symbolischer Darstellung

Die Buchstaben dienen zur Bezeichnung der Funktionselemente. Bild 1.11 zeigt ein Beispiel für die Anwendung der Symbole an einem SCARA-Roboter mit 4 Freiheitsgraden.

1.6 Sensoren zur Messung von Gelenkbewegungen

Bei Verschiebegelenken kommen inkrementale oder absolute Wegmeßsysteme zum Einsatz, bei Drehgelenken die entsprechenden Winkelmeßsysteme [2]. Verschiebungen können über Spindeln oder Riementriebe mit Winkelmeßgeräten erfaßt werden.

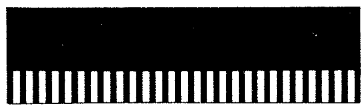
a) Wegmeßsysteme (Bild 1.12)

inkremental

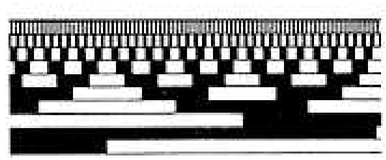
Glasmaßstäbe mit Strichen (a)
Magnetische Maßstäbe

absolut

Glasmaßstäbe mit Kodespuren (b)



a) Strichlineal



b) Kodelineal

Bild 1.12: Linearmaßstäbe

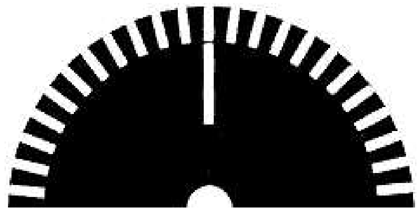
b) Winkelmeßsysteme (Bild 1.13)

inkremental

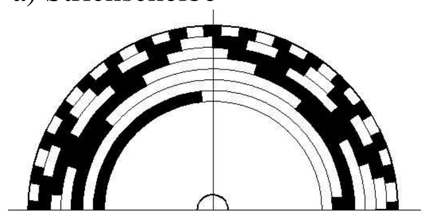
Winkelschrittgeber (Teilscheibe mit Strichen) (a)

Absolut

Winkelkodegeber (Teilscheibe mit Kodespuren) (b)



a) Strichscheibe



b) Kodescheibe

Bild 1.13: Teilscheiben zur Winkelmessung

Multiturngeber sind Winkelkodegeber, die mit Hilfe von Zusatzscheiben mehrere Umdrehungen erfasssen können.

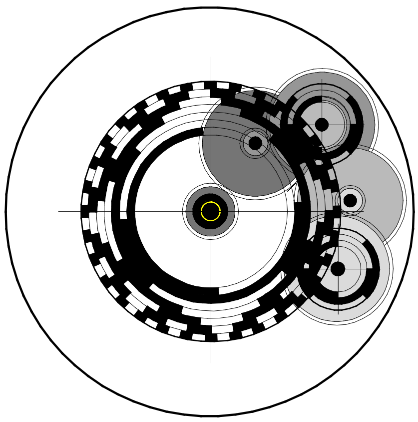


Bild 1.14: Multiturngeber

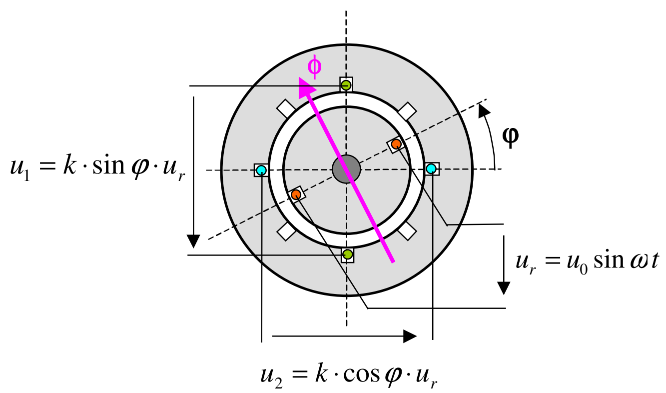


Bild 1.15: Resolver

Resolver gleichen eher einer elektrischen Maschine. Sie sind Drehtransformatoren mit zwei um 90° versetzten Spulen, die sinusförmige Signale liefern, deren Amplitude vom Drehwinkel moduliert werden. Dieser kann daher aus den Signalen ermittelt werden.

1.6.1 Strichlineal zu Wegmessung

Die Fenster des Strichlineals werden von hinten beleuchtet. Das durchfallende Licht wird über vier Fenstergruppen einer Abtastplatte auf vier großflächige Fotoempfänger gelenkt. Die obere Fensterreihe 1, 2 ist gegen die untere 3, 4 um Δx/4 versetzt. Die rechten Fenstergruppen 2, 3 sind gegen die linken 1, 4 um nΔx + Δx/2 versetzt, so daß sich hier, bei gleichen Mittelwerten, invertierte Signale ergeben. Wenn die Abtastplatte in x-Richtung bewegt wird, liefern die Fotoempfänger die elektrischen Dreiecksignale u_1 bis u_4 als Funktion des Weges x.

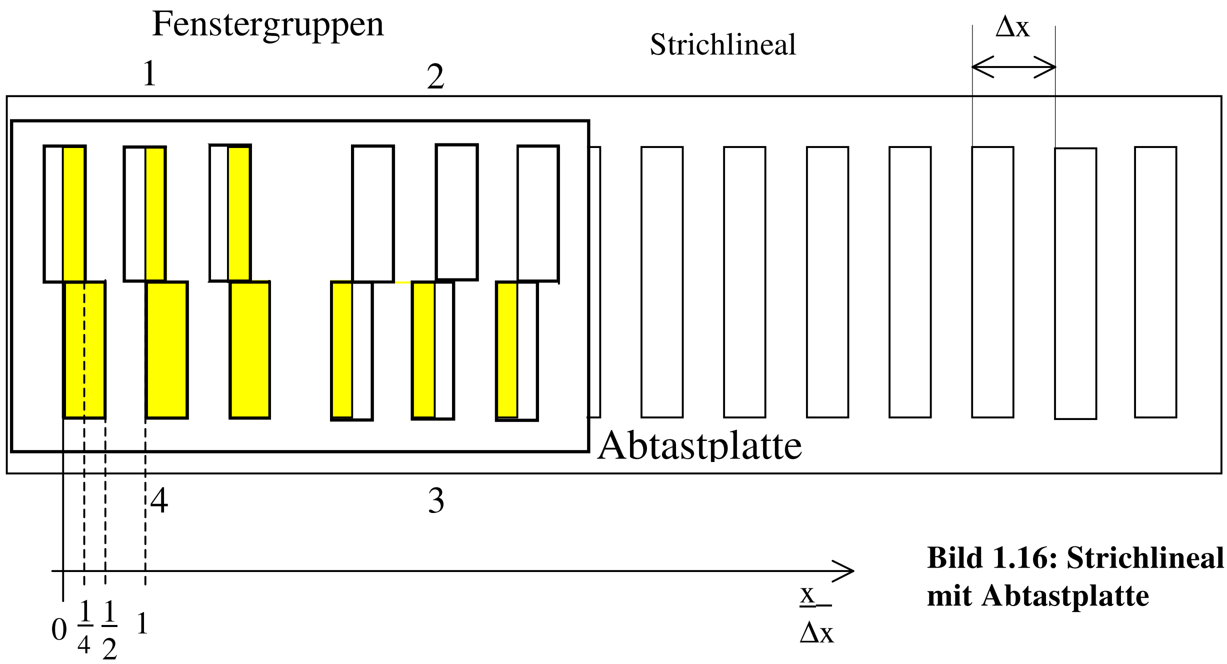


Bild 1.16: Strichlineal mit Abtastplatte

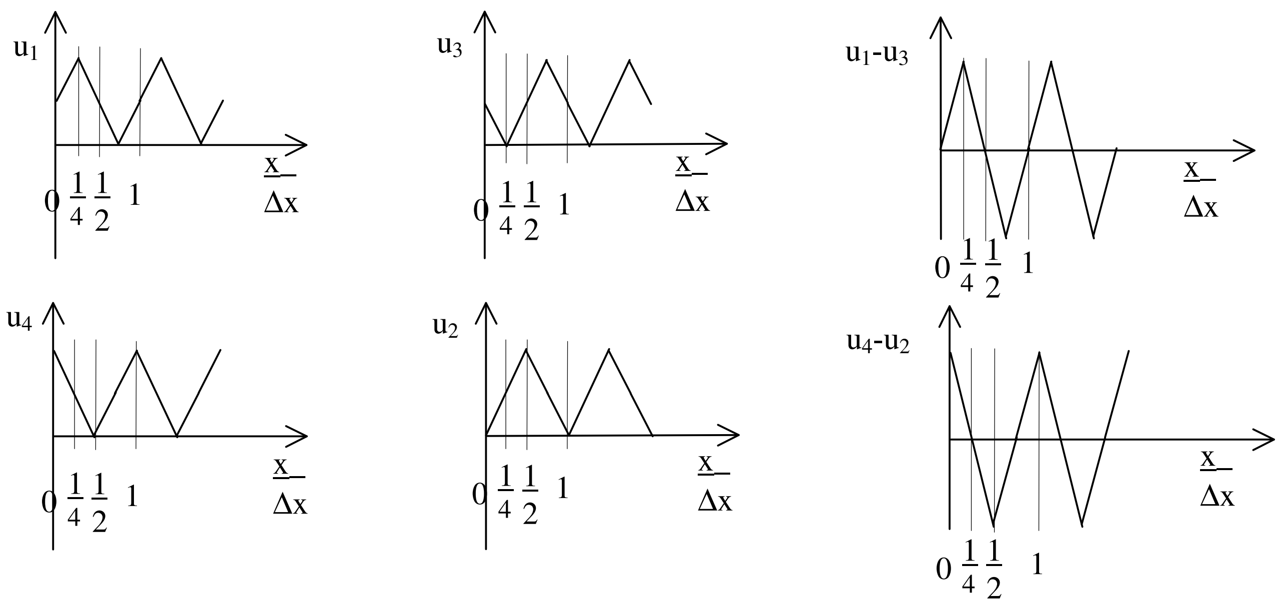


Bild 1.17: Spannungssignale der Fotoempfänger und Differenzsignale

Durch Subtraktion erhält man die Signale u_1-u_3 und u_4-u_2. Diese sind nun mittelwertfrei und weisen eine Phasenverschiebung von Δx/4 auf.

Mittels Schwellwertoperationen können aus den Dreiecksignalen Rechtecksignale A und B gemacht werden, deren Flanken zur Wegmessung gezählt werden.

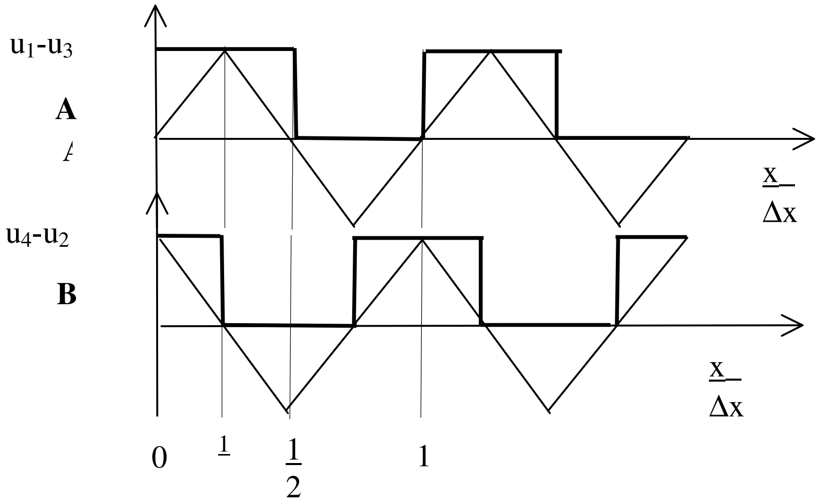


Bild 1.18: Rechtecksignale A und B mit Δx/4 Versatz

Zwischen den Flanken ist die Position unbestimmt. Wenn der Sensor Sinus- und Kosinussignale a und b liefert, kann zwischen den Flanken eines Meßintervalles Δx interpoliert werden.

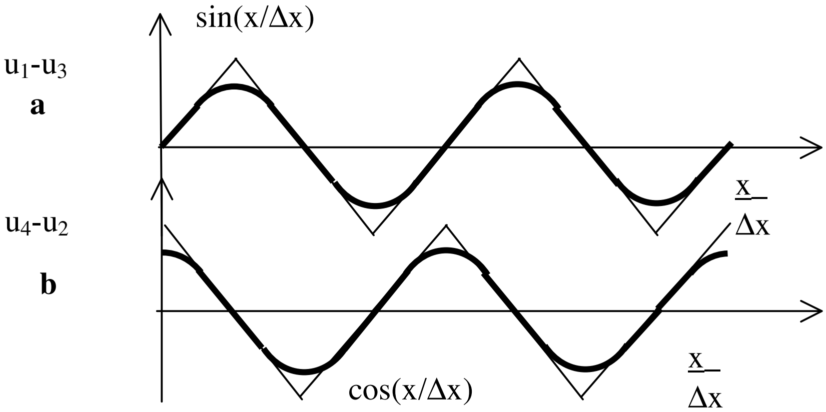


Bild 1.19: Sinus- und Kosinussignale a und b

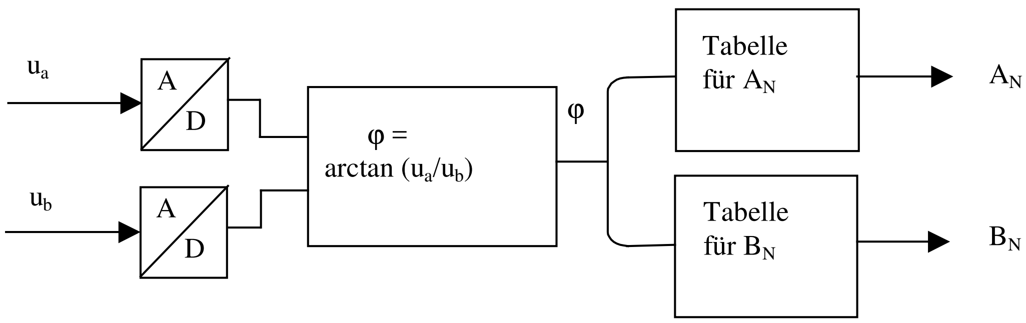


Bild 1.20: Blockschaltbild zur Interpolation

Unter Interpolation versteht man die Vervielfachung der Perioden eines Rechtecksignales zwischen den Flanken des Strichlineals um den Faktor N.

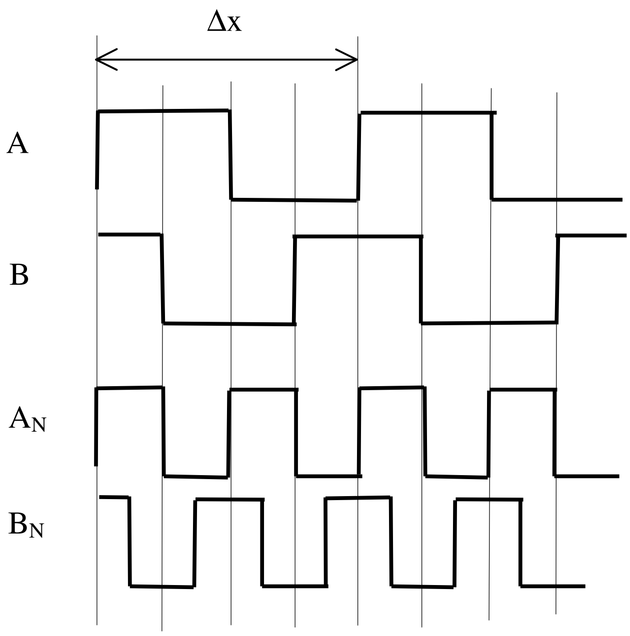


Bild 1.21: Interpolierte Signale A_N und B_N für N = 2

Bild 1.23 zeigt das Beispiel einer Interpolation mit dem Faktor N = 2, also mit zweifacher Interpolation. Dabei werden in einem Intervall Δx zwei Perioden der Rechtecksignale A_N und B_N erzeugt.

1.6.2 Wegmessung

Wenn das Wegintervall Δx bekannt ist, kann der Weg durch Zählung der Intervalle gemessen werden, wobei noch eine Abweichung A berücksichtigt werden muß. Wurden n Intervalle gezählt, so lautet das Meßergebnis

$$x = n \cdot \Delta x \pm A \tag{1.1}$$

Die größten Abweichungen, die auftreten können, ergeben sich aus Bild 1.22. Die Intervalle werden an den positiven Flanken gezählt. Die Wegmessung beginnt im Fall 1 am Anfang des Nullintervalls

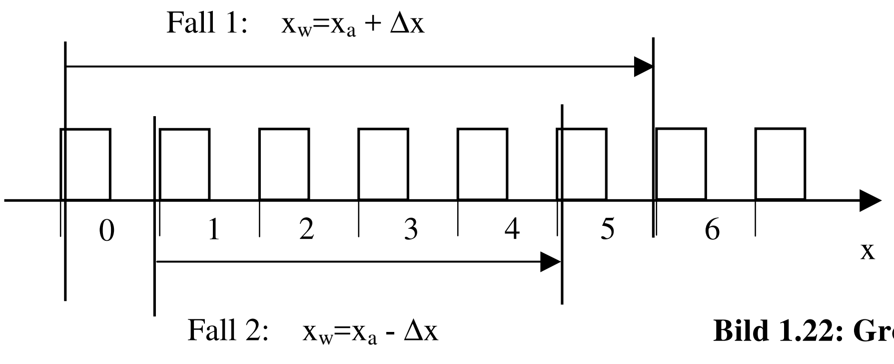


Bild 1.22: Größte Abweichungen

und endet vor der Flanke zu Intervall 6. Gezählt wird daher n = 5. Der angezeigte Weg beträgt daher

$x_a = n\,\Delta x$ (1.2)

Da der Weg nach dem Eintritt in das 5te Intervall um Δx fortgesetzt wurde, ergibt sich ein wahrer Weg der Größe

$x_w = x_a + \Delta x = n\,\Delta x + \Delta x$ (1.3)

Im 2. Fall von Bild 1.22 beginnt der Weg unmittelbar vor der positiven Flanke zu Intervall 1 und endet unmittelbar hinter der positiven Flanke zu Intervall 5. Daher werden wieder n = 5 Flanken gezählt. Wie zuvor gilt daher allgemein

$x_a = n\,\Delta x$ (1.4)

Da aber der Weg am Ende des Nullintervalls begann, wurde ein Intervall weniger durchfahren. Der wahre Weg lautet daher

$x_w = x_a - \Delta x = n\,\Delta x - \Delta x$ (1.5)

Die Abweichung beträgt daher

$A = x_w - x_a = \pm\,\Delta x$ (1.6)

und das Meßergebnis lautet

$x = n\,\Delta x \pm \Delta x$ (1.7)

1.6.3 Richtungserkennung

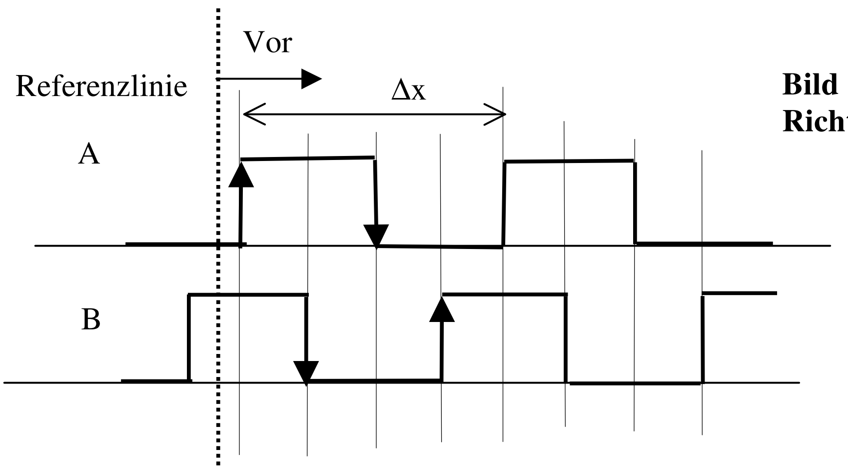


Bild 1.23: Signale A und B zur Richtungserkennung

Die Bewegungsrichtung kann erkannt werden, wenn zwei um etwa Δx/4 versetzte Rechtecksignale nach Bild 1.23 vorliegen. Die Richtung Vorwärts sei gegeben, wenn sich die Referenzlinie nach rechts bewegt. Die boolsche Variable Vor läßt sich dann wie folgt schreiben:

$Vor = (\uparrow A \wedge B) \vee (\downarrow B \wedge A) \vee (\downarrow A \wedge \overline{B}) \vee (\uparrow B \wedge \overline{A})$ (1.8)

Der Pfeil gibt dabei die Flankenrichtung an und der Überstrich die Inversion. Die Richtung kann nur an den Flanken der Signale festgestellt werden. Da es vier Flanken innerhalb eines Intervalles gibt, ergeben sich vier Terme, die als ODER-Verknüpfung die Richtung angeben.

1.6.4 Strichscheibe zur Winkelmessung

Zur Winkelmessung wird eine Strichscheibe eingesetzt. Statt des Wegintervalls Δx ergibt sich ein Winkelintervall $\Delta\alpha$. Mit der Anzahl von N Strichen auf der Scheibe erhält man

$$\Delta\alpha = \frac{2\pi}{N} \qquad (1.9)$$

Wenn n Winkelintervalle gezählt wurden, lautet das Meßergebnis

$$\alpha = n \cdot \Delta\alpha \pm \Delta\alpha \qquad (1.10)$$

Anforderung

Ein gängiges Maß für die Positionierabweichung eines Roboters beträgt etwa A= ±50 µm. Ein kinematischer Arm mit der Länge l läßt sich auf dem Kreis um seine Achse mit der Abweichung

$$\Delta u = l \cdot \Delta\alpha \qquad (1.11)$$

positionieren. Den Zusammenhang veranschaulicht Bild 1.24 für ein positives Winkelintervall $\Delta\alpha$.

Für eine Armlänge von l = 500 mm folgt aus Gl. 1.11 ein Winkelintervall von

$$\Delta\alpha = \frac{\Delta u}{l} = \frac{0{,}05\,mm}{500\,mm} = 0{,}0001\,rad \qquad (1.12)$$

Aus Gl. 1.9 erhält man die dafür erforderliche Teilung

$$N = \frac{2\pi}{\Delta\alpha} = \frac{2\pi}{\Delta u} l = \frac{2\pi \cdot 500\,mm}{0{,}05\,mm} = 63157 \qquad (1.13)$$

Auf der Scheibe müssen also mehr als 63000 Striche untergebracht werden. Aus optischen Gründen beträgt die maximale Strichdichte 100 Striche pro mm.

Der Umfang U der Scheibe muß daher dort, wo sich die Striche befinden, 630 mm betragen. Der dafür erforderliche Radius beträgt

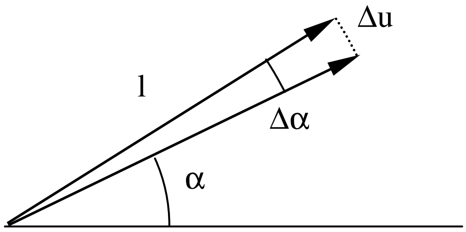


Bild 1.24: Winkelintervall und Positionierabweichung

$$r = \frac{U}{2\pi} = \frac{630\,mm}{2\pi} = 100{,}3\,mm \qquad (1.14)$$

Die Scheibe müßte also einen Durchmesser von etwa 200 mm haben, was aus Platzgründen völlig unrealistisch ist. Die erforderliche Auflösung kann daher nicht über eine ausreichende Zahl von Strichen auf der Scheibe erreicht werden, sondern muß über andere Maßnahmen wie Interpolation oder Zwischengetriebe sichergestellt werden.

1.6.5 Kodelineal zur Wegmessung

Das Kodelineal besitzt M Spuren, die für jedes Wegintervall einen Kode liefern. Meistens erfolgt die Kodierung mit einem einschrittigen Graykode, um gleichzeitige Pegelwechsel mehrerer Spuren zu vermeiden. Bild 1.25 zeigt ein Beispiel mit 3 Spuren. Den Graykode erhält man, indem man in jedem Intervall die Spuren als Stellen einer Binärzahl auffaßt. Die Intervall-Nummer ist der dezimale Wert Z_{10} dieses Kodes.

Die größte Intervall-Nummer im Dezimalsystem beträgt

$$\hat{Z}_{10} = 2^M - 1 \qquad (1.15)$$

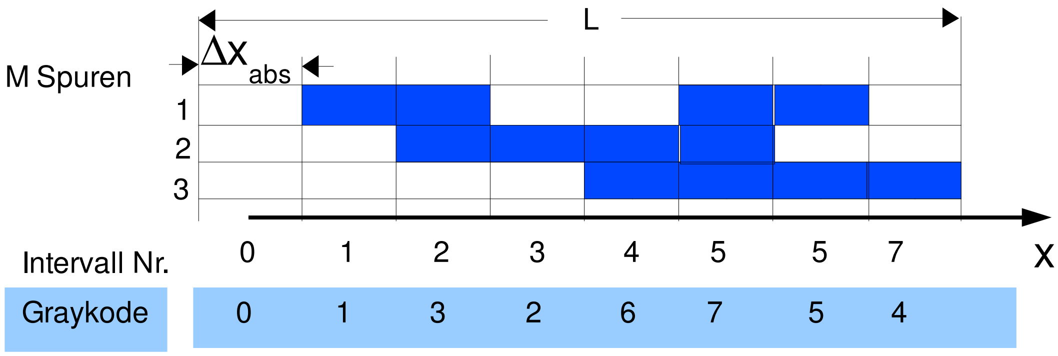


Bild 1.25: Graxkode-Lineal

Bei einer Länge L ergibt sich ein Wegintervall der Größe

$$\Delta x_{abs} = \frac{L}{2^M} \qquad (1.16)$$

Der gemessene Weg ergibt sich aus der Intervall-Nummer multipliziert mit dem Wegintervall.

$$x = \Delta x_{abs} Z_{10} \qquad (1.17)$$

Der gemessene Weg hat die Abweichung A = $\pm\Delta x_{abs}$. Das Meßergebnis lautet daher

$$x = \Delta x_{abs} Z_{10} \pm \Delta x_{abs} \qquad (1.18)$$

Zu beachten ist auch, daß links vom 0 Intervall und rechts vom 2^M-1 Intervall keine Meßwerte verfügbar sind.

