EASY-ROB

3D Robot Simulation Tool

Janko Härtig

4. März 2004

Inhaltsverzeichnis:

1. Allgemein

1.1. Allgemeine Beschreibungen

Kinematik = Bewegungslehre

Teilgebiet der Mechanik:
Die Kinematik befasst sich mit der geometrischen Beschreibung von Bewegungsverhältnissen. Die Kräfte, die diese Bewegung verursacht, bleiben dabei unberücksichtigt.

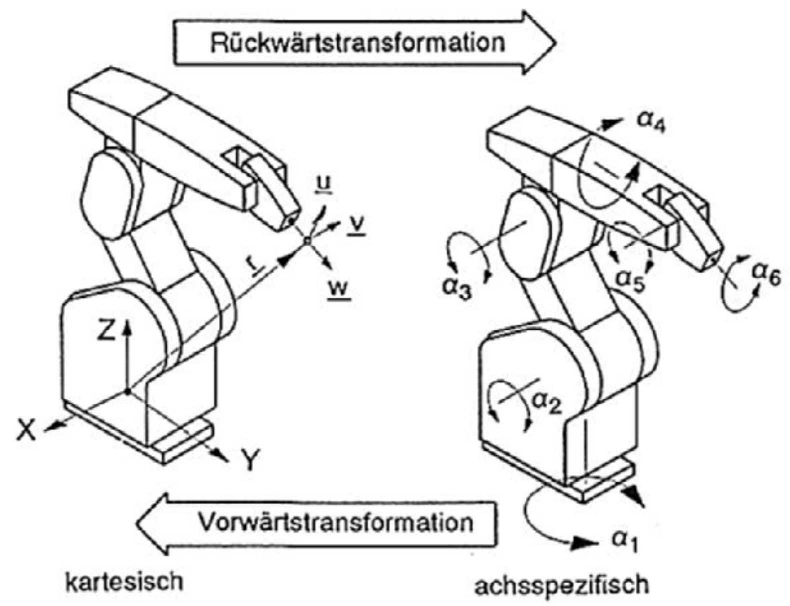
Beschreibung:
Roboter sind aus mehreren Gliedern zusammen gesetzt. Um eine bestimmte Stellung des Roboters zu erreichen müsste man jedes einzelne Glied in die gewünschte Lage bringen.
Vorwärts Kinematik (Vorwärtstransformation)
Die *inverse Kinematik (Rückwärtstransformation)* vereinfacht das, mit ihrer Hilfe Verhält sich der Roboter wie eine Gliederkette.

Zum Beispiel:
Es genügt an einer Fingerkuppe zu ziehen um den ganzen Finger oder sogar den gesamten Arm zu strecken.

Koordinatentransformation:
Ist ein numerisches Verfahren um die Steuerinformationen für den Roboter von kartesischen Koordinaten in geräteeigene Koordinaten umzurechen. Denn es ist für den Menschen anschaulicher im Kartesischen Raum „zu denken" als in Roboter- oder Winkelkoordinaten.

1.2. Direkte Koordinatentransformation (Vorwärtstransformation)

Es sind die Winkelstellungen der einzelnen Gelenke und die Länge der einzelnen Arme bekannt. Aus diesen *Gelenkkoordinaten* wird der resultierende Endpunkt berechnet.
Die Vorwärtstransformation wird z.B. zur Anzeige gebraucht und kommt bei der Teach-in Programmierung vor, Sie ist auch zum Einrichten und Überprüfen eines Roboters von Bedeutung.



1.3. Inverse Koordinatentransformation (Rückwärtstransformation)

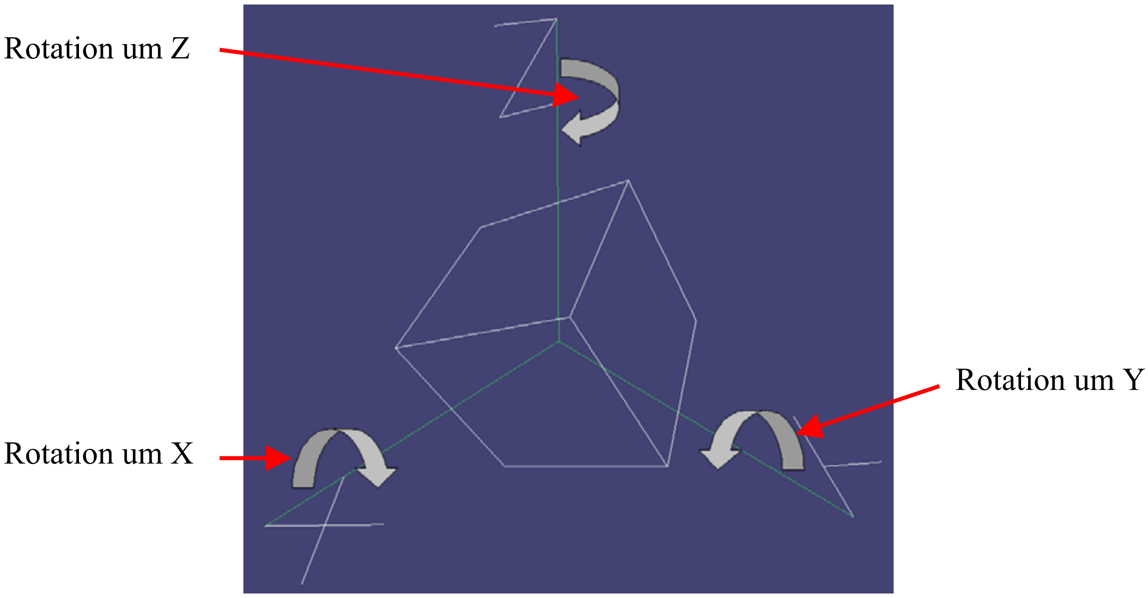
Bei der Rückwärtstransformation ist der Endpunkt bekannt und es müssen dazu die resultierenden Winkelstellungen berechnet werden.
Diese Transformation ist sehr rechenintensiv und kann daher zu zeitlichen Engpässen in der Echtzeitberechnung führen.

1.4. Framentransformation (Homogene Transformation)

In der Homogenen Transformation, welches ein einziges festes *Welt-Koordinatensystem* besitzt, werden die einzelnen Gelenke zueinander dargestellt. Also die relative Lage zweier benachbarter Gelenke in Abhängigkeit der jeweiligen Winkelkoordinate.

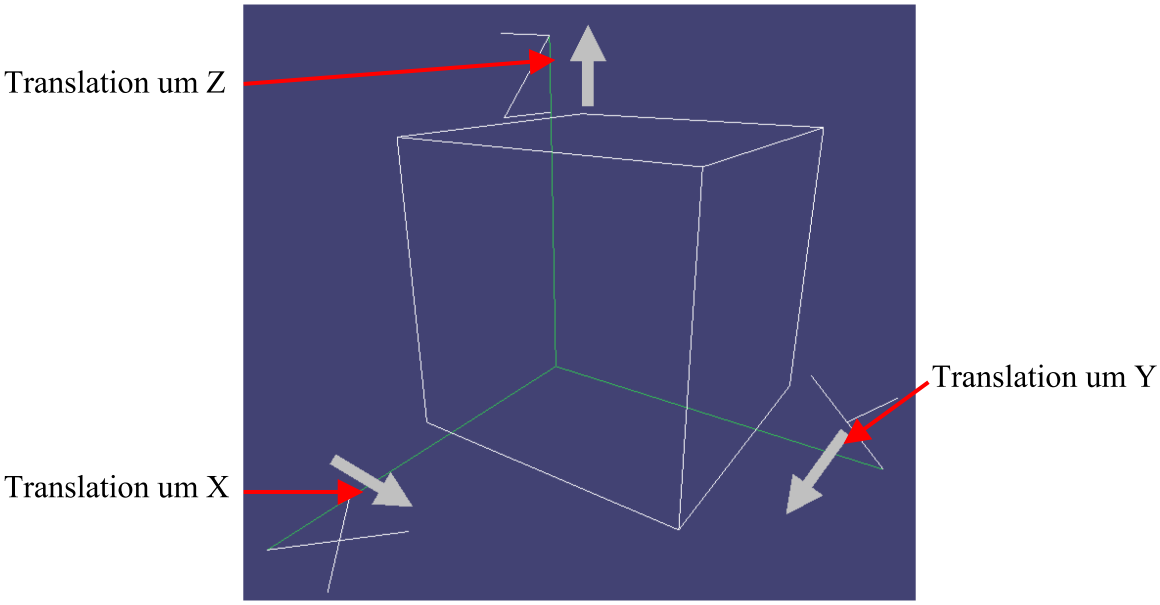
Eine Gelenkbewegung besteht im allgemeinen aus Rotation und Translation in einem körperfesten Koordinatensystem.

Bei einer reinen Rotation ist folgende 3x3 Matrix definiert:



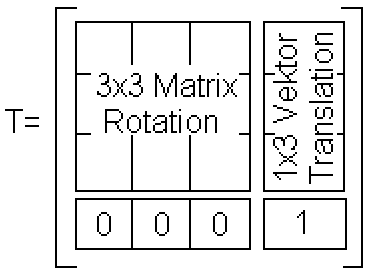
$$Rotation(T) = \begin{bmatrix} u_x & v_x & w_x \\ u_y & v_y & w_y \\ u_z & v_z & w_z \end{bmatrix} = \left[Rot_x(T), Rot_y(T), Rot_z(T)\right]$$

Bei einer reinen Translation ist folgender 1x3 Vektor definiert:



$$Position(T) = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \left[p_x, p_y, p_z\right]^T$$

Um die Berechnung der zwei völlig verschiedenen Operationen (Matrizen- und Vektorberechnung) zu vereinfachen wird von Gleichungssystemen in homogene 4x4 Matrizen umgewandelt.



$$T = \left[\begin{array}{ccc|c} u_x & v_x & w_z & p_x \\ u_y & v_y & w_y & p_y \\ u_z & v_z & w_z & p_z \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

Durch die Erweiterung der letzten Zeile (0,0,0,1) verändert sich der Wert nicht, es vereinfacht aber die Berechnung.

Vom festen Weltkoordinatensystem können einzelne Bewegungen durchgeführt werden.

Translation um X, Y, Z

$$T_{x,y,z} = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation um X mit dem Winkel α

$$R_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation um Y mit dem Winkel β

$$R_{y,\beta} = \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation um Z mit dem Winkel γ

$$R_{z,\gamma} = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 & 0 \\ \sin\lambda & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Eine resultierende Transformation ergibt sich nun aus der Multiplikation von den einzelnen Translations und Rotationsmatrizen.

$$T = \left[T_{x,y,z}\right] * \left[R_{x,\alpha}\right] * \left[R_{y,\beta}\right] * \left[R_{z,\gamma}\right]$$

1.5. Denavit-Hartenberg

Definition:

Beschreibung zweier benachbarter Bezugssysteme durch vier Transformationen, diese sind immer eindeutig (2 Rotationen, 2 Translationen).

Um mehrere Gelenkachsen, die beliebig zueinander verlaufen, zu berücksichtigen wird für die homogene 4x4 Matrix das Denavit-Hartenberg verfahren angewandt.
Damit lassen sich Transformationen für eine Kinematische Kette über mehrere Koordinatensysteme hinweg durchführen.

Das Koordinatensystem *n-1* wird durch Transformation in das Koordinatensystem *n* durch vier nacheinander auszuführende Transformationsschritte beschrieben.

1. Rotation um Θ_n um die Achse z_{n-1} bis x_{n-1} parallel zu x_n liegt
2. Verschiebung um d_n in Richtung z_{n-1} bis sich x_{n-1} und x_n decken
3. Verschiebung um a_n in Richtung x_n bis Koordinatenursprünge gleich sind
4. Rotation um α_n um die Achse x_n bis die Koordinatensysteme identisch sind.

Die Gesamttransformation einer Achse lautet:

$$T_{n-1,n} = [Rot(z_{n-1},\Theta)] * [Trans(0,0,d_n)] * [Trans(a_n,0,0)] * [Rot(x_n,\alpha_n)]$$

$$T_{n-1,n} = \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 & 0 \\ \sin\Theta & \cos\Theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

Diese Matrix kann mit jeder weitern Gelenkachse fortgeführt werden.

Die allgemeine Gesamttransformation lautet:

$${}^{m}_{n}T_{gesamt} = \left({}^{n+1}_{n}T\right) * \left({}^{n+2}_{n+1}T\right) \ldots \left({}^{m}_{m-1}T\right)$$ (wie ist T_{n+1} aus T_n entstanden usw.)

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_0 = \left({}^{n+1}_{n}T\right) * \left({}^{n+2}_{n+1}T\right) \ldots \left({}^{m}_{m-1}T\right) * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_m$$

Daraus ergibt sich: X_{TCP}
Y_{TCP}
Z_{TCP} im Bezug zum Basiskoordinatensystem (0,0,0).

Aus der Denavit-Hartenberg Transformation (Vorwärtstransformation) T_{gesamt} lässt sicht jetzt die inverse Transformation (Rückwärtstransformation) T^{-1}_{gesamt} erstellen.

Übung an Beispielen siehe dazu:

3.1.1. Allgemeine Erläuterung zum erstellen von Denavit-Hartenberg
3.1.2. Denavit-Hartenberg - SCARA Roboter
3.1.3. Denavit-Hartenberg - KUKA Roboter

1.6. Jakobi-Matrix

$$D = J * D_\Theta$$

Die Jakobi Matrix stellt die Geschwindigkeit der kartesischen Koordinaten D und die Geschwindigkeit der gelenk Koordinaten D_Θ in Beziehung.

