ROBOTIK-Einführung

5AHTA/2003
5CHTA/2003
Dr. Hinterreiter
Version 1.0

Höhere Technische Bundeslehranstalt - Linzer Technikum

A-4020 Linz, Paul-Hahn-Straße 4, Tel.: +43 70 / 77 03 01, Fax: +43 70 / 78 14 92, e-mail:office.litec@eduhi.at

Vorwort

Zweite Karriere für Kuki

Ganz im Sinne der im Beitrag des neuen Direktors geforderten Zusammenarbeit zwischen Wirtschaft und Schule ist es dem LiTec als einziger HTL in Oberösterreich gelungen, einen ausgewachsenen Industrieroboter für den Laborunterricht aller Abteilungen anzubieten.

Ob bei Mercedes, VW, oder Opel, in allen Fertigungsstraßen sind Schweißroboter der in Augsburg ansässigen Firma KUKA im Einsatz. Vor allem die größeren Exemplare sind wahre Kraftlackel, die in der Lage sind, bis zu 2,5 kN Kopflast zu heben und „punktgenau" zu manövrieren. Das heißt, dass jeder Raumpunkt innerhalb einer Halbkugel mit 5,2 m Durchmesser mit einer Toleranz von ± 0,2 mm(!) reproduzierbar angesteuert werden kann. Somit eignen sich diese Roboter ideal zur Handhabung von Punktschweißgeräten, die am Kopf des Roboters montiert Tausende Autokarosserien zusammenschweißen.

Hat der Roboter dies zwei bis drei Jahre lang brav gemacht, geht es ihm wie Schirennläufern: Die Gelenke werden etwas ausgeleiert, und er schafft die Ausscheidungsrennen nicht mehr so wie früher. Nun ist es höchste Zeit, für ihn ein „Ausgedinge" zu suchen. So gelang es unserer Schule einen nach einer technischen Überholung hervorragend in Schuss gehaltenen Roboter des Typs KR 125 zu ergattern. Hiezu war es allerdings notwendig, eine Rettungscrew zu motivieren unseren Frührentner wieder zu aktivieren.

Zunächst wurde nach Einwilligung des Versicherungsunternehmens ein ursprünglich beschädigter Roboter zum Schrottwert dem LiTec überlassen. Danach wurde durch die Fa. KUKA ein nicht beschädigter Eintauschroboter ausgewählt, in KUKA-Orange neu gespritzt und schließlich kostenlos von Augsburg nach Linz transportiert

Auch die nächste Fürbitte nach dem Erwerb einer möglichst günstigen Steuerung wurde von KUKA um DM 9 000,- äußerst caritativ erhört, indem man einen gebrauchten Steuerungsschrank aus einem Autowerk in Südkorea per Schiff herankarrte.
Der digital anzusteuernde Greiferkopf wurde dann von der Fa. SCHUNK gesponsert, und schließlich stellte sich die VA–TECH TMS mit dem letzten, aber sehr wichtigen Geschenk in Form des Aluschutzgitters ein.

Mit vereinten LiTec-Kräften – hier seien die Werkstättenfachlehrer Willnauer, Wolfsegger und Hackl besonders erwähnt – gelang es schließlich, die Inbetriebnahme planmäßig abzuschließen. Die Einschulung der Lehrer, die zukünftig Laborübungen veranstalten werden, ist zur Zeit im Gange.

Dank dieser vorbildlichen Zusammenarbeit zwischen Wirtschaft und Schule dürfen unsere Schüler ab sofort etwas „Autoindustrieluft" schnuppern und Erfahrungen mit den neuesten Technologien sammeln.

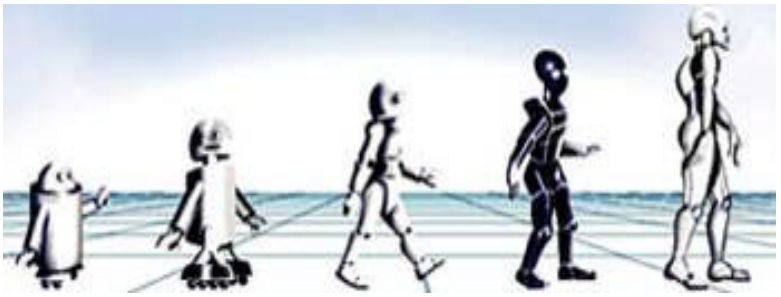
Prof. Albert Cuchiero, Frühjahr 2001

Inhaltsverzeichnis

1. Einführung

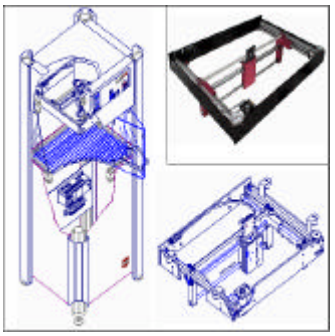
1.1 Entwicklung

Die historische Entwicklung der Robotertechnik begann im 18. Jahrhundert. Erst in den 50er Jahren des 20. Jhdt. Wurden erste roboterähnliche Geräte entwickelt. Bei den ersten Robotern benutzte man hydraulische Antriebe. Heute verwendet man standardgemäß einen drehzahlgeregelten Elektromotor. Die Steuerung der moderneren Robotern war anfangs, ähnlich wie bei den Werkzeugmaschinen, eine Numerische. Erst als der Computer im Bereich der Steuerung von Werkzeugmaschinen eingesetzt wurde, fand auch diese Steuerung in der Robotertechnologie ihre Anwendung.