1.6.6 Kodescheibe zur Winkelmessung

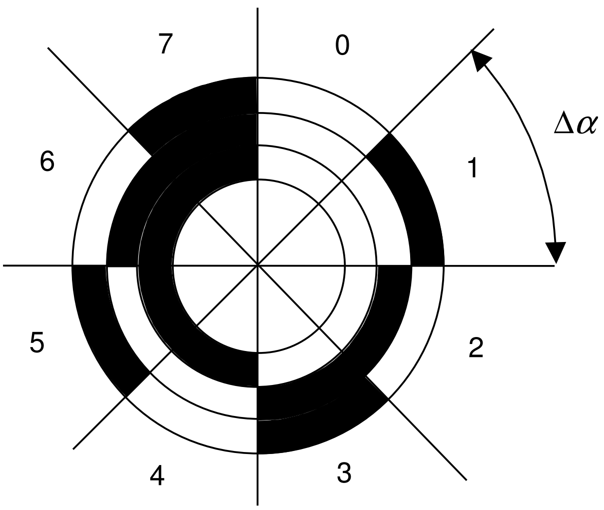


Bild 1.26: Kodescheibe mit 3 Spuren im Binärkode

Bei M = 3 Spuren gibt es 8 Intervalle. Da das 0-Intervall als Startintervall gebraucht wird, können 7 Intervalle für die Winkelmessung verwendet werden. Eine Darstelung zeigt Bild 1.26.

Wird über das letzte Intervall 7 hinaus gedreht, kommt man wieder in das 0-Intervall. Bei einer Linksdrehung gelangt man von 0 nach 7. Im Gegensatz zum Kodelineal gibt es bei der Kodescheibe keine Kodeausfälle bei Bereichsüberschreitungen.

Die Scheibe besitzt M Spuren, die für jedes Winkelintervall Δα einen Kode liefern. Die Anzahl der Kodes beträgt

$$\hat{Z}_{K10} = 2^M \qquad (1.19)$$

Das Dach über dem Zeichen bedeutet den größten Wert. Die größte Intervallnummer im Dezimalsystem ist

$$\hat{Z}_{10} = 2^{M} - 1 \tag{1.20}$$

Die Größe eines Winkelintervalls beträgt

$$\Delta\alpha = \frac{2\pi}{\hat{Z}_{K10}} = \frac{2\pi}{2^{M}} \tag{1.21}$$

Die Intervall-Nummer ist wieder der dezimale Wert Z_{10} dieses Kodes. Damit ergibt sich für den Winkel

$$\alpha = \Delta\alpha \cdot Z_{10} \pm \Delta\alpha \tag{1.22}$$

1.6.7 Multiturngeber

Wenn die Position eines Linearpositioniersystems über die Drehung der Antriebswelle gemessen werden soll, wie das bei Systemen mit Zahnriemen oder Spindeln möglich ist, müssen mehrere Umdrehung gemessen werden können. Absolute Drehgeber müssen dafür mit weiteren Scheiben versehen werden, mit denen die Umdrehungen gezählt werden. Diese Nebenscheiben werden über Getriebe an die Welle der Hauptscheibe angekoppelt. Sie ändern íhren Kode immer dann, wenn die Hauptscheibe eine Umdrehung vollendet hat. Siehe auch Bild 1.14.

Die nachfolgende Darstellung zeigt das Modell des Multiturngebers mit Kodes der einzelnen Scheiben. Die Hauptscheibe 1 ist für eine große Auflösung ausglegt und besitzt M_1 Spuren, mit der jede einzelne Umdrehung erfaßt wird. Die Nebenscheiben besitzen mit meistens nur $M_2 = M_3 = 4$ Spuren eine geringe Auflösung. Sie sind über Getriebe in Reihe geschaltet und dienen zur Zählung der Umdrehungen der Hauptscheibe. Die Übersetzung der Getriebe ist so ausgelegt, daß mit ü Umdrehungen der antreibenden Scheibe der Kodevorrat der Nebenscheibe voll ausgenutzt wird.

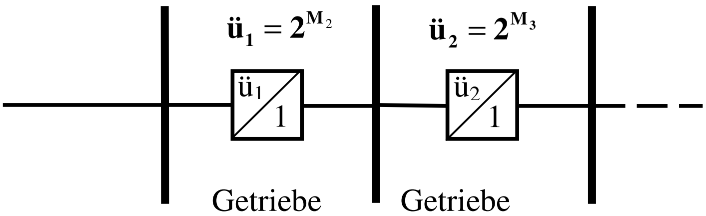
Üblicherweise werden bis zu drei Nebenscheiben verwendet. Drei Scheiben mit 4 Spuren können 2^{12} Kodes liefern. Da der Kode 0 keine Umdrehung zählt, muß von der Zahl der möglichen Kodes dieser Kode abgezogen werden, so daß sich $2^{12} - 1 = 4095$ Umdrehungen zählen lassen.
Die nachfolgende Darstellung zeigt das Prinzip des Multiturngeber an einem Beispiel.

1.6.8 Beispiel zum Multiturngeber

Scheiben:	Hauptscheibe	Nebenscheiben	
Nummer:	1	2	3
Kodespuren:	M_1	M_2	M_3

Übersetzungen:

Hauptwelle (Antrieb)



Umdrehungen:	$\ddot{u}_1\,\ddot{u}_2$	$\ddot{u}_2$	1

← D → ← 4 → ← 7 → ← 0 → ← 2 →

Register: 1 0 1 1 0 0 1 0 1 1 1 0 0 | 0 0 0 0 | 1 0 0 0

LSB MSB

Stellen: M_1 M_2 M_3

M_3 (4 ------------ 1)	M_2 (4 ------------ 1)	M_1 (13 ------------ 1)
0 0 0 1	0 0 0 0	0 0 1 1 1 0 1 0 0 1 1 0 1

MSB LSB

Mit den Nebenscheiben können $\ddot{u}_1 * \ddot{u}_2 - 1$ Umdrehungen gezählt werden.

Zahl der Kodes der Nebenscheiben: $\ddot{u}_1 = 2^{M_2}$ $\ddot{u}_2 = 2^{M_3}$

Zahl der Kodes der Nebenscheiben: $\hat{Z}_N = \ddot{u}_1 \cdot \ddot{u}_2 = 2^{M_2} \cdot 2^{M_3} = 2^{M_2+M_3}$

Das Dach über dem Buchstaben bezeichnet feste Maximalwerte an.

Zählbare Umdrehungen mit den Nebenscheiben $\hat{Z}_U = 2^{M_2+M_3} - 1$
da der Kode 0 zur Zählung ausfällt.

Die Hauptscheibe löst eine Umdrehung genau auf.

Die Zahl der Kodes beträgt $\hat{Z}_H = 2^{M_1}$

Die Gesamtzahl der Kodes beträgt daher $\hat{Z} = \hat{Z}_H \cdot \hat{Z}_N$

Beispiel 1: $M_1 = 13$, $M_2 = M_3 = 4$

$$\hat{Z} = \hat{Z}_H \cdot \hat{Z}_N = 2^{13} \cdot 2^4 \cdot 2^4 = 2^{13} \cdot 2^8 = 8192 \cdot 256 = 2\,097\,152$$

Die Winkelauflösung beträgt:

$$\Delta\alpha = \frac{360°}{\hat{Z}_H} = \frac{360°}{2^{M_1}} = \frac{360°}{8192} = 0{,}043945°$$

Beispiel 2: Die im Register dargestellte Binärzahl hat den hexadezimalen Wert $2074D_H$. Sie entspricht der Zahl der Winkelinkremente und hat den Dezimalwert:
$2074D_H = 132\,941$

Gemessen wurden nach Registerinhalt $Z_N = 10_H = 16$ Umdrehungen.
Die Hauptscheibe steht auf dem Kode $Z_H = 74D_H = 1869$
Der gemessene Winkel beträgt daher

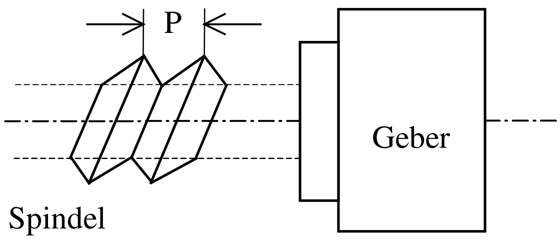
$$\alpha = Z_H * \Delta\alpha + Z_N * 360° = 1869 * 0{,}043945° + 16 * 360° = 82{,}1332° + 5760° = 5842{,}133°$$

Teilt man diesen Winkelwert durch die Winkelauflösung, so ergibt sich mit einer kleinen Abweichung wieder der Dezimalwert des Registers.

$$\frac{\alpha}{\Delta\alpha} = \frac{5842{,}133°}{0{,}043945°} = 132\,941{,}9$$

Beispiel 3: Positioniersystem mit Spindel

Ein Positioniersystem arbeitet mit einer Spindel und einem Multiturngeber.



Spindel
Gewindesteigung P

Spindelsteigung $P = 3$ mm
Multiturngeber $M_1 = 13$ $M_2 = M_3 = 4$

Fragen:

1. Welche Wegauflösung Δx besitzt das System?
2. Wie groß ist der maximale Meßweg L?
3. Wieviel Kodes stehen insgesamt zur Verfügung?

Lösungen:

1. $\Delta x = \frac{P}{2^{M_1}} = \frac{3\,mm}{2^{13}} = 0{,}37 \cdot 10^{-3}\,mm$
1. $L = P \cdot \left(2^{M_2+M_3} - 1\right) + \Delta x \cdot (2^{M_1} - 1) = P \cdot \left(2^{M_2+M_3} - 1\right) + P - \Delta x = P \cdot 2^{M_2+M_3} - \Delta x$
   volle Umdrehungen der Nebenscheiben +Umdrehung der Hauptscheibe bis vor Überlauf
   $L = 768\,mm - 0{,}37 \cdot 10^{-3}\,mm = 767{,}99963\,mm$
2. $\hat{Z} = \hat{Z}_H \cdot \hat{Z}_N = 2^{M_1} \cdot 2^{M_2+M_3} = 2\,097\,152$ Kodes der Hauptscheibe mal Kodes der Nebenscheiben

Zahnriementrieb:
Beim Zahnriementrieb beträgt der Vorschub pro Umdrehung πD, wobei D der Durchmesser des Antriebsrades ist.

1.7 Transformation von Koordinaten

Positionen und Orientierungen werden mit Hilfe von Koordinatensystemen beschrieben. Das Bezugskoordinatensystem (Weltkoordinatensystem) ist dabei immer ein orthogonales, rechtsdrehendes System. Für die Beschreibung der Kinematik von Industrie-Robotern sowie für das Zusammenspiel zwischen Roboter und externen Geräten ist die Umrechnung von Positionen und Orientierungen von einem Koordinatensystem in ein anderes von großer Bedeutung (Koordinatentransformation).

In einer Arbeitszelle können mehrere lokale Koordinatensysteme vorkommen, die in das allen übergeordnete Bezugssystem umgerechnet werden müssen. Dazu zeigt Bild 1.8 einige Beispiele. Das Basiskoordinatensystem des Roboters stimmt hier mit dem absoluten Weltkoordinatensystem überein. Die Effektor- und Bandkoordinatensysteme sind bewegliche, lokale Koordinatensysteme und das Sensorkoordinatensystem ist ein festes lokales Koordinatensystem. Alle kartesischen Koordinatensysteme sind rechtwinklige, rechtsorientierte Systeme. Mit anderen Worten, die Achsen stehen senkrecht aufeinander, und man kann mit Daumen, Zeigefinger und Mittelfinger der rechten Hand in die Richtung der x, y und z-Achse zeigen.

Koordinaten-Systeme werden im folgenden mit dem Bezeichner S (für System) und gegebenenfalls einem Index bezeichnet. Der Bezeichner wird dem jeweiligen System durch eine Beziehung der folgenden Art zugewiesen:

$S = (0, x, y, z)$ für ein System mit den Koordinatenachsen x, y, z mit dem Ursprung 0 bzw.

$S^* = (0^*; x^*, y^*, z^*)$ für ein System mit den Koordinatenachsen x^*, y^*, z^* und dem Ursprung 0^*.

Der Ursprung muß nicht immer explizit ausgewiesen werden.

Zum Übergang von einem Koordinatensystem zum anderen kann eine Translation, eine Drehung oder beides erforderlich sein.

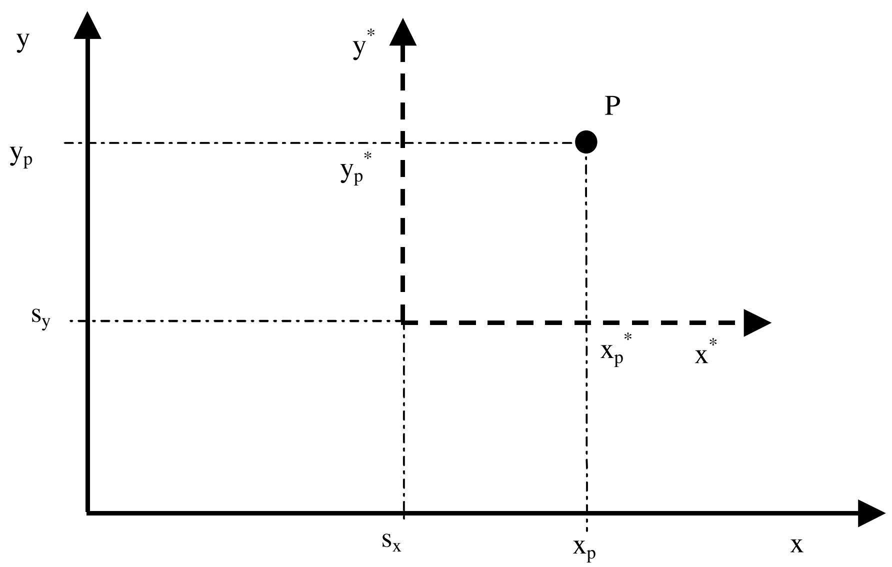


Bild 1.27: Translation eines Koordinatensystems

1.7.1 Translation

Bei einer Translation sind die Koordinatensysteme S und S* achsenparallel verschoben. Gemäß Bild 1.26 ist der Zusammenhang zwischen den Koordinatenachsen wie folgt gegeben:

$$x_P = x_P{*} + s_x \qquad y_P = y_P{*} + s_y \qquad z_P = z_P{*} + s_z \qquad (1.23)$$

Dabei wurde die nicht dargestellte z-Koordinate in passender Weise ergänzt. Umgekehrt ergeben sich die Koordinaten des Punktes P im verschobenen System aus

$$\begin{aligned} x_p^* &= x_p - s_x \\ y_p^* &= y_p - s_y \\ z_p^* &= z_p - s_z \end{aligned} \qquad (1.24)$$

Dieses Gleichungssystem kann in Matrizenform geschrieben werden

$$\begin{Bmatrix} x_p^* \\ y_p^* \\ z_p^* \end{Bmatrix} = \begin{Bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{Bmatrix} \begin{Bmatrix} x_p \\ y_p \\ z_p \end{Bmatrix} - \begin{Bmatrix} s_x \\ s_y \\ s_z \end{Bmatrix} \quad \text{kürzer:} \quad x^* = A\,x - s \qquad (1.25)$$

Eine vorteilhaftere Darstellung bezieht die Verschiebeparameter in die Matrix ein. Dies wird möglich, wenn die Koordinatenspalten um ein Element mit Inhalt 1 erweitert werden und die Matrix eine 4 x 4 Matrix wird. Die so erweiterten Koordinaten heißen dann homogen.

$$\begin{Bmatrix} x_p^* \\ y_p^* \\ z_p^* \\ 1 \end{Bmatrix} = \begin{Bmatrix} 1 & 0 & 0 & -s_x \\ 0 & 1 & 0 & -s_y \\ 0 & 0 & 1 & -s_z \\ 0 & 0 & 0 & 1 \end{Bmatrix} \begin{Bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{Bmatrix} \quad \text{kürzer:} \quad x_h^* = A_h\, x_h \qquad (1.26)$$

Die 3 x 3 Einheitsmatrix neben dem Verschiebevektor kann später durch eine Orientierungsmatrix ersetzt werden. Die Transformation bewirkt dann gleichzeitig eine Verdrehung und eine Verschiebung.

1.7.2 Drehung

Das Problem der Drehung sei zunächst am ebenen Fall betrachtet für die Drehung um eine Koordinatenachse. In Bild 1.28 sind zwei gegeneinander verdrehte Koordinatensysteme zu sehen.

Passive Transformation:

Der Vektor $\vec{p}$ hat im nichtverdrehten Koordinatensystem S die Koordniaten x_p und y_p. Im Verdrehten Koordimatensystem S^* hat er die Koordinaten x_p* ind y_p*. Zum Zeichen dafür, daß er in den Koordinaten des verdrehten System angegeben ist, heißt er $\vec{p}\,*$. Ansonsten gilt $\vec{p} = \vec{p}\,*$. Der Vektor $\vec{p}$ bleibt also am Ort stehen und das Koordinatensystem dreht sich. Daher heißt die Transformation passiv. Aus Bild 1.28 ergibt sich für die Koordinaten die folgende Transformation:

$$\begin{aligned} x_p* &= \;\; x_p\cos\gamma + y_p\sin\gamma + \mathit{z_p\cdot 0} \\ y_p* &= -x_p\sin\gamma + y_p\cos\gamma + z_p\cdot 0 \\ \mathit{z_p*} &= \;\; \mathit{x_p\cdot 0} \quad + \mathit{y_p\cdot 0} \quad + \mathit{z_p\cdot 1} \end{aligned} \qquad (1.27)$$

Dabei wurde die z-Komponente wieder passend ergänzt und kursiv eingetragen. Das obige Gleichungssystem kann in Matrizenform geschrieben werden:

$$\begin{Bmatrix} x_p^* \\ y_p^* \\ z_p^* \end{Bmatrix} = \begin{Bmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{Bmatrix} \begin{Bmatrix} x_p \\ y_p \\ z_p \end{Bmatrix} \qquad (1.28)$$

Diese Transformation rechnet also die Koordinaten eines Vektors aus dem Bezugssystem in ein verdrehtes Koordinatensystem um.

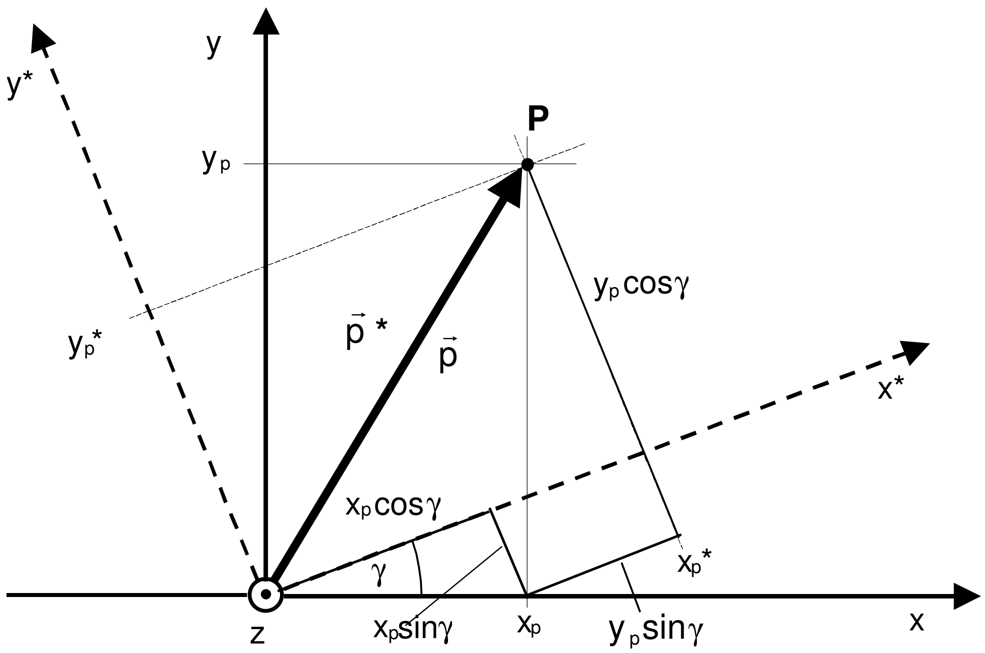


Bild 1.28: Passive Transformation eines Vektors

Aktive Transformation:

In Bild 1.28 besitzt der Vektor $\vec{p}$ die Koordniaten x_p und y_p. Der verdrehte Vektor p* besitzt die Koordinaten x_p* ind y_p* infolge einer Drehung um die z-Achse mit dem Winkel γ. Dieses Mal wurde also der Vektor gedreht und das Koordinatensystem blieb fest. Die Transformation heißt daher aktiv. Die Koordinaten des verdrehten Vektors lassen sich wie folgt beschreiben:

$$\begin{aligned} x_p^* &= p\cos(\gamma+\varphi) = p\cos\varphi\cos\gamma - p\sin\varphi\sin\gamma = x_p\cos\gamma - y_p\sin\gamma + z_p\,0 \\ y_p^* &= p\sin(\gamma+\varphi) = p\sin\varphi\cos\gamma + p\cos\varphi\sin\gamma = x_p\sin\gamma + y_p\cos\gamma + z_p\,0 \\ z_p^* &= x_p\cdot 0 \quad + y_p\cdot 0 \quad + z_p\cdot 1 \end{aligned} \qquad (1.29)$$

Die Gleichungen 1.29 lassen sich nun wieder in Matrixform schreiben:

$$\begin{Bmatrix} x_p^* \\ y_p^* \\ z_p^* \end{Bmatrix} = \begin{Bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{Bmatrix} \begin{Bmatrix} x_p \\ y_p \\ z_p \end{Bmatrix} \qquad (1.30)$$

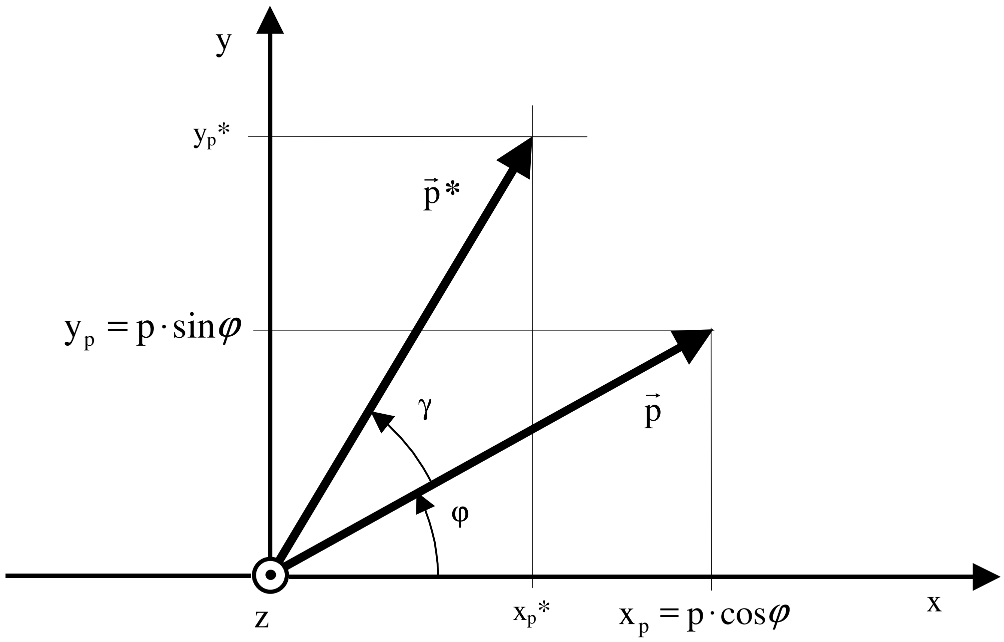


Bild 1.29: Aktive Transformation eines Vektors

Der Vergleich der Transformationsmatrizen der Gln. 1.28 und 1.30 zeigt, daß diese sich nur durch eine Transposition unterscheiden.

1.7.3 Orientierung eines Koordinatensystems

Die Drehlage eines Koordinatensystems S^* wird im System S beschrieben durch die Lage seiner Einheitsvektoren $\vec{e}_x^*, \vec{e}_y^*, \vec{e}_z^*$, die die Richtungen der x*, y* und z*-Achsen angeben. Diese sind Ortsvektoren, die stets im Ursprung beginnen und die Länge $|\vec{e}| = 1$ besitzen. Beide Koordinatensysteme haben daher einen gemeinsamen Ursprung 0 = 0*. Bild 1.29 zeigt der Einfachheit halber zwei gegeneinander um die z-Achse verdrehte Koordinatensysteme in der Ebene.

Damit gilt die Beziehung

$$\begin{aligned}
\vec{e}_x^* &= \vec{e}_x \cos\gamma + \vec{e}_y \sin\gamma + \vec{e}_z \cdot 0 \\
\vec{e}_y^* &= \vec{e}_x \sin\gamma + \vec{e}_y \cos\gamma + \vec{e}_z \cdot 0 \\
\vec{e}_z^* &= \vec{e}_x \cdot 0 + \vec{e}_y \cdot 0 + \vec{e}_z \cdot 1
\end{aligned} \tag{1.31}$$

In Matrizenschreibweise ergibt das

$$\begin{Bmatrix} \vec{e}_x^* \\ \vec{e}_y^* \\ \vec{e}_z^* \end{Bmatrix} = \begin{bmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} \vec{e}_x \\ \vec{e}_y \\ \vec{e}_z \end{Bmatrix} \tag{1.32}$$

In der allgemeinen Form kann die Rotationslage auch durch mehrere Rotationen um verschiedene Achsen verursacht worden sein und soll dann als *Orientierung* bezeichnet werden. Die Spalten und die Matrix können schließlich noch durch Symbole ersetzt und, wie in Gl. 1.33 geschehen, ganz kompakt als Matrizengleichung geschrieben werden.