Allgemeine Jakobi-Matrix:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} dx/d\Theta_1 & \cdots & dx/d\Theta_n \\ \vdots & \ddots & \vdots \\ d\delta_z/d\Theta_1 & \cdots & d\delta_z/d\Theta_n \end{bmatrix} * \begin{bmatrix} \dot{\Theta}_1 \\ \vdots \\ \vdots \\ \dot{\Theta}_n \end{bmatrix}$$

Die drei Elemente $\dot{x}, \dot{y}, \dot{z}$ stehen für die Translationsgeschwindigkeit, die drei Elemente $\dot{\omega}_x, \dot{\omega}_y, \dot{\omega}_z$ stehen für die Rotationsgeschwindigkeit. Um die jeweilige Hauptachsen des Koordinatensystems.

Häufig werden Roboter mit f=6 Achsen eingesetzt:

$$D = \begin{bmatrix} d_x \\ d_y \\ d_z \\ \delta_x \\ \delta_y \\ \delta_z \end{bmatrix} \quad \text{abgeleitet nach der Zeit} \quad d/dt = \dot{D} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix}$$

$$D_\Theta = \begin{bmatrix} d\Theta_1 \\ d\Theta_2 \\ d\Theta_3 \\ d\Theta_4 \\ d\Theta_5 \\ d\Theta_6 \end{bmatrix} \quad \text{abgeleitet nach der Zeit} \quad d/dt = \dot{D}_\Theta = \begin{bmatrix} \dot{\Theta}_1 \\ \dot{\Theta}_2 \\ \dot{\Theta}_3 \\ \dot{\Theta}_4 \\ \dot{\Theta}_5 \\ \dot{\Theta}_6 \end{bmatrix}$$

Damit ergibt sich folgende 6x6 Matrix:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} & J_{13} & J_{14} & J_{15} & J_{16} \\ J_{21} & J_{22} & J_{23} & J_{24} & J_{25} & J_{26} \\ J_{31} & J_{32} & J_{33} & J_{34} & J_{35} & J_{36} \\ J_{41} & J_{42} & J_{43} & J_{44} & J_{45} & J_{46} \\ J_{51} & J_{52} & J_{53} & J_{54} & J_{55} & J_{55} \\ J_{61} & J_{62} & J_{63} & J_{64} & J_{65} & J_{65} \end{bmatrix} * \begin{bmatrix} \dot{\Theta}_1 \\ \dot{\Theta}_2 \\ \dot{\Theta}_3 \\ \dot{\Theta}_4 \\ \dot{\Theta}_5 \\ \dot{\Theta}_6 \end{bmatrix}$$

Aus der Jakobi-Matrix (Vorwärtstransformation) D lässt sicht jetzt die inverse Transformation (Rückwärtstransformation) J^{-1} erstellen.

$$D_\Theta = J^{-1} * D$$

Übung an einem Beispiel siehe dazu:

3.1.4. Jakobi-Matrix - einfacher RT Roboter

1.7. Die Zentralhand

Ein Standart Roboter besteht im allgemeinen aus f=6 Achsen Rotatorisch RRR:RRR.

$$\underbrace{R\ R\ R}_{\text{Positionierung}} : \underbrace{R\ R\ R}_{\text{Orientierung}}$$

der Zentralhand

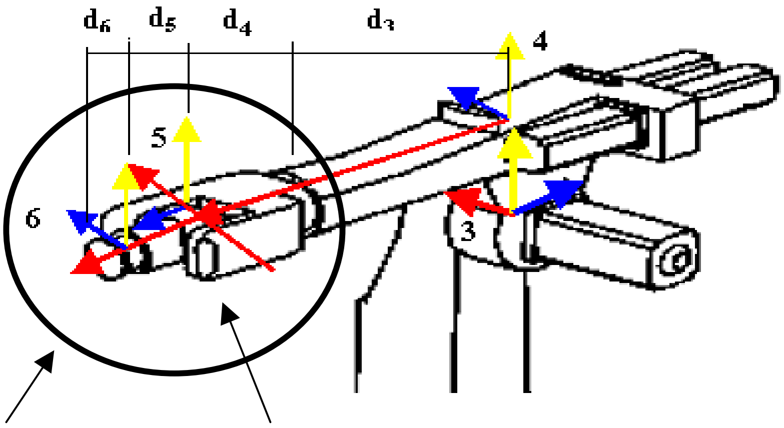
Die ersten drei Rotationen dienen zur Positionierung der *Zentralhand* im Raum.
z.B. Körper, Oberarm und Unterarm.

Die anderen drei Rotationen dienen zur Orientierung der *Zentralhand* im Raum.
z.B. Hand

Zentralhand:

- Alle drei Achsen (Achse 4,5 und 6) schneiden sich durch Verlängerung ihrer z-Achsen in einem zentralen Punkt.
- Aber die Koordinatensysteme 4,5 und 6 müssen nicht auf einem Punkt liegen.

Am Beispiel KUKA wird dies verdeutlicht:



Zentralhand mit zentralem Punkt
in dem sich die drei z-Achsen schneiden.

Da es sich nur um eine Verlängerung der z-Achsen handelt kann am KUKA Roboter angenommen werden, das:

- d_3 mit d_4 als eine Länge angesehen wird
- d_5 mit d_6 als eine Länge angesehen wird

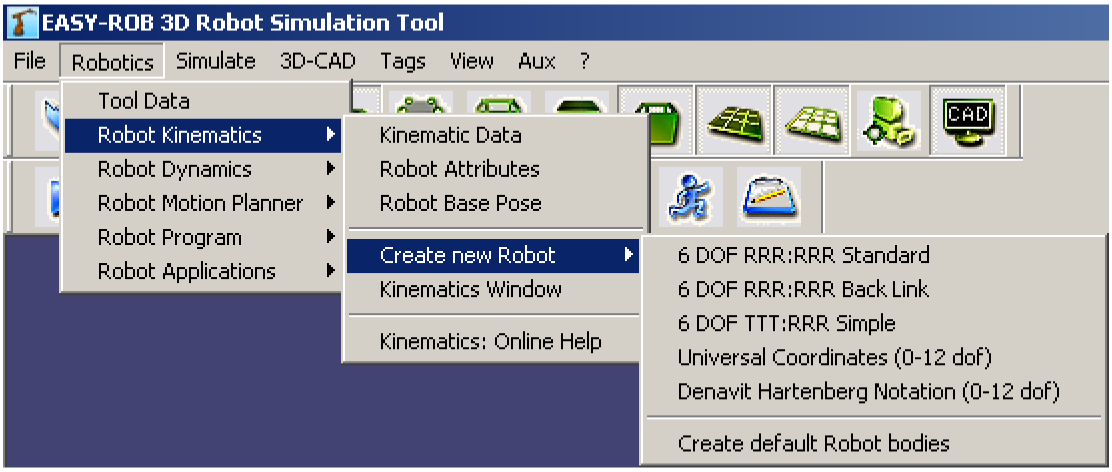
2. Anwendung in EASY-ROB

2.1. Direkte Koordinatentransformation

2.1.1. Allgemeine Erklärungen

- Es sind die Winkelstellungen und die Länge der einzelnen Achsen bekannt.
  Aus diesen Gelenkkoordinaten wird der resultierende Endpunkt des TIP berechnet.

In EASY-ROB kann eine Direkte Koordinatentransformation in mehreren Varianten berechnet und dargestellt werden.



Es wird im folgenden auf die drei wesentlichen Transformationen eingegangen.

- 6 DOF RRR:RRR Standart
- Denavit-Hartenberg
- Universal Koordinaten

2.1.1.1. Standard RRR:RRR

Ein Standart Roboter besteht im allgemeinen aus f=6 Achsen Rotatorisch RRR:RRR.

$$\underbrace{R_Z\ R_Y\ R_Y}_{\text{Positionierung}} : \underbrace{R_Z\ R_Y\ R_Z}_{\text{Orientierung}}$$

der Zentralhand

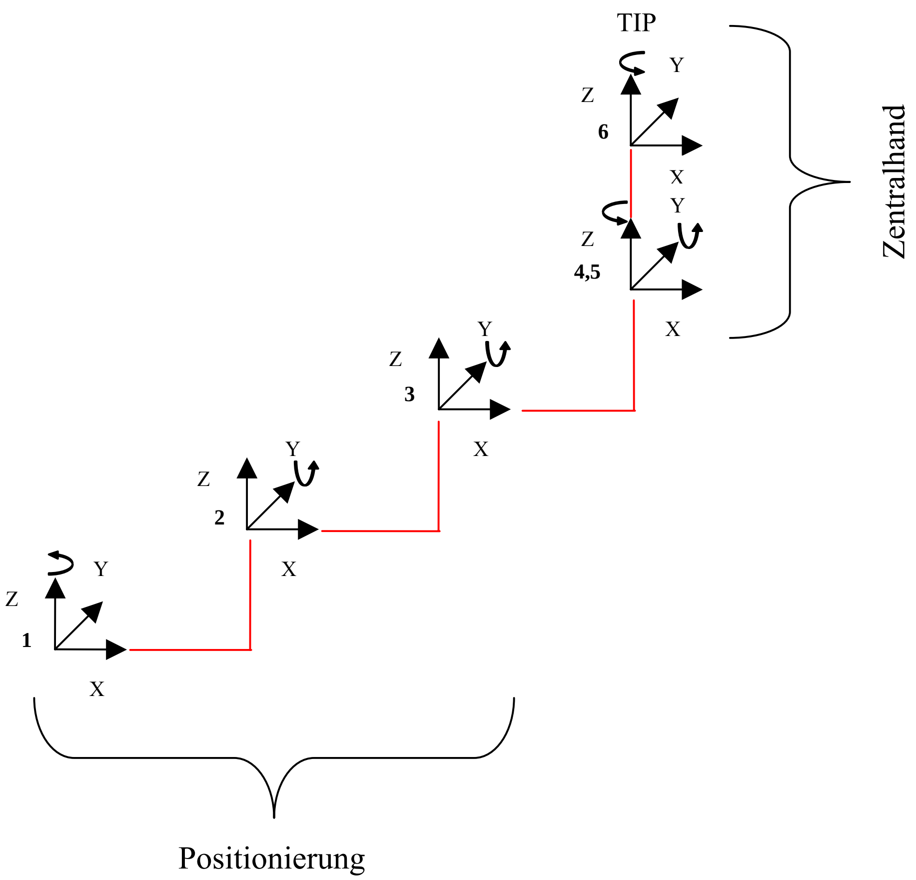
Hinweise zur Zentralhand siehe 1.7. Die Zentralhand.

Bei der Standard Transformation RRR:RRR werden die Maße angegeben, welche zur Positionierung der Zentralhand benötigt werden, Koordinatensystem 1 bis 3.

Außerdem das Maß zur Bestimmung des TIP von der Zentralhand aus, Koordinatensystem 4 bis 6.

Zur Vereinfachung und für eine bessere Übersicht kann folgende Tabelle dienen:

Koordinatensystem	Bewegung	Länge in x in mm	Länge in z in mm	Länge in y in mm
1	R_Z	l1x	l1z	l1y
2	R_Y	l2x	l2z	l2y
3	R_Y	l3x	l3z	l3y
4	R_Z	/	/	/
5	R_Y	/	l5z	/
6	R_Z	/	/	/



2.1.1.2. Denavit-Hartenberg

Mittels Denavit-Hartenberg wird eine Tabelle erstellt und diese Parameter werden dann in EASY-ROB eingetragen.

Achse	Bewegung	Θ in °	d in mm	a in mm	α in °
1	R oder T	Θ_1	d_1	a_1	α_1
2	R oder T	Θ_2	d_2	a_2	α_2
3	R oder T	Θ_3	d_3	a_3	α_3
4	R oder T	Θ_4	d_4	a_4	α_4
usw.					

Hinweise dazu siehe: 1.5. Denavit-Hartenberg

2.1.1.3. Universal Koordinaten

Bei der Universellen Koordinatentransformation wird ein festes Koordinatensystem erstellt (Base) und an den Achsen entlang von Drehachse zu Drehachse verschoben. Ohne Beachtung von Vorschriften nach Denavit-Hartenberg und Standard RRR:RRR.

Somit kann an jeder Achse Rotation oder Translation beschrieben werden und es kann das Koordinatensystem n-1 an jeder Achse zum Koordinatensystem n verschoben werden.