Die heute im Einsatz befindlichen Roboter sind vermehrt als starre Roboter ausgeführt. Jede einzelne Achse, Arm wird als starr betrachtet. Die Entwicklungen gehen nun in die Richtung des elastischen Roboters, wobei Gesetzmäßigkeiten der Mechanik berücksichtigt werden, als Beispiel die menschliche Hand.

1.2 Verschiedene Bauarten von Robotern:

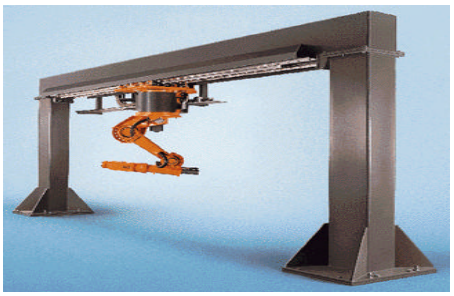


Ein typischer Industrieroboter mit drei Linearachsen ist der **Portalroboter**. Der Miniportalroboter zeichnet sich durch seine kleinen Inbauabmessungen im Verhältnis zu seinem großen Arbeitsbereich aus. Durch den Einsatz von stationären Motoren können die bewegten Massen besonders klein gehalten werden. Hohe Beschleunigung bei gleichzeitig kompaktem Design sind die wesentlichen Vorteile.

Dieser Industrieroboter mit **drei Linearachsen** und **einer Drehachse** hat gegenüber dem Portalroboter den Vorteil, auch schräg liegende Werkstücke auf der Plattform zu erfassen. Ein großer Nachteil ist hier aber der enorme Platzbedarf.

Bei der Automobilfertigung gibt es viele Aufgaben, bei denen es nicht so sehr auf allerhöchste Positionsgenauigkeit ankommt, sondern eher darauf, mit einen Werkzeug entlang beliebiger Bahnen im Raum bei beliebiger Orientierung des Werkzeuges zu verfahren. Typisch hierfür ist das Bahnschweißen, Punktschweißen, Lackieren, Kleberauftragen und ähnliches. Dass der Vertikalknick-armroboter für solche Aufgaben sehr gut geeignet ist, und in der Automobilindustrie ein sehr hoher Automationsgrad herrscht, erklärt seine große Verbreitung.

6-achsiger Vertikal-Knickarmroboter

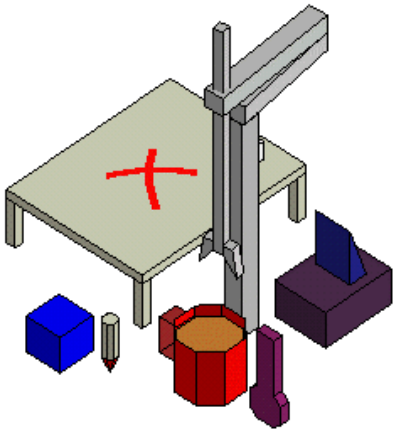


6-achsiger Knickarmroboter mit zusätzlicher Linearachse

Um den Arbeitsbereich bei dem oben angeführten Knickarmroboter zu erweitern, wird eine 7. Linear oder Rotationsachse ergänzt. Dazu wird z.B. der sonst feste Standort auf eine Linearachse ergänzt. Dies ermöglicht auch Arbeiten an Teilen die sich auf Einer Transferstrasse stetig vorwärtsbewegen.

1.3 Einsatzbereich:

Der Einsatzbereich von Robotern erstreckt sich von einfachen transportieren bis zu hoch komplexen Operationen am menschlichen Körper. Roboter werden hauptsächlich, nur um ein paar Beispiele zu nennen, bei Werkzeugmaschinen, Pressen und Schmieden, Schweissen und Lackieren und Druck-/Spritzguss verwendet.



2. Bauarten von KUKA – Roboter:

KR 30 K
In platzsparender Konsolbauweise für das „Arbeiten von oben“.

KR 125 W/2
In platzsparender Wandbau-Weise für das „Arbeiten von der Wand“.

Palletierer
Speziell zugeschnitten für Ver-Packungs- und Palletieraufgaben bieten diese Modelle eine unschlag-bare Flexibilität.

Pressenverketter
Sind konzipiert für die Verkettung von Produktionsstrassen mit Einzelpressen, eignen sich auch für Maschinen-Beschickung sowie zum Abstappeln und Paletieren.

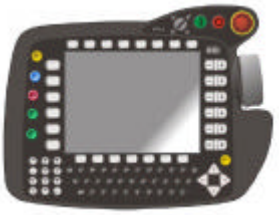
3. KUKA KR 125/3:

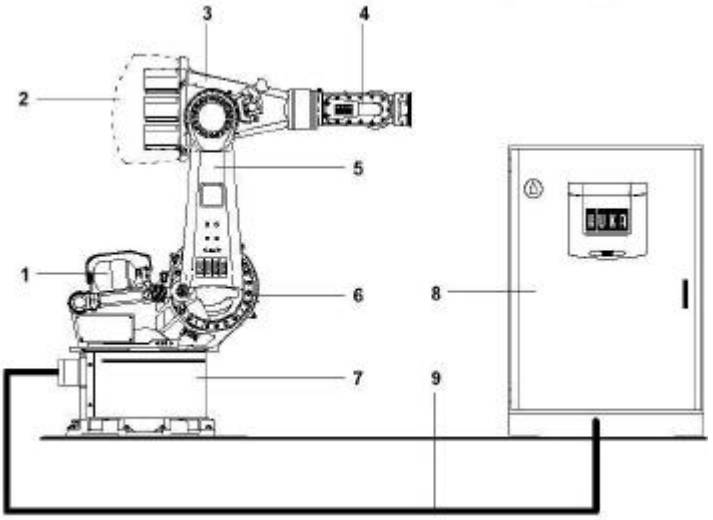
3.1 Bestandteile:

M2/Tr.Bl.1,2,3,4,5,7
Steuerschrank KR C1

Roboter KUKA KR 125

M2/Tr.Bl.6/...
KUKA Control Pannel



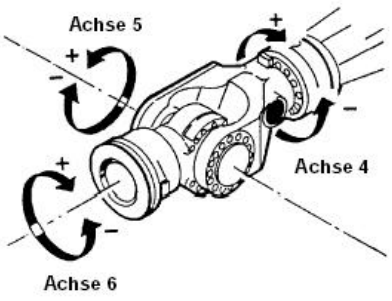


1...Hydropneumatisches Ausgleichssystem
2...Gegengewicht
3...Arm
4...Zentralhand
5...Schwinge
6...Karussell
7...Grundgestell
8...Steuerschrank
9...Verbindungsleitung

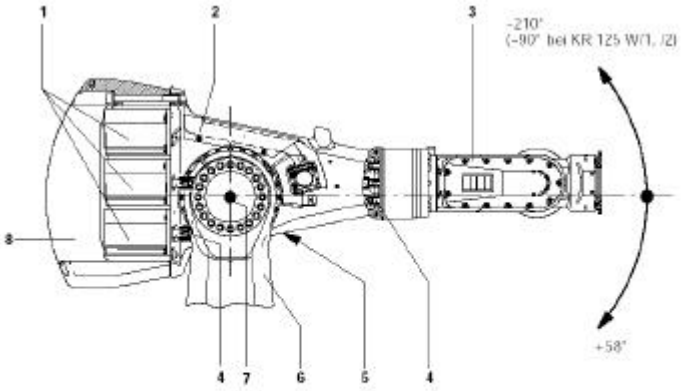
- **Hydropneumatisches Ausgleichssystem:**
  Zum Ausgleich des herrschenden Lastmoments.

- **Gegengewicht:**
  Gewichtsausgleich bei hohen Handlingmassen.

- **Zentralhand:**
  Ist über 3 Achsen beweglich und kann bis zu 125 kg Traglast aufnehmen. Adapter zur Aufnahme von Greifer oder Schweisszange.



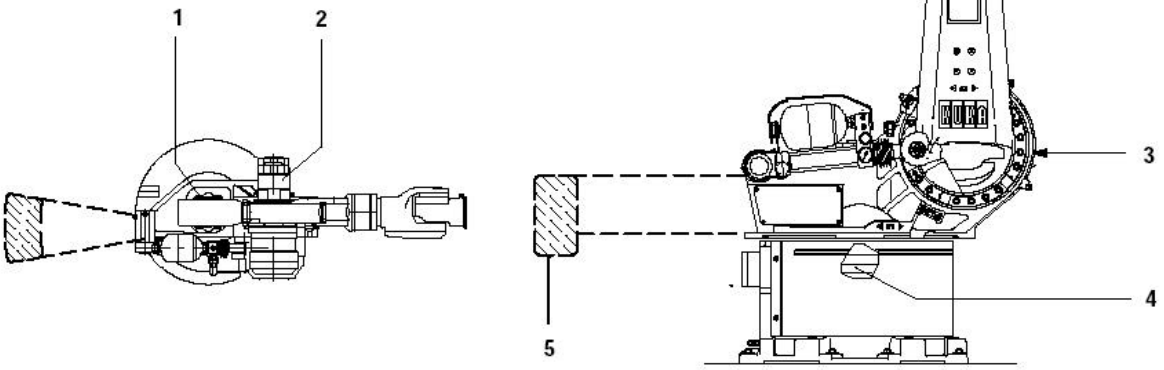
- **Arm mit Drehbereich und Zentralhand:**
  Der Arm ist seitlich an der Schwinge angeflanscht und wird vom Grundachsantrieb angetrieben.



1...Antriebseinheit für Handachse
2...Arm
3...Zentralhand
4...Welle
5...Grundachsantrieb A3
6...Schwinge
7...Drehachse 3
8...Gegengewicht A3

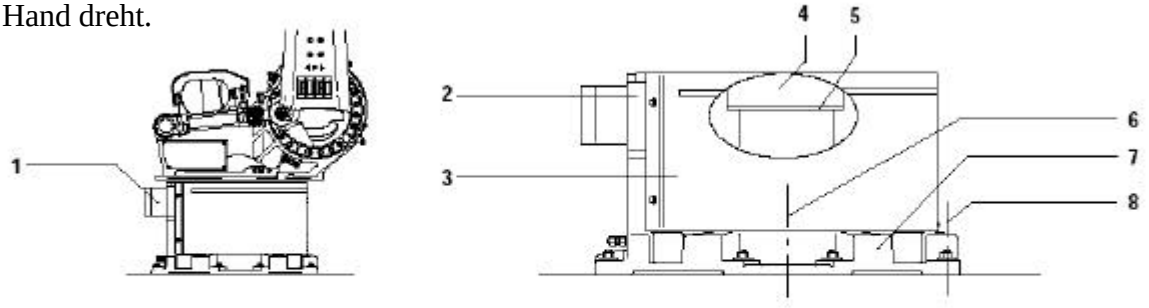
- **Schwinge:**
  Abtriebselement der 2. Achse.