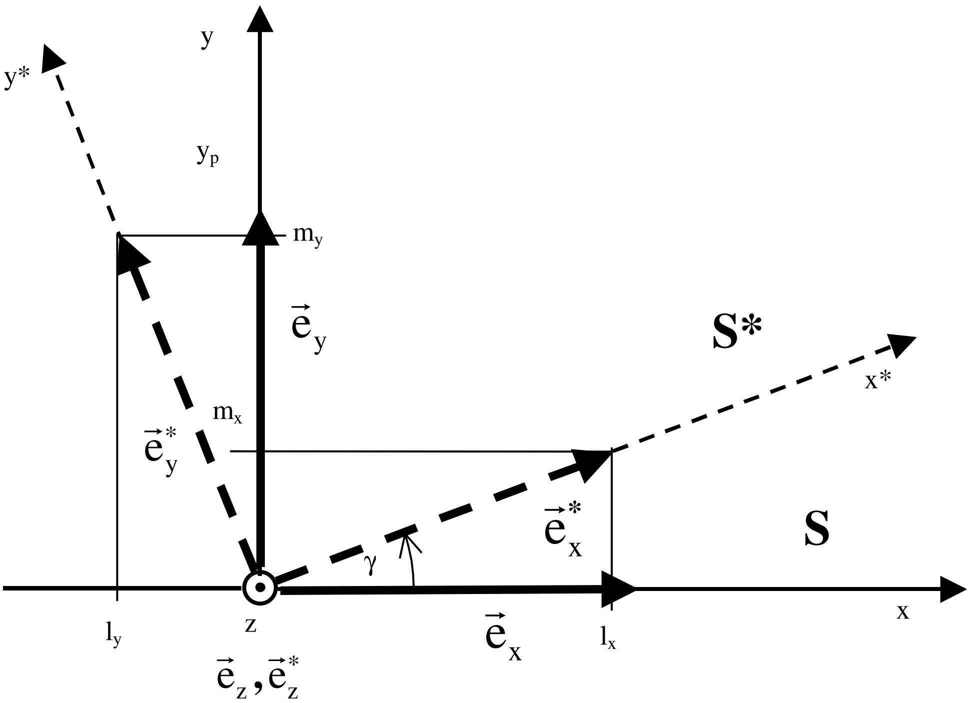


Bild 1.30: Beschreibung der Drehlage durch die Komponenten der Einheitsvektoren

$$\begin{Bmatrix} \vec{e}_x^{\,*} \\ \vec{e}_y^{\,*} \\ \vec{e}_z^{\,*} \end{Bmatrix} = \begin{Bmatrix} l_x & m_x & n_x \\ l_y & m_y & n_y \\ l_z & m_z & n_z \end{Bmatrix} \begin{Bmatrix} \vec{e}_x \\ \vec{e}_y \\ \vec{e}_z \end{Bmatrix} \qquad \mathbf{e^*} = \mathbf{A} \cdot \mathbf{e} \tag{1.33}$$

Die Einheitsvektoren beschreiben die Richtung der Koordinatenachsen und damit die Orientierung des Koordinatensystems. Der Begriff des Koordinatensystems S wird im weiteren mit den drei Einheitsvektoren gleichgesetzt, und zwar speziell mit ihrer Anordnung in einem Zeilenvektor und dann Fett geschrieben:

$$\mathbf{S}^* = \{\vec{e}_x^{\,*} \;\; \vec{e}_y^{\,*} \;\; \vec{e}_z^{\,*}\} \tag{1.34}$$

Das Bezugssystem S wird ebenso beschrieben:

$$\mathbf{S} = \{\vec{e}_x \;\; \vec{e}_y \;\; \vec{e}_z\} \tag{1.35}$$

Aus der Transposition von Gl. 1.33 erhält man

$$\mathbf{e}^{*\mathbf{T}} = \mathbf{e}^{\mathbf{T}} \cdot \mathbf{A}^{\mathbf{T}} \tag{1.36}$$

Ausgeschrieben ergibt sich

$$\{\vec{e}_x^{\,*} \;\; \vec{e}_y^{\,*} \;\; \vec{e}_z^{\,*}\} = \{\vec{e}_x \;\; \vec{e}_y \;\; \vec{e}_z\} \cdot \begin{Bmatrix} l_x & l_y & l_z \\ m_x & m_y & m_z \\ n_x & n_y & n_z \end{Bmatrix} \tag{1.37}$$

Die Matrix $\mathbf{A}^{\mathbf{T}}$ enthält nun die Koordinaten der verdrehten Einheitsvektoren als *Spalten*, d. h.

$$\vec{e}_x^{\,*} = \begin{Bmatrix} l_x \\ m_x \\ n_x \end{Bmatrix} \quad \vec{e}_y^{\,*} = \begin{Bmatrix} l_y \\ m_y \\ n_y \end{Bmatrix} \quad \vec{e}_z^{\,*} = \begin{Bmatrix} l_z \\ m_z \\ n_z \end{Bmatrix} \tag{1.38}$$

Da die Matrix $\mathbf{A}^\mathbf{T}$ im weiteren häufig gebraucht wird, erhält sie das Symbol **B**. Es gilt also

$$\mathbf{B} = \mathbf{A}^\mathbf{T} = \begin{Bmatrix} l_x & l_y & l_z \\ m_x & m_y & m_z \\ n_x & n_y & n_z \end{Bmatrix} \tag{1.39}$$

Die Gl. 1.36 kann nun mit den Gln. 1.34, 1.35 und 1.39 kompakt geschrieben werden:

$$\mathbf{S}^* = \mathbf{S} \cdot \mathbf{B} \tag{1.40}$$

Damit ist die grundlegende Beziehung für die Orientierung von Koordinatensystem gefunden. Das verdrehte System ergibt sich aus dem Bezugssystem durch eine Rechtsmultiplikation mit B. Dabei sind zwei Fälle zu unterscheiden:

1. Das Bezugssystem ist ein Basissystem

In diesem Fall sind die Einheitsvektoren des Bezugssystems wie folgt gegeben

$$\vec{e}_x = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix} \quad \vec{e}_y = \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix} \quad \vec{e}_z = \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix} \tag{1.41}$$

Das System S ist damit durch eine Einheitsmatrix gegeben und Gl. 1.40 vereinfacht sich zu

$$\mathbf{S}^* = \mathbf{B} \tag{1.42}$$

2. Das Bezugssystem ist ein bereits verdrehtes Systeme

Das Basissystem sei S_0. Die Transformation B_{01} transformiert von S_0 nach S_1, die Transformation B_{12} von S_1 nach S_2 usw., bis $B_{n-1,n}$ von S_{n-1} nach S_n transformiert. Das bedeutet

$$\mathbf{S}_1 = \mathbf{S}_0\, \mathbf{B}_{01}$$
$$\mathbf{S}_2 = \mathbf{S}_1\, \mathbf{B}_{12}$$
$$\vdots$$
$$S_n = S_{n-1}\, B_{n-1,n} \tag{1.43}$$

Nacheinander einsetzen ergibt eine Transformation von $\mathbf{S}_0$ nach $\mathbf{S}_n$

$$\mathbf{S}_n = \mathbf{S}_0\, \mathbf{B}_{01}\, \mathbf{B}_{12} \ldots \mathbf{B}_{n-1,n} \tag{1.44}$$

Die Gesamttransformation $\mathbf{B}_{0n}$ (von 0 nach n) ist also durch das Produkt der Teiltransformationen gegeben. Sie wird durch Aufwärtstransformation vom Basissystem bis zum Endsystem gefunden. Die Matrizen der Teiltransformationen werden dabei in aufsteigender Reihenfolge von rechts an die erste Matrix heranmultipliziert.

$$\mathbf{B}_{0n} = \mathbf{B}_{01}\, \mathbf{B}_{12} \ldots \mathbf{B}_{n-1,n} \tag{1.45}$$

Beispiel 4: Zwei Teiltransformationen

Bild 1.31 zeigt zwei gegen das Bezugssystem S_0 verdrehte Koordinatensysteme. Das System S_1 ist im Bezugssystem S_0 gegeben und das System S_2 dagegen im System S_1.

$$\mathbf{S}_1 = \mathbf{S}_0\mathbf{B}_{01} \tag{1.46}$$

$$\text{mit } \mathbf{B}_{01} = \begin{Bmatrix} l_{x01} & l_{y01} & l_{z01} \\ m_{x01} & m_{y01} & m_{z01} \\ n_{x01} & n_{y01} & n_{z01} \end{Bmatrix} \tag{1.47}$$

$$\mathbf{S}_2 = \mathbf{S}_1\mathbf{B}_{12} \tag{1.48}$$

$$\text{mit } \mathbf{B}_{12} = \begin{Bmatrix} l_{x12} & l_{y12} & l_{z12} \\ m_{x12} & m_{y12} & m_{z12} \\ n_{x12} & n_{y12} & n_{z12} \end{Bmatrix} \tag{1.49}$$

Einsetzen von Gl. 1.46 in Gl. 1.48 ergibt

$$S_2 = S_0B_{01}B_{12} \tag{1.50}$$

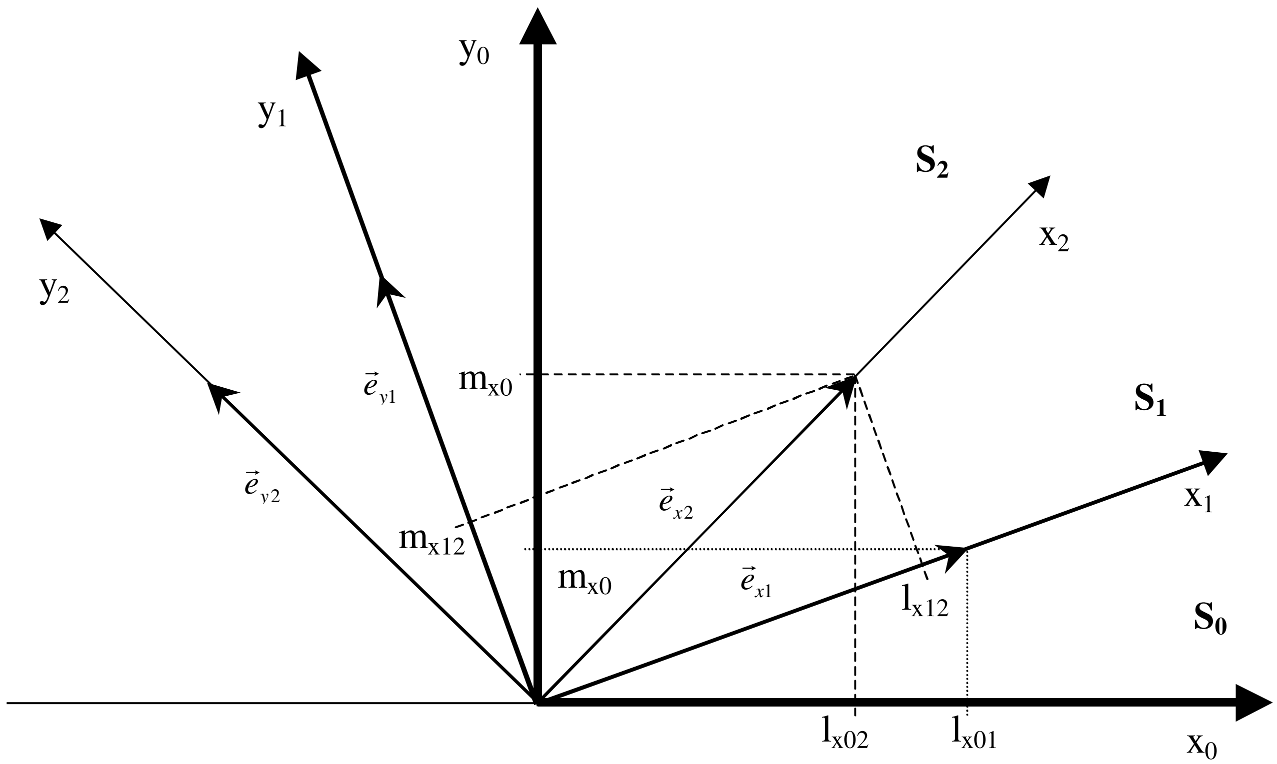


Bild 1.31: Zwei gegen das Bezugssystem verdrehte Kordinatensysteme

Damit ist die Transformation von S_0 nach S_2 bekannt. Mit anderen Worten, man kennt die Matrix

$$\mathbf{B}_{02} = \mathbf{B}_{01}\,\mathbf{B}_{12} = \begin{Bmatrix} l_{x02} & l_{y02} & l_{z02} \\ m_{x02} & m_{y02} & m_{z02} \\ n_{x02} & n_{y02} & n_{z02} \end{Bmatrix} = \begin{Bmatrix} l_{x01} & l_{y01} & l_{z01} \\ m_{x01} & m_{y01} & m_{z01} \\ n_{x01} & n_{y01} & n_{z01} \end{Bmatrix} \begin{Bmatrix} l_{x12} & l_{y12} & l_{z12} \\ m_{x12} & m_{y12} & m_{z12} \\ n_{x12} & n_{y12} & n_{z12} \end{Bmatrix} \qquad (1.51)$$

1.7.4 Passive Transformation von Vektoren

Die passive Transformation von Gl. 1.28 wird nun allgemein hergeleitet. Die drei Einheitsvektoren $\vec{e}_x, \vec{e}_y, \vec{e}_z$ spannen das Bezugskoordinatensystem auf. In diesem System können Punkte und Vektoren angegeben werden. Ein Vektor kann z. B. als Ortsvektor vom Ursprung zu einem Punkt P mit den Koordinaten (x_p, y_p, z_p) zeigen (siehe Bild 1.28):

$$\vec{p} = x_p\,\vec{e}_x + y_p\,\vec{e}_y + z_p\,\vec{e}_z \qquad (1.52)$$

In einem verdrehten Koordinatensystem S* gibt es einen Vektor $\vec{p}^{\,*}$, der auf den gleichen Punkt zeigt.

$$\vec{p}^{\,*} = x_p\,\vec{e}_x^{\,*} + y_p\,\vec{e}_y^{\,*} + z_p\,\vec{e}_z^{\,*} \qquad (1.53)$$

Da die Vektoren vom Ursprung auf den gleichen Punkt zeigen, sind sie gleich, d. h. $\vec{p} = \vec{p}\,*$, nur daß die Koordinaten in unterschiedlichen Koordinatensystemen angegeben worden sind:

$$x_p^*\vec{e}_x^{\,*} + y_p^*\vec{e}_y^{\,*} + z_p^*\vec{e}_z^{\,*} = x_p\vec{e}_x + y_p\vec{e}_y + z_p\vec{e}_z \qquad (1.54)$$

Die Vektoren können auch mit Hilfe von Spalten- und Zeilenmatrizen auch geschrieben werden:

$$\begin{Bmatrix} \vec{e}_x^{\,*} & \vec{e}_y^{\,*} & \vec{e}_z^{\,*} \end{Bmatrix} \begin{Bmatrix} x_p^* \\ y_p^* \\ z_p^* \end{Bmatrix} = \begin{Bmatrix} \vec{e}_x & \vec{e}_y & \vec{e}_z \end{Bmatrix} \begin{Bmatrix} x_p \\ y_p \\ z_p \end{Bmatrix} \qquad (1.55)$$

Die Einheitsvektoren transformieren nach Gl. 1.36, d. h.

$$\begin{Bmatrix} \vec{e}_x^{\,*} & \vec{e}_y^{\,*} & \vec{e}_z^{\,*} \end{Bmatrix} = \begin{Bmatrix} \vec{e}_x & \vec{e}_y & \vec{e}_z \end{Bmatrix} \cdot \mathbf{A}^T$$

Damit und mit Gl. 1.39, nach der $A^T = B$ gilt, ergibt sich

$$\begin{Bmatrix} \vec{e}_x & \vec{e}_y & \vec{e}_z \end{Bmatrix} \cdot \mathbf{B} \cdot \begin{Bmatrix} x_p^* \\ y_p^* \\ z_p^* \end{Bmatrix} = \begin{Bmatrix} \vec{e}_x & \vec{e}_y & \vec{e}_z \end{Bmatrix} \begin{Bmatrix} x_p \\ y_p \\ z_p \end{Bmatrix} \qquad (1.56)$$

Beide Seiten dieser Gleichnung haben die gleiche Zeilenmatrix als linksseitigen Faktor. Daher muß für die übrigen Terme gelten

$$\begin{Bmatrix} x_p \\ y_p \\ z_p \end{Bmatrix} = \mathbf{B} \cdot \begin{Bmatrix} x_p^* \\ y_p^* \\ z_p^* \end{Bmatrix} \qquad \text{oder} \qquad \vec{p} = \mathbf{B} \cdot \vec{p}^{\,*} \qquad (1.57)$$

Diese Beziehung besagt folgendes:

Die Koordinaten eines beliebigen Ortsvektors $\vec{p}^*$, angeordnet in einer Spaltenmatrix, werden von der Matrix **B** vom System **S*** in das System **S** transformiert. Da der Vektor dabei nicht bewegt wird, heißt diese Transformation *passiv*.

Die Elemente der Spaltenmatrix bedeuten auch die Koordinaten des Ortsvektors bei einer Zerlegung entlang der Koordinatenachsen. Man kann den Vektor daher auch als Spaltenmatrix auffassen und schreiben

$$\mathbf{p} = \begin{Bmatrix} x_p \\ y_p \\ z_p \end{Bmatrix} \qquad \text{bzw.} \qquad \mathbf{p}^* = \begin{Bmatrix} x_p^* \\ y_p^* \\ z_p^* \end{Bmatrix} \tag{1.58}$$

Mit den Definitionen von Gl. 1.58 erhält man die Gl. 1.57 in der Form

$$\mathrm{p} = \mathbf{B}\ \mathrm{p}^* \tag{1.59}$$

B transformiert also den Vektor $\vec{p}^*$ von S* nach S oder anders ausgedrückt: Durch Multiplikation mit B werden die Koordinaten des Vektors vom *-system in das Bezugssystem *umgerechnet*. Die Transformation ist also *passiv*.

Die Tranformation kann auch in die andere Richtung erfolgen, wenn die Inverse B^{-1} von B bekannt ist. Multipliziert man beide Seiten von Gl. 1.59 mit B^{-1} so ergibt sich

$$B^{-1}p = B^{-1}B \tag{1.60}$$

Aufgrund der Eigenschaft der Inverse gilt $B^{-1}B = I$, ist also eine Einheitsmatrix, so daß sich ergibt

$$p^* = B^{-1}\ p \tag{1.61}$$

Die Frage ist nun, wie man die Inverse einer Orientierungsmatrix findet.

1.7.5 Berechnung der Inversen einer Orientierungsmatrix

Die Vektoren $\vec{p}$ und $\vec{p}^*$ sind gleich lang (siehe Bild 1.28). Daher muß gelten

$$p^{*T}\ p^* = p^Tp \tag{1.62}$$

Diese Ausdrücke stellen die Summe der Koordinatenquadrate dar. Laut Gl. 1.59 gilt p = B p*

Damit wird aus Gl. 1.62

$$p^{*T}\ p^* = p^{*T}B^TB\ p^* \tag{1.63}$$

Die Gültigkeit dieser Gleichung erfordert, daß das Produkt der Matrizen eine Einheitsmatrix ergibt, daß also gilt

$$B^TB = I \tag{1.64}$$

Daraus folgt nun unmittelbar

$$B^T = B^{-1} \tag{1.65}$$

Bei allen Orientierungsmatrizen ist daher die Inverse durch die Transponierte gegeben. Diese Eigenschaft bezeichnet man als orthogonal.

Orthogonal

Eine Matrix ist orthogonal, wenn gilt: $\mathbf{A}^{-1} = \mathbf{A}^{T}$ und $|\mathbf{A}| \neq 0$.

Aus der Beziehung $\mathbf{A}\,\mathbf{A}^{T} = \mathbf{I}$ folgt

$$\begin{Bmatrix} l_x & m_x & n_x \\ l_y & m_y & n_y \\ l_z & m_z & n_z \end{Bmatrix} \begin{Bmatrix} l_x & l_y & l_z \\ m_x & m_y & m_z \\ n_x & n_y & n_z \end{Bmatrix} = \begin{Bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{Bmatrix} \tag{1.66}$$

Diese Matrizengleichung führt zu den linearen Gleichungssystemen

$$\begin{aligned} l_x^2 + m_x^2 + n_x^2 &= 1 \\ l_y^2 + m_y^2 + n_y^2 &= 1 \\ l_z^2 + m_z^2 + n_z^2 &= 1 \end{aligned} \tag{1.67}$$

und

$$\begin{aligned} l_x l_y + m_x m_y + n_x n_y &= 0 \\ l_x l_z + m_x m_z + n_x n_z &= 0 \\ l_y l_z + m_y m_z + n_y n_z &= 0 \end{aligned} \tag{1.68}$$

Aus Symmetriegründen treten die Gln. 1.68 zweimal auf. Die Bedeutung der Gleichungen läßt sich erhellen, wenn man Zwischenwinkel einführt. Das sind die Winkel zwischen den Achsen des Bezugssystems und dem verdrehten System [4].

Die Winkel α_x, α_y, α_z mögen zwischen der x-Achse und den im Index angegebenen Achsen des *-Systems liegen. Entsprechendes gilt für Winkel β und γ für die anderen Achsen des Systems. Die Koordinaten lassen sich nun durch die Kosinuswerte der Zwischenwinkel ausdrücken, so daß sich die folgenden Gleichungen ergeben:

$$\begin{aligned} \cos^2\alpha_x + \cos^2\beta_x + \cos^2\gamma_x &= 1 \\ \cos^2\alpha_y + \cos^2\beta_y + \cos^2\gamma_y &= 1 \\ \cos^2\alpha_z + \cos^2\beta_z + \cos^2\gamma_z &= 1 \end{aligned} \tag{1.69}$$

$$\begin{aligned} \cos\alpha_x \cos\alpha_y + \cos\beta_x \cos\beta_y + \cos\gamma_x \cos\gamma_y &= 0 \\ \cos\alpha_x \cos\alpha_z + \cos\beta_x \cos\beta_z + \cos\gamma_x \cos\gamma_z &= 0 \\ \cos\alpha_y \cos\alpha_z + \cos\beta_y \cos\beta_z + \cos\gamma_y \cos\gamma_z &= 0 \end{aligned} \tag{1.70}$$

Da es für neun Winkel nur sechs Gleichungen gibt, sind die Zwischenwinkel voneinander abhängig.

1.7.6 Drehungen um Koordinatenachsen

In der Praxis muß eine allgemeine Orientierung über die Drehung um drei unabhängige Achsen hergestellt werden (Eulermethode). Die dafür erforderlichen Rotationsmatrizen werden nachfolgend hergeleitet. Eine Matrix, die eine Rotation um eine Achse des Koordinatensystems bewirkt, ist eine spezielle Form der Orientierungsmatrix B und soll als Drehmatrix mit dem Buchstaben R bezeichnet werden. Im weiteren werden folgende Bezeichnungen verwendet:

$\mathbf{R_x}(\alpha)$	Drehung um die x-Achse mit Drehwinkel α
$\mathbf{R_y}(\beta)$	Drehung um die y-Achse mit Drehwinkel β
$\mathbf{R_z}(\gamma)$	Drehung um die z-Achse mit Drehwinkel γ

Bei der Drehung eines Koordinatensystems um eine Achse fällt der Einheitsvektor der Drehachse des gedrehten Systems mit einer Achse des Bezugssystems zusammen. und hat dort die Koordinate 1 und auf allen anderen Achsen des Bezugssystems die Koordinate 0. Die anderen Eiheitsvektoren des gedrehten Systems stehen auf dieser Achse senkrecht und haben daher dort ebenfalls die Koordinaten 0. Von den neun Koordinaten einer Rotationsmatrix liegen damit schon fünf fest.

Drehung um die z-Achse

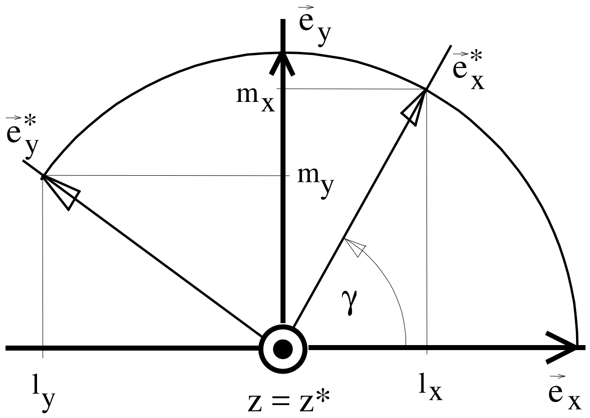


Bild 1.32: Drehung um die z-Achse

Die Drehung um die z-Achse zeigt Bild 1.32. Die Koordinaten der Einheitsvektoren können daraus abgelesen werden:

$l_x = \cos\gamma \quad l_y = -\sin\gamma \quad l_z = 0$
$m_x = \sin\gamma \quad m_y = \cos\gamma \quad m_z = 0$
$n_x = 0 \quad n_y = 0 \quad n_z = 0$

Damit erhält man die Drehmatrix

$$\mathbf{R_z}(\gamma) = \begin{Bmatrix} l_x & l_y & l_z \\ m_x & m_y & m_z \\ n_x & n_y & n_z \end{Bmatrix} = \begin{Bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{Bmatrix} \tag{1.71}$$

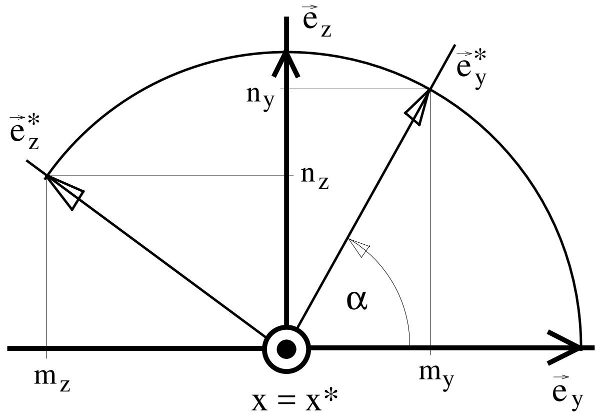


Bild 1.33: Drehung um die x-Achse

Die Drehung ist eine gerichtete Größe. Der Drehwinkel ist positiv, wenn die Drehung rechtsdrehend um die positive Achsrichtung erfolgt und negativ, wenn sie linksdrehend ist. Beispielsweise erfolgt die Drehung in Bild 1.32 rechtsdrehend um die positive z-Achse.

Zur Rücktransformation dient daher die gleiche Rotationsmatrix. Nur die Drehrichtung kehrt sich dabei um.

Drehung um die x-Achse

Aus Bild 1.33 liest man folgende Koordinaten der Einheitsvektoren ab:

$l_x = 1 \quad l_y = 0 \quad l_z = 0$
$m_x = 0 \quad m_y = \cos\alpha \quad m_z = -\sin\alpha$
$n_x = 0 \quad n_y = \sin\alpha \quad n_z = \cos\alpha$

Damit ergibt sich die Rotationsmatrix

$$R_X(\alpha) = \begin{Bmatrix} 1 & 0 & 0 \\ 0 & \mathbf{cos}\,\alpha & -\mathbf{sin}\,\alpha \\ 0 & \mathbf{sin}\,\alpha & \mathbf{cos}\,\alpha \end{Bmatrix} \tag{1.72}$$

Drehung um die y-Achse

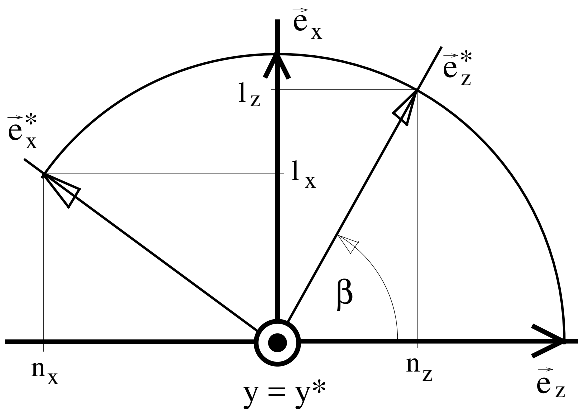


Bild 1.34: Drehung um die y-Achse

Die Komponenten der Einheitsvektoren von Bild 1.34 haben die Werte:

$l_x = \mathbf{cos}\beta$	$l_y = 0$	$l_z = \mathbf{sin}\beta$
$m_x = 0$	$m_y = 1$	$m_z = 0$
$n_x = \mathbf{-sin}\beta$	$n_y = 0$	$n_Z = \mathbf{cos}\beta$

Dazu gehört die Rotationsmatrix

$$\mathbf{R}_Y(\beta) = \begin{Bmatrix} \mathbf{cos}\,\beta & 0 & \mathbf{sin}\,\beta \\ 0 & 1 & 0 \\ -\mathbf{sin}\,\beta & 0 & \mathbf{cos}\,\beta \end{Bmatrix} \tag{1.73}$$

Die Ergebnisse der Drehungen um die drei Koordinatenachsen lauten zusammengefaßt:

$$\mathbf{R}_X(\alpha) = \begin{Bmatrix} 1 & 0 & 0 \\ 0 & \mathbf{cos}\,\alpha & -\mathbf{sin}\,\alpha \\ 0 & \mathbf{sin}\,\alpha & \mathbf{cos}\,\alpha \end{Bmatrix} \quad \mathbf{R}_Y(\beta) = \begin{Bmatrix} \mathbf{cos}\,\beta & 0 & \mathbf{sin}\,\beta \\ 0 & 1 & 0 \\ -\mathbf{sin}\,\beta & 0 & \mathbf{cos}\,\beta \end{Bmatrix} \quad \mathbf{R}_Z(\gamma) = \begin{Bmatrix} \mathbf{cos}\,\gamma & -\mathbf{sin}\,\gamma & 0 \\ \mathbf{sin}\,\gamma & \mathbf{cos}\,\gamma & 0 \\ 0 & 0 & 1 \end{Bmatrix} \tag{1.74}$$

1.7.7 Mehrere Drehungen in Folge

Jede beliebige Drehlage läßt sich durch aufeinanderfolgende Drehungen um drei verschiedene Achsen eines Koordinatensystems herbeiführen. Dies bedeutet, daß eine allgemeine Drehlage durch die Angabe von drei Winkeln α, β, γ (Eulerwinkel) eindeutig bestimmt ist.