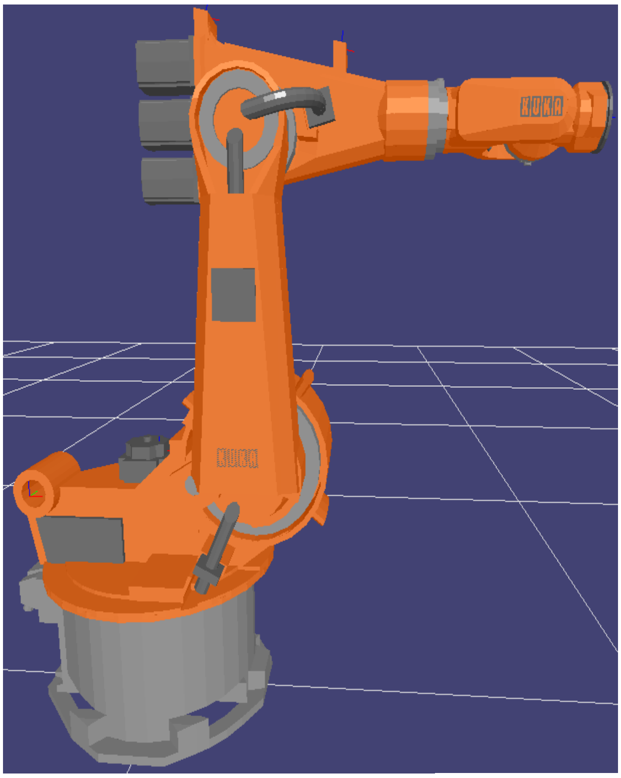
Zur Vereinfachung und für eine bessere Übersicht kann folgende Tabelle dienen:

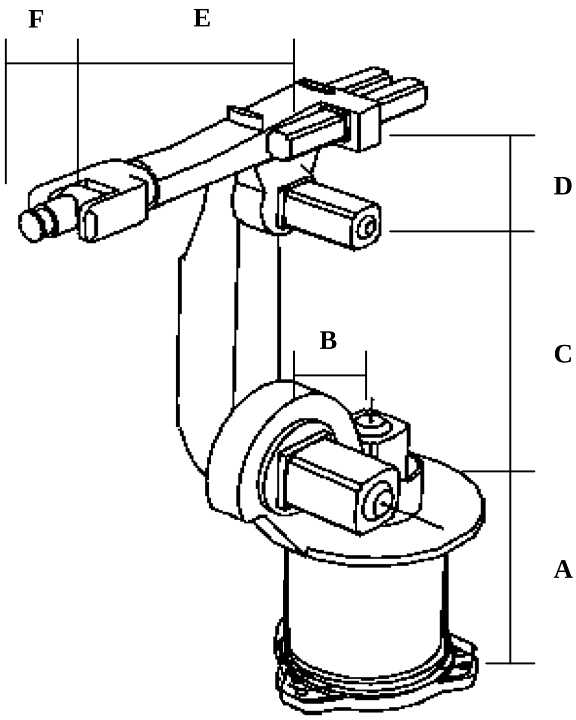
Achse	Bewegung	Länge in x in mm	Länge in z in mm	Länge in y in mm
1	R oder T	x_1	z_1	y_1
2	R oder T	x_2	z_2	y_2
3	R oder T	x_3	z_3	y_3
4	R oder T	x_4	z_4	y_4
usw.				

2.1.2. KUKA 6 Achs Roboter

Im folgenden wird der KUKA Roboter KR 125-2 Schritt für Schritt in EASY-ROB erstellt.

In den Transformationen:
- Standard RRR:RRR
- Denavit-Hartenberg
- Universal Koordinaten





KUKA2

KUKA KR 125-2

Die Parameter in der Tabelle wurden aus dem Datenblatt der Roboters entnommen.

Achse	Maß in mm
A	865
B	410
C	1000
D	45
E	1000
F	210

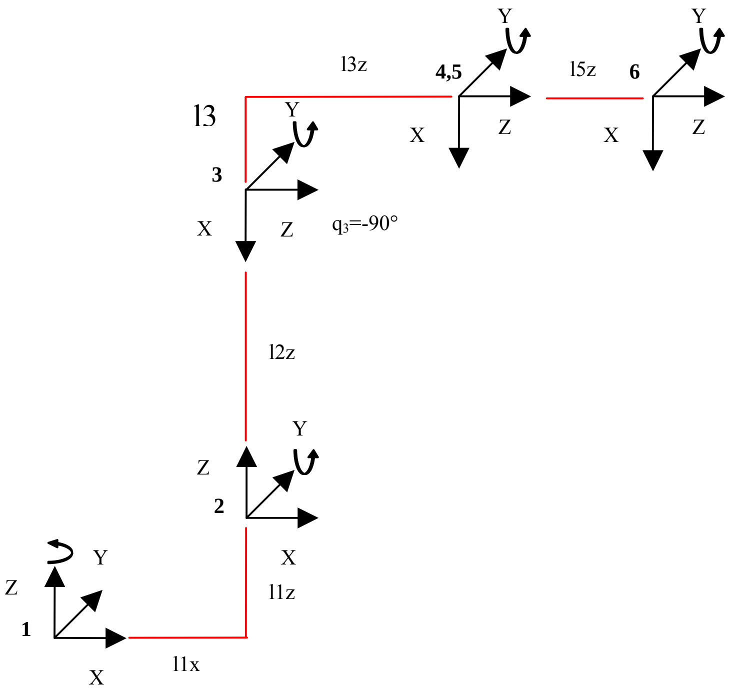
2.1.2.1. Standard RRR:RRR

Aus dem Datenblatt werden die Parameter für die einzelnen Längen bestimmt, wobei die Rotation vorgeschrieben ist.

$R_Z\ R_Y\ R_Y : R_Z\ R_Y\ R_Z$

Koordinatensystem	Rotation	Länge in x in mm	Länge in z in mm	Länge in y in mm
1	$R_Z=0°$	410	865	0
2	$R_Y=0°$	0	1000	0
3	$R_Y=-90°$	-45	1000	0
4	$R_Z=0°$	0	0	0
5	$R_Y=0°$	0	210	0
6	$R_Z=0°$	0	0	0

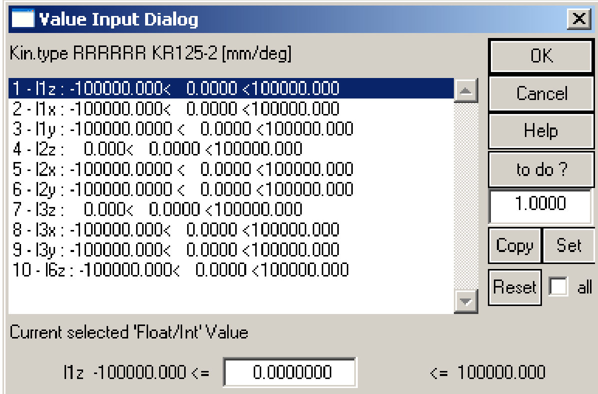
Um eine korrekte Stellung für den KUKA zu bekommen muss das Koordinatensystem 3 mit R_Y um -90° gedreht werden.



Erstellung mit EASY-ROB:

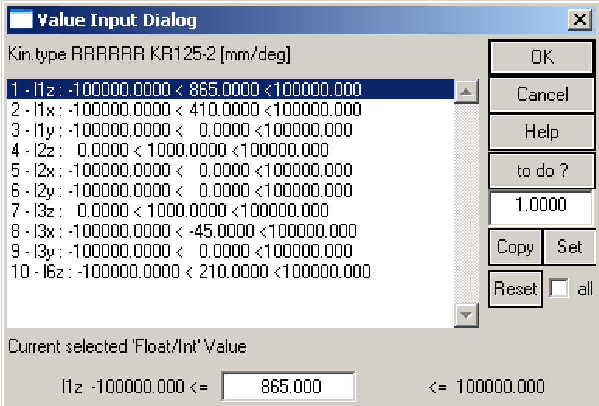
Schritt 1:
Auswahl der Transformation:

Robotics/ Robot Kinematics/ Create new Robot/ 6 DOF RRR:RRR Standard



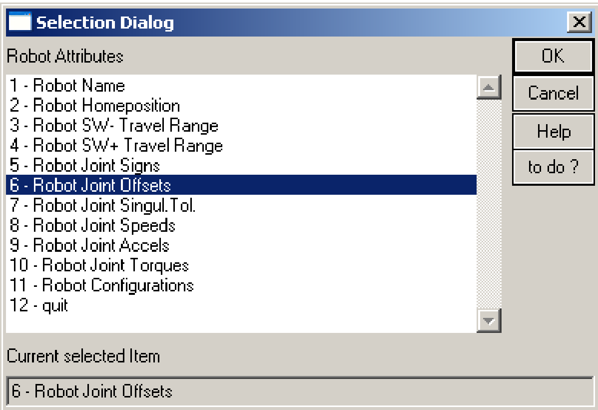
Schritt 2:
Hier werden nun die Parameter eingetragen, für die drei Koordinatensysteme der Positionierung. Und die Länge zum TIP.

Hinweis: Bis zur Version 3.6 steht l6z anstatt l5z geschrieben.

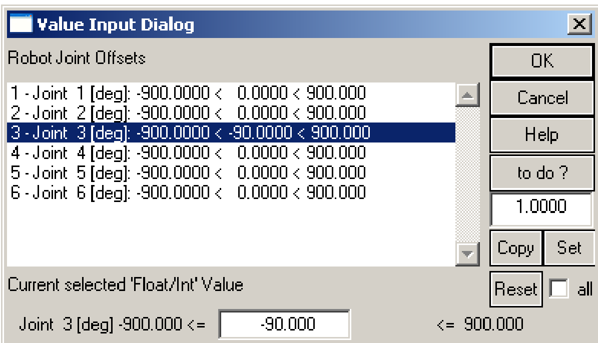


Schritt 3:
Die Achse 3 mit -90° definieren:

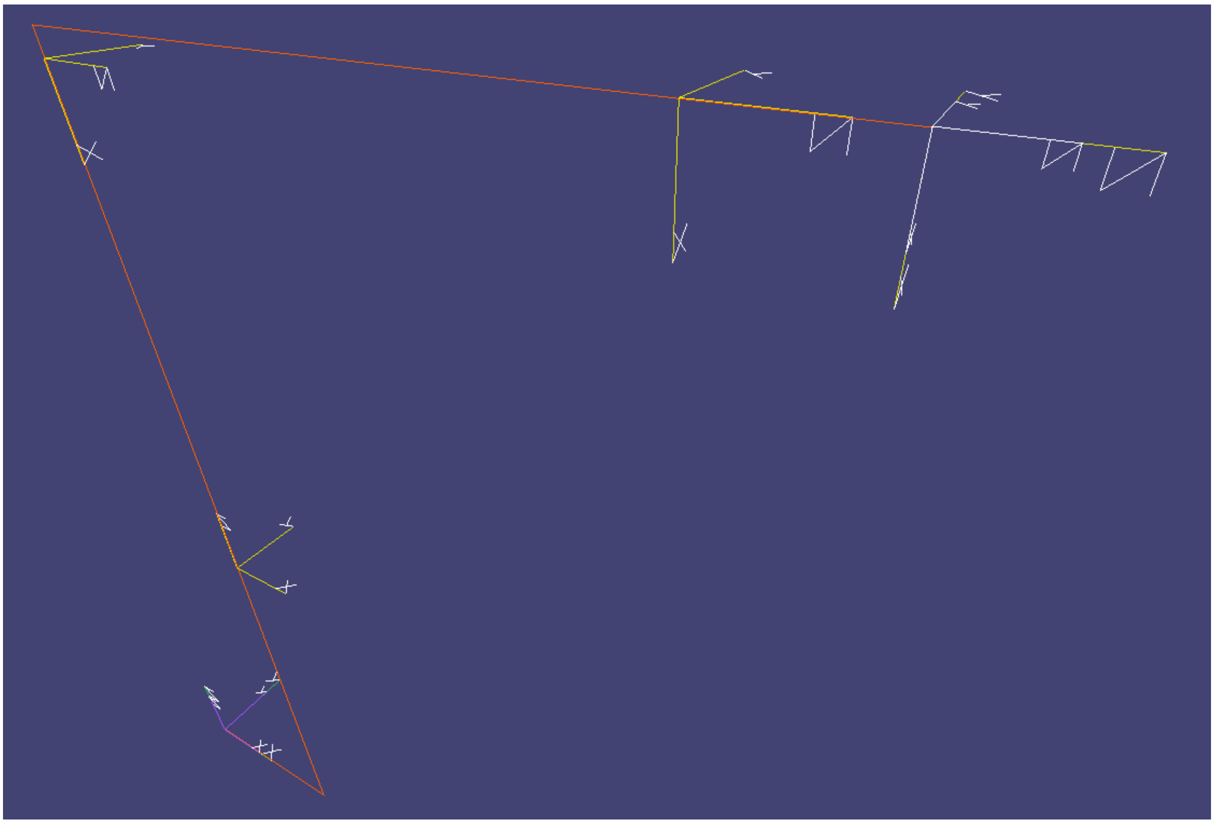
Robotics/ Robot Kinematics/ Robot Attributes/ 6-Robot Joint Offset



Schritt 4:
Hier wird die Achse 3 (*3 - Joint 3*) definiert.



Nach der Eingabe aller Parameter erscheinen die Koordinatensysteme mit ihren Längen in EASY-ROB.



Mit dem Menü *Robot Joints* können Sie zur Kontrolle alle Aschen einzeln bewegen.

Achse	Bewegung
1	linke Maustaste
2	mittlere Maustaste
3	rechte Maustaste
4	Strg + linke Maustaste
5	Strg + mittlere Maustaste
6	Strg + rechte Maustaste

2.1.2.2. Denavit-Hartenberg

Die Tabelle wurde mittels Denavit-Hartenberg erstellt.

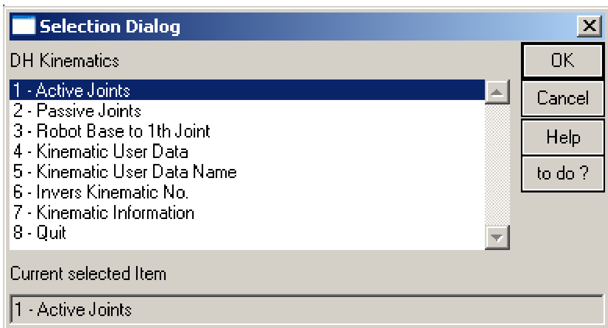
Hinweise dazu siehe: 1.5. Denavit-Hartenberg
3.1.3. Denavit-Hartenberg - KUKA Roboter.