- **Karussell:**
  Führt die Bewegung um die Drehachse 1 aus.



1 Grundachsantrieb A 1
2 Grundachsantrieb A 2
3 Spezial-Untersetzungsgetriebe A 2
4 Spezial-Untersetzungsgetriebe A 1
5 Gegengewicht A 1 (nur für KR 125 W/1, /2)

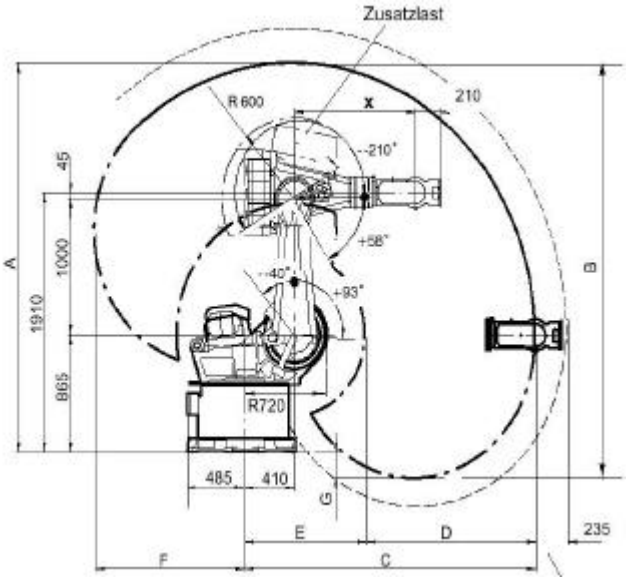
- **Grundgestell:**
  Ist der feststehende Teil des Roboters auf dem sich das Karussell mit Schwinge, Arm und Hand dreht.

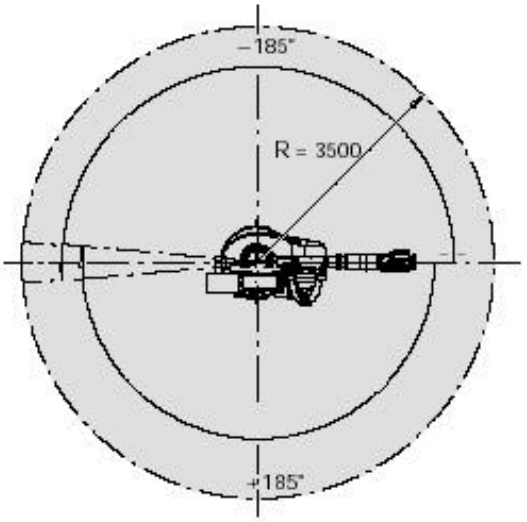


1 Anschlußkästen (2x)
2 Grundgestell-Gehäuse
3 Abdeckung
4 Spezial-Untersetzungsgetriebe A 1
5 Flansch
6 Paßbohrungen (2x)
7 Fußflansch
8 Befestigungsbohrungen (8x)

- **Verbindungsleitung:**
  Übertragung der Steuerungssignale sowie der Messwerte zwischen Roboter und Steuerschrank.

- **Arbeitsbereich:**





A:	2688 mm	E:	1005 mm
B:	3054 mm	F:	1234 mm
C:	2410 mm	G:	188 mm
D:	1405 mm	X:	1000 mm

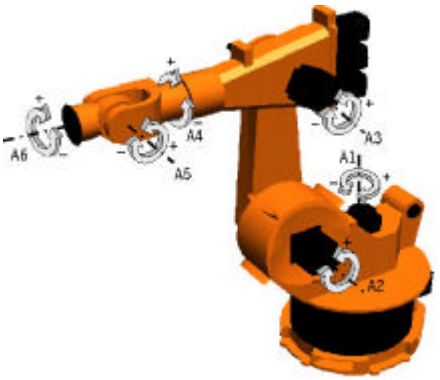
3.2 Technische Daten:

M2-Anfang

- Nennleistung 4KW
- Datensicherung Lithium Batterie
- Maximalverfahrgeschw. 1,45 m/s
- Traglast 125kg
- Zusatzlast 120kg
- Max. Gesamtlast 245kg
- Eigengewicht 975kg
- Max. Reichweite 2410 mm
- Armlänge 1000 mm
- Wiederholgenauigkeit <± 0,2 mm
- Achsenzahl 6
- Einbaulagen Boden, Decke
- Zulässige mechanische und klimatische Beanspruchung bei Betrieb:
  - mit Kühlgerät: 0° bis 55°C
  - ohne Kühlgerät: 0° bis 45°C

3.3 Koordinatensysteme:

M2/Tr.Bl.8/ S32/66



Wie man in dieser Abbildung sieht, besitzt der Kuka-Roboter 6 rotatorische Achsen, die es ihm ermöglichen, einen starren Körper im Raum eindeutig zu bestimmen.