Das Basissystem S_0 soll also durch $R_x(\alpha)$ zunächst in das System S_1 dann durch $R_y(\beta)$ in das System S_2 und schließlich durch $R_z(\gamma)$ in das System S_3 transformiert werden. Dazu wird von Gl. 1.45 Gebrauch gemacht. Die erste Drehung R_x erfolgt um die x-Achse des Bezugssystems:

$$B_{xyz} = R_X(\alpha)\, R_Y(\beta)\, R_Z(\gamma) \tag{1.75}$$

Durch Einsetzen der Drehmatrizen erhält man

$$\mathbf{B}_{xyz} = \begin{Bmatrix} 1 & 0 & 0 \\ 0 & \mathbf{cos}\,\alpha & -\mathbf{sin}\,\alpha \\ 0 & \mathbf{sin}\,\alpha & \mathbf{cos}\,\alpha \end{Bmatrix} \begin{Bmatrix} \mathbf{cos}\,\beta & 0 & \mathbf{sin}\,\beta \\ 0 & 1 & 0 \\ -\mathbf{sin}\,\beta & 0 & \mathbf{cos}\,\beta \end{Bmatrix} \begin{Bmatrix} \mathbf{cos}\,\gamma & -\mathbf{sin}\,\gamma & 0 \\ \mathbf{sin}\,\gamma & \mathbf{cos}\,\gamma & 0 \\ 0 & 0 & 1 \end{Bmatrix} \tag{1.76a}$$

Die schrittweise Ausmultiplikation liefert

$$\mathbf{B}_{xyz} = \begin{Bmatrix} \cos\beta & 0 & \sin\beta \\ \sin\alpha\sin\beta & \cos\alpha & -\sin\alpha\cos\beta \\ -\cos\alpha\sin\beta & \sin\alpha & \cos\alpha\cos\beta \end{Bmatrix} \begin{Bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{Bmatrix} \qquad \text{(b)}$$

$$\mathbf{B}_{xyz} = \begin{Bmatrix} \cos\beta\cos\gamma & -\cos\beta\sin\gamma & \sin\beta \\ \sin\alpha\sin\beta\cos\gamma + \cos\alpha\sin\gamma & -\sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & -\sin\alpha\cos\beta \\ -\cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma & \cos\alpha\sin\beta\sin\gamma + \sin\alpha\cos\gamma & \cos\alpha\cos\beta \end{Bmatrix} \qquad \text{(c)}$$

Die Reihenfolge der Indizes von B von links nach rechts gibt die Reihenfolge der Drehungen an. Je nach der Reihenfolge, ergibt sich eine andere Matrix B, da die Matrizenmultiplikation nicht kommutativ ist. Damit sich die gleiche Orientierung ergibt, müssen die Winkel α, β, γ bei jeder Änderung der Reihenfolge anders gewählt werden. Bei gleichen Winkeln α, β, γ gilt daher

$$B = \begin{Bmatrix} l_x & l_y & l_z \\ m_x & m_y & m_z \\ n_x & n_y & n_z \end{Bmatrix} = B_{xyz} \neq B_{zxy} \neq B_{yzx} \neq B_{xzy} \neq B_{yxz} \neq B_{zyz} \qquad (1.77)$$

Dabei sind l_x bis n_z die Richtungskosinusse für die endgültige Orientierung bzw. die Koordinaten der verdrehten Einheitsvektoren. Aus der Orientierungsmatrix können die Winkel α, β, γ berechnet werden, die zu einer gewünschten Orientierung führen.

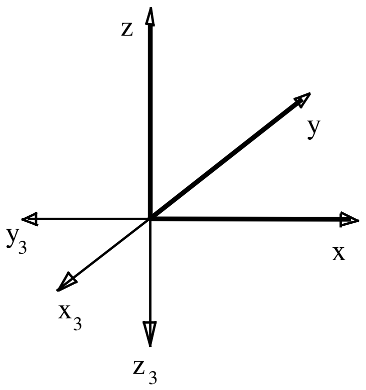


Bild 1.35: Verdrehtes Koordinatensystem

Beispiel 5: Drehungen um mehrere Achsen

Welche Drehwinkel α, β, γ sind erforderlich, um die Orientierung eines Effektors nach Bild 1.35 zu erreichen, wenn folgende *Zwischen*winkel gegeben sind?

$$\alpha_x = 90^\circ \quad \alpha_y = 180^\circ \quad \alpha_z = 90^\circ$$
$$\beta_x = 180^\circ \quad \beta_y = 90^\circ \quad \beta_z = 90^\circ$$
$$\gamma_x = 90^\circ \quad \gamma_y = 90^\circ \quad \gamma_z = 180^\circ$$

Aus den Zwischenwinkeln ergeben sich die Koordinaten der Eiheitsvektoren

$$l_x = \cos\alpha_x = 0, \quad l_y = \cos\alpha_y = -1, \quad l_z = \cos\alpha_z = 0$$
$$m_x = \cos\beta_x = -1, \quad m_y = \cos\beta_y = 0, \quad m_z = \cos\beta_z = 0$$
$$n_x = \cos\gamma_x = 0, \quad n_y = \cos\gamma_y = 0, \quad n_z = \cos\gamma_z = -1$$

Diese Einheitsvektoren müssen nun mit den Winkeln der Drehmatrix erzeugt werden.

$B_{xyz} =$

$$\begin{Bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{Bmatrix} = \begin{Bmatrix} \cos\beta\cos\gamma & -\cos\beta\sin\gamma & \sin\beta \\ \sin\alpha\sin\beta\cos\gamma + \cos\alpha\sin\gamma & -\sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & -\sin\alpha\cos\beta \\ -\cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma & \cos\alpha\sin\beta\sin\gamma + \sin\alpha\cos\gamma & \cos\alpha\cos\beta \end{Bmatrix}$$

(1.78)

Aus dieser Matrizengleichung können die Drehwinkel α, β, γ bestimmt werden. Dazu entnimmt man folgende Bestimmungsgleichungen:

1. $\sin\beta = 0 \quad \rightarrow \beta = 0^o \quad (180^o)$
2. $\cos\alpha\cos\beta = -1 \quad \rightarrow \alpha = 180^o \quad (0^o)$
3. $-\cos\beta\sin\gamma = -1 \quad \rightarrow \gamma = 90^o \quad (-90^o)$

Aus der ersten Gleichung folgt $\beta = 0°$, aus der zweiten damit $\alpha = 180°$ und aus der dritten $\gamma = 90°$. Eine Drehung um die y-Achse ist also nicht erforderlich, da $\beta = 0°$ gilt. Die neue Lage wird erreicht, indem zunächst mit Winkel $\alpha = 180°$ um die x-Achse und dann mit $\gamma = 90°$ um die z-Achse gedreht wird (Reihenfolge für B: α, β, γ). Die Lösung ist nicht eindeutig. In Klammern ist eine alternative Löung angegeben.

Die beiden Drehungen sind in Bild 1.36 und 1.37 dargestellt.

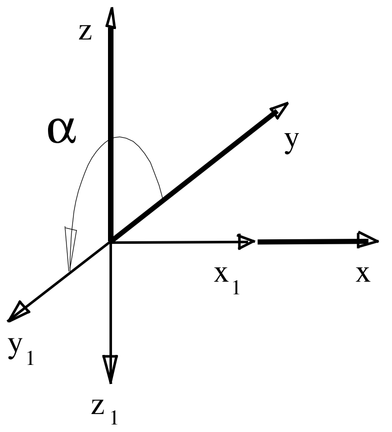


Bild 1.36

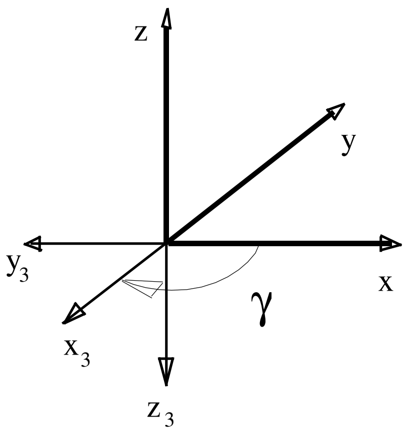


Bild 1.37

1.7.8 Drehungen um die Achsen des Bezugssystems

Gelegentlich wird auch verlangt, ein Koordinatensystem um die Achsen des Bezugssystems zu drehen. Wenn ein Koordinatensystem, das anfänglich parallel zum Bezugssystem liegt, erstmalig gedreht wird, erfolgt die Drehung immer um eine Achse des Bezugssystems. Wie kann man aber erreichen, daß dies bei der nächste Drehung wieder der Fall ist?

Man betrachte dazu Gl. 1.43. Dort ist $\mathbf{B}_{01}$ die erste Drehmatrix. Sie dreht die um die Achse des Bezugssystems. Man muß also ein Schema verwenden, bei dem die nächste Drehmatrix links von $\mathbf{B}_{01}$ steht und damit sozusagen zu einer „ersten" Drehmatrix wird. Wenn man schreibt

$$\mathbf{S}_1 = \mathbf{R}_x \cdot \mathbf{S}_0 \qquad (1.79)$$

dann wird $\mathbf{S}_0$ um die x-Achse des Bezugssystems ins System $\mathbf{S}_1$ gedreht. Die Vermutung ist daher, daß eine linksseitige Multiplikation von $\mathbf{S}_1$ mit einer Drehmatrix auch eine Drehung um eine Achse des Bezugssystems ausführt, z. B. mit $\mathbf{R}_y$,

$$\mathbf{S}_2 = \mathbf{R}_y \cdot \mathbf{S}_1 \qquad (1.80)$$

Die Gesamtdrehung ergibt sich damit zu

$$\mathbf{S}_2 = \mathbf{R}_y \cdot \mathbf{S}_1 = \mathbf{R}_y \cdot \mathbf{R}_x \cdot \mathbf{S}_0 \qquad (1.81)$$

Diese Ergebnis soll an einem Beispiel überprüft werden. Für $\alpha = 90°$ und $\beta = 90°$ gilt

$$\mathbf{R}_x = \begin{Bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{Bmatrix} = \begin{Bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{Bmatrix} \quad (1.82)$$

$$\mathbf{R}_y = \begin{Bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{Bmatrix} = \begin{Bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{Bmatrix} \quad (1.83)$$

Damit erhält man aus Gl. 1.81

$$\mathbf{S}_2 = \mathbf{R}_y \mathbf{R}_x \mathbf{S}_0 = \begin{Bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{Bmatrix} \begin{Bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{Bmatrix} \mathbf{S}_0 = \begin{Bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{Bmatrix} \mathbf{S}_0 \quad (1.84)$$

Da $\mathbf{S}_0$ eine Einheitsmatrix ist, ist $\mathbf{S}_2$ gegeben durch

$$\mathbf{S}_2 = \begin{Bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{Bmatrix} \quad (1.85)$$

Diese Ergebnis läßt sich graphisch überprüfen. Die erste Drehung um 90° um die x-Achse des Bezugssystems ist in Bild 1.38 dargestellt. Der Index 1 kennzeichnet das verdrehte System $\mathbf{S}_1$.

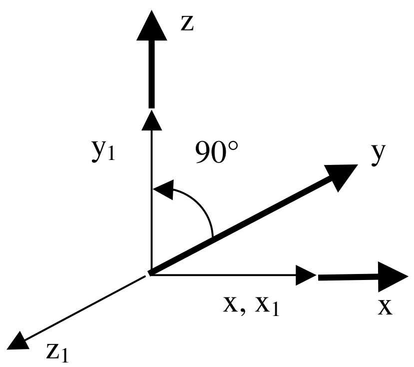


Bild 1.38: Drehung um die x-Achse

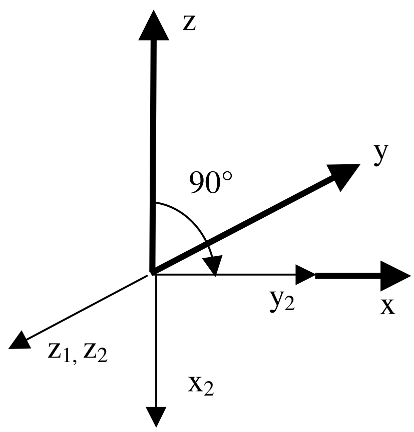


Bild 1.39: Drehung um die y-Achse

In Bild 1.39 wurde $\mathbf{S}_1$ um 90° um die y-Achse des Bezugssystems gedreht. Diese zweite Drehung führt zum System $\mathbf{S}_2$. Für dieses System liest man aus Bild 1.39 nun ab

$$\mathbf{S}_2 = \begin{Bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{Bmatrix} \quad (1.86)$$

Der Vergleich der Ergebnisse der Gln. 1.85 und 1.86 zeigt Übereinstimmung. Das Schema der Gln. 1.79 bis 1.81 bewirkt also tätsächlich eine Drehung der Koordinatensysteme um die Achsen des Bezugssystems.

1.8 Homogene Koordinaten

Homogene Koordinaten [4] sind ein Begriff aus der Projektiven Geometrie. Sie dienen zur einheitlichen behandlung von Translation und Rotation mit Hilfe einer Matrixmultiplikation. Die Koordinaten x, y, z eines Raumpunktes P werden dabei nach den Beziehungen

$$x = \frac{x_h}{t} \quad y = \frac{y_h}{t} \quad z = \frac{z_h}{t} \qquad (1.87)$$

definiert. Diese gehen aus den Koordinaten x_h, y_h, z_h, und der Größe t hervor, die als homogene Koordinaten bezeichnet werden. Die Größe t stellt offensichtlich einen Skalierungsfaktor dar. Homogene Koordinaten können durch eine Spaltenmatrix mit 4 Elementen dargestellt werden.

$$\begin{Bmatrix} x_h \\ y_h \\ z_h \\ t \end{Bmatrix} \qquad (1.88)$$

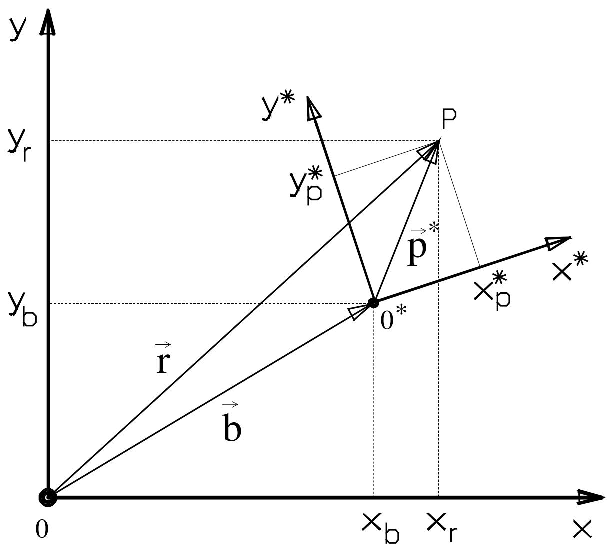


Bild 1.40: Rotation und Verschiebung eines Koordinatensystems

Im Folgenden werden homogene Koordinaten verwendet, um Drehung und Verschiebung von Koordinatensystemen in <u>einer</u> Transformationsmatrix zusammenzufassen. Dazu betrachten wir Bild 1.38. Es zeigt ein verdrehtes Koordinatensystem, dessen Ursprung 0* gegenüber dem Ursprung 0 des Bezugssystems um einen Vektor $\vec{b}$ verschoben ist.

Ein Punkt P des Raumes wird im Bezugssystem S = (x, y, z) beschrieben durch den Vektor $\vec{r}$.

Vom Ursprung 0* des Systems **S*** = (x*, y*, z*) zeigt ein Vektor $\vec{p}^*$ zum Punkt P. Verschiebt man S* in den Ursprung von S, dann kann nach Gl. 1.57 $\vec{p}^*$ mit Hilfe der Transformation

$$\vec{p} = \mathbf{B} \cdot \vec{p}^* \qquad (1.89)$$

in das Bezugssystem umgerechnet werden. Der Vektor $\vec{r}$ ergibt sich dann aus der Beziehung

$$\vec{r} = \vec{p} + \vec{b} = \mathrm{B} \ \vec{p}^* + \vec{b} \qquad (1.90)$$

Gesucht wird nun eine Transformationsmatrix, die den Vektor $\vec{p}^*$ in <u>einem</u> Schritt in den Vektor $\vec{r}$ überführt.

Dies erreicht man durch den Übergang auf homogene Koordinaten mit t = 1. Die Transformationsmatrix wird zu einer 4x4 Matrix erweitert, die in der 4ten Spalte die Komponenten des Veschiebevektors enthält. Gl. 1.83 zeigt den Aufbau dieser Transformationsmatrix T und die Vektoren mit den homogenen Koordinaten.

$$\vec{p}_h^{\,*} = \begin{Bmatrix} x_p^* \\ y_p^* \\ z_p^* \\ 1 \end{Bmatrix} \qquad \vec{r}_h = \begin{Bmatrix} x_r \\ y_r \\ z_r \\ 1 \end{Bmatrix} \qquad \mathbf{T} = \left\{\begin{array}{c:c} \mathbf{B} & \mathbf{b} \\ \hdashline 000 & 1 \end{array}\right\} \qquad \text{mit} \qquad \mathbf{b} = \begin{Bmatrix} x_b \\ y_b \\ z_b \end{Bmatrix} \tag{1.91}$$

Die Transformation läßt sich damit als Matrixprodukt wie folgt Schreiben.

$$\vec{r}_h = \mathrm{T}\ \vec{p}_h^{\,*} \qquad \text{oder ausführlich} \qquad \begin{Bmatrix} x_r \\ y_r \\ z_r \\ 1 \end{Bmatrix} = \left\{\begin{array}{ccc:c} l_x & l_y & l_z & x_b \\ m_x & m_y & m_z & y_b \\ n_x & n_y & n_z & z_b \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right\} \begin{Bmatrix} x_p^* \\ y_p^* \\ z_p^* \\ 1 \end{Bmatrix} \tag{1.92}$$

Diese 4x4 Transformationsmatrix für homogene Koordinaten führt die Rotation und die Verschiebung in einem Schritt aus. Sie enthält enthält die B-Matrix, die die Orientierung des Koordinatensystems S* im Bezugssystem beschreibt sowie einen Spaltenvektor mit den Komponenten des Verschiebevektors, angegeben im Bezugssystem. Auf die Kennzeichnung der homogenen Koordinaten mit dem tiefgestellten h wird im weiteren verzichtet. Ob ein Vektor rein kartesisch oder homogen ist, muß aus dem Zusammenhang entnommen werden.

Die T-Matrix enthält damit alle Informationen über die Stellung des Koordinatensystems S*, d. h. über die Position des Ursprungs und der Orientierung von S*.

Mit einer Neudefinition des Koordinatensystems in der homogenen Form

$$\mathbf{S} = \left\{\begin{array}{c} \vec{e}_x \vec{e}_y \vec{e}_z \vec{b} \\ \hdashline 0\ 0\ 0\ 1 \end{array}\right\} \tag{1.93}$$

kann die Koordinatentransformation nun analog zu Gl. 1.40 dargestellt werden

$$\mathbf{S}^* = \mathbf{S} \cdot \mathbf{T} \tag{1.94}$$

Vergleicht man dies mit Gl. 1.84, so fällt auf, daß die Matrix T links vom Vektor steht und vom *-System in das Bezugssystem transformiert (passive Transformation), währen in Gl. 1.86 die Matrix rechts vom System S steht und vom Bezugssystem in das *-system transformiert.

Wenn **S** weder verdreht noch verschoben ist, also das übergeordnete Bezugssystem darstellt, wird **S** durch eine Einheitsmatrix **I** dargestellt. Dann gilt analog zu Gl. 1.42

$$\mathbf{S}^* = \mathbf{T} \tag{1.95}$$

und das verdrehte und verschobene System wird mit **T** identisch.

Daher wird T auch als *Frame* bezeichnet.

Frame bedeutet auf deutsch *Rahmen* und steht hier für die Stellung eines Koordinatensystems, also für seine Position und Orientierung im Bezugssystem.

Mit der neuen Transformationsmatrix lassen sich auch reine Drehungen oder Translationen darstellen. Für eine reine Drehung braucht nur der Verschiebevektor Null gesetzt zu werden.

Reine Drehung

Der Verschiebevektor ist Null

$$\mathbf{T} = \left\{\begin{array}{ccc:c} l_x & l_y & l_z & 0 \\ m_x & m_y & m_z & 0 \\ n_x & n_y & n_z & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right\} \tag{1.96}$$

Reine Verschiebung

Bei einer reinen Verschiebung muß die B-Matrix eine Einheitsmatrix sein.

$$\mathbf{T} = \left\{\begin{array}{ccc:c} 1 & 0 & 0 & x_b \\ 0 & 1 & 0 & y_b \\ 0 & 0 & 1 & z_b \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right\} \tag{1.97}$$

Beispiel 7: Drehung und Verschiebung mit homogener Transformation

a) Reine Drehung

Ein System wird relativ zum Basissystem S_0 gedreht nach S_1

$$\mathbf{T}_{01} = \begin{Bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \tag{1.98}$$

b) Reine Verschiebung

Verschieben des gedrehten Systems S_1 nach S_2 parallel zu den Achsen von S_1

$$\mathbf{T}_{12} = \begin{Bmatrix} 1 & 0 & 0 & x_{b1} \\ 0 & 1 & 0 & y_{b1} \\ 0 & 0 & 1 & z_{b1} \\ 0 & 0 & 0 & 1 \end{Bmatrix} \tag{1.99}$$

Als Gesamttransformation ergibt sich

$$\mathbf{S}_1 = \mathbf{S}_0\mathbf{T}_{01} \qquad \text{Drehung} \tag{1.100}$$

$$\mathbf{S}_2 = \mathbf{S}_1\mathbf{T}_{12} \qquad \text{Verschiebun} \tag{1.101}$$

$$\mathbf{S}_2 = \mathbf{S}_0\mathbf{T}_{01}\mathbf{T}_{12} = \mathbf{S}_0\mathbf{T}_{02} \qquad \text{Drehung und Verschiebung} \tag{1.102}$$

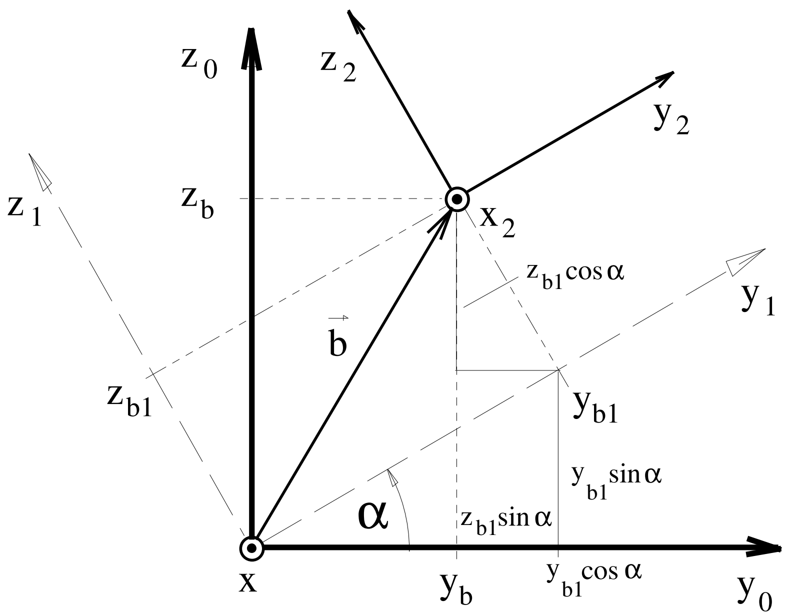
$$\mathbf{T_{02} = T_{01}T_{12}} = \begin{Bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \begin{Bmatrix} 1 & 0 & 0 & x_{b1} \\ 0 & 1 & 0 & y_{b1} \\ 0 & 0 & 1 & z_{b1} \\ 0 & 0 & 0 & 1 \end{Bmatrix} = \begin{Bmatrix} 1 & 0 & 0 & x_{b1} \\ 0 & \cos\alpha & -\sin\alpha & y_{b1}\cos\alpha - z_{b1}\sin\alpha \\ 0 & \sin\alpha & \cos\alpha & y_{b1}\sin\alpha + z_{b1}\cos\alpha \\ 0 & 0 & 0 & 1 \end{Bmatrix} \tag{1.103}$$

Für $\alpha = 30°$, $x_{b1} = 0$, $y_{b1} = 1$ und $z_{b1} = 0{,}5$ zeigt Bild 1.39 das Ergebnis. Aus dem geometrischen Zusammenhang des Bildes kann der Verschiebevektor in T_{02} nachvollzogen werden

$y_b = y_{b1}\,\mathbf{cos}\,\alpha - z_{b1}\,\mathbf{sin}\,\alpha$
$z_b = y_{b1}\,\mathbf{sin}\,\alpha + z_{b1}\,\mathbf{cos}\,\alpha$

Daraus wird folgendes deutlich:

Die Koordinaten des Verschiebevektors werden im Frame (bzw. in der Transformationsmatrix) stets im Bezugssystems angegeben.



Mit anderen Worten, das System S_2 wurde parallel zu den Achsen von S_1 verschoben. In der Verschiebematrix T_{12} stehen die Verschiebekoordinaten x_{b1}, y_{b1} und z_{b1} im System S_1, das für diese Verschiebung das Bezugssystem darstellt.