Achse	Θ in °	d in mm	a in mm	α in °
1	0	865	410	90
2	90	0	1000	0
3	0	0	45	90
4	0	1000	0	-90
5	0	0	0	-90
6	0	210	0	0

Erstellung mit EASY-ROB.

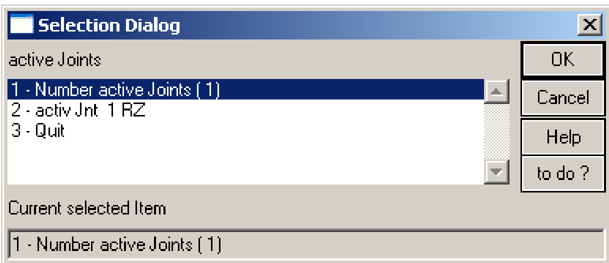
Schritt 1:
Auswahl der Transformation:

Robotics/ Robot Kinematics/ Create new Robot/ Denavit-Hartenberg Notation

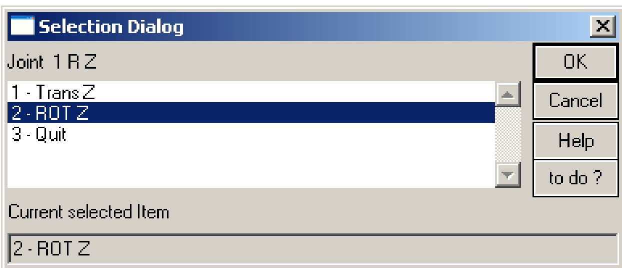


Schritt 2:
Auswahl der Aktive Achsen (1 - *Active Joints*).

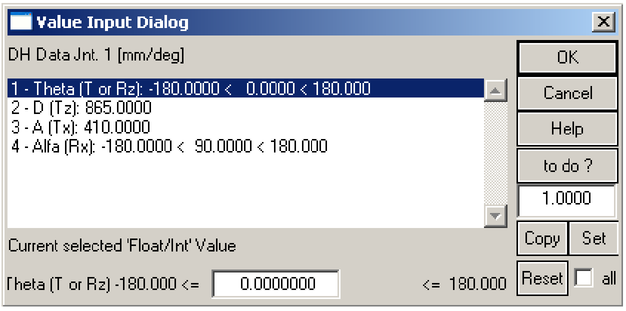
Schritt 3:
Aktive Achse 1 (2 - *activ Jnt 1*) wählen.



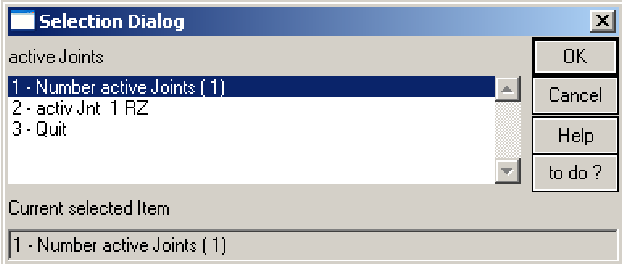
Schritt 4:
Rotation um Z



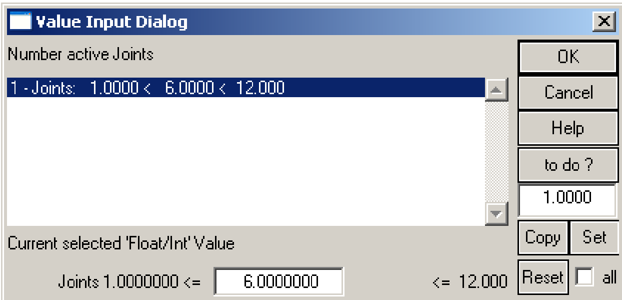
Schritt 5:
Parameter für die Achse 1 eintragen.



Schritt 6:
1 - Number active Joints wählen.

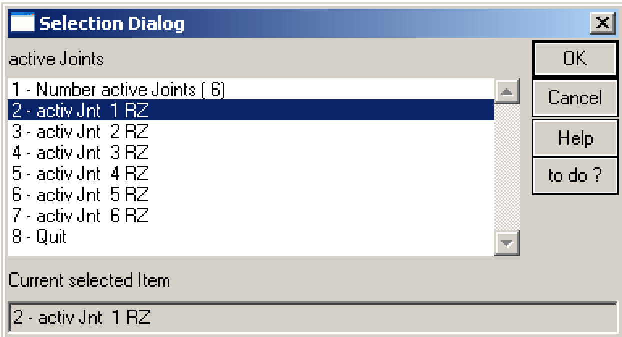


Schritt 7:
Der Roboter besitzt 6 Aktive Achsen

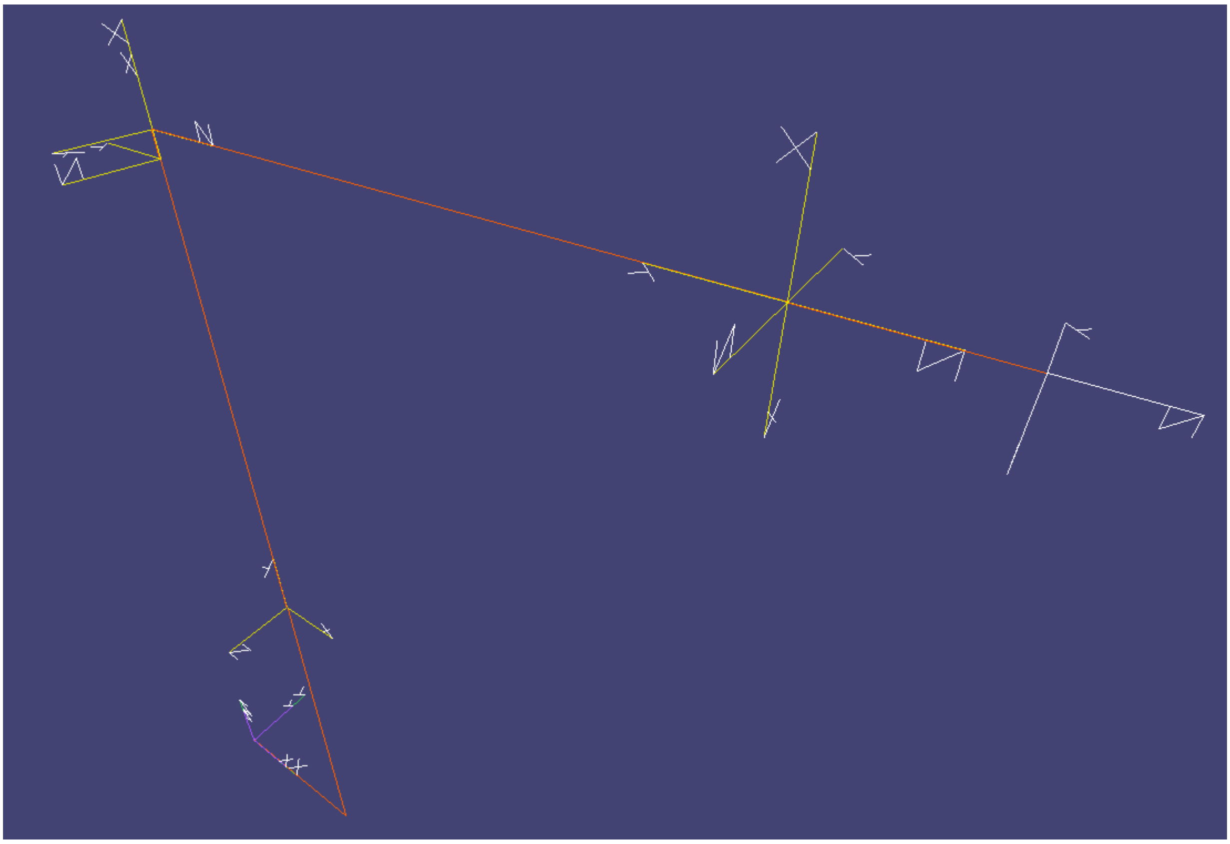


Nach dem die Achsanzahl angegeben wurde kann man nach und nach alle Achsen bestimmten wie die Achse 1 (Schritt 4 und Schritt 5).

Zum Schluss werden alle Achsen dargestellt und durch Auswahl können sie verändert werden.



Nach der Eingabe aller Parameter erscheinen die Koordinatensysteme mit ihren Längen in EASY-ROB.



Mit dem Menü *Robot Joints* können Sie zur Kontrolle alle Aschen einzeln bewegen.

Achse	Bewegung
1	linke Maustaste
2	mittlere Maustaste
3	rechte Maustaste
4	Strg + linke Maustaste
5	Strg + mittlere Maustaste
6	Strg + rechte Maustaste

2.1.2.3. Universal Koordinaten

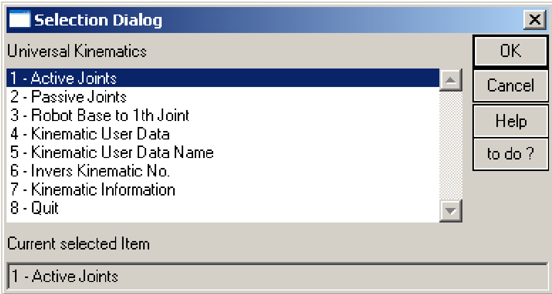
Bei der Universellen Koordinatentransformation kann um jede Achse gedreht und an jeder Achse verschoben werden.

Achse	Rotation	Länge in x in mm	Länge in z in mm	Länge in y in mm
1	$R_Z=0$	410	865	0
2	$R_Y=0$	0	1000	0
3	$R_Y=-90°$	-45	1000	0
4	$R_Z=0$	0	0	0
5	$R_Y=0$	0	0	0
6	$R_Z=0$	0	210	0

Erstellung mit EASY-ROB.

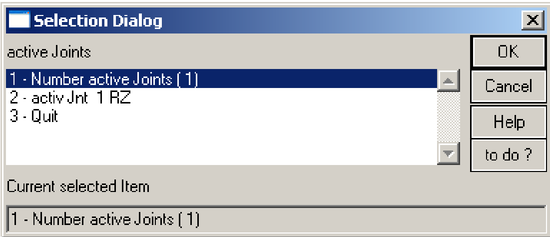
Schritt 1:
Auswahl der Transformation:

Robotics/ Robot Kinematics/ Create new Robot/ Universal Coordinates

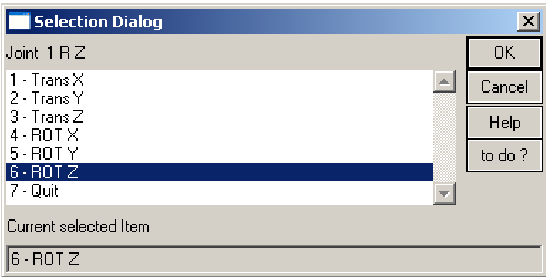


Schritt 2:
Auswahl der Aktive Achsen (1 - *Active Joints*).

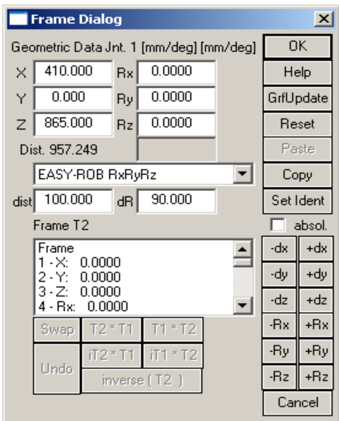
Schritt 3:
Aktive Achse 1 (2 - *activ Jnt 1*) wählen.



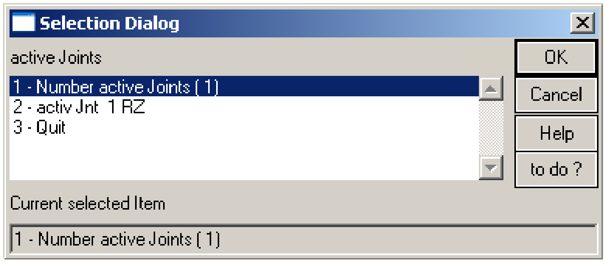
Schritt 4:
Rotation um Z



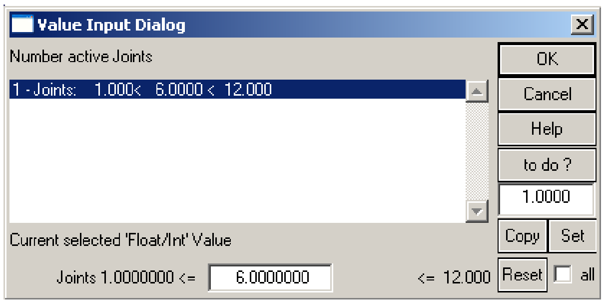
Schritt 5:
Parameter für Achse 1 eintragen.



Schritt 6:
1 - Number active Joints wählen.