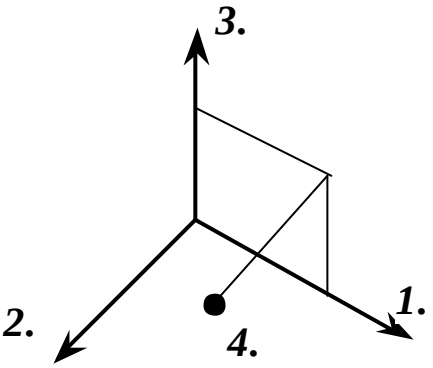
3.3.1 Allgemein

- **Globales Koordinatensystem:**

Bei diesem liegt der Ursprung im Zentrum der Hauptachse.

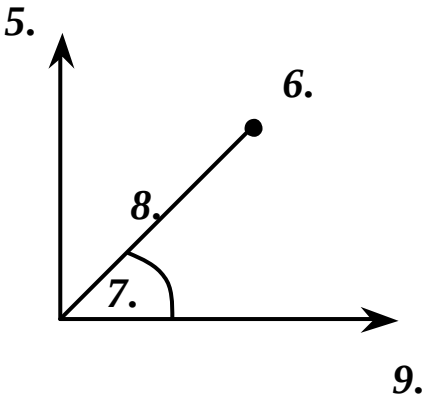
- **Rechtwinklige (kartesische) Koordinaten**

entstehen dadurch, dass durch den Ursprung drei senkrecht aufeinander stehende Geraden gelegt werden. Sie heißen Koordinatenachsen und werden in der Regel als x-, y- und z-Achse bezeichnet. Im allgemeinen wird ein **rechts orientiertes** Koordinatensystem benutzt. (Wird die x-Achse auf dem kürzesten Weg auf die y-Achse gedreht, zeigt die z-Achse in Richtung der Vorwärtsbewegung einer rechtsdrehenden Schraube.) Für ein **links orientiertes** System gilt entsprechendes. Um einen Punkt im Raum eindeutig zu bestimmen sind 3 Parameter notwendig, nämlich x-, y- und z-Koordinate.



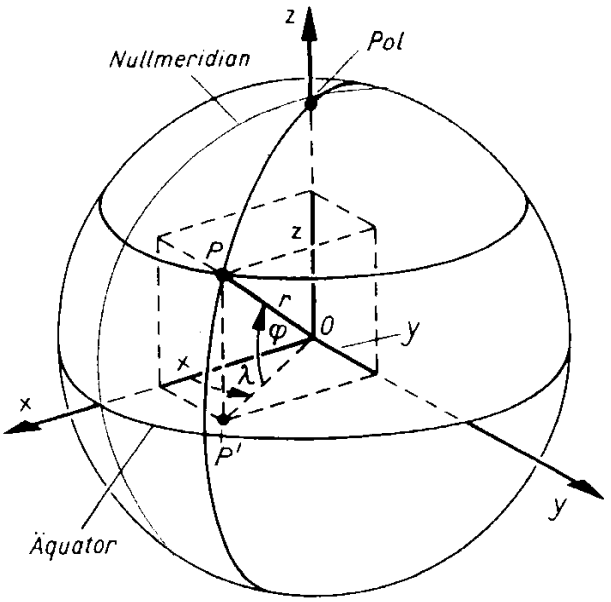
- **Polares Koordinatensystem**

Mit Hilfe der polaren Koordinaten lässt sich die Position eines Punktes mit Hilfe des Radius r und eines Winkels (φ) eindeutig definieren.



- **Kugelkoordinaten**

sind dann zweckmäßig, wenn bestimmte Probleme, die sich auf der Oberfläche einer Kugel (Erde) abspielen, berechnet werden sollen (Navigation im Schiffs- und Flugverkehr).
Ein Punkt wird darin festgelegt durch den Kugelradius r, durch den Winkel f von r zur Äquatorialebene und den Winkel ? zwischen Nullmeridian und der Projektion von r in die Äquatorialebene.



Kugelkoordinaten eines Raumpunktes

3.3.2 Kuka-spezifisch

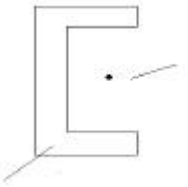
Bei dem Kuka-Roboter KR 125/3 ist es möglich, mit 4 verschiedenen Koordinatensystemen zu arbeiten.

- **Globales Koordinatensystem:**
  Bei diesem liegt der Ursprung im Zentrum der Hauptachse.

- **Lokales Koordinatensystem:**
  Bei diesem ist der Ursprung im Raum frei wählbar.

- **Achsenbezogenes Koordinatensystem:**
  Jede Achse ist einzeln steuerbar.

- **Tool Koordinaten (TCP):**
  Der Nullpunkt liegt im Greiferbereich.



Greifer

3.4 ONLINE – OFFLINE - Betrieb:

siehe Controlpanel M2/Tr.Bl.6/...

- **Online - Programmierung mittels Kuka Control Panel (KCP):**
  Die Programmierung erfolgt direkt am Roboter mit Hilfe des Teach in, Touch up – Verfahrens (Punkte werden manuell angefahren, und deren Position dann abgespeichert).

  Vorteil: Relativ einfach zu programmieren, Sofortkontrolle der jeweilig programmierten Schritte.

  Nachteil: Roboter kann neben programmieren keine Arbeit verrichten; je größer das Programm, desto mühsamer wird programmieren.