Durch die Ausmultiplikation der Transformationsmatrizen werden diese Koordinaten umgerechnet in das Bezugssystem S_0.

Bild 1.41: Drehung und Verschiebung

Beispiel 8: Bestückungsroboter mit Kamera

Der Bestückungs-Roboter von Bild 1.40 holt Teile mit einem Sauger aus einem Magazin und läßt jedes Teil von einem Kamerasensor einmessen. Dieser bestimmt in seinem Koordinatensystem (Index K) die Position des Werkstück-Koordinatensystems (Index W) und den Verdrehwinkel γ. Welchen Frame hat das Werkstück im Basis-Koordinatensystem (Index 0)?

Man bestimmt zunächst den Frame des K-Systems im 0-System und dann den Frame des W-Systems im K-System. Die Multiplikation ergibt dann den gesuchten Frame bzw., was das gleiche ist, die Transformation vom 0- ins W-System. Die Transformation vom 0-System ins K-System erfordert zwei Verschiebungen s_{x0} und $-s_{y0}$, also entlang der positiven x- und der negativen y-Achse. Außerdem muß mit Winkel $\beta = -90°$ um die y_K-Achse gedreht werden.

Damit lautet die Transformationsmatrix

$$\mathbf{T_{0K}} = \begin{Bmatrix} \cos\beta & 0 & \sin\beta & s_{x0} \\ 0 & 1 & 0 & -s_{y0} \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \tag{1.104}$$

Mit β = -90° ergibt sich

$$\mathbf{T_{0K}} = \begin{Bmatrix} 0 & 0 & -1 & s_{x0} \\ 0 & 1 & 0 & -s_{y0} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \tag{1.105}$$

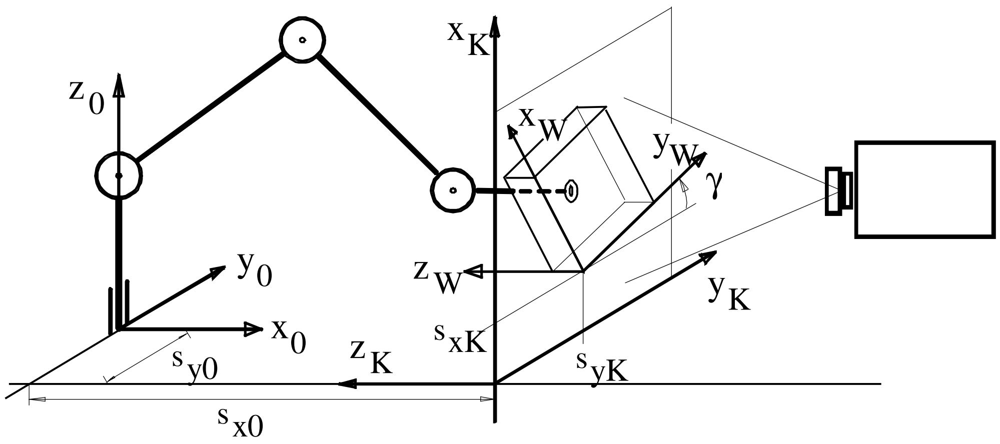


Bild 1.42: Lagevermessung an Werkstücken

Vom K-System zum W-System gelangt man über die Verschiebung s_{xK} entlang der x_K-Achse und der Verschiebung s_{yK} entlang der y_K-Achse. Außerdem ist um den Winkel γ um die z_W-Achse zu drehen. Daher ergibt sich

$$\mathbf{T_{KW}} = \begin{Bmatrix} \cos\gamma & -\sin\gamma & 0 & s_{xK} \\ \sin\gamma & \cos\gamma & 0 & s_{yK} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \tag{1.106}$$

Der eingezeichnete Drehwinkel γ stellt eine Linkdrehung dar. Beim Einsetzen eines Zahlenwertes wäre dies noch zu berücksichtigen. Der gesuchte Frame (und gleichbedeutend die gesuchte Transformation) folgt nun aus der Beziehung

$$\mathbf{S}_K = \mathbf{S}_0 \mathbf{T}_{0K} \tag{1.107}$$

$$\mathbf{S}_W = \mathbf{S}_K \mathbf{T}_{KW} \tag{1.108}$$

$$\mathbf{S}_W = \mathbf{S}_0 \mathbf{T}_{0K} \mathbf{T}_{KW} = \mathbf{S}_0 \mathbf{T}_{0W} \tag{1.109}$$

$$\mathbf{T}_{0\mathrm{W}} = \mathbf{T}_{0\mathrm{K}}\mathbf{T}_{\mathrm{KW}} = \begin{Bmatrix} 0 & 0 & -1 & s_{x0} \\ 0 & 1 & 0 & -s_{y0} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \begin{Bmatrix} \cos\gamma & -\sin\gamma & 0 & s_{xK} \\ \sin\gamma & \cos\gamma & 0 & s_{yK} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} = \begin{Bmatrix} 0 & 0 & -1 & s_{x0} \\ \sin\gamma & \cos\gamma & 0 & s_{yK} - s_{yK} \\ \cos\gamma & -\sin\gamma & 0 & s_{xK} \\ 0 & 0 & 0 & 1 \end{Bmatrix} \tag{1.110}$$

2 Kinematische Gleichungen

Die Bewegungen von IR werden durch die Verstellung der Gelenke erzeugt. Die Bewegungsgleichungen sind Funktionen der Gelenkvariablen. Sie werden benötigt, um die erforderlichen Sollwerte der Gelenkvariablen für Bahnfahraufgaben zu berechnen. An einem einfachen Beispiel soll dieses Problem zunächst auf direktem Wege gelöst werden. Danach wird gezeigt, daß die Lösung auch mit Hilfe von Koordinatentransformationen gefunden werden kann. Dieser Weg ist in dem einfachen Beispiel zwar aufwendiger, aber für allgemeinere Problemstellungen der geeignetere Lösungsweg.

2.1 Kinematische Gleichungen eines Roboterarmes

Gegeben sei ein Roboterarm nach Bild 2.1, dessen Aufgabe darin bestehen soll, den Effektor unter einem festem Winkel φ auf einer geraden Bahn an der Oberfläche des schraffiert gezeichneten Werkstückes entlangzuführen. Man kann sich vorstellen, daß ein solches Problem bei der Bearbeitung von Oberflächen auftritt (z. B. Auftrag von Kleber oder Farbe, Abschleifen von Graten).

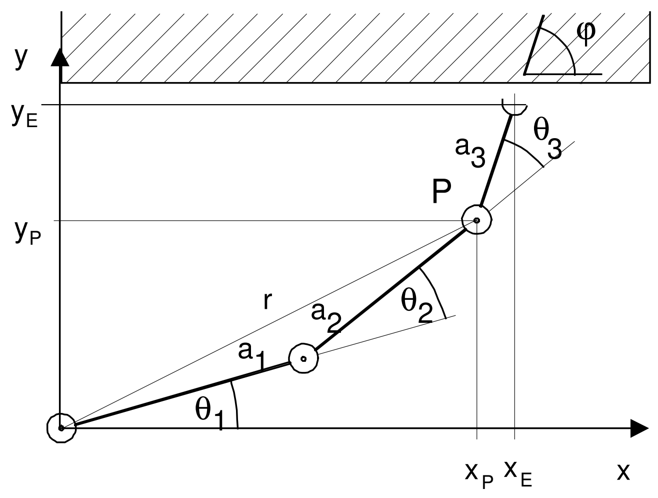


Bild 2.1: Gelenkarm mit 3 Freiheitsgraden

Gesucht wird nun der zeitliche Verlauf der Gelenkwinkel θ_1, θ_2 und θ_3 als Funktion der Zeit für eine konstante Geschwindigkeit v des Effektorbezugspunktes (x_E, y_E) entlang dem Werkstück. Der Effektorbezugspunkt wird auch als Tool Centre Point (TCP) bezeichnet wird. Seine Koordinaten lassen sich wie folgt angeben:

$$x_E = a_1 \cos\theta_1 + a_2 \cos(\theta_1 + \theta_2) + a_3 \cos(\theta_1 + \theta_2 + \theta_3) \tag{2.1}$$

$$y_E = a_1 \sin\theta_1 + a_2 \sin(\theta_1 + \theta_2) + a_3 \sin(\theta_1 + \theta_2 + \theta_3) \tag{2.2}$$

Diese Gleichungen liefern abhängig von den Gelenkparametern - den Drehwinkeln θ_1, θ_2 und θ_3 - die Koordinaten des TCP im Bezugssystem. Eine solche Transformation wird als *Vorwärtstransformation* bezeichnet.

Die Berechnung der Gelenkparameter aus den den Koordinaten des Bezugssystems dagegen heißt *Rückwärtstransformation*.

Diese soll nun bestimmt werden. Dazu werden zunächst die Winkel θ_1 und θ_2 aus den Koordinaten x_P und y_P des Punktes P berechnet. Die Koordinaten von P sind wie folgt gegeben:

$$x_P = a_1 \cos\theta_1 + a_2 \cos(\theta_1 + \theta_2) \tag{2.3}$$
$$y_P = a_1 \sin\theta_1 + a_2 \sin(\theta_1 + \theta_2) \tag{2.4}$$

Quadriert man beide Gleichungen und addiert sie anschließend, so erhält man:

$$x_P^2 + y_P^2 = a_2^2 \cos^2(\theta_1 + \theta_2) + 2a_1 a_2 \cos(\theta_1 + \theta_2) \cos\theta_1 + a_1^2 \cos^2(\theta_1) + a_2^2 \sin^2(\theta_1 + \theta_2) + 2a_1 a_2 \sin(\theta_1 + \theta_2) \sin\theta_1 + a_1^2 \sin^2(\theta_1) \tag{2.5}$$

Unter Berücksichtigung der Beziehungen

$\cos^2\alpha + \sin^2\beta = 1$ und $\cos\alpha \cos\beta + \sin\alpha \sin\beta = \cos(\alpha-\beta)$ ergibt sich

$$x_P^2 + y_P^2 = a_2^2 + a_1^2 + 2a_1 a_2 \cos(\theta_2) \tag{2.6}$$

Hieraus erhält man

$$\cos\theta_2 = \frac{x_p^2 + y_p^2 - a_1^2 - a_2^2}{2a_1 a_2} \tag{2.7}$$

Multipliziert man weiterhin die erste der obigen Gleichung mit y_P und die zweite mit x_P, so folgt

$$y_P x_P = y_P\left[a_1 \cos\theta_1 + a_2 \cos(\theta_1 + \theta_2)\right] \tag{2.8}$$
$$x_P y_P = x_P\left[a_1 \sin\theta_1 + a_2 \sin(\theta_1 + \theta_2)\right] \tag{2.9}$$

Da die linken Seiten gleich sind, gilt also

$$y_P\left[a_1 \cos\theta_1 + a_2 \cos(\theta_1 + \theta_2)\right] = x_P\left[a_1 \sin\theta_1 + a_2 \sin(\theta_1 + \theta_2)\right] \tag{2.10}$$

Mit Hilfe der Additionstheoreme

$$\cos(x + y) = \cos x \cos y - \sin x \sin y$$
$$\sin(x + y) = \sin x \cos y + \cos x \sin y$$

wird daraus

$$y_P\left[a_1 \cos\theta_1 + a_2 \cos\theta_1 \cos\theta_2 - \sin\theta_1 \sin\theta_2\right] = x_P\left[a_1 \sin\theta_1 + a_2 \sin\theta_1 \cos\theta_2 + \cos\theta_1 \sin\theta_2\right] \tag{2.11}$$

Teilt man nun durch $\cos\theta_1$, so ergibt sich eine Gleichung, die nach $\tan\theta_1$ aufgelöst werden kann.

$$y_P\left[a_1 + a_2(\cos\theta_2 - \tan\theta_1 \sin\theta_2)\right] = x_P\left[a_1 \tan\theta_1 + a_2(\tan\theta_1 \cos\theta_2 + \sin\theta_2)\right] \tag{2.12}$$

Die einzige Unbekannte in dieser Gleichung ist $\tan\theta_1$. Die Auflösung ergibt $\mathbf{cos}\,\theta_2 = \dfrac{x_p^{\;2} + y_p^{\;2} - a_1^{\;2} - a_2^{\;2}}{2a_1a_2}$

$$\mathbf{tan}\,\theta_1 = \frac{y_P(a_1 + a_2\,\mathbf{cos}\,\theta_2) - x_P a_2\,\mathbf{sin}\,\theta_2}{x_P(a_1 + a_2\,\mathbf{cos}\,\theta_2) + y_P a_2\,\mathbf{sin}\,\theta_2} \tag{2.13}$$

Damit liegen zwei Gleichungssystem vor, aus denen die Gelenkwinkel θ_1 und θ_2 berechnet werden können. Wenn der Effektor mit konstanter Geschwindigkeit an dem Werkstück entlangfahren soll, sind $x_E(t) = v_{xE}t$ und $y_E(t)$ = konst vorgegeben, und die Gelenkwinkel ergeben sich durch folgenden Satz von Gleichungen:

$$x_P(t) = x_E(t) - a_3\,\mathbf{cos}\,\varphi = v_{xE}t - a_3\,\mathbf{cos}\,\varphi \qquad y_E = y_P = konst \tag{2.14}$$

$$\cos\theta_2(t) = \frac{x_P^2(t) + y_P^2 - a_1^2 - a_2^2}{2a_1a_2} \tag{2.15}$$

$$\mathbf{tan}\,\theta_1(t) = \frac{y_P(a_1 + a_2\,\mathbf{cos}\,\theta_2(t)) - x_P(t)a_2\,\mathbf{sin}\,\theta_2(t)}{x_P(t)(a_1 + a_2\,\mathbf{cos}\,\theta_2(t)) + y_P a_2\,\mathbf{sin}\,\theta_2(t)} \tag{2.16}$$

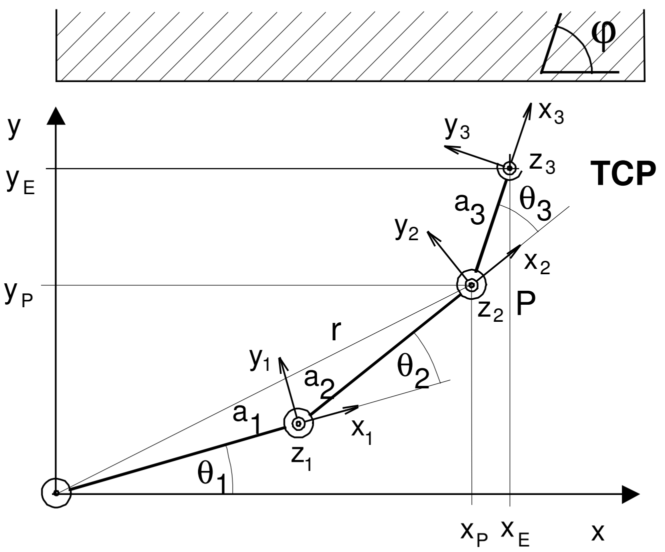


Bild 2.2: Gelenkarm mit Achskoordinaten

Für den Winkel θ_3 muß außerdem gelten:

$$\theta_3(t) = \varphi - \theta_1(t) - \theta_2(t) \tag{2.17}$$

Diese Gleichungen beschreiben, wie sich die Gelenkwinkel als Funktion der Zeit ändern müssen, damit der Effektor mit gleichmäßiger Geschwindigkeit parallel zur x-Achse an dem Werkstück entlangfährt.

Das einfache Beispiel soll nun nochmal verwendet werden, um zu zeigen, wie man die Lösung mit Hilfe von Transformationen finden kann. Danach soll diese Vorgehensweise dann als Methode zur Beschreibung der Kinematik von IR untermauert werden.

In Bild 2.2 wurden Koordinatensysteme in die Gelenke und in den TCP des Effektors eingezeichnet. Diese wurden so gelegt, daß die z-Achse jeweils mit der Gelenkachse zusammenfällt und die x-Achse die gleiche Richtung hat wie die Verbindungslinie zwischen zwei Gelenken. Im ersten Gelenk liegt das Bezugssystem $\mathbf{S}_0$ = (x, y, z). Gesucht ist also die Stellung von $\mathbf{S}_3$ im System $\mathbf{S}_0$, d. h. der Frame von $\mathbf{S}_3$. Diese Stellung besteht aus den Koordinaten x_E und y_E des TCP (Ursprung von $\mathbf{S}_3$) und aus der Matrix $\mathbf{B}_{03}$ von $\mathbf{S}_3$, die die Orientierung zu $\mathbf{S}_0$ beschreibt.

Beide Größen sind in der Transformationsmatrix $\mathbf{T}_{03}$ enthalten, die von $\mathbf{S}_0$ nach $\mathbf{S}_3$ transformiert. Die Transformationsmatrix wird gefunden, indem man schrittweise von $\mathbf{S}_0$ über $\mathbf{S}_1$ und $\mathbf{S}_2$ nach $\mathbf{S}_3$

transformiert. Dazu sind drei Rotationen und drei Verschiebungen eforderlich. Diese lauten im einzelnen:

$$\mathbf{T_{01}} = \begin{Bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & a_1\cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & 0 & a_1\sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \quad (2.18) \qquad \mathbf{T_{12}} = \begin{Bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & a_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & a_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \quad (2.19)$$

$$\mathbf{T_{23}} = \begin{Bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & a_3\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & a_3\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \quad (2.20)$$

Die vollständige Transformationsmatrix vom Basissystem S_0 nach S_3 lautet $T_{03} = T_{01}\ T_{12}\ T_{23}$. Die schrittweise Multiplikation der Matrizen ergibt:

$$\mathbf{T_{01}T_{12}} = \begin{Bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & a_1\cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & 0 & a_1\sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \begin{Bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & a_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & a_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} =$$

$$\begin{Bmatrix} \cos(\theta_1+\theta_2) & -\sin(\theta_1+\theta_2) & 0 & a_2\cos(\theta_1+\theta_2)+a_1\cos\theta_1 \\ \sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) & 0 & a_2\sin(\theta_1+\theta_2)+a_1\sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \quad (2.21)$$

Darin bedeuten

$$a_2\cos(\theta_1+\theta_2)+a_1\cos\theta_1 = x_P \quad (2.22)$$
$$a_2\sin(\theta_1+\theta_2)+a_1\sin\theta_1 = y_P$$

Damit läßt sich die obige Matrix kürzer schreiben. Man erhält die vollständige Transformation nun aus dem Matrizenprodukt

$$\mathbf{T_{02}T_{23}} = \begin{Bmatrix} \cos(\theta_1+\theta_2) & -\sin(\theta_1+\theta_2) & 0 & x_P \\ \sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) & 0 & y_P \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \begin{Bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & a_3\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & a_3\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \quad (2.23a)$$

$$T_{03} = \begin{Bmatrix} \cos(\theta_1+\theta_2+\theta_3) & -\sin(\theta_1+\theta_2+\theta_3) & 0 & x_P+a_3\cos(\theta_1+\theta_2+\theta_3) \\ \sin(\theta_1+\theta_2+\theta_3) & \cos(\theta_1+\theta_2+\theta_3) & 0 & y_P+a_3\sin(\theta_1+\theta_2+\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \quad (b)$$

Diese Matrix beschreibt die Stellung des Koordinatensystems $\mathbf{S}_3$ im System $\mathbf{S}_0$, und zwar dargestellt durch die Achskoordinaten θ_1, θ_2 und θ_3. Diese Stellung kann andererseits aber auch unabhängig von den Achskoordinaten dargestellt werden mit Hilfe der Richtungskosinusse und des Verschiebevektors, wie die nachfolgend Darstellung zeigt:

$$T = \left\{\begin{array}{ccc:c} l_x & l_y & l_z & x_E \\ m_x & m_y & m_z & y_E \\ n_x & n_y & n_z & z_E \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right\} \qquad (2.24)$$

Die beiden Matrizen T und T_{03} sollen die gleiche Information enthalten. Sie sind dann elementweise gleich:

$$\left\{\begin{array}{ccc:c} l_x & l_y & l_z & x_E \\ m_x & m_y & m_z & y_E \\ n_x & n_y & n_z & z_E \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right\} = \left\{\begin{array}{cccc} \mathbf{cos}(\theta_1+\theta_2+\theta_3) & -\mathbf{sin}(\theta_1+\theta_2+\theta_3) & 0 & x_P + a_3\,\mathbf{cos}(\theta_1+\theta_2+\theta_3) \\ \mathbf{sin}(\theta_1+\theta_2+\theta_3) & \mathbf{cos}(\theta_1+\theta_2+\theta_3) & 0 & y_P + a_3\,\mathbf{sin}(\theta_1+\theta_2+\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right\} \qquad (2.25)$$

Da lediglich eine Drehung um die z-Achse mit dem Winkel $\gamma = \varphi$ vorliegt, haben die Elemente der Richtungskosinusse die Werte

$$\begin{array}{lll} l_x = \cos\varphi & l_y = -\sin\varphi & l_z = 0 \\ m_x = \sin\varphi & m_y = \cos\varphi & m_z = 0 \\ n_x = 0 & n_y = 0 & n_z = 1 \end{array} \qquad (2.26)$$

Über die Gleichheit der Elementen erhält man Gleichungen für die Bestimmung der Gelenkvariablen.

Aus $l_x = \cos\varphi = \cos(\theta_1 + \theta_2 + \theta_3)$ folgt $\varphi = \theta_1 + \theta_2 + \theta_3$ (2.27)

Für die Elemente des Verschiebevektors ergibt sich

$$x_E = x_P + a_3 \cos(\theta_1 + \theta_2 + \theta_3) \qquad (2.28)$$
$$y_E = y_P + a_3 \sin(\theta_1 + \theta_2 + \theta_3)$$

Die Beziehungen von Gln. 2.3 und 2.4 können für x_P und y_P eingesetzt werden. Mit der Gl. 2.27 hat man dann die folgenden drei Gleichungen für die Unbekannten θ_1, θ_2 und θ_3 zur Verfügung.

$$\varphi = \theta_1 + \theta_2 + \theta_3 \qquad (2.29a)$$
$$x_E = a_1 \cos\theta_1 + a_2 \cos(\theta_1 + \theta_2) + a_3 \cos(\theta_1 + \theta_2 + \theta_3) \qquad (b)$$
$$y_E = a_1 \sin\theta_1 + a_2 \sin(\theta_1 + \theta_2) + a_3 \sin(\theta_1 + \theta_2 + \theta_3) \qquad (c)$$

Mit Hilfe der Tranformationsmethode wurden damit auch die Gln. 2.1, 2.2 und 2.17 bestätigt, wobei sich die geometrischen Überlegungen aber auf die Aufstellung der Transformationsgleichungen beschränkt haben. Trotz des scheinbar größeren Aufwandes, sollte sich diese Methode in komplexeren geometrischen Verhältnissen als überlegen erweisen.

2.2 Allgemeine Transformation von Gelenkkoordinaten

Mit Kapitel 2.1 wurde deutlich, daß sich durch den Frame des Toolkoordinatensystems die Kinematik des Roboterarms beschreiben läßt, wenn dieser Frame durch Transformation über alle Gelenkkoordinatensysteme gebildet wird. Dann enthält er nämlich die Gelenkvariablen und die geometrischen Abmessungen des kinematischen Arms. Dabei ist es von Vorteil, wenn in die Transformation von einem Gelenk zum nächsten die geometrischen Eigenschaften des Arms schon eingearbeitet sind. Dies führt zur Transformation von Denavit und Hartenberg, deren Methode zu einem Standard zur Beschreibung kinematischer Arme geworden ist [6].

2.2.1 Transformation von Denavit und Hartenberg

Für die Transformation von einem Gelenk zum nächsten wurde von Denavit und Hartenberg eine allgemeingültige Transformation angegeben. Jedem Gelenk i der kinematischen Kette wird dabei ein Koordinatensystem $S_i = (x_i, y_i, z_i)$ in der Weise zugeordnet, daß dessen z-Achse auf der Gelenkachse i+1 liegt und sich die x-Achse auf der *gemeinsamen Normalen* zwischen der i- Achse und i+1-Achse befindet. Die gemeinsame Normale steht auf beiden Achsen senkrecht, angedeutet durch rechte Winkelzeichen. Diese Anordnung der Koordinatensysteme verdeutlicht Bild 2.3.

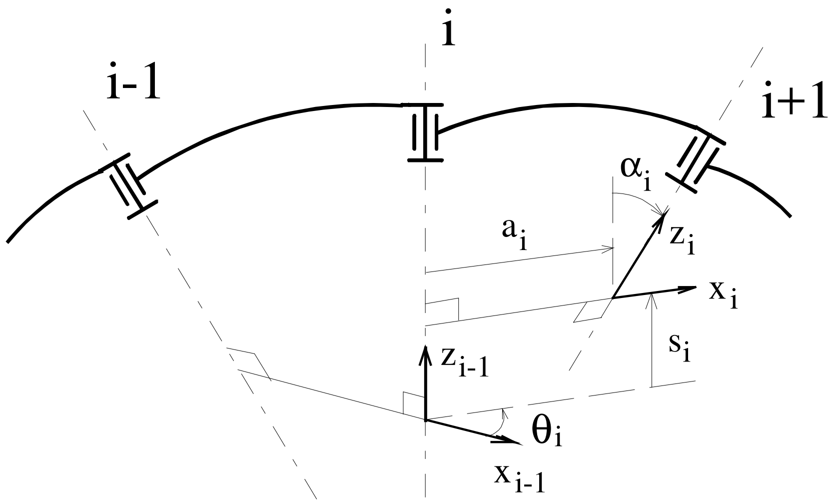


Bild 2.3: Gelenkparameter von Denavit und Hartenberg

Die Stellung des i-ten Koordinatensystems wird bezüglich des i+1-ten durch 4 Parameter beschrieben:

1. θ_i ist der Winkel von der x_{i-1}-Achse zur x_i-Achse. Er ist positiv zu nehmen, wenn er rechts um die positive z_{i-1}-Achse dreht.

2. $\mathbf{s}_i$ ist die Verschiebung des i-ten Koordinatensystems entlang der zi-1-Achse. Sie ist positv in Richtung der zi-1-Achse.

3. $\mathbf{a}_i$ ist der Abstand zwischen der z_{i-1}-Achse und der z_i-Achse.

4. α_i ist der Winkel, um den die z_i-Achse gegen die z_{i-1}-Achse verdreht ist. Er ist positiv, wenn die Drehung rechts um die positive x_i-Achse erfolgt.