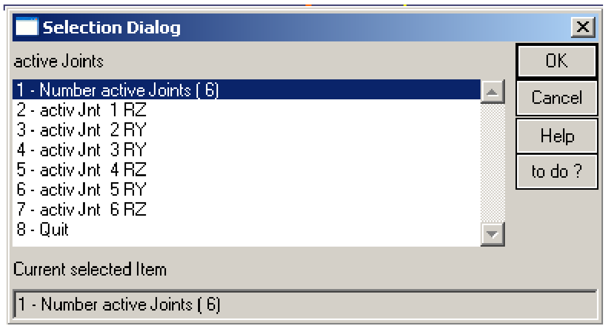


Schritt 7:
Der Roboter besitzt 6 Aktive Achsen



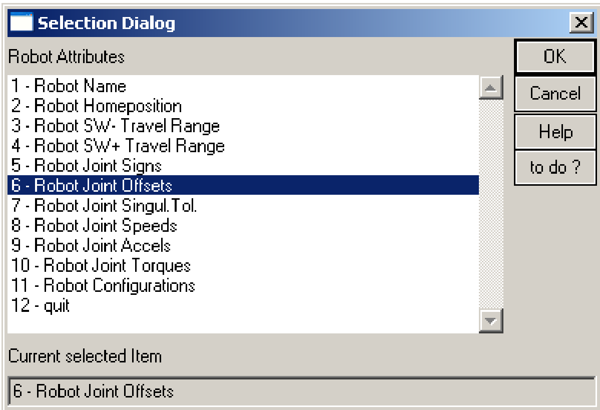
Nach dem die Achsanzahl angegeben wurde kann man nach und nach alle Achsen bestimmten wie die Achse 1 (Schritt 4 und Schritt 5).

Zum Schluss werden alle Achsen dargestellt und durch Auswahl können sie auch nachträglich verändert werden.



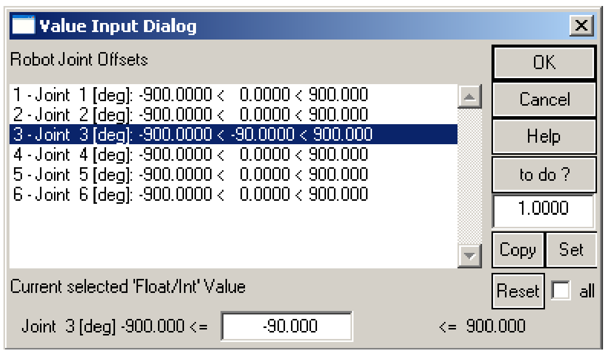
Schritt 8:
Die Achse 3 mit -90° definieren:

Robotics/ Robot Kinematics/ Robot Attributes/ 6-Robot Joint Offset

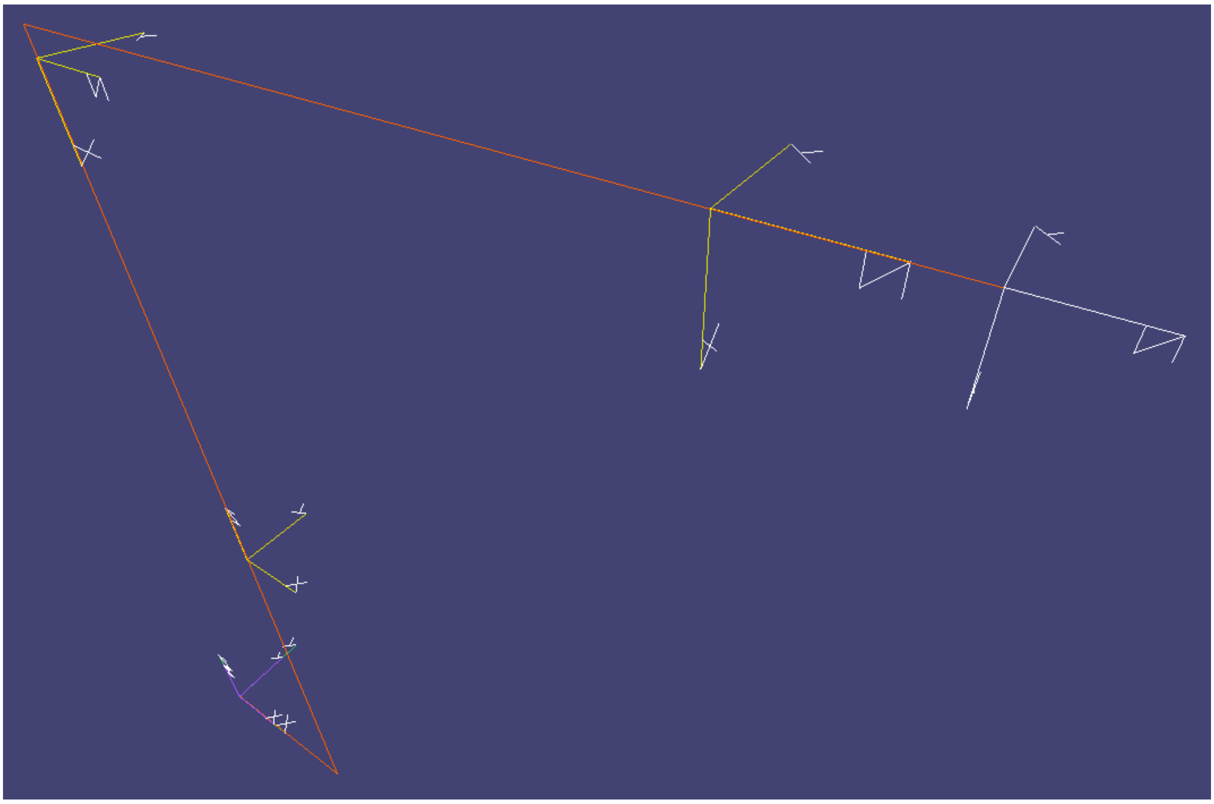


Schritt 9:

Hier wird die Achse 3 (*3 - Joint 3*) definiert.



Nach der Eingabe aller Parameter erscheinen die Koordinatensysteme mit ihren Längen in EASY-ROB.



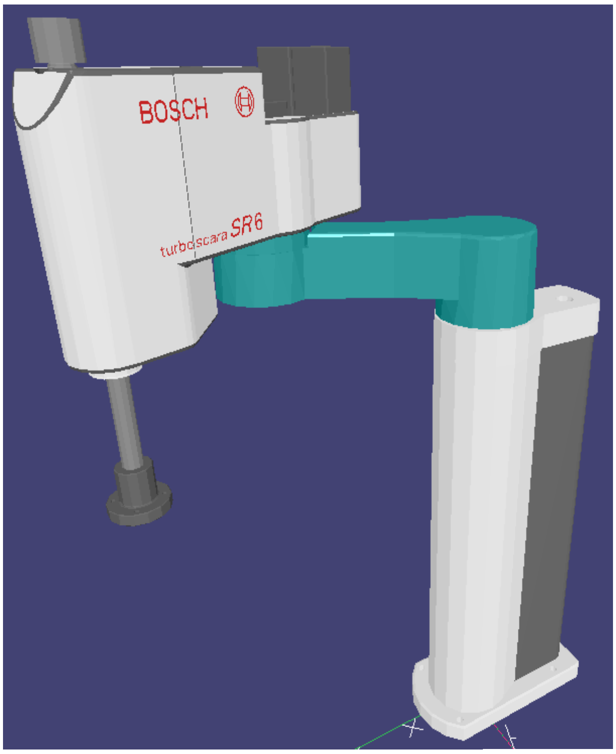
Mit dem Menü *Robot Joints* können Sie zur Kontrolle alle Aschen einzeln bewegen.

Achse	Bewegung
1	linke Maustaste
2	mittlere Maustaste
3	rechte Maustaste
4	Strg + linke Maustaste
5	Strg + mittlere Maustaste
6	Strg + rechte Maustaste

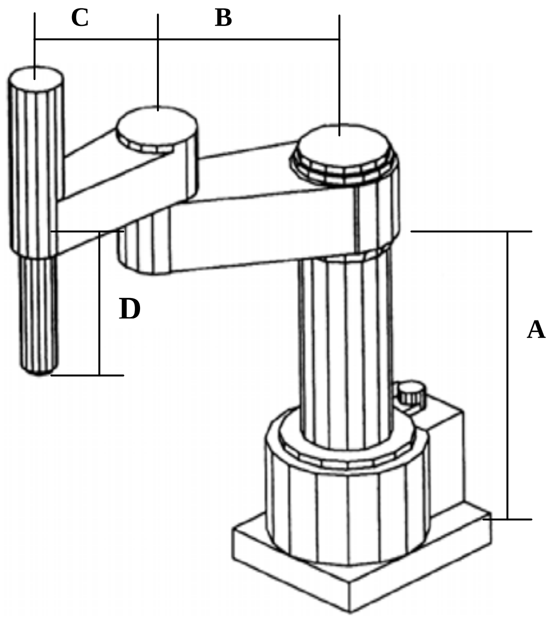
2.1.3. turbo Scara 4 Achs Roboter:

Im folgenden wird der turbo SCARA Roboter Schritt für Schritt in EASY-ROB erstellt.

In den Transformationen:
- Standard RRR:RRR
- Denavit-Hartenberg
- Universal Koordinaten



SCARA SR6



SCARA1

Die Parameter in der Tabelle wurden aus dem Datenblatt der Roboters entnommen.

Achse	Maß in mm
A	700
B	330
C	270
D	70

2.1.3.1. Standard RRR:RRR

Die Transformation Standard RRR:RRR ist für einen Standard Roboter mit 6 Achsen geeignet.
Der Bosch turbo SCARA hat 4 Achsen RRTR, deshalb kann die Transformation Standard RRR:RRR nicht angewandt werden.

2.1.3.2. Denavit-Hartenberg

Die Tabelle wurde mittels Denavit-Hartenberg erstellt.

Hinweise dazu siehe: 1.5. Denavit-Hartenberg
3.1.2. Denavit-Hartenberg - SCARA Roboter.

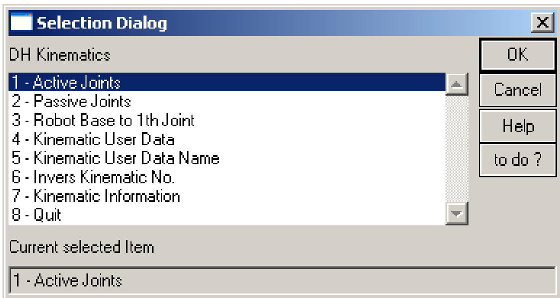
Achse	Θ in °	d in mm	a in mm	α in °
1	R_Z=0°	0	330	0
2	R_Z=0°	0	270	0
3	T_Z=0°	0	0	0
4	R_Z=0°	-70	0	

Bei turbo SCARA ist die Stativ Länge A optional. Darum wird sie bei Denavit-Hartenberg in EASY-ROB nicht beachtet. Sie wird gesondert angegeben.

Erstellung mit EASY-ROB.

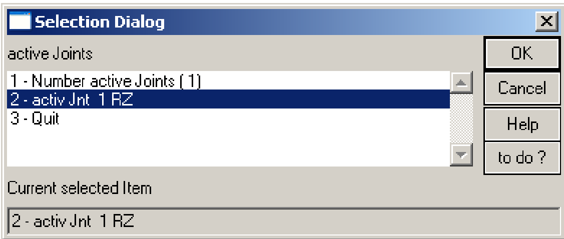
Schritt 1:
Auswahl der Transformation:

Robotics/ Robot Kinematics/ Create new Robot/ Denavit-Hartenberg Notation

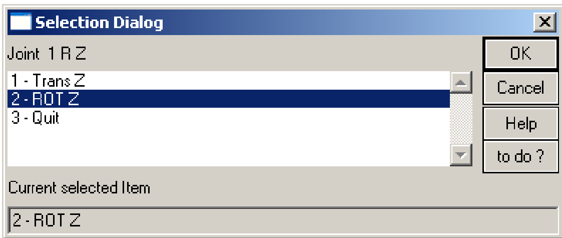


Schritt 2:
Auswahl der Aktive Achsen (1 - *Active Joints*).

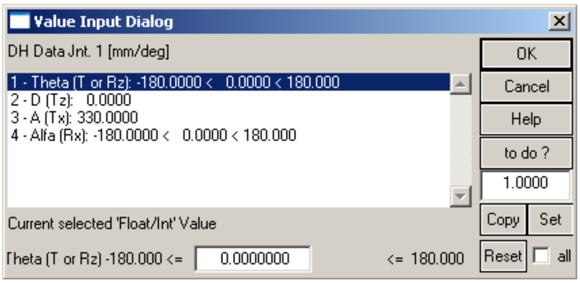
Schritt 3:
Aktive Achse 1 (2 - *activ Jnt 1*) wählen.



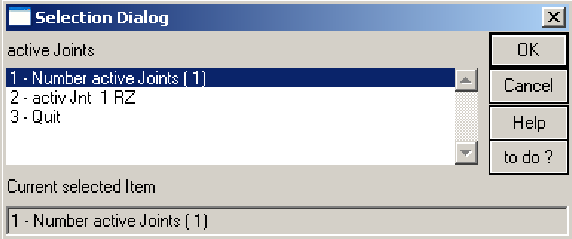
Schritt 4:
Rotation um Z



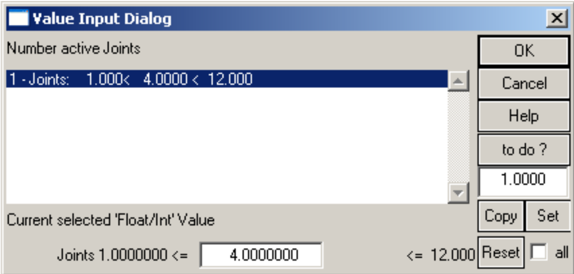
Schritt 5:
Parameter für die Achse 1 eintragen.



Schritt 6:
1 - Number active Joints wählen.