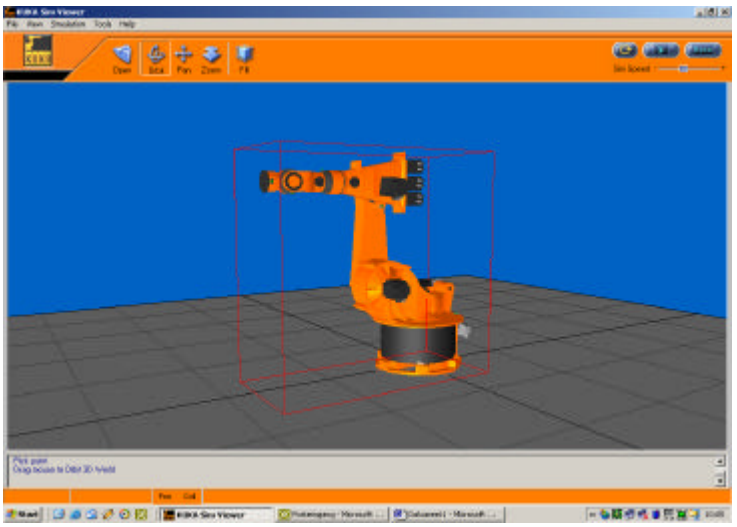
- **Offline - Programmierung mittels PC:**
  Die Programmierung erfolgt am PC und wird danach über die Steuerung in den Roboter geladen. Programmiert wird entweder mit Hilfe einer 3D Simulation oder eines einfachen Text-Editors.

  Vorteil: Dem Roboter wird ermöglicht, auch während dem Programmieren Arbeit auszuführen. Änderungen des Programm leicht möglich.

  Nachteil: Hoher Speicherbedarf

  Programme zur 3D Offline – Programmierung mittels PC:

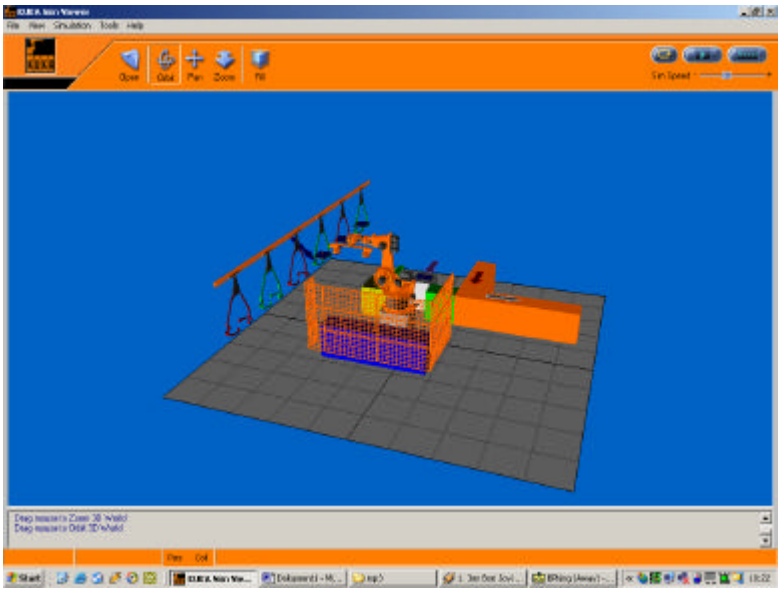
  - **KUKA Sim Viewer:**



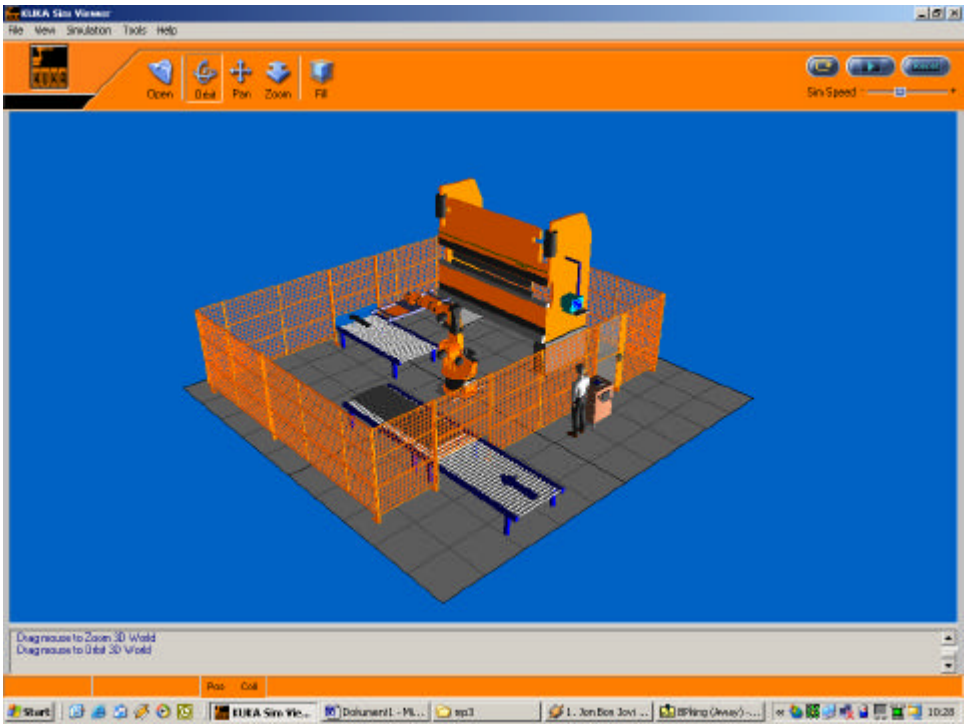
Dieses Programm ist zum gratis download auf der KUKA – Homepage (www.kuka.de) als Shareware angeboten.