Mit Hilfe dieser Festlegungen wird das Koordinatensystem $\mathbf{S}_{i-1}$ schrittweise in das System $\mathbf{S}_i$ transformiert. Dabei werden Zwischenkoordinatensysteme durchlaufen, die mit hochgestelltem Index gekennzeichnet werden:

1. Das System $\mathbf{S}_{i-1}$ wird um den Winkel θ_i in das System $\mathbf{S}^1$ gedreht. Damit liegen die x_{i-1}-Achse und die x_i-Achse parallel. Das System $\mathbf{S}^1$ hat den Frame

$$\mathbf{T}_{i-1,1} = \begin{Bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \tag{2.30}$$

2. $\mathbf{S}^1$ wird entlang der z_{i-1}-Achse um den Weg s_i verschoben. Damit liegt der Ursprung von $\mathbf{S}^2$ auf der gemeinsamen Normalen. Im System $\mathbf{S}^1$ hat $\mathbf{S}^2$ den Frame

$$\mathbf{T}_{1,2} = \begin{Bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & s_i \\ 0 & 0 & 0 & 1 \end{Bmatrix} \tag{2.31}$$

3. Die Verschiebung von $\mathbf{S}^2$ entlang der gemeinsamen Normalen um den Weg a_i führt nach $\mathbf{S}^3$. Die Ursprünge und die x-Achsen von $\mathbf{S}^3$ und $\mathbf{S}_i$ fallen damit zusammen. $\mathbf{S}^3$ hat in $\mathbf{S}^2$ den Frame

$$\mathbf{T}_{2,3} = \begin{Bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \tag{2.32}$$

4. Durch Drehung um den Winkel α_i um die positive x_i-Achse wird das Zielsystem $\mathbf{S}_i$ erreicht. $\mathbf{S}_i$ hat in $\mathbf{S}^3$ den Frame

$$\mathbf{T}_{3,i} = \begin{Bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \tag{2.33}$$

Mit diesen Teiltransformationen ergibt sich nun die folgende Gesamttransformation $\mathbf{T}_{i-1,i}$ vom System $\mathbf{S}_{i-1}$ in das System $\mathbf{S}_i$:

$$\mathbf{T}_{i-1,\,i} = \mathbf{T}_{i-1,\,1}\,\mathbf{T}_{1,\,2}\,\mathbf{T}_{2,\,3}\,\mathbf{T}_{3,\,i} \tag{2.34}$$

$$\mathbf{T}_{i-1,\,i} = \begin{Bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \begin{Bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & s_i \\ 0 & 0 & 0 & 1 \end{Bmatrix} \mathbf{T}_{2,\,3}\,\mathbf{T}_{3,\,i} = \begin{Bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & s_i \\ 0 & 0 & 0 & 1 \end{Bmatrix} \mathbf{T}_{2,\,3}\,\mathbf{T}_{3,\,i}$$

$$\mathbf{T}_{i-1,\,i} = \begin{Bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & s_i \\ 0 & 0 & 0 & 1 \end{Bmatrix} \begin{Bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \mathbf{T}_{3,\,i} =$$

$$\begin{Bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i & 0 & a_i\sin\theta_i \\ 0 & 0 & 1 & s_i \\ 0 & 0 & 0 & 1 \end{Bmatrix} \begin{Bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} =$$

$$\boxed{\mathbf{T}_{i-1,\,i} = \begin{Bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & s_i \\ 0 & 0 & 0 & 1 \end{Bmatrix}} \qquad (2.35)$$

Diese Transformations-Matrix wurde erstmals von Denavit und Hartenberg angegeben. Sie transformiert in einem Schritt von einem Gelenk zum nächsten.

Vorgehen bei der Anwendung von Denavit und Hartenberg

Bei der Anwendung der Methode von Danavit und Hartenberg geht man nach folgendem Schema vor:

1 Symbolische Darstellung des kinematischen Arms

2 Für jedes Gelenk ein Koordinatensystem einzeichnen nach dem Schema:

- z-Achsen auf die Gelenkachsen legen
  - z-Achse des Bezugssystems auf die Achse von Gelenk 1
  - z-Achse von Gelenk i auf die Gelenkachse von Gelenk i+1
  - z-Achse von Gelenk n in den TCP legen
- x-Achsen auf die jeweilige gemeinsame Normale legen. (Ausnahmen x_0 und x_n, liegt im TCP)

3 Parameter nach Denavit und Hartenberg ermitteln (s, a, α, θ)

4 Transformationen von D + H bestimmen

5 Ausmultiplizieren: $\mathbf{T_{0n}} = \mathbf{T_{01}}\mathbf{T_{12}} \cdots \mathbf{T_{n-1,n}}$

2.2.2 Anwendung der Transformation von Denavit und Hartenberg auf einen SCARA-Roboter

In Bild 2.4 ist ein SCARA-Roboters symbolisch dargestellt. Die Transformationsmatrix von Denavit und Hartenberg wird dazu verwendet, die Stellung des Effektor-Koordinatensystem S_4 im Basiskoordinatensystem S_0 herzuleiten.

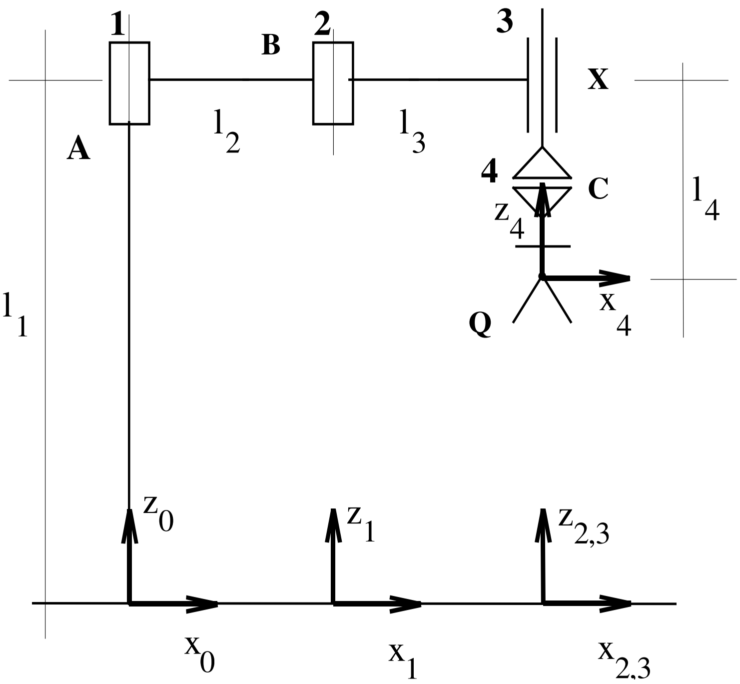
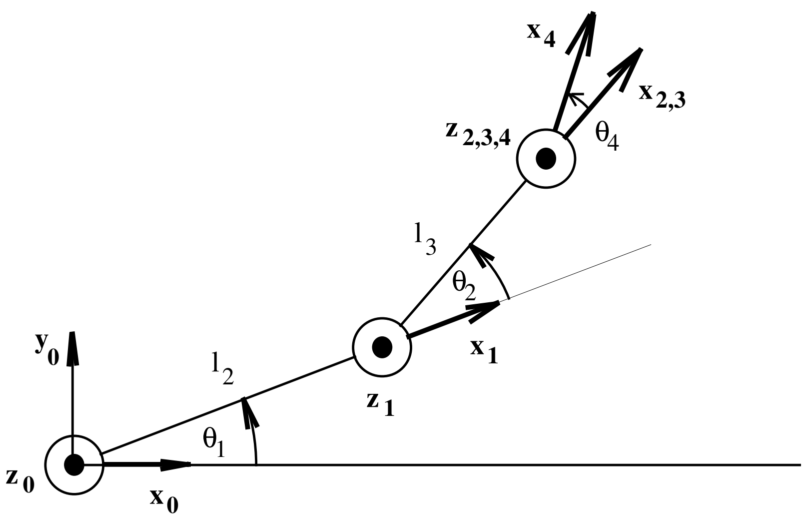


Bild 2.4: Scara-Roboter

Aus Bild 2.4 ergeben sich folgende Parameter nach Denavit und Hartenberg:

Gelenk Nr.	s	a	α/°	θ/°
1	0	l_2	0	θ_1
2	0	l_3	0	θ_2
3	s_3	0	0	0
4	l_1-l_4	0	0	θ_4



Die Bedeutung der Gelenkwinkel θ_1 bis θ_4 wird durch Bild 2.5 anschaulich gemacht, das die Ansicht des Gelenkarms von oben zeigt.

Bild 2.5: Bedeutung der Gelenkwinkel θ_1 bis θ_4

Der Frame des Effektorkoordinatensystems 4 kann über die 4-malige Anwendung der Transformation von D-H und multiplizieren der Matrizen bestimmt werden:

$$\mathbf{T}_{04} = \mathbf{T}_{01}\mathbf{T}_{12}\mathbf{T}_{23}\mathbf{T}_{34} \tag{2.36}$$

Die Teiltransformationen ergeben sich nun aus der Transformation von Denavit und Hartenberg nach Gl. 2.34 durch Einsetzen der Parameter aus der obigen Tabelle:

$$\mathbf{T}_{01} = \begin{Bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & l_2\cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & 0 & l_2\sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \qquad \mathbf{T}_{12} = \begin{Bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & l_3\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & l_3\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \tag{2.37}$$

$$\mathbf{T}_{23} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & s_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{T}_{34} = \begin{Bmatrix} \cos\theta_4 & -\sin\theta_4 & 0 & 0 \\ \sin\theta_4 & \cos\theta_4 & 0 & 0 \\ 0 & 1 & 0 & l_1 - l_4 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \tag{2.38}$$

Die Multiplikationen werden im einzelnen ausgeführt:

$$\mathbf{T}_{01}\mathbf{T}_{12}\mathbf{T}_{23}\mathbf{T}_{34} = \begin{Bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & l_2\cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & 0 & l_2\sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \begin{Bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & l_3\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & l_3\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \mathbf{T}_{23}\mathbf{T}_{34} \tag{2.39}$$

Das Produkt der ersten beiden Matrizen ergibt:

$$\begin{Bmatrix} \cos\theta_1\cos\theta_2 - \sin\theta_1\sin\theta_2 & -\cos\theta_1\sin\theta_2 - \sin\theta_1\cos\theta_2 & 0 & l_3(\cos\theta_1\cos\theta_2 - \sin\theta_1\sin\theta_2) + l_2\cos\theta_1 \\ \sin\theta_1\cos\theta_2 + \cos\theta_1\sin\theta_2 & -\sin\theta_1\sin\theta_2 + \cos\theta_1\cos\theta_2 & 0 & l_3(\sin\theta_1\cos\theta_2 + \cos\theta_1\sin\theta_2) + l_2\sin\theta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix}$$

Mit Hilfe von Additionstheoremen können die Matrixelemente zusammengefaßt werden, so daß die folgende Beziehung erhält:

$$\mathbf{T}_{01}\mathbf{T}_{12}\mathbf{T}_{23}\mathbf{T}_{34} = \begin{Bmatrix} \cos(\theta_1+\theta_2) & -\sin(\theta_1+\theta_2) & 0 & l_3\cos(\theta_1+\theta_2) + l_2\cos\theta_1 \\ \sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) & 0 & l_3\sin(\theta_1+\theta_2) + l_2\sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \mathbf{T}_{23}\mathbf{T}_{34} \tag{2.40}$$

Die Matrix $\mathbf{T}_{23}$ stellt eine reine Verschiebeoperation dar

$$\mathbf{T}_{01}\mathbf{T}_{12}\mathbf{T}_{23}\mathbf{T}_{34} = \begin{Bmatrix} \begin{bmatrix} \cos(\theta_1+\theta_2) & -\sin(\theta_1+\theta_2) & 0 & l_3\cos(\theta_1+\theta_2)+l_2\cos\theta_1 \\ \sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) & 0 & l_3\sin(\theta_1+\theta_2)+l_2\sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & s_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{Bmatrix} \mathbf{T}_{34} \qquad (2.41)$$

Durch die Matrixmultiplikation wird der Verschiebeterm s_3 in die linksstehende Matrix eingetragen. Nun muß noch mit der Matrix T_{34} multipliziert werden:

$$\mathbf{T}_{01}\mathbf{T}_{12}\mathbf{T}_{23}\mathbf{T}_{34} = \begin{Bmatrix} \begin{bmatrix} \cos(\theta_1+\theta_2) & -\sin(\theta_1+\theta_2) & 0 & l_3\cos(\theta_1+\theta_2)+l_2\cos\theta_1 \\ \sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) & 0 & l_3\sin(\theta_1+\theta_2)+l_2\sin\theta_1 \\ 0 & 0 & 1 & s_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_4 & -\sin\theta_4 & 0 & 0 \\ \sin\theta_4 & \cos\theta_4 & 0 & 0 \\ 0 & 0 & 1 & l_1-l_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{Bmatrix} \qquad (2.42)$$

Nach der Ausmultiplikation folgt mit Hilfe der Additionstheoreme das endgültige Ergebnis:

$$\mathbf{T}_{04} = \mathbf{T}_{01}\mathbf{T}_{12}\mathbf{T}_{23}\mathbf{T}_{34} = \begin{Bmatrix} \cos(\theta_1+\theta_2+\theta_4) & -\sin(\theta_1+\theta_2+\theta_4) & 0 & l_3\cos(\theta_1+\theta_2)+l_2\cos\theta_1 \\ \sin(\theta_1+\theta_2+\theta_4) & \cos(\theta_1+\theta_2+\theta_4) & 0 & l_3\sin(\theta_1+\theta_2)+l_2\sin\theta_1 \\ 0 & 0 & 1 & l_1-l_4+s_3 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \qquad (2.43)$$

Diese Transformation stellt den Frame von Koordinatensystem 4 dar. In der 4. Spalte stehen die Komponenten des Verschiebevektors. Die Größe $l_1 - l_4 + s_3$ bedeutet z. B. die Verschiebung entlang der z-Achse. Die Orientierung des Greifers wird durch den Rotationsteil des Frames beschrieben. Es handelt sich um eine reine Rotation um die z_0-Achse des Bezugssystems bestimmt durch die Summe der Drehwinkel θ_1, θ_2 und θ_4.

Zusammenfassend kann festgehalten werden, daß die Beschreibungsmethode von Denavit und Hartenberg eine formale, klare Beschreibung der kinematischen Zusammenhänge von Gelenkarmen erlaubt. Ohne eine solche standardisierte Beschreibung ist der Vergleich der kinematischen Eigenschaften von Industrie-Robotern nur schwer möglich.

2.2.3 Auflösung der Transformationsmatrix nach den Gelenkkoordinaten

Die Bestimmung der Gelenkkoordinaten (im Fall des SCARA die Variablen θ_1, θ_2, θ_4 und s) für eine vorgegebene Stellung des Effektors wird als Rückwärts-Transformation bezeichnet. Wenn die Gelenkkoordinaten bekannt sind, können die Stellantriebe angesteuert werden. Bei der TEACH-IN-Programmierung lernt der IR die erforderlichen Gelenkkoordinaten dadurch kennen, daß die Stellung mit dem Roboter handgesteuert angefahren wird. Beim Bahnfahren dagegen muß die Steuerung die Achskoordinaten für die einzelnen Bahnpunkte laufend berechnen, da die programmierten Punkte im kartesischen System interpoliert werden müssen. Die Berechnung durch die Steuerung ist auch beim Punkt-zu-Punkt-Fahren erforderlich, wenn während der Ausführung externe Sensorinformationen berücksichtigt werden müssen. Die Gleichungen für die Rückwärts-Transformation müssen dann entweder analytisch zur Verfügung stehen oder die Transformation muß numerisch berechnet werden. Die Elemente der vollständige Transformationsmatrix für einen IR bedeuten bekanntlich die Richtungskosinuswerte im Rotationsteil und den Verschiebevektor für den TCP im Verschiebeteil. Für den SCARA läßt sich daher folgende Matrizengleichung angeben:

$$\begin{Bmatrix} l_x & l_y & l_z & x_E \\ m_x & m_y & m_z & y_E \\ n_x & n_y & n_z & z_E \\ 0 & 0 & 0 & 1 \end{Bmatrix} = \begin{Bmatrix} \cos(\theta_1+\theta_2+\theta_4) & -\sin(\theta_1+\theta_2+\theta_4) & 0 & l_3\cos(\theta_1+\theta_2)+l_2\cos\theta_1 \\ \sin(\theta_1+\theta_2+\theta_4) & \cos(\theta_1+\theta_2+\theta_4) & 0 & l_3\sin(\theta_1+\theta_2)+l_2\sin\theta_1 \\ 0 & 0 & 1 & l_1-l_4+s_3 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \quad (2.44)$$

Betrachtet man die Terme der links stehenden Matrix als vorgegeben, dann können die Gelenkvariablen der korrespondierenden Terme der rechten Matrix bestimmt werden. Da ein SCARA nur 4 Freiheitsgrade hat, gilt es 4 Gelenkvariable mit Hilfe von 4 Gleichungen zu bestimmen. Drei Gleichungen beziehen sich auf die Position des Effektorkoordinatensystems und eine auf die Rotation.

Die Komponenten des Verschiebevektors liefern die Gleichungen

$$x_E = l_3\cos(\theta_1+\theta_2)+l_2\cos\theta_1 \quad (2.45)$$

$$y_E = l_3\sin(\theta_1+\theta_2)+l_2\sin\theta_1 \quad (2.46)$$

$$z_E = l_1 - l_4 + s_3 \quad (2.47)$$

Die gewünschte Drehung des Effektors kann über den Parameter θ_4 eingestellt werden. Zu seiner Bestimmung sind 4 gleichwertige Beziehungen verfügbar, von denen eine beliebige ausgewählt werden kann. Gewählt wird die Beziehung

$$m_x = \sin(\theta_1+\theta_2+\theta_4) \quad (2.48)$$

Aus Gl. 2.44 erhält man

$$s_3 = z_4 - l_1 + l_4 \quad (2.49)$$

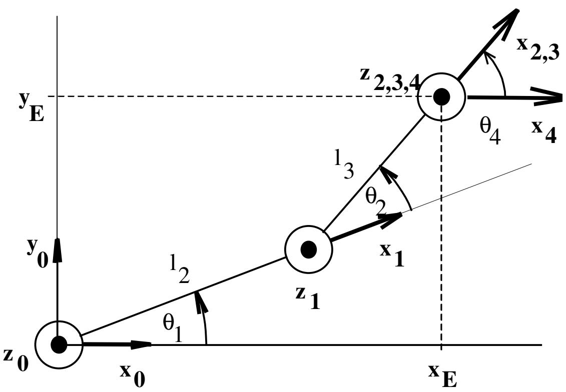


Bild 2.6: SCARA mit Stellungsvorgaben

Die Winkel θ_1 und θ_2 erhält man aus den Gln. 2.43 und 2.44 d. h.

$$\cos\theta_2 = \frac{x_E^2 + y_E^2 - l_2^2 - l_3^2}{2\, l_2\, l_3} \quad (2.50)$$

$$\tan\theta_1 = \frac{y_E(l_2 + l_3\cos\theta_2) - x_E l_3 \sin\theta_2}{x_E(l_2 + l_3\cos\theta_2) + y_E l_3\sin\theta_2} \quad (2.51)$$

Der Winkel θ_4 ergibt sich aus Gl. 2.46 wie folgt:

$$\theta_4 = arc\sin(m_x) - \theta_1 - \theta_2 \quad (2.52)$$

2.2.4 Beispiel 9: Berechnung der Gelenkkoordinaten

Die Drehung des Effektors in Bild 2.6 wurde so gewählt, daß die x_4-Achse parallel zur x_0-Achse steht. Diese Orientierung bedeutet

$$l_x = 1 \quad m_x = 0 \quad n_x = 0 \quad (2.53)$$

Mit den Winkeln

$$\theta_1 = 21° \quad \text{und} \quad \theta_2 = 27° \tag{2.54}$$

ergibt sich daher

$$\theta_4 = -\theta_1 - \theta_2 = -48° \tag{2.55}$$

Die arctan2-Funktion

Die Umkehrfunktionen der trigonometrischen Funktionen sin, cos und tan sind mehrdeutig. Ihre Hauptwerte liegen in den Grenzen

$$\begin{aligned} -\frac{\pi}{2} &\leq \mathbf{arctan}\, x \leq \frac{\pi}{2} \\ -\frac{\pi}{2} &\leq \mathbf{arcsin}\, x \leq +\frac{\pi}{2} \\ 0 &\leq \mathbf{arccos}\, x \leq +\pi \end{aligned} \tag{2.56}$$

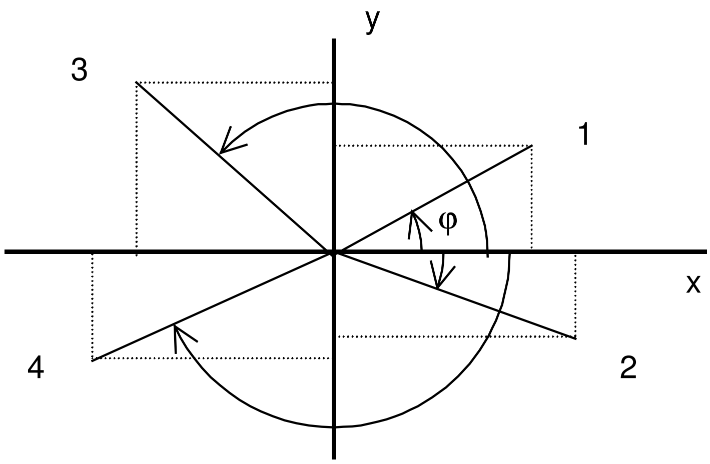


Bild 2.7: Die Funktion arctan2, Fälle 1 - 4

Diese Hauptwerte stellen für die Bestimmung der Gelenkwinkel eine nicht hinnehmbare Einschränkung dar, die mit Hilfe der arctan2-Funktion beseitigt wird. Sie besitzt 2 Argumente, deren Vorzeichen bei der Bestimmung des Winkels berücksichtigung finden:

$$\varphi = \mathbf{arctan}\, 2(x, y) \tag{2.57}$$

Dazu zeigt Bild 2.7 vier Fälle. Im Fall 1 gilt

$$\mathbf{tan}\, \varphi = \frac{y}{x}$$

Den Winkel φ erhält man dann aus

$$\varphi = \mathbf{arctan}\left(\frac{y}{x}\right) \tag{2.58}$$

Der Winkel φ ist negativ im Fall 2, wenn y kleiner Null ist, aber positiv im Fall 3, wenn x kleiner Null ist. Je nach Vorzeichen der zwei Argumente sind daher die folgendenFunktionswerte definiert:

$$\mathbf{arctan}\,2(x,y)=\begin{cases}1 \quad \mathbf{arctan}(\frac{y}{x}) & \textit{für} \quad x>0,\ y>0 \\ 2 \quad \mathbf{arctan}(\frac{y}{x}) & \textit{für} \quad x>0,\ y<0 \\ 3 \quad \mathbf{arctan}(\frac{y}{x})+\pi & \textit{für} \quad x<0,\ y>0 \\ 4 \quad \mathbf{arctan}(\frac{y}{x})-\pi & \textit{für} \quad x<0,\ y<0 \\ 5 \quad +\frac{\pi}{2} & \textit{für} \quad x=0,\ y>0 \\ 6 \quad -\frac{\pi}{2} & \textit{für} \quad x=0,\ y<0 \\ 7 \quad \textit{nicht definiert} & \textit{für} \quad x=,\ y=0 \end{cases} \tag{2.59}$$

2.3 Methoden zur Orientierungseinstellung der Hand

Roboter-Steuerung erlauben die Angabe der Orientierung der Hand mit Hilfe von drei Eulerwinkeln für unterschiedliche Achsfolgen. Dies können z. B. folgende Drehsequenzen sein:

xyz, Drehung um die x-, y2-, z3-Achsen
Roll-, Nick- und Gierwinkeln, zyx
Drehung um die XYZ-Achsen des Bezugssystems

a) Orientierung mit Eulerwinkeln

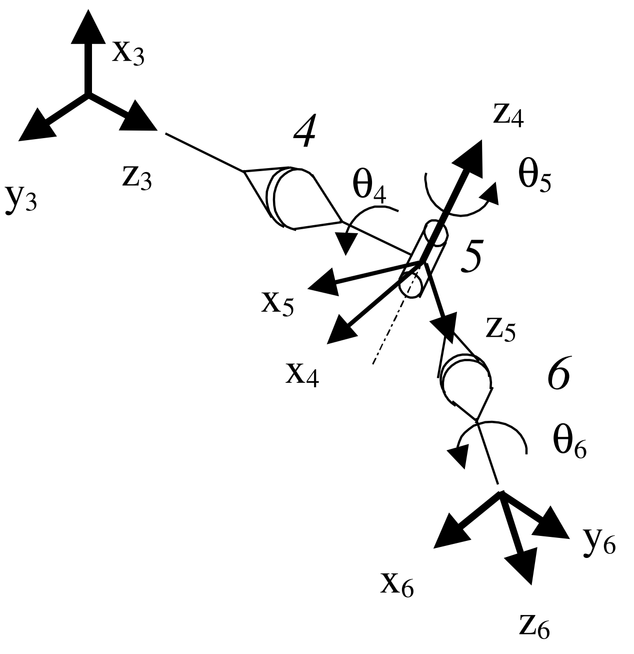


Bild 2.8: Gelenke einer Hand

Weit verbreitet ist die Einstellung der Orientierung mit Eulerwinkeln der Folge zxz (siehe Bild 1.6 vorn). Sie entsprechen am ehesten dem mechanischen Aufbau der Hand. Man betrachte dazu Bild 2.8. Die Einstellung der Orientierung erfolgt dadurch, daß zuerst mit θ_4 um die z_3-Achse gedreht wird, danach mit θ_5 um die verdrehte z_4-Achse und schließlich mit θ_6 um die verdrehte z_5-Achse.

Die den Gelenkwinkeln θ_4, θ_5 und θ_6 entsprechen den Eulerwinkel φ, ϑ und ψ.
Das Koordinatensystem (x_6, y_6, z_6) stellt das Effektorkoordinatensystem (x_E, y_E, z_E) dar.
Mit $\varphi = \theta_4$, $\vartheta = \theta_5$ und $\psi = \theta_6$ ergeben sich daher folgende Rotationsmatrizen:

$$R_{z3}(\varphi) = \begin{Bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{Bmatrix} \quad R_{z4}(\vartheta) = \begin{Bmatrix} \cos\vartheta & 0 & \sin\vartheta \\ 0 & 1 & 0 \\ -\sin\vartheta & 0 & \cos\vartheta \end{Bmatrix} \quad R_{z5}(\psi) = \begin{Bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{Bmatrix} \tag{2.60}$$

Die Orientierungsmatrix erhält damit die Form

$$\mathbf{B}(\varphi, \vartheta, \psi) = \mathbf{R}_{z3}(\varphi)\mathbf{R}_{z4}(\vartheta)\mathbf{R}_{z5}(\psi) =$$

$$\begin{Bmatrix} \cos\varphi\cos\vartheta\cos\psi - \sin\varphi\sin\psi & -\cos\varphi\cos\vartheta\sin\psi - \sin\varphi\cos\psi & \cos\varphi\sin\vartheta \\ \sin\varphi\cos\vartheta\cos\psi - \cos\varphi\sin\psi & -\sin\varphi\cos\vartheta\sin\psi + \cos\varphi\cos\psi & \sin\varphi\sin\vartheta \\ -\sin\vartheta\cos\psi & \sin\vartheta\sin\psi & \cos\vartheta \end{Bmatrix} \tag{2.61}$$

Dies entspricht, auf ein Koordinatensystem bezogen, den Drehungen um dessen z-Achse, danach um die verdrehte y_2-Achse und anschließend wieder um die verdrehte z_3-Achse, wie es Bild 1.6 zeigt.