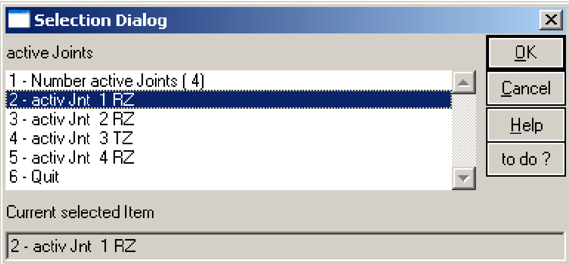


Schritt 7:
Der Roboter besitzt 4 Aktive Achsen



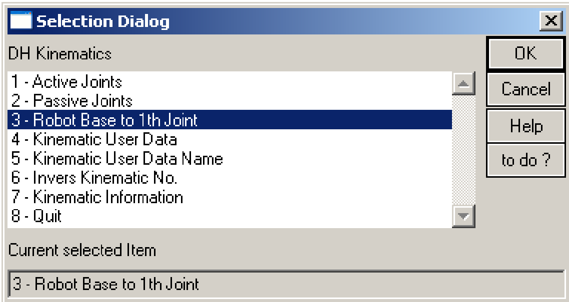
Nach dem die Achsanzahl angegeben wurde, kann man nach und nach alle Achsen bestimmten wie die Achse 1 (Schritt 4 und Schritt 5).

Zum Schluss werden alle Achsen dargestellt und durch Auswahl können sie verändert werden.

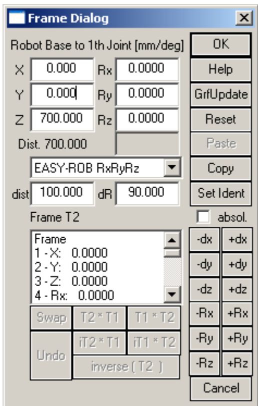


Schritt 8:
Das Stativ mit der Länge A einstellen.

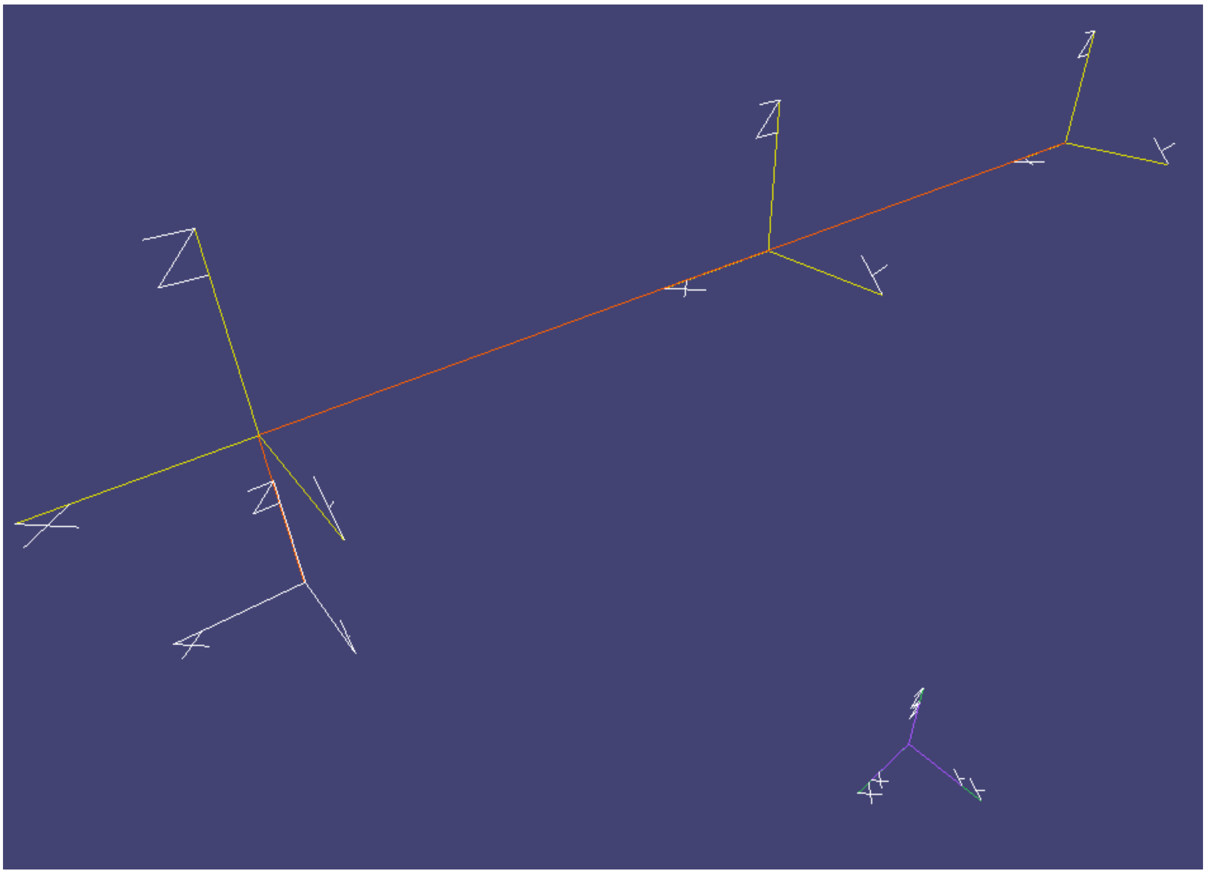
3 – Robot Base to 1th Joint



Schritt 9:
Länge A ist der Abstand von Base an der z-Achse zum ersten Koordinatensystem.



Nach der Eingabe aller Parameter erscheinen die Koordinatensysteme mit ihren Längen in EASY-ROB.



Mit dem Menü *Robot Joints* können Sie zur Kontrolle alle Aschen einzeln bewegen.

Achse	Bewegung
1	linke Maustaste
2	mittlere Maustaste
3	rechte Maustaste
4	Strg + linke Maustaste
5	Strg + mittlere Maustaste
6	Strg + rechte Maustaste

2.1.2.3. Universal Koordinaten

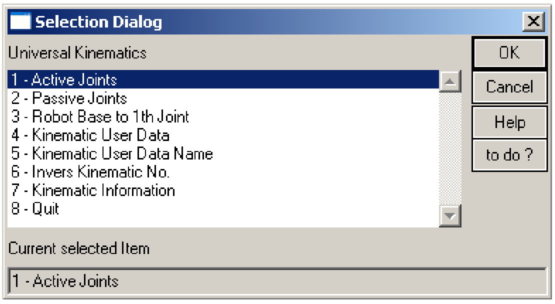
Bei der Universellen Koordinatentransformation kann um jede Achse gedreht und an jeder Achse verschoben werden.

Achse	Rotation	Länge in x in mm	Länge in z in mm	Länge in y in mm
1	R_Z=0	0	330	0
2	R_Z=0	0	270	0
3	T_Z=0	0	0	0
4	R_Z=0	0	-70	0

Bei turbo SCARA ist die Stativ Länge A optional. Darum wird sie bei Universal Koordinaten in EASY-ROB nicht beachtet. Sie wird gesondert angegeben.

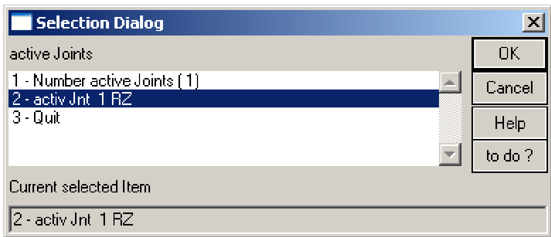
Erstellung mit EASY-ROB.

Schritt 1:
Auswahl der Transformation:
Robotics/ Robot Kinematics/ Create new Robot/ Universal Coordinates

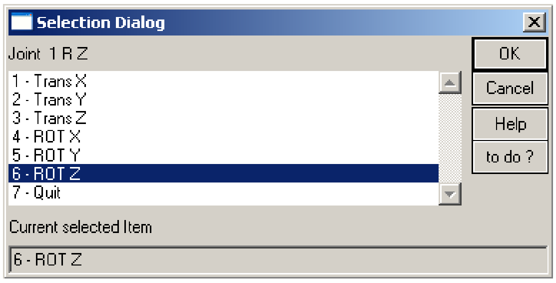


Schritt 2:
Auswahl der Aktive Achsen (1 - *Active Joints*).

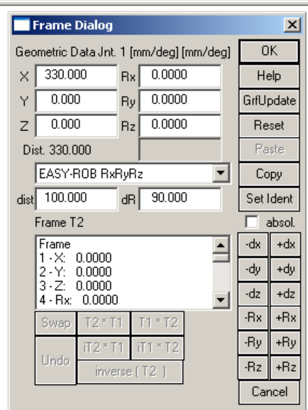
Schritt 3:
Aktive Achse 1 (2 - *activ Jnt 1*) wählen.



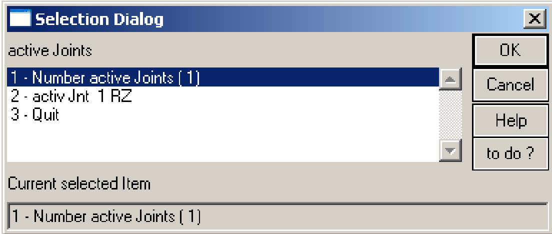
Schritt 4:
Rotation um Z



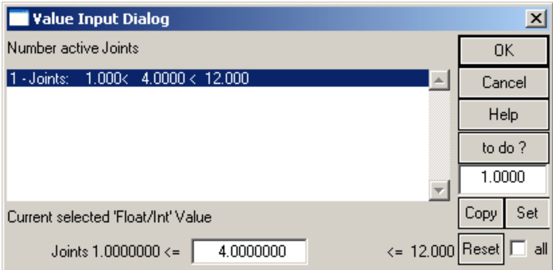
Schritt 5:
Parameter für Achse 1 eintragen.



Schritt 6:
1 - Number active Joints wählen.

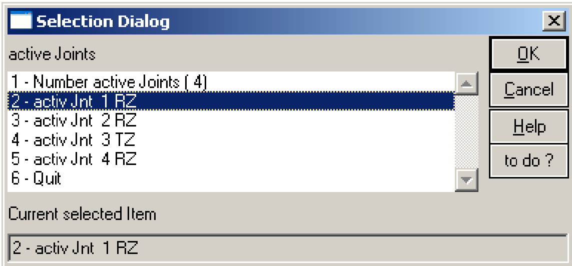


Schritt 7:
Der Roboter besitzt 4 Aktive Achsen



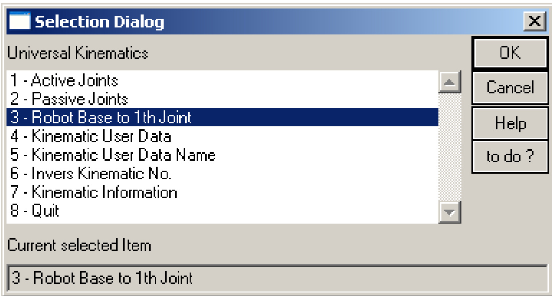
Nach dem die Achsanzahl angegeben wurde, kann man nach und nach alle Achsen bestimmten wie die Achse 1 (Schritt 4 und Schritt 5).

Zum Schluss werden alle Achsen dargestellt und durch Auswahl können sie verändert werden.

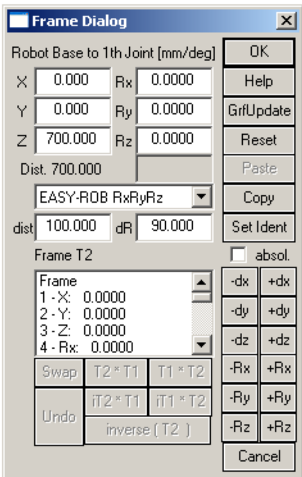


Schritt 8:
Das Stativ mit der Länge A einstellen.

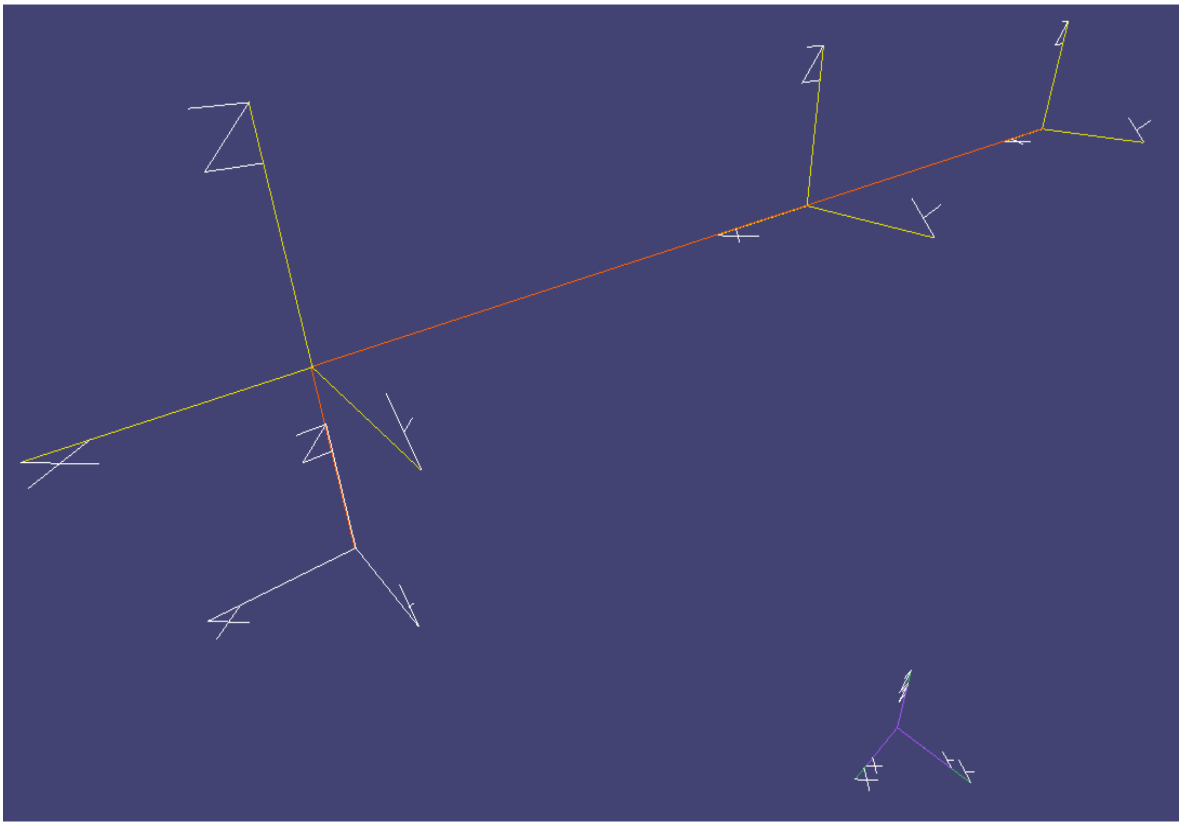
3 – Robot Base to 1th Joint



Schritt 9:
Länge A ist der Abstand von Base an der z-Achse zum ersten Koordinatensystem.