Das Programm erlaubt aufgrund der Tatsache, dass es nur ein Shareware Programm ist, nur sehr geringe Anwendungen. Man kann immer nur einen Roboter einbauen, und das Programm erlaubt einem nicht mehrere Komponenten zu einer funktionierenden Einheit zusammenzubauen. Es kann immer nur ein Element auf dem Computermonitor erscheinen.

Im lizenzierten Programm könnte man wahrscheinlich wirklich gut simulieren, denn es ist vom Roboter Zubehör wirklich alles vorhanden. Man kann vom KCP bis zum Schaltkasten bis zu den verschiedenen Robotern alles einfügen.

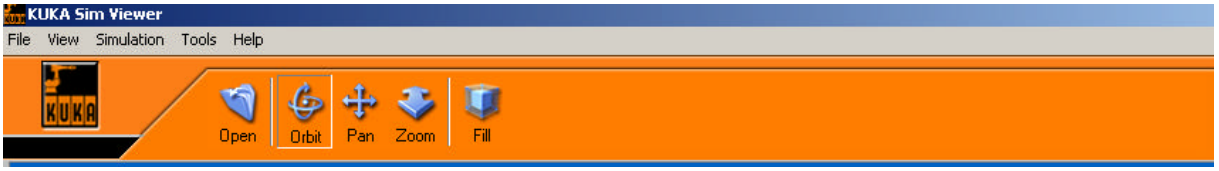


Dies ist ein Probebeispiel, das bei der Shareware mit dabei war. Diese Probebeispiele kann man jedoch simulieren.

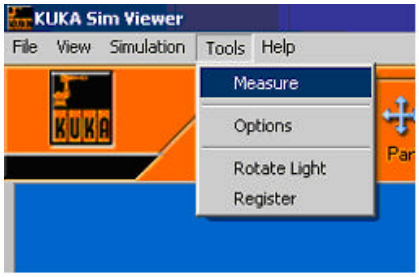


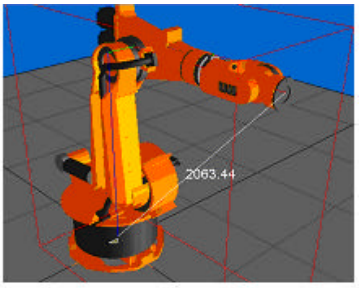
Die diversen Roboter werden im Programm über den Programm Button *„Open“* hereingeladen. Man trifft dann auf ein gut gestaffeltes, übersichtliches Ordnersystem in dem vom Roboter bis zu den Schaltkästen usw... alles verzeichnet ist.

Des weiteren kann man im Programm zoomen und verschiedene Seitenansichten einstellen. Dies funktioniert über die Buttons *„Zoom“* und *„Orbit“*.



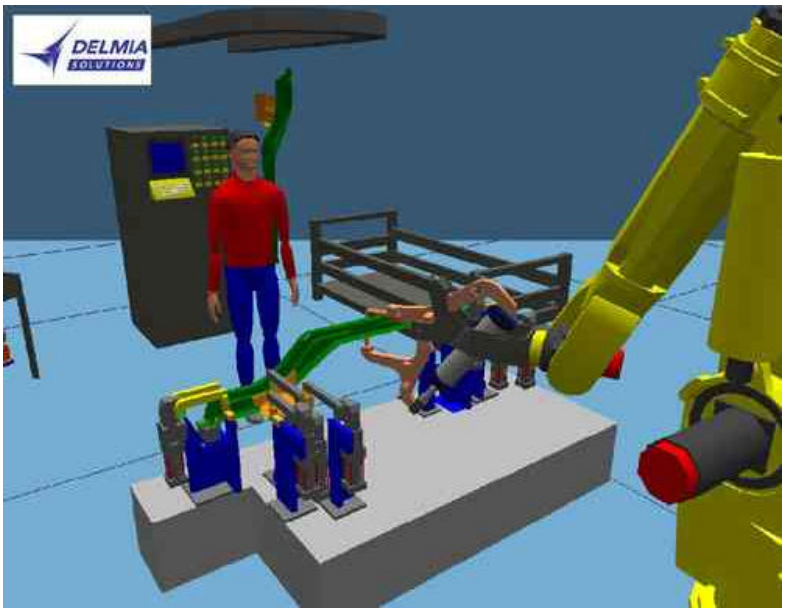
In der Menüführung unter Tools gibt es einen Unterpunkt *„Measure“* mit dem man verschiedene Punkte am Roboter vermessen werden können. Dies ist ganz praktisch, denn damit sind die Achsabstände immer bekannt.



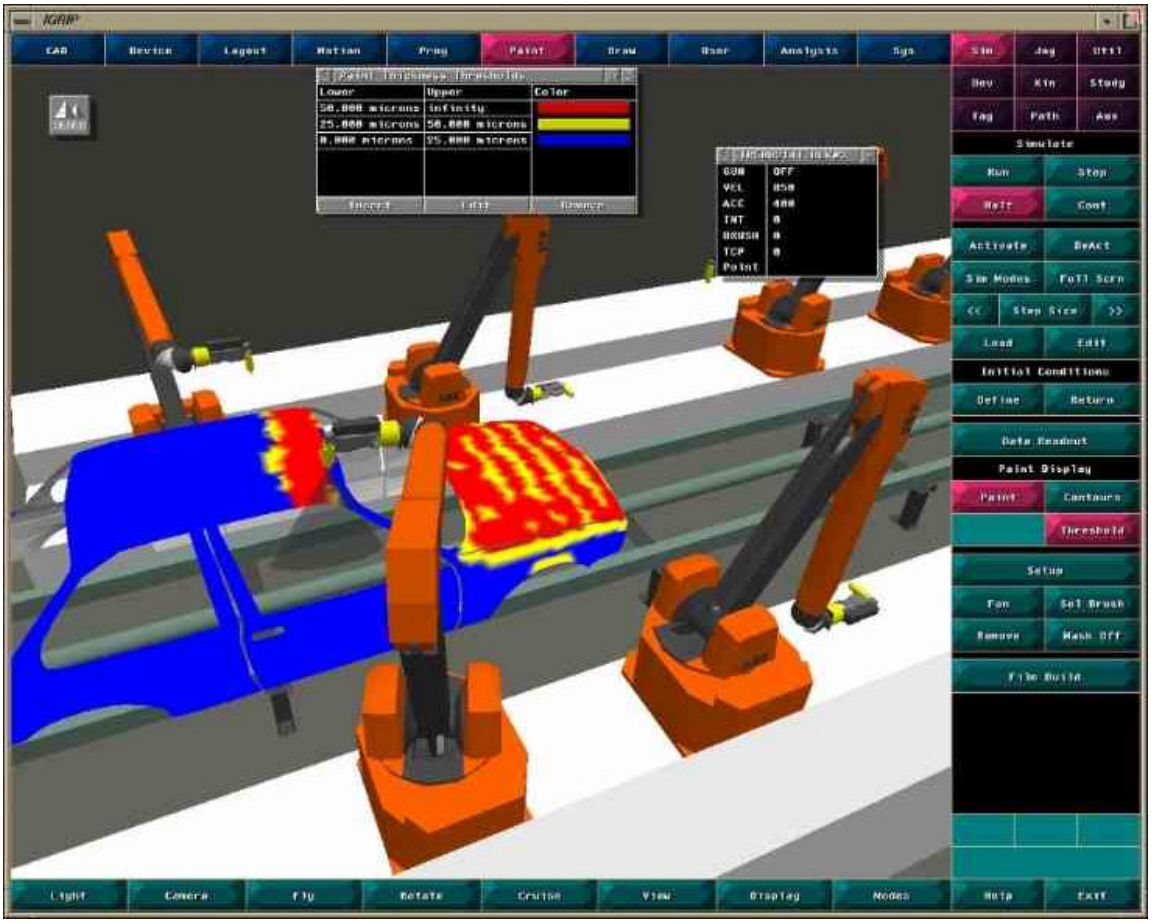


- **IGRIP (Interactive Graphics Robot Instruction Program)**

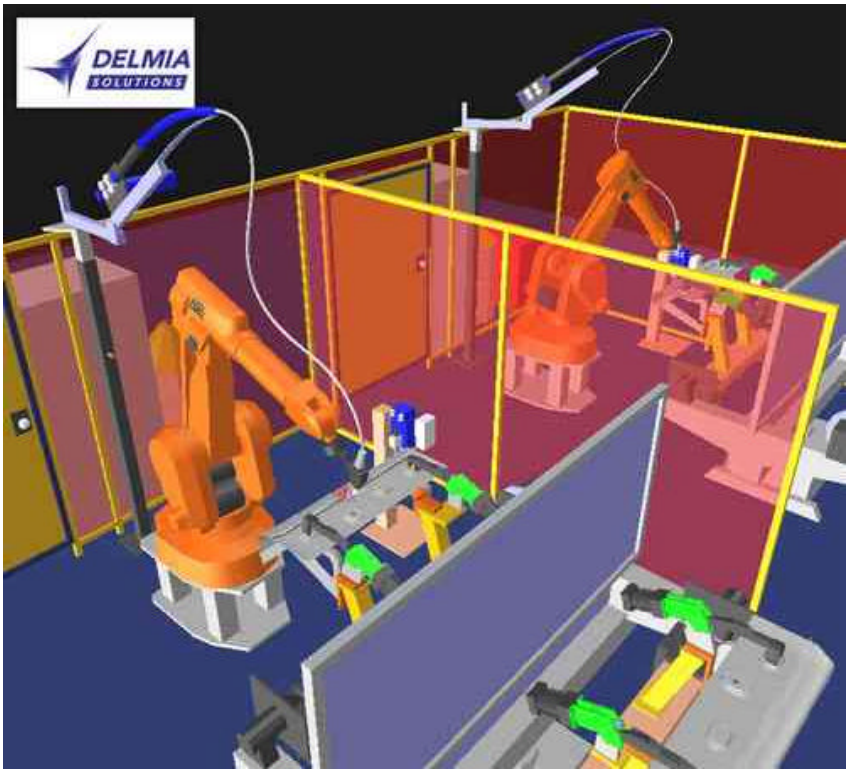
Anwendungsbeispiele (siehe www.delmia.de ,www.daneb.com .....) :



Handhabung von Bauteilen - Ergonomie