Zu einer gegebenen Orientierungsmatrix

$$B = \begin{Bmatrix} l_x & l_y & l_z \\ m_x & m_y & m_z \\ n_x & n_y & n_z \end{Bmatrix}$$

können die Eulerwinkel nach folgender Methode ermittelt werden. Man setzt

$$\mathbf{B} = \mathbf{R}_{z1}(\varphi)\mathbf{R}_{y1}(\vartheta)\mathbf{R}_{z3}(\psi) \tag{2.62}$$

und multipliziert linksseitig mit $R_z(\varphi)^{-1}$. Damit ergibt sich

$$\mathbf{R}_{z1}(\varphi)^{-1}\,\mathbf{B} = \mathbf{R}_{y1}(\vartheta)\mathbf{R}_{z2}(\psi) \tag{2.63}$$

oder ausführlich geschrieben

$$\begin{Bmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{Bmatrix} \begin{Bmatrix} l_x & l_y & l_z \\ m_x & m_y & m_z \\ n_x & n_y & n_z \end{Bmatrix} = \begin{Bmatrix} \cos\vartheta & 0 & \sin\vartheta \\ 0 & 1 & 0 \\ -\sin\vartheta & 0 & \cos\vartheta \end{Bmatrix} \begin{Bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{Bmatrix} \tag{2.64}$$

Mit Hilfe von Gl. 2.61 wurde also die Variable φ auf die linke Seite der Gleichung gebracht, was die Lösung wesentlich vereinfacht. Die Matrizen beider Seiten werden ausmultipliziert.

$$\begin{Bmatrix} l_x\cos\varphi + m_x\sin\varphi & l_y\cos\varphi + m_y\sin\varphi & l_z\cos\varphi + m_z\sin\varphi \\ -l_x\sin\varphi + m_x\cos\varphi & -l_y\sin\varphi + m_y\cos\varphi & -l_z\sin\varphi + m_z\cos\varphi \\ n_x & n_y & n_z \end{Bmatrix} =$$

$$\begin{Bmatrix} \cos\vartheta\cos\psi & -\cos\vartheta\sin\psi & \sin\vartheta \\ \sin\psi & \cos\psi & 0 \\ -\sin\vartheta\cos\psi & \sin\vartheta\sin\psi & \cos\vartheta \end{Bmatrix} \tag{2.65}$$

Durch Koeffizientenvergleich können die Drehwinkel der Reihe nach bestimmt werden.

Bestimmung des Winkels φ

Aus dem Vergleich der Terme (2,3) findet man

$$-l_z \sin\varphi + m_z \cos\varphi = 0 \tag{2.66}$$

$$\frac{m_z}{l_z} = \frac{\sin\varphi}{\cos\varphi} = \tan\varphi \tag{2.67}$$

Mit Hilfe der arctan2-Funktion erhält man nun im Bereich $\pm\pi$

$$\varphi = \arctan 2\left(\frac{m_z}{l_z}\right) \tag{2.68}$$

Bestimmung des Winkels ϑ

Nachdem φ bekannt ist, wird aus den Elementen (1,3) und (3,3) der Winkel ϑ bestimmt.

$$+l_z \cos\varphi + m_z \sin\varphi = \sin\vartheta \tag{2.69}$$

$$n_z = \cos\vartheta \tag{2.70}$$

Damit ergibt sich

$$\frac{l_z \cos\varphi + m_z \sin\varphi}{n_z} = \frac{\sin\vartheta}{\cos\vartheta} = \tan\vartheta \tag{2.71}$$

Das Ergebnis lautet daher

$$\vartheta = \arctan 2\left(\frac{l_z \cos\varphi + m_z \sin\varphi}{n_z}\right) \tag{2.72}$$

Bestimmung des Winkels ψ

Die Terme (2,1) und (2,2) werden gleich gesetzt:

$$l_x \sin\varphi + m_x \cos\varphi = \sin\psi \tag{2.73}$$

$$l_y \sin\varphi + m_y \cos\varphi = \cos\psi \tag{2.74}$$

Damit erhält man für ψ *das Ergebnis*

$$\psi = \arctan 2\left(\frac{-l_x \sin\varphi + m_x \cos\varphi}{-l_y \sin\varphi + m_x \cos\varphi}\right) \tag{2.75}$$

Orientierung nach Roll-, Nick- und Gierwinkeln

Die Begriffe Roll-, Nick- und Gierwinkel stammen aus der Fahrzeugtechnik. Die Bedeutung der Winkel geht aus Bild 2.9 beim Flugzeug hervor. Bei Landfahrzeugen wird die z-Achse nach oben angesetzt. Die englischsprachigen Ausdrücke sind Roll, Pitch und Yaw. Häufig wird daher auch die Abkürzung RPY-Winkel verwendet. Auch Roll-, Nick- und Gierwinkel sind Eulerwinkel allerdings für die spezielle Drehfolge zyx.

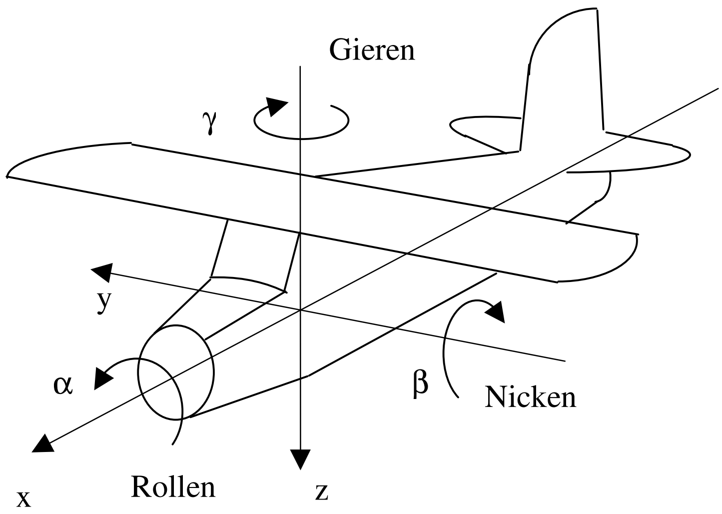


Bild 2.9: Roll-, Nick- und Gierwinkel beim Flugzeug

Die Orientierungseinstellung erfolgt erst durch eine Drehung um die z-Achse (Flug-, Fahrrichtung), zweitens um die verdrehte y_1-Achse und drittens um die verdrehte x_2-Achse.

Mit den Drehmatrizen

$$R_z(\gamma) = \begin{Bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{Bmatrix} \qquad R_{y1}(\beta) = \begin{Bmatrix} \mathbf{\cos\beta} & 0 & \mathbf{\sin\beta} \\ 0 & 1 & 0 \\ \mathbf{-\sin\beta} & 0 & \mathbf{\cos\beta} \end{Bmatrix} \qquad R_{x2}(\alpha) = \begin{Bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{Bmatrix} \tag{2.76}$$

erhält man die Gesamtdrehmatrix

$$\mathbf{R}(\gamma,\beta,\alpha) = \begin{Bmatrix} \mathbf{\cos\gamma\cos\beta} & \mathbf{-\sin\gamma\cos\alpha+\cos\gamma\sin\beta\sin\alpha} & \mathbf{\sin\gamma\sin\alpha+\cos\gamma\sin\beta\cos\alpha} \\ \mathbf{\sin\gamma\cos\beta} & \mathbf{\cos\gamma\cos\alpha+\sin\gamma\sin\beta\sin\alpha} & \mathbf{-\cos\gamma\sin\alpha+\sin\gamma\sin\beta\cos\alpha} \\ \mathbf{-\sin\beta} & \mathbf{\cos\beta\sin\alpha} & \mathbf{\cos\beta\cos\alpha} \end{Bmatrix} \tag{2.77}$$

Nach dem gleichen Schema wie bei der zyz Folge zuvor ergeben sich folgende Lösungen:

$$\gamma = \mathbf{arctan}\,2\left(\frac{m_x}{l_x}\right) \tag{2.78}$$

$$\beta = \mathbf{arctan}\,2\left(\frac{-n_x}{l_x\,\mathbf{\cos}\,\gamma + m_x\,\mathbf{\sin}\,\gamma}\right) \tag{2.79}$$

$$\alpha = \mathbf{arctan}\,2\left(\frac{l_z\,\mathbf{\sin}\,\gamma - m_z\,\mathbf{\cos}\,\gamma}{-l_y\,\mathbf{\sin}\,\gamma + m_y\,\mathbf{\cos}\,\gamma}\right) \tag{2.80}$$

2.4 Roboterkonfigurationen

Industrie-Roboter können in einigen Fällen die gleiche Stellung mit zwei unterschiedlichen Sätzen von Gelenkvariablen anfahren. Welcher der zwei Sätze zu verwenden ist, muß über eine Konfigurationsangabe festgelegt werden. Die Berechnung der Gelenkvariablen aus den Rotationsmatrizen muß darauf Rücksicht nehmen.

Solche Konfigurationen sind

- Ellbogen oben oder unten
- Arm links oder rechts
- Handmittelachse geflippt (um 180° geschwenkt)

Die nachfolgenden Bilder veranschaulichen die drei Konfigurationen.

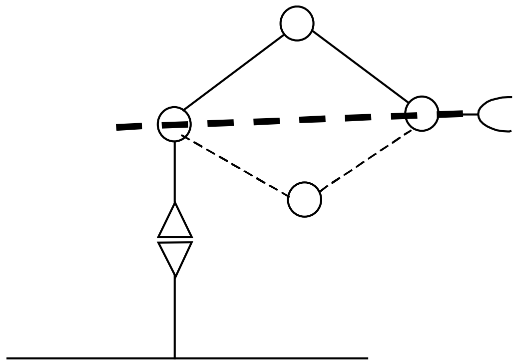


Bild 2.9: Konfiguration *oben und unten*

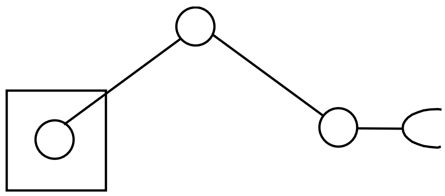


Bild 2.10: Konfiguration *links und rechts*

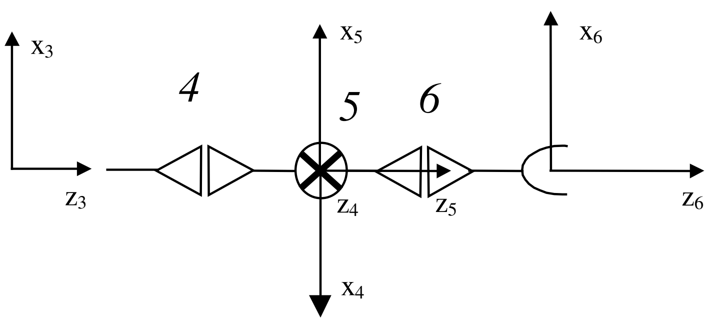


Bild 2.11: Die Konfigurationen geschwenkt und nicht geschwenkt (flip, no flip)

Gelenk 5 kann sich um 180° drehen, ohne daß sich die Koordinatensysteme 3 und 6 bewegen.

Konfigurationsparameter des Movemaster RV-E2

Eine Stellung wird über drei Positionen x, y und z sowie über die Roll-, Nick- und Gierwinkel γ, β und α definiert. Bei einer Untermenge aller Stellungen (also nicht immer) kann man die Stellung aber mit drei unterschiedlichen Gelenkeinstellungen anfahren. Die Unterscheidung der Konfigurationen erfolgt am RV-E2 mit Hilfe der Konfigurationsparameter

A/B above/below
L/R left/right
F/N flip/noflip

3 Programmierung

Dieses Kapitel behandelt die textuelle Off-Line-Programmierung von Industrierobotern, d. h. die Programmierung der Bewegungsabläufe an einem Rechner (CAP-System*) mit Hilfe von Programmiersprachen. Diese Art der Programmierung bietet folgende Vorteile:

- Während der Programmierung wird der Roboter nicht benötigt
- Komplexe Handhabungsaufgaben lassen sich leichter und schneller lösen
- Das Zusammenspiel mit Komponenten zur integrierten Fertigung (CIM, Computer integratet Manufactoring) wird ermöglicht
- Die Programmiersprachen werden durch Simulationsprogramme unterstützt.

Programmiersprachen für Industrieroboter benötigen folgende Sprachelemente:

a) Elemente üblicher Programmiersprachen
- Konstanten, Variablen, Datenobjekte
- Programmsteuerung und Unterprogramme
- mathematische Operationen

b) Robotertypische Sprachelemente
- Bewegungsbefehle
- Endeffektor- und Sensorbefehle
- Monitorfunktionen

c) Kommunikation
- Steuerung von Maschinen
- Ausgaben auf Terminals

Einige verfügbare Programmiersprachen:

AML	A Manufactoring Language, (IBM, Pascal ähnlich)
BAPS	Bewegungs- und Ablauf-Programmier-Sprache (BOSCH)
IRDATA	Industriel Robot DATA (herstellerunabhängige Zwischensprache, (DIN 66314)
IRL	Industrial Robot Language (pascalähnliche Hochsprache, (DIN 66312)
MRL	Mitsubishi Robot Language (BASIC Programmiersprache)
MELFA BASIC III + IV + V	Mitsubishi Robot Language mit verbesserten Funktionen.
RAIL	Programmiersprache der Firma AUTOMATIX mit den Komponenten Robovision, Cybervision (Montage), Autovision
SRCL	Siemens Robot Control Language
VAL II, V^+	Basic ähnliche Programmiersprachen der Firmen ADEPT TECHNOLOGY und STÄUBLI- UNIMATION
RAPL	Robot Automation Programming Language (Cobra)

* CAP steht für *Computer Aided Planing*, d. h. rechnerunterstützte Planung

3.1 Die Mitsubishi Robot Language MELFA BASIC III + IV + V

MELFA BASIC ist die Programmiersprache der MELFA-Roboter, wobei MELFA für Mitsubishi Electric Factory Automation steht. Die Sprache basiert, ebenso wie die Vorgängersprache MRL, auf der BASIC-Programmiersprache, ist aber wesentlich flexibler und leistungsfähiger. Die Sprache stellt ein umfangreiches Instrumentarium von Befehlen zur Verfügung. Eine Besonderheit ist die Trennung von Programm und Positionen. Die Positionen können in einer eigenen Datei, der Positionsliste, abgespeichert werden. Jedem Befehl muß eine Zeilennummer vorausgehen.

Im weiteren wird nur eine Auswahl von Befehlen behandelt, die von didaktischer Bedeutung sind. Weitere Einzelheiten können dem Handbuch zum RV-E2 entnommen werden [7] für MELFA BASIC III und dem CONTROLLER INSTRUCTION MANUAL [8] für MELFA BASIC IV. MELFA BASIC IV bringt als Erweiterungen Multitasking Eigenschaften, Förderbandsynkronisation und weitere Befehle. MELFA BASIC V unterscheidet Groß- und Kleinschreibung und benötigt keine Zeilennummern. Statt dessen werden Schrittnummern automatisch beim Laden des Programmes vergeben. Sprungziele müssen als LABEL (*NAME) angegeben werden [9].

Vergleichs- und logische Operationen

Befehl	*Funktion*	*Bedeutung*
=	gleich	Datenvergleich auf Gleichheit
<>, ><	ungleich	Datenvergleich auf Ungleichheit
<	kleiner	Datenvergleich auf kleiner
>	glößer	Datenwertvergleich auf größer
<=, =<	kleiner oder gleich	Datenvergleich auf kleiner oder gleich
>=, =>	größer oder gleich	Datenvergleich auf größer oder gleich
NOT	Negation	Dient zur Erzeugung des 2er-Komplements
AND	Logisches UND	UND-Verknüpfung
OR	Logisches ODER	ODER-Verknüpfung
XOR	Exklusives ODER	Exklusive ODER-Verknüpfung

Mathematische Funktionen (einige)

DEG (<Numerischer Ausdruck:rad>)	Wandelt die Einheit des Winkels von Radiant (rad) in Grad (deg) um.
DEGRAD (<NumerischerAusdruck: degree>)	Wandelt die Einheit des Winkels von Grad (deg) in Radiant (rad) um.
RAD (<Numerischer Ausdruck: deg>)	Wandelt die Einheit des Winkels von Grad (deg) in Radiant (rad) um.
RADDEG (<Numerischer Ausdruck: rad>)	Wandelt die Einheit des Winkels von Radiant (rad) in Grad (deg) um.
EXP (<Numerischer Ausdruck>)	Berechnet den Wert der Exponentialfunktion.
LN (<Numerischer Ausdruck>)	Berechnet den natürlichen Logarithmus.
LOG (<Numerischer Ausdruck>)	Berechnet den dekadischen Logarithmus.
SQR (<Numerischer Ausdruck>)	Berechnet die Quadratwurzel.
ATN2 (<Numerischer Ausdruck>	Berechnet den Arcustangens. (ATN2 = ATN(X,Y)) Definitionsbereich: -180° bis +180° (außer 0)
COS (<Numerischer Ausdruck>)	Berechnet den Kosinus. Definitionsber.: -1 bis +1
SIN (<Numerischer Ausdruck>)	Berechnet den Sinus. Definitionsbereich: -1 bis +1 Einheit: Grad

Programmsteuerung

Befehl	*Bedeutung*
HLT	Programmablauf stoppen
STOP	Programmablauf stoppen
GOSUB	Sprung zu einem Unterprogramm
ON a GOSUB b1, b2, ..	Sprung zum Sprungziel a (z. B. b2 für a = 2)
RETURN	Rücksprung zum Hauptprogramm
CALLP "a", b1, b2, ...	Programm "a" aufruf en und Paramter b1, b2, .. übergeben
GOTO	Sprung zu einer Programmzeile oder -marke
FOR-NEXT	Programmschleife (Wiederholung)
IF THEN ELSE	Bedingte Verzweigung
WHILE WEND	Wiederholschleife
END	Programmende

Variablentypen

Erstes Zeichen ist

P	Positionsvariable	(P1 ... P900)
J	Gelenkvariable	(J1, J2, J3, ...)
M	Arithmetische Variable	(M1 ... M90, M steht für Mathematik), aber auch Variablen, die mit anderen Buchstaben als P, J, M, und C beginnen, werden als numerische Variablen angesehen.
C	Zeichenkettenvariable	(C1$, C2$, ...)

Zweites Zeichen ist

_	Globale Variable	(P_1, J_1; M_1)

Beispiel Konstanten:

M_PI	Kreiszahl (3.1415...)
M_EXP	Basis des natürlichen Logarithmus (2.71828...)
M_G	Erdbeschleunigung (9.80665)
P_CURR	Aktuelle Position
P_Safe	Sichere Position

Definition von Variablen

Position definieren

DEF POS P-sicher

P-Sicher = P_Safe Einer Position einen besonderen Namen geben, maximal 8 Zeichen.
P_Safe ist Systemvariable (unter CIROS nicht bekannt)

Bogen definieren

DEF ARCH Nr, AUF-Inkrement, AB-Inkrement, Aufwärtsschritt, Abwärtsschritt

Der Bogen wird mit dem Bewegungsbefehlt MVA (Move Arch) abgefahren

Funktion definieren

DEF FNeigenefunktion(M1, M2,...) = f(M1, M2,...)

Beispiel:

DEF FNMittelwert (M1, M2) = (M1 + M2)/2

Positionierung und Roboterbewegung

Diese Gruppe umfaßt die Befehle zum Definieren von Positionen und Koordinaten. Desweiteren lassen sich mit den Befehlen dieser Gruppe die Interpolationsmethode, die Geschwindigkeit, die Timer, die Werkzeuglänge , die Gitterkoordinaten für Paletten usw. einstellen.

Positionsvariablen:

Befehl	*Bedeutung*
HRE	Aktuelle Position zuweisen, z. B. P1 = HRE
P1 = (x, y, z, α, β, γ)(FL1, FL2)	P1 werden die Positionskoordinaten x, y, z in mm und die Gier-, Nick-, Rollwinkel α, β, γ in Grad zugewiesen. FL1 sind Flags für Konfiguration: 7 = &B 0000 0111 (Binärzahl) | 1/0 Non Flip/Flip | 1/0 Above/Below | 1/0 Right/Left FL2 sind Flags für ganze Umdrehungen (hier immer 0)
P1.X	X-Komponente von P1. Andere Komponenten analog, aber die **Winkelkomponenten werden in Rad** angegeben.
J7 = (W, S, E, T, P, R)	Drehwinkel der Gelenke (W = Waist, S = Shoulder, E = Elbow , T = Twist, P = Pitch, R = Roll). Einheit Grad.
J7.J4	Twistwinkel von J7 in **Rad**. Andere Gelenkwinkel analog.

Bewegungen mit Gelenkinterpolationen:

MOV (Move)

Beispiele:

MOV P1 Stellung P1 anfahren
MOV P1, -100 Stellung P1 mit -100 mm in Toolrichtung anfahren
MOV P1, -100 WTH M_HND(1) = 1
wie oben, aber mit Hand 1 geöffnet
MOV J1 Gelenkstellung anfahren

Relative Positionierung:

MOV P1 + P2 Stellung anfahren, die durch die Summe der Koordinaten von P1 und P2 gegeben ist

Linenarbewegungen:

MVS (Move Straight) *Geradlinige Bewegung*

Beispiele:

MVS P1 Stellung P1 anfahren
MVS P1, 100 Stellung P1 mit 100 mm in Toolrichtung anfahren
MVS ,-100 von der aktuellen Position -100 mm in Toolrichtung

Bewegung mit Zusatz:

MVS P1, -100 WTH M_HND(1) = 0 wie oben, aber mit Hand 1 geschlossen

Kreisbewegungen:

MVC (Move C) ***Kreis-Interpolation***
Beispiel:
MVC P1, P2, P3
Kreis durch die Punkte P1, P2, P3

MVR (Move Radius) ***Kreissegment mit drei Punkten***
Beispiel:
MVR P1, P2, P3
Kreisbogen von P1 über P2 nach P3

MVR2 (Move R2) ***Kreissegment von P1 nach P3 mit P3 als Referenzposition***
Beispiel:
MVR2 P1, P2, P3
Kreisbogen vom Startpunkt P1 zum Endpunkt P2 mit P3 als Referenzpunkt dahinter

Pallettenbefehle

DEF PLT (Define Pallet) ***Palette definieren***

DEF PLT a, P1, P2, P3, [P4], Z, S, B
a Palettennummer 1 ... 8
P1 Anfangspunkt der Palette
P2 Zeilenendpunkt
P3 Spaltenendpunkt
[P4] Optionaler Punkt diagonal zu P1 (bessere Genauigkeit)
Z Zahl der Zeilenplätze
S Zahl der Spaltenplätze
B Bewegungsrichtung: von P1 nach P2 und zurück (B=1), Richtung P1 nach P2 (B=0)

PLT (Pallet) Koordinaten für Palettenpunkt berechnen
Beispiele:
P1 = PLT 3, 7 **P1** erhält die Position von Gitterpunkt 7, oder
M4 = 7
MOV (PLT 3, M4), -50 Gitterpunkt 7 in 50 mm Abstand anfahren

Geschwindigkeit einstellen

SPD (Speed) Geschwindigkeit für lineare und kreisförmige Bewegungen.
Beispiele:
SPD M_NSPD Standardwert 63,3 mm/s
Mögliche Bereich SPD 1....650 mm/s

OVRD (Override) Prozentualer Übersteuerungswert bei Linearinterpolation
1 <= Übersteuerungswert <= 200

JOVRD (J Override) Prozentualer Übersteuerungswert bei Gelenkinterpolation
1 <= Übersteuerungswert <= 200

Die aktuelle Arbeitsgeschwindigkeit (AG) ergibt sich aus:

Linear- und Kreisbewegungungen: AG = Override(Teach-Box) x OVRD x SPD
Gelenkinterpolation: AG = Override(Teach-Box) x OVRD x JOVRD

Der Standardwert der Arbeitsgeschwindigkeit beträgt 100% der Standardeinstellung (M_NOVRD).

(Zum Testen von Programmen Override(Teach-Box) < 10% wählen!)

Befehle für die Handsteuerung

M_HND()	(Hand)	M_HND(a) = b a = Handnummer 1, 2, 3; b = 1 offen, b = 0 zu Beispiel: **M_HND(1) = 1** Hand öffnen M_HND ist ein Systemparameter)
HCLOSE 1	(Hand)	Hand 1 schließen
HOPEN 1	(Hand)	Hand 1 schließen
TOOL	(Tool)	Werkzeuglänge einstellen. Beispiel: **TOOL P1** mit **P1** = (Δx, Δy, Δz, $\Delta\alpha$, $\Delta\beta$, $\Delta\gamma$) mit Deltas im Tool-Koordinatensystem
DLY	(Delay)	Wartezeit. Beispiel: **DLY 0.5** 0,5 Sek. warten

3.2 Anwendungsbeispiele

Position definieren

Eine Position kann in einer Positionsliste definierte werden oder aber durch Zuweisung im Programm. Der Begriff *Position* umfaßt hier neben den Raumkoordinaten auch noch die Orientierung und die Konfiguration des Roboters. Solche Position werden wie folgt definiert (siehe oben unter Positionsvariablen):

P1 = (-150, 300, 250, 180, 0, 0)(7,0)
P2 = (150, 300, 250, 180, 0, 90)(7,0)

Bewegung zu einem Punkt

Der TCP befinde sich an der Position P1. Er kann mit folgenden Befehlen zur Position P2 bewegt werden. Siehe Bild 3.1:

MOV P2 WTH M_HND(1) = 0 (MOVE)
Punkt-zu-Punkt-Bewegung (Gelenkinterpolation) zu P2

MVS P2 WTH M_HND(1) = 0 (MOVE STRAIGHT) Linearbewegung zu P2.
In beiden Fällen ist die Hand geschlossen.

Bewegung auf einem Kreis

MVC	**P1, P2, P3**	(MOVE CIRCLE)	Kreis durch drei Position fahren, Bild 3.2
MVR	**P1, P2, P3**	(MOVE RADIUS)	Kreis**bogen** durch drei Position fahren, Bild 3.3
MVR2	**P1, P2, P3**	(MOVE RADIUS)	Kreis**bogen** von P1 nach P2 mit P3 als Referenz,

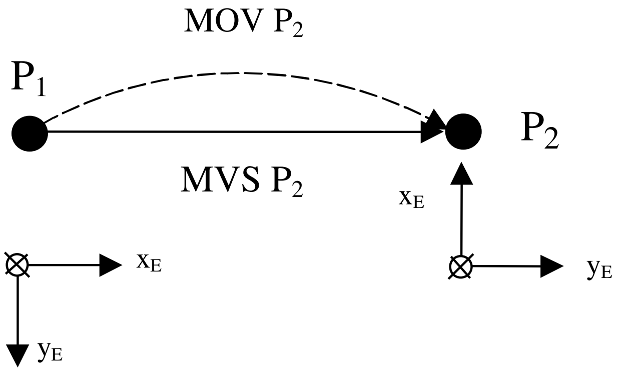


Bild 3.1: Anfahren einer Position

Der Punkt-zu-Punkt-Befehl **MOV** bewegt den TCP zur Position P2. Die Bahn dorthin ist nicht spezifiert. Meistens kommt eine Art Kreisbogen zustande. Im Punkt P1 erhält das Tool die durch P2 vorgeschrie-bene Orientierung.