Nach der Eingabe aller Parameter erscheinen die Koordinatensysteme mit ihren Längen in EASY-ROB.



Mit dem Menü *Robot Joints* können Sie zur Kontrolle alle Aschen einzeln bewegen.

Achse	Bewegung
1	linke Maustaste
2	mittlere Maustaste
3	rechte Maustaste
4	Strg + linke Maustaste
5	Strg + mittlere Maustaste
6	Strg + rechte Maustaste

3. Anhang

3.1. Beispiele:

3.1.1 Allgemeine Erläuterungen zum erstellen von Denavit-Hartenberg

• Es gibt ein Stützpunktkoordinatensystem (Base) in Verlängerung zur ersten Achse.
• Koordinatensystem **1 (BASE)** wird in das erste dreh Gelenk verschoben bzw. verlängert.
• Die Symbole Θ und d bezeihen sich auf die Bewegung der z-Achse.
• Die Symbole a und α beziehen sich auf die Bewegung der x-Achse.
• Die z-Achse schaut aus dem dreh Gelenk heraus.
• Hilfsmittel ist die rechte Hand (Daumen = x, Zeigefinger = y, Mittelfinger = z)
• Jeder Schritt bzw. jede Achse wird in ein Tabelle eingetragen.

Achse	Θ_n	d_n	a_n	α_n

• Es wird nach folgenden vier schritten Verfahren:

1. Rotation um Θ um die z-Achse des Koordinatensystems **1** bis die x-Achse mit der x-Achse des Koordinatensystems **2** parallel liegen
2. Verschiebung um d der z_1-Achse bis sich die x_1-Achse und x_2-Achse decken.
3. Verschiebung um a, entlang der x Achse bis die Koordinatenursprünge **1** und **2** sich decken.
4. Rotation um α um die x Achse bis die Koordinatensysteme sich decken.

• Diese vier Punkte werden für jede einzelne Achse abgearbeitet.

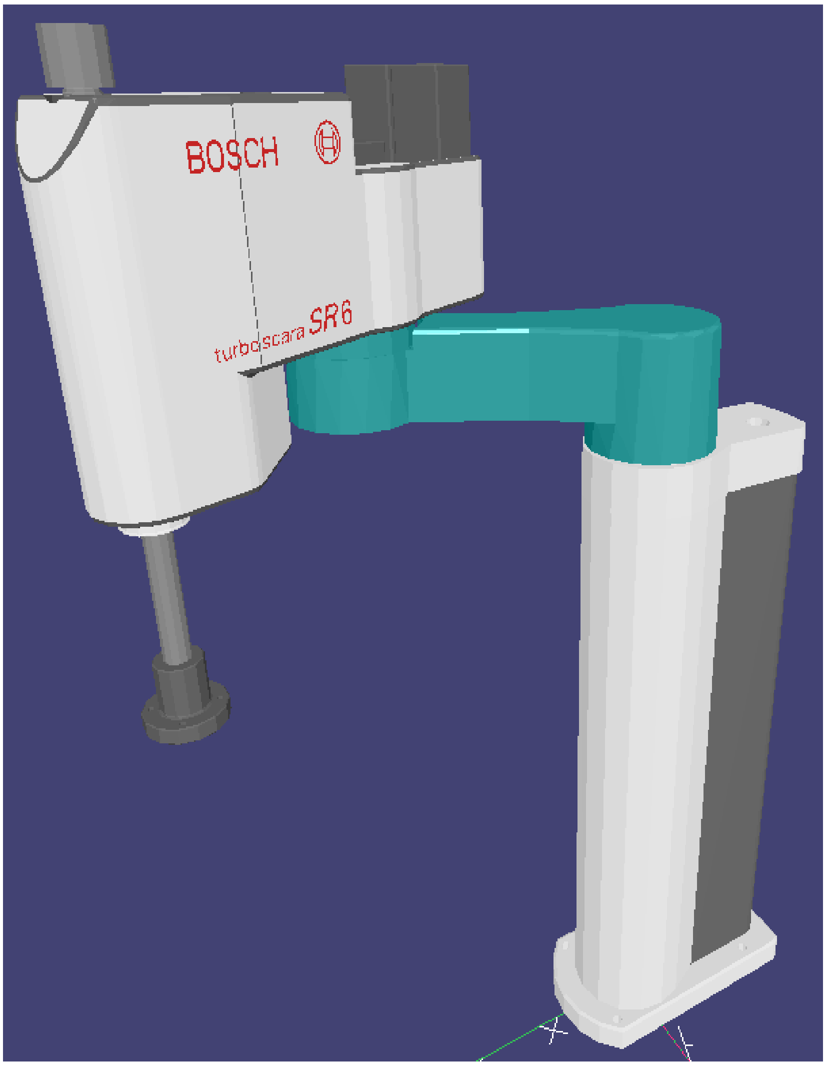
• Die Schritte 1 und 2 oder 3 und 4 können ggf. vertauscht werden.
Aber die Reihefolge, erst z dann x, muss beibehalten werden.

Mathematischer beweiß:

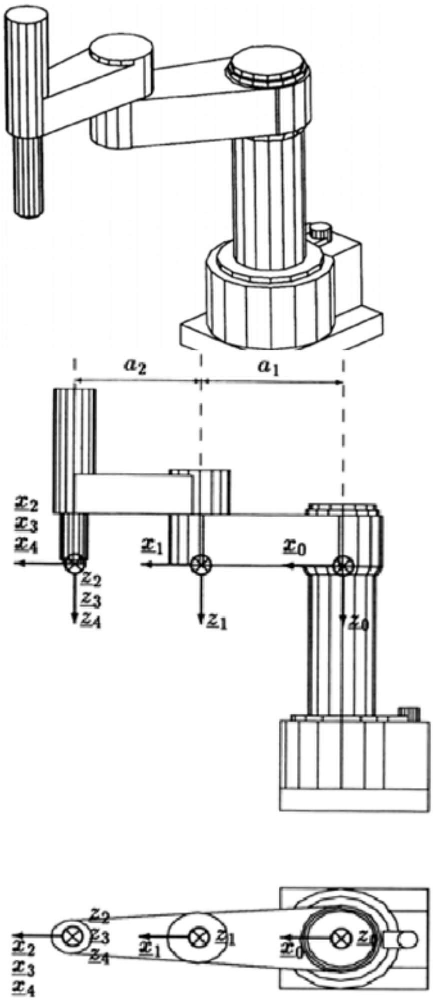
$$R_z = \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 & 0 \\ \sin\Theta & \cos\Theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad T_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_z * T_z = \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 & 0 \\ \sin\Theta & \cos\Theta & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} = T_z * R_z = \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 & 0 \\ \sin\Theta & \cos\Theta & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3.1.2. Denavit-Hartenberg Transformation am Bosch turbo SCARA (4 Achs)



SCARA SR6



SCARA 2

Die Parameter des turbo SCARA Roboters können in eine Tabelle eingetragen werden.

1. Rotation um Θ_n um die Achse zn-1 bis xn-1 parallel zu x_n liegt
2. Verschiebung um d_n in Richtung z_{n-1} bis sich x_{n-1} und x_n decken
3. Verschiebung um a_n in Richtung x_n bis Koordinatenursprünge gleich sind
4. Rotation um α_n um die Achse x_n bis die Koordinatensysteme identisch sind

Achse	Θ	d	a	α
1	$R_Z = \Theta_1$	d_1	a_1	0
2	$R_Z = \Theta_2$	0	a_2	0
3	$T_Z = \Theta_3$	0	0	0
4	$R_Z = \Theta_4$	$-d_3$	0	0

Die Parameter d und a können aus dem Datenblatt entnommen werden und Θ ist abhängig von der Home Position des Roboters.

Nach dem die Tabelle erstellt worden ist, können jetzt die Parameter für jede Achse in die Matrix eingesetzt und berechnet werden.

$$T_{0,1} = [Rot(z1,\Theta)] * [Trans(0,0,d_1)] * [Trans(a_1,0,0)] * [Rot(x_1,\alpha_1)]$$

$$T_1 = \begin{bmatrix} \cos\Theta & -\cos\Theta\sin\alpha & \sin\Theta\sin\alpha & a\cos\Theta \\ \sin\Theta & \cos\Theta\cos\alpha & -\cos\Theta\sin\alpha & a\sin\Theta \\ 0 & \sin\alpha & \cos\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Für die anderen vier Achsen gilt die selbe Matrix.

Somit ergibt sich die Gesamttransformation:

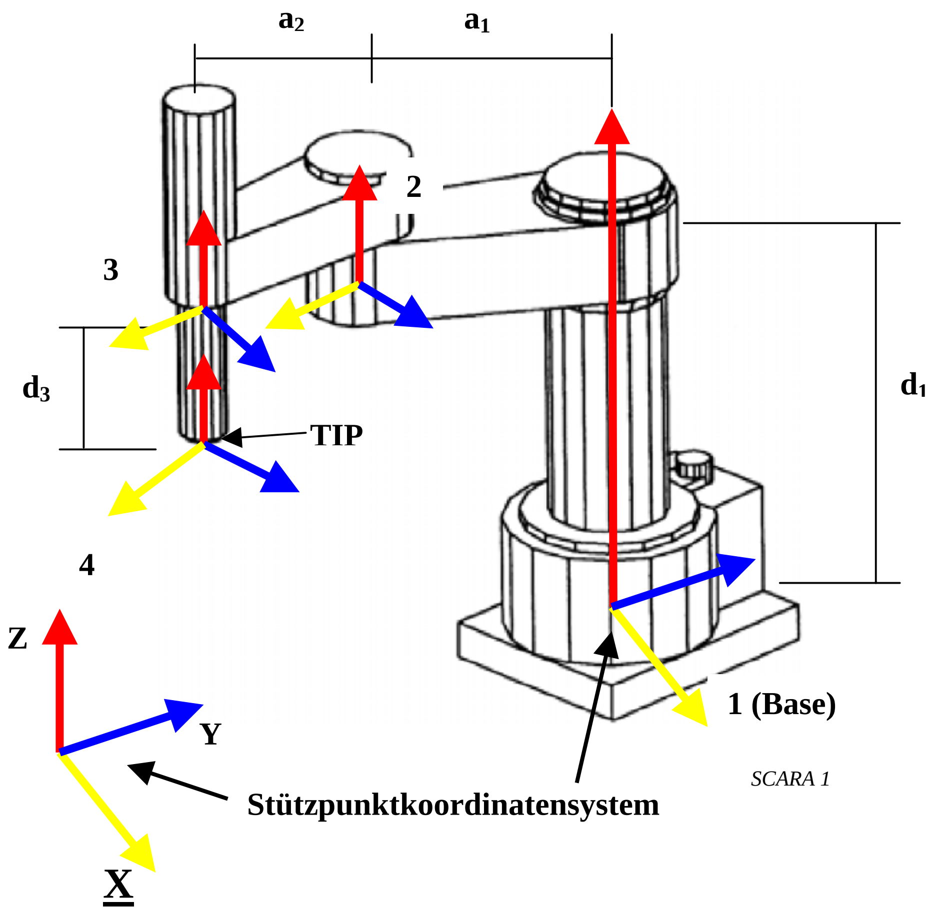
$$^{m}_{n}T_{gesamt} = \left(^{n+1}_{n}T\right) * \left(^{n+2}_{n+1}T\right) \ldots \left(^{m}_{m-1}T\right)$$

Für den SCARA:

$$^{4}_{0}T_{gesamt} = {}^{1}_{0}T * {}^{2}_{1}T * {}^{3}_{2}T * {}^{4}_{3}T$$

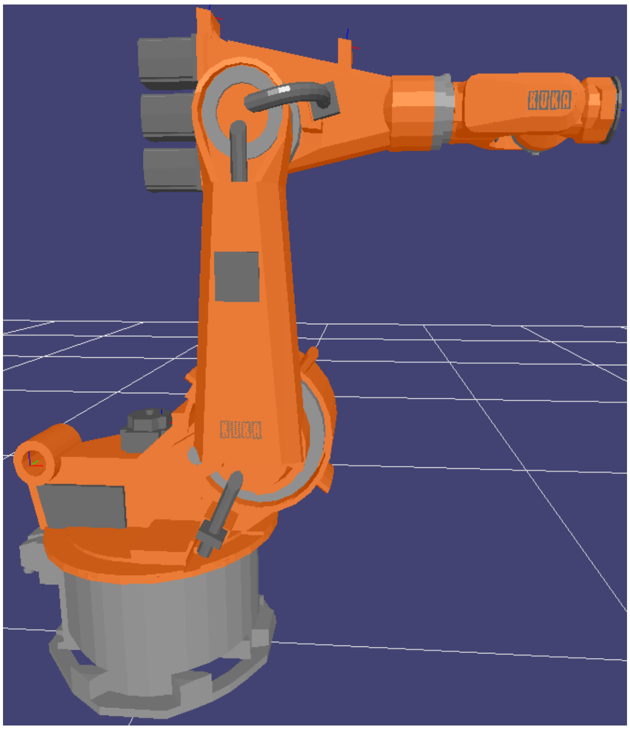
Aus der Denavit-Hartenberg Transformation (Vorwärtstransformation) T_{gesamt} lässt sicht jetzt die inverse Transformation (Rückwärtstransformation) T^{-1}_{gesamt} erstellen.

Bosch turbo SCARA SR6

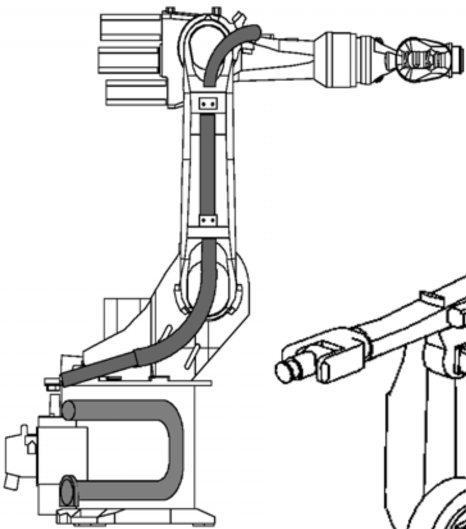


SCARA 1

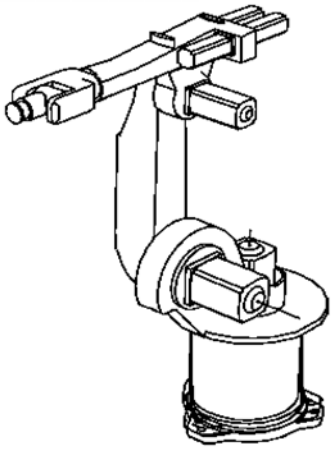
3.1.3. Denavit-Hartenberg Transformation am KUKA Roboter (6 Achs)



KUKA KR 125-2



KUKA 1



KUKA 2

Die Parameter des KUKA Roboters können in eine Tabelle eingetragen werden.

1. Rotation um Θ_n um die Achse z_{n-1} bis x_{n-1} parallel zu x_n liegt
2. Verschiebung um d_n in Richtung z_{n-1} bis sich x_{n-1} und x_n decken
3. Verschiebung um a_n in Richtung x_n bis Koordinatenursprünge gleich sind
4. Rotation um α_n um die Achse x_n bis die Koordinatensysteme identisch sind

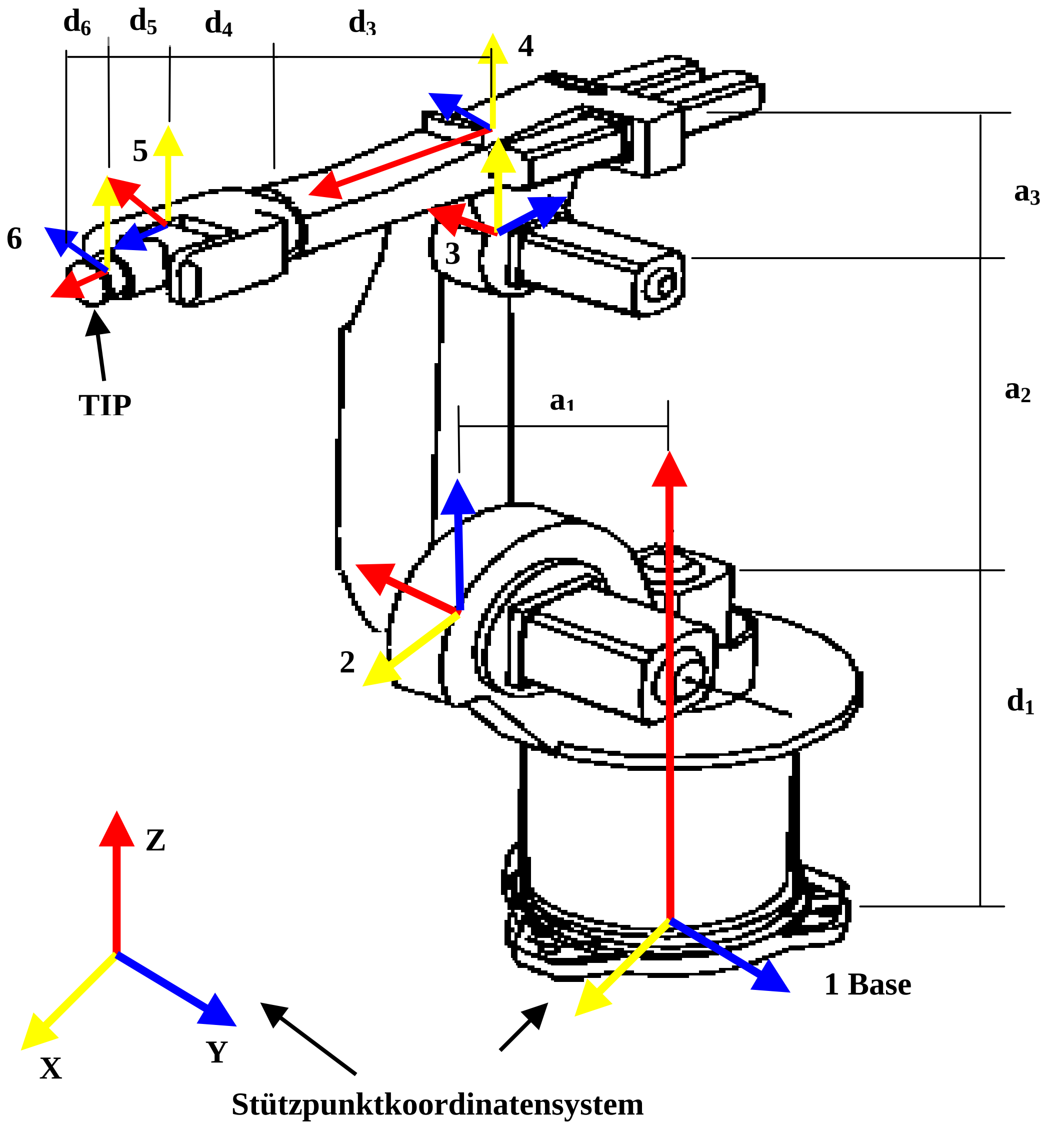
Hinweise zur Erstellung der Achse 4,5 und 6 siehe dazu 1.7. Die Zentralhand

Achse	$R_Z = \Theta$	d	a	α
1	Θ_1	d_1	a_1	90°
2	90°	0	a_2	0
3	Θ_3	0	a_3	90°
4	Θ_4	d_3+d_4	0	-90°
5	Θ_5	0	0	90°
6	Θ_6	d_5+d_6	0	0

Die Parameter d und a können aus dem Datenblatt entnommen werden und Θ ist abhängig von der Home-Position des Roboters.

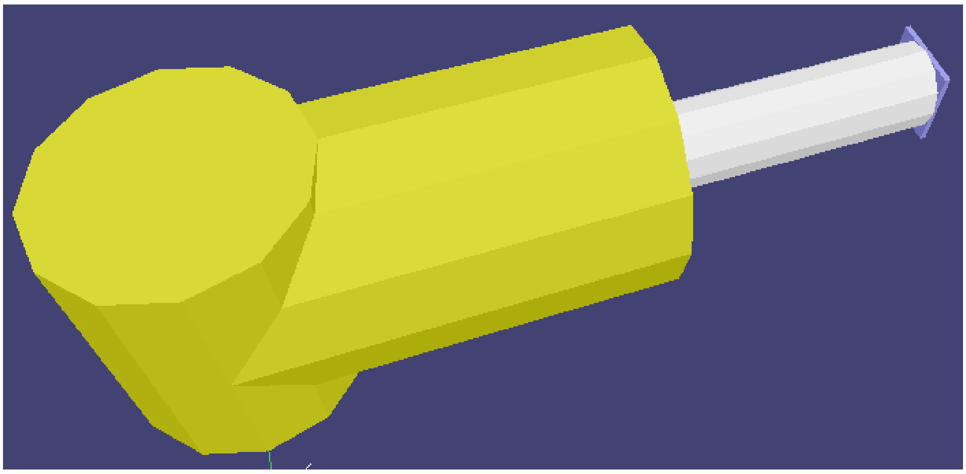
Nach dem die Tabelle erstellt worden ist, können jetzt die Parameter für jede Achse in die Matrix eingesetzt und berechnet werden.

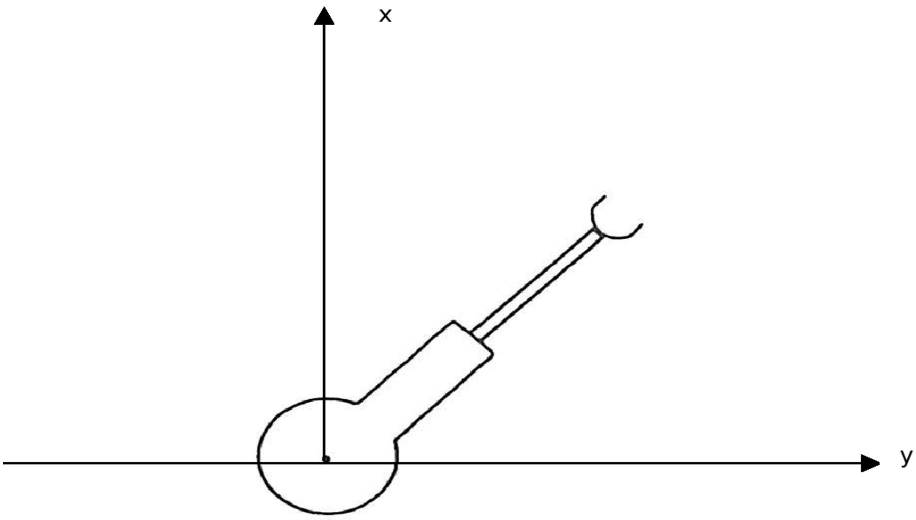
KUKA KR 125-2


d6
d5
d4
d3
4
5
3
6
a3
a2
a1
TIP
2
d1
Z
1 Base
X
Y
Stützpunktkoordinatensystem

3.1.4. Berechnung der Jakobi-Matrix

Für einen Roboter mit 6 Achsen wird es eine recht komplexe Rechnung darum wird es am einfachen RT Roboter (2 Achsen) erläutert und durch geführt.





Die Geschwindigkeit in kartesischen Raum,

$$D = \begin{bmatrix} d_x \\ d_y \end{bmatrix} \quad \text{mit} \quad {}^{d}\!/_{dt} = \dot{D} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}$$

und die Geschwindigkeit der Gelenke:

$$D_\Theta = \begin{bmatrix} d_r \\ d_\Theta \end{bmatrix} \quad \text{mit} \quad {}^{d}\!/_{dt} = \dot{D}_\Theta = \begin{bmatrix} \dot{r} \\ \dot{\Theta} \end{bmatrix}$$

Als Jakobi-Matrix einsetzen:

$$D = J * D_\Theta \rightarrow \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} * \begin{bmatrix} \dot{r} \\ \dot{\Theta} \end{bmatrix}$$

Es gilt:

$x = r \cdot \cos\Theta$ und $y = r \cdot \sin\Theta$ x und y Zeitlich ableiten ergibt:

$$\dot{x} = \dot{r} \cdot \cos\Theta - \dot{\Theta} \cdot r \cdot \sin\Theta \quad \text{und} \quad \dot{y} = \dot{r} \cdot \sin\Theta + \dot{\Theta} \cdot r \cdot \cos\Theta$$

$$\rightarrow \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos\Theta & -r \cdot \sin\Theta \\ \sin\Theta & r \cdot \sin\Theta \end{bmatrix} \cdot \begin{bmatrix} \dot{r} \\ \dot{\Theta} \end{bmatrix} \rightarrow J(r,\Theta) = \begin{bmatrix} \cos\Theta & -r \cdot \sin\Theta \\ \sin\Theta & r \cdot \cos\Theta \end{bmatrix}$$

Aus der Jakobi-Matrix (Vorwärtstransformation) D lässt sicht jetzt die inverse Transformation (Rückwärtstransformation) J^{-1} erstellen.

$$D_{\Theta} = J^{-1} * D$$

Quellen:

Internet:

www.glossar.de
www.it.fht-esslingen.de/~schmidt/vorlesungen/vr/seminar/ws9899/kinmod-invkinem.html
www.worldwidewolf.de/diplom.pdf
http://www.kuka-roboter.de/deutsch/index.html
http://www.boschrexroth.com/BoschRexroth/business_units/brl/de/produkte/roboter/turboscara_sr_plus/index.jsp

Buch:

Stefan Hesse	Industrieroboter
Wei Li	Grafische Simulation
Wesley E.Snyder	Computergesteuerte Industrieroboter

Vorlesungsscript Herr Prof. Müller