Der Linearbefehl **MVS** bewegt den TCP ebenfalls zur Position P2, aber auf einer geraden Bahn. Die Orientierung wird vom Startpunkt ausgehend linear in die Zielorientierung überführt, d. h. sie wird auf der Geraden linear *interpoliert*. Dabei wird die Orientierung von P1 linear in die Orientierung von P2 überführt.

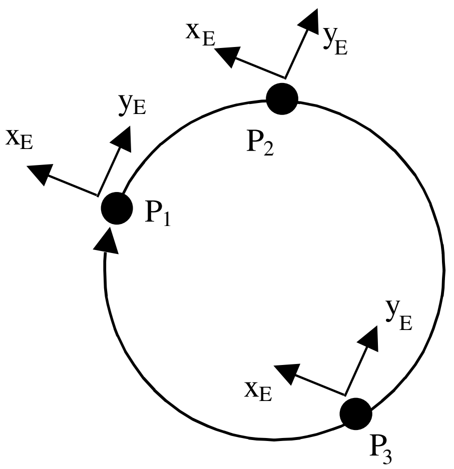


Bild 3.2: Kreis durch drei Positionen

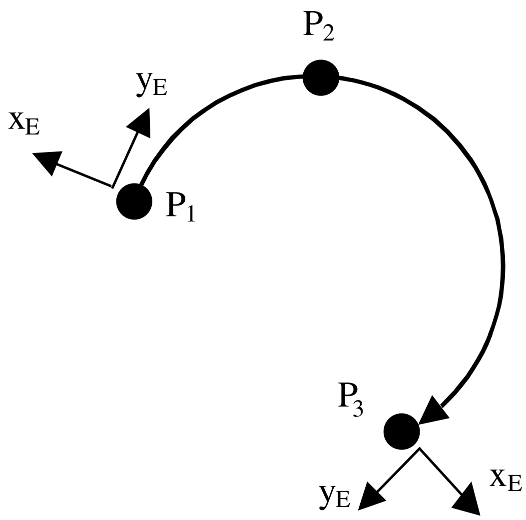


Bild 3.3:Kreisbogen über drei Punkte

Der Befehl MVC legt einen Kreis durch drei Positionen. Die Orientierung von Punkt P1 wird beibehalten.

Der Befehl **MVR** legt einen Kreisbogen von P1 durch P2 nach P3 (Bild 3.3). Die Orientierung wird *interpoliert*. Die Orientierung von P2 spielt keine Rolle.

Der Befehl MVR2 legt einen Kreisbogen von P1 nach P2 mit P3 als Referenzpunkt. Die Orientierung wird interpoliert. Die Orientierung von P3 spielt keine Rolle (Bild 3.4)

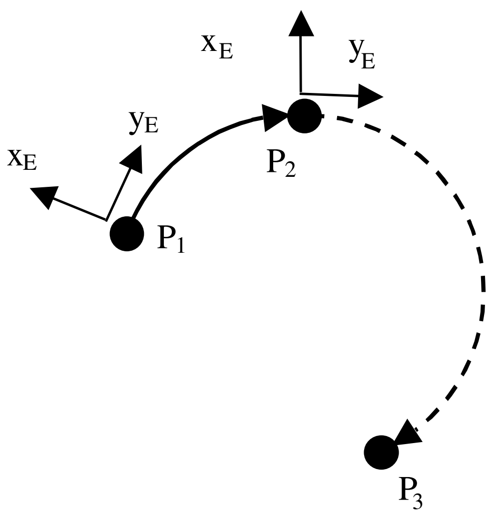


Bild 3.4: Kreis-2 durch drei Positionen

Relative Bewegungen

Diese Bewegungen finden relativ zum einer bestehenden Position statt. Damit kann man Positionen anfahren, die nicht als absolute Position definiert sind. Dadurch lassen sich absolute Positionen einsparen. Vorteilhaft ist auch,daß sich die relativen Positionen mit ihren absoluten Bezugspositionen verschieben, falls diese ihre Position ändern. Das folgende Beispiel macht dies deutlich.

Holen und Ablegen von Objekten

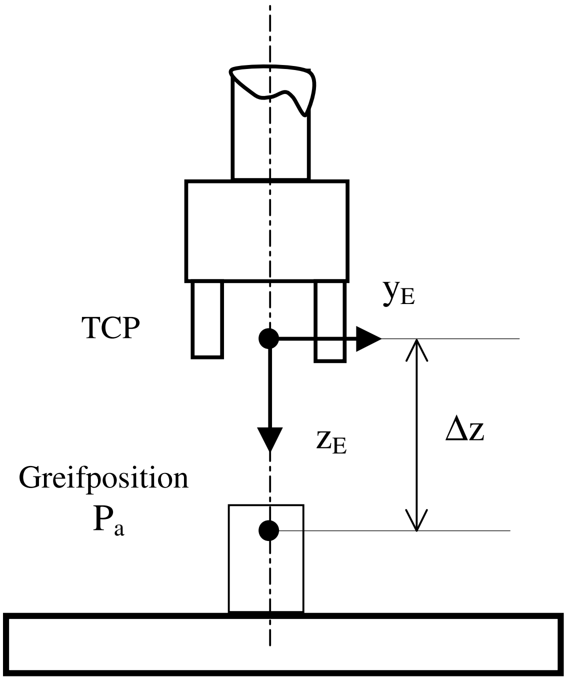


Bild 3.5: Greifen eines Objektes

Ein Greifer soll über ein Objekt mit der Greifposition P_a fahren. Das Toolkoordinatensystem ist mit seiner z_E-Achse nach unten gerichtet.

Mit dem Befehl

MOV Pa, -Δz

fährt man eine relative Position senkrecht über P_a an. Diese Position ist um Δz in negativer z_E-Richtung verschoben.

Der Befehl

MVS Pa WTH M_HND(1) = 1

führt den TCP mit geöffnetem Greifer auf einer Geraden zur Position P_a. Dann kann der Greifer geschlossen werden:

HCLOSE 1

Nach diesem Befehl ist eine Wartezeit erforderlich, damit der Greifer schließen kann. Der folgende Timerbefehl erzeugt 0,5 s Wartezeit.

DLY 0.5

Nach dem Greifen soll das Objekt zunächst ein kurzes Stück linear angehoben werden. Dazu dient der Befehl:

MVS Pa, -Δz

Annäherung in mehreren Richtungen

In komplizierteren Fällen muß die Annäherung in Richtung mehrerer Toolachsen erfolgen. Dazu dienen die MOV- und MVS-Befehle mit Positionsaddition.

MOV Pa + Pb

MVS Pa + Pb

In Bild 3.7 befindet sich der TCP an der Position P_a. Von dort soll er linear zur Position P_b bewegt werden. Dazu definiert man eine Deltaposition

Pb = (-Δx, -Δy, -Δz, 0, 0, 0)

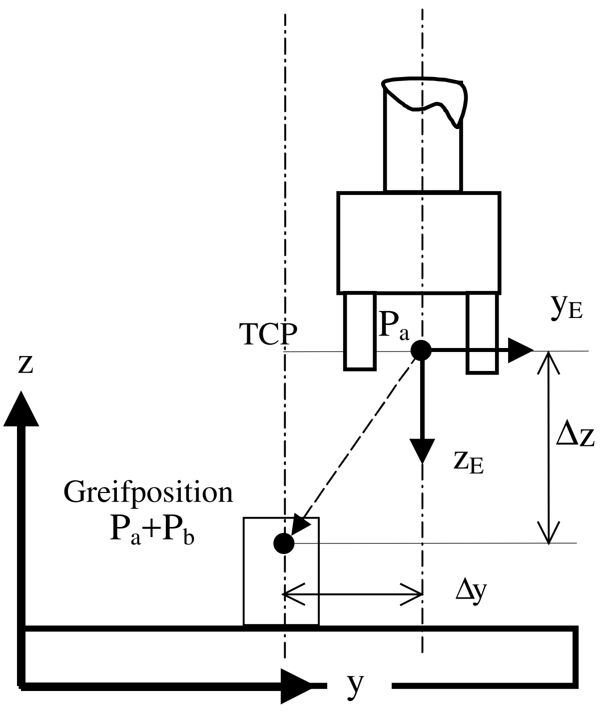


Bild 3.6: Annäherung in 2 Koordinaten-richtungen

Die Deltawerte beziehen sich auf die Achsen des Roboterkoordinatensystems x, y, z.

Die Greifposition P_a + P_b existiert nur relativ zur Position P_a und wird von dort aus angefahren. Mit dem MVS-Befehlt erfolgt die Annäherung linear.

MVS P_a+P_b

Die Annäherung kann also in alle drei Achsrichtungen (und auch mit Deltawerten für die Orientierungswinkel) erfolgen. Dann dreht sich der Greifer bei seiner Annäherung in eine neue Orientierung.

Verschiebung von Bahnverläufen

Das Muster von Bild 3.8 läßt sich mit relativen Befehlen sehr einfach abfahren. Eine mögliche Programmsequenz dafür zeigt das Programm von Bild 3.9.

In Zeile 10 wird die Position P_a definiert. Die Orientierung ist so gewählt, daß die z-Achse des Effektors nach unten zeigt. Der Greifer ist geschlossen.
In Zeile 40 wird die Verschiebung des Rechteckes in X-Richtung definiert (großer Wert ΔX).

Zeile 60 bildet zusammen mit Zeile 240 eine Wieder-holungsschleife mit 2 Durchläufen. In Zeile 60 wird der TCP über die Position P_a gefahren, d. h. um -Δz in Richtung der Toolachse relativ zu

P_a. Da die Toolachse nach unten geneigt ist, erfolgt die relative Verschiebung nach oben. Der Befehl von Zeile 70 senkt geradlinig auf die Startposition P_a des Recht-eckes ab.
Damit befindet sich der TCP in Arbeitshöhe. Nun wird die x-Komponente von Pa um Δx vergrößert und der veränderte Punkt in Zeile 110 linear angefahren. In gleicher Weise werden die weite-ren Seiten des Rechteckes abgefahren. Damit ist der TCP wieder am Startpunkt des Rechtecks angekommen. Nach jedem Bewegungsbefehlt steht eine Wartezeit, damit die Positionen auch erreicht werden.
In Zeile 240 erfolgt die Verschiebung des Rechteckes in X-Richtung um den Betrag ΔX. Nachdem beide Rechtecke abgefahren worden sind, wird der TCP geradlinig um Δz angehoben und dann der Roboter in eine sichere Position gefahren. P_Safe steht als Roboterstatusvariable zur Verfügung.
Für das Abfahren der Rechtecke wird also nur eine Position Pa benötigt. Alle anderen Positionen sind relativ zu Pa definiert.

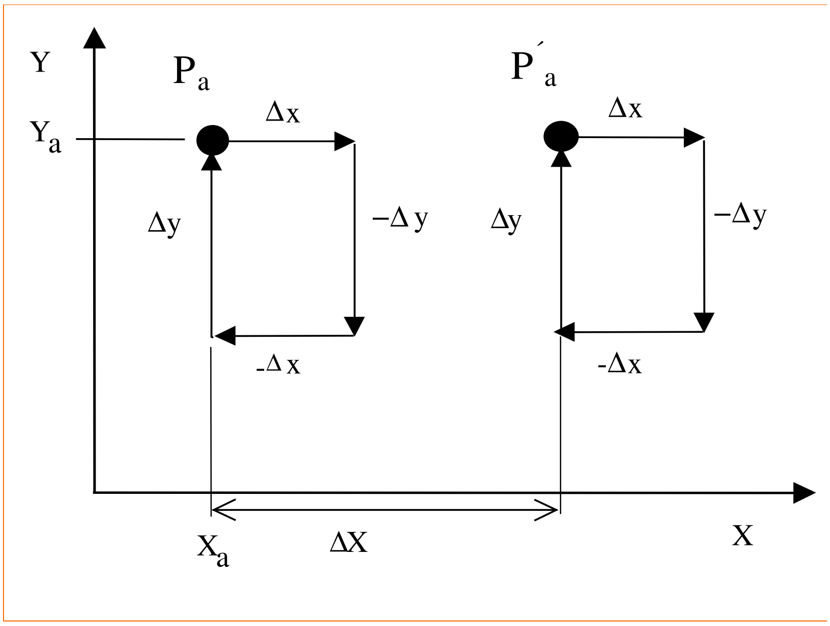


Bild 3.7: Verschobene Rechtecke

```
10      Pa = (Xa, Ya, Za, 180, 0, 90)(7, 0)
20      M1 = Δx
30      M2 = Δy
40      M3 = ΔX
50      FOR M4 = 1 TO 2
60      MVS  Pa, -Δz
70      MVS  Pa                  ' Startposition Rechteck
80      DLY  0.5
90      ´Rechteck abfahren
100     Pa.X = Pa.X + M1
110     MVS  Pa
120     DLY  0.5
130     Pa.Y = Pa.Y - M2
140     MVS  Pa
150     DLY  0.5
160     Pa.X = Pa.X - M1
170     MVS  Pa
180     DLY  0.5
190     Pa.Y = Pa.Y + M2
200     MVS  Pa         ' zurück an Startposition
210     DLY  0.5
220     MVS  Pa, -Δz
230     ´***************
240     Pa.X = Pa.X + M3      'Verschiebung Startpo-
sition Rechteck
250     NEXT
260     MVS  Pa, -Δz
270     MOV  P_Safe
250     END
```

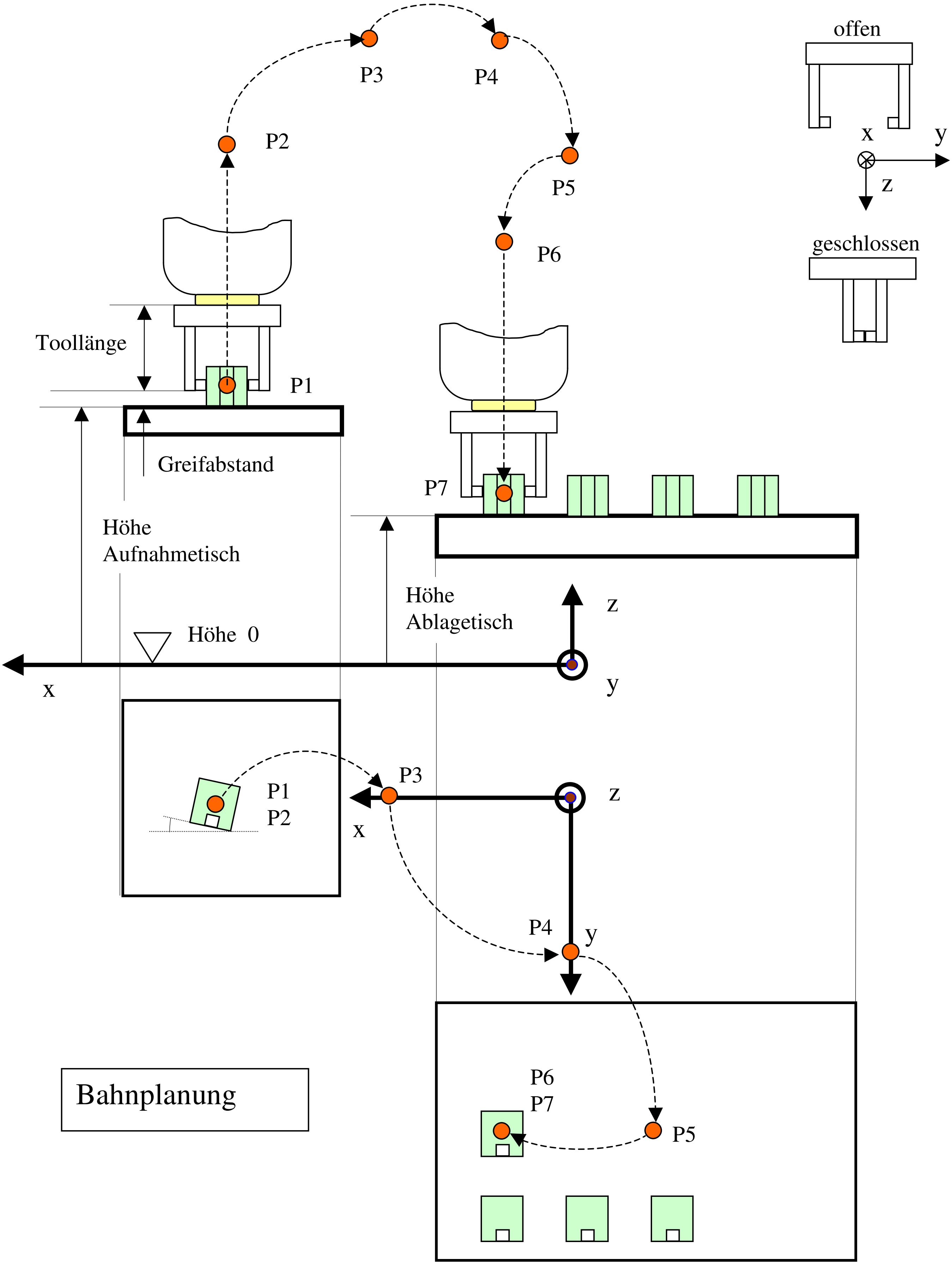
Bild 3.9: Programm für verschobene Rechtecke

3.2.1 Aufstellung eines Bewegungsplans

Ein Programm läßt sich nur dann sinnvoll schreiben, wenn die Aufgabe bekannt ist und die Positionen und Bahnverläufe definiert sind. Diese Festlegungen können mit einem Bewegungsplan getroffen werden. Folgende Angaben sollten enthalten sein:

1. Parkposition (Sichere Position)
2. Startposition
3. Zielpositionen
4. Zwischenpunkte,
   um Kollisionen zu vermeiden und Bewegungsarten wechseln zu können
5. Geschwindigkeiten: vor und nach dem Greifen langsam fahren
6. Wartezeiten:
   - nach Greifvorgängen
   - vor abrupten Richtungswechseln
7. Greifoperationen, mit Linearbewegungen durchführen
8. Gelenkbewegungen für schnelle Transportvorgänge
9. Tollänge, je nach Länge des Effektors
10. Endpunkt ist die Parkposition (Sichere Position)

3.2.2 Programmierbeispiel



Programm in MELFA-Basic III und IV

```
10    ‘Programm in der MELFABASIC Roboterprogrammiersprache
20    '
30    ‘Definition von Positionen:
40    JOVRD 10                                  ' PTP auf 10 % einstellen (zur Vorsicht)
50    P1=(0.00, 0.00, -60.00, 0.00, 0.00, 0.00)  ' Absenkung in z-Richtung
60    P6 = (60, 240, 210, 180, 0, 90)(7,0)      ' Ablegepunkt
70    P7 = (0, 0, 168, 0, 0, 0)                 ' Toollänge
80    P8 = (324.40, 0.00, 633.26, 0.00, 90.00, 0.00)(7,0)    'Sichere Position
90    PABLAGE = P6                              ' Ablegepunkt definieren
100   PSICHER = P8                              ' Sichere Position definieren

110   ‘Parameter initialisieren
120   TOOL  P7          ' Toolverschiebung  168 mm in Richtung der z-Toolachse
130   M1 = 12           ' Zählvariable für 12 Teile
140   SPD 222           ' 222 mm/s eistellen
150   OVRD 100          ' Override auf 100%, volle gewählte Geschwindigkeit 222 mm/s
160   JOVRD 100         ' PTP schnell

170   ‘Bewegungsbefehle
180   MOV PSICHER        ' die sichere Position anfahren
190   WHILE M1 > 1       ' Zählschleife
200   HOPEN 1            ' Greifer öffnen
210   MOV                'P3    ' In Wartestellung fahren

215   *WARTEN                             ' Marke für Wartefunktion
220   IF M_IN(5) = 0 THEN *WARTEN         ' warten bis  Bit 5 gesetzt ist (Teil ist da)

230   MOV   P2          ' Punkt 2 anfahren, PTP
240   OVRD 10           ' Geschwindigkeit auf 10 %  reduzieren
250   MVS  P2, 100 ' absenken auf Greifhöhe, entspricht P1, der nicht gebraucht wird
260   HCLOSE 1          ' Greifer schließen
270   DLY 0,5           ' 0,5 s Warten
280   MVS   P2          ' Linearbewegung zu Punkt 2
290   OVRD 100          ' volle Geschwindigkeit
300   MOV   P3          ' Zwischenpunkt P3 anfahren
310   MOV   P4          ' Zwischenpunkt P4 anfahren
320   MOV   P5          ' Verteilpunkt über der Palette anfahren
330   MOV   PABLAGE     ' Ablegepunkt anfahren (die Palette ist hier nicht programmiert)
340   OVRD 10           ' Geschwindigkeit auf 10 % reduzieren
350   MVS   ABLAGE + P1        ' 60 mm gegen die Richtung der z-Achse absenken
360   HOPEN 1           ' Greifer öffnen und Teil ablegen
370   DLY    0,5        ' 0,5 s warten
380   MVS, -50          ' linear 50 mm über die aktuelle Position fahren
390   OVRD 100          '  schnell
400   MOV   P4          '  zu P4 mit PTP
410   M1 = M1 -1        '  Palettenplatz runterzählen (Palette hier nicht definiert)
420   WEND              '  Ende der WHILE-Schleife
430   MOV PSICHER
440   END               '  Programmende
```

4 Literatur

[1] Bögelsack, G.; Kallenbach, E., Linnemann, G.:
Roboter in der Gerätetechnik
Hüthig Verlag, Heidelberg, 1984

[2] Bartenschläger, J.; Hebel, H.; Schmidt, G.:
Handhabungstechnik mit Robotertechnik – Funktion, Arbeitsweise, Programmierung.
Vieweg Verlag, 1998

[3] Hesse, Stefan: Industrieroboterpraxis.
- Automatisierte Handhabung in der Fertigung -
Vieweg Verlag, 1998

[4] Stöcker, H.: Taschenbuch mathematischer Formeln und moderner Verfahren.
Verlag Harri Deutsch, Frankfurt, 1995

[5] Güsmann, B.:
Einführung in die Roboter-Programmierung
Vieweg Verlag 1992

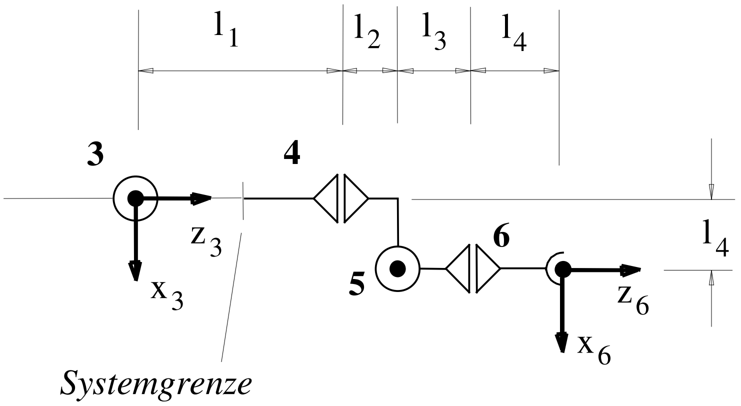
[6] Weber, Wolfgang: Industrieroboter – Methoden der Steuerung und Regelung.
Hanser Verlag 2009

[7] RV-E2 Bedienungs- und Programmieranleitung. Mitsubishi Electric Europe, 1998

[8] CR1-CR9 Controller Instruction Manual (Detailed explanations of functions and operations). Mitsubishi Electric Corporation, 1999

5 Aufgaben

Aufgabe 1



Geben ist die Hand an einem Roboter mit den Drehgelenken (4, 5 und 6). Vom Roboter ist noch das Gelenk 3 dargestellt.

Bestimmen Sie gemäß Denavit und Hartenberg

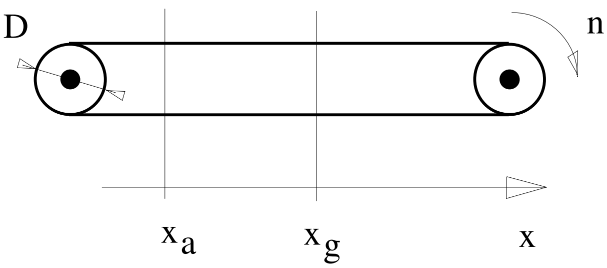
a) die Gelenkparameter 4, 5 und 6
b) die Transformationsmatrizen $\underline{T}_{34}$ und $\underline{T}_{56}$.

Aufgabe 2

$$\mathbf{B} = \begin{Bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & -1 \end{Bmatrix}$$

Ein Effektor besitzt zunächst die Orientierung des Basissystems. Nach einer Drehung α um die x-Achse und nachfolgender Drehung γ um seine z-Achse hat er die Orientierung B:

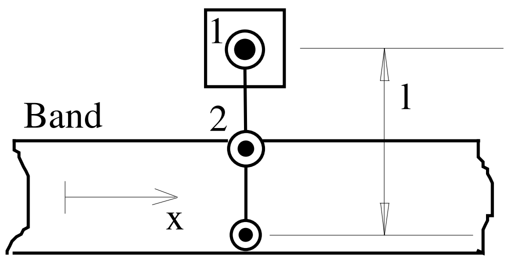
a)Ermitteln Sie α und γ mit Hilfe der Rotationsmatrizen $\mathbf{R}_X(\alpha)$ und $\mathbf{R}_Z(\gamma)$.
b)Zeichnen Sie das gedrehte Effektorsystem im Basissystem



Aufgabe 3

Ein Förderband transportiert Teile zu einem Roboter. Der Greifpunkt liegt bei x_g, der Auflegepunkt bei x_a. Die Position wird an einer Bandwalze mit einem Inkrementalgeber mit N = 2000 Inkrementen gemessen.
Folgende Daten gelten: D = 12 cm
$x_g - x_a = 150$ cm n = 80 min^{-1}

a) Wieviele Inkremente Z werden gezählt, bis ein Teil im Greifpunkt angekommen ist.
b) Mit welcher Frequenz f_N werden die Inkremente geliefert?
c) Mit welcher Genauigkeit wird das Teil bei x_g positioniert?



Aufgabe 4

Auf das Band greift ein SCARA-Roboter zu. Seine Armlänge beträgt l = 850 mm. Der Drehwinkel wird direkt an Gelenk 1 mit einem Resolver gemessen. (Gelenk 2 bleibt unberücksichtigt)
Welche Auflösung muß der RDC von Gelenk 1 mindestens haben, wenn die Positions-abweichnung zwischen Band und Greifer ±1 mm nicht überschreiten soll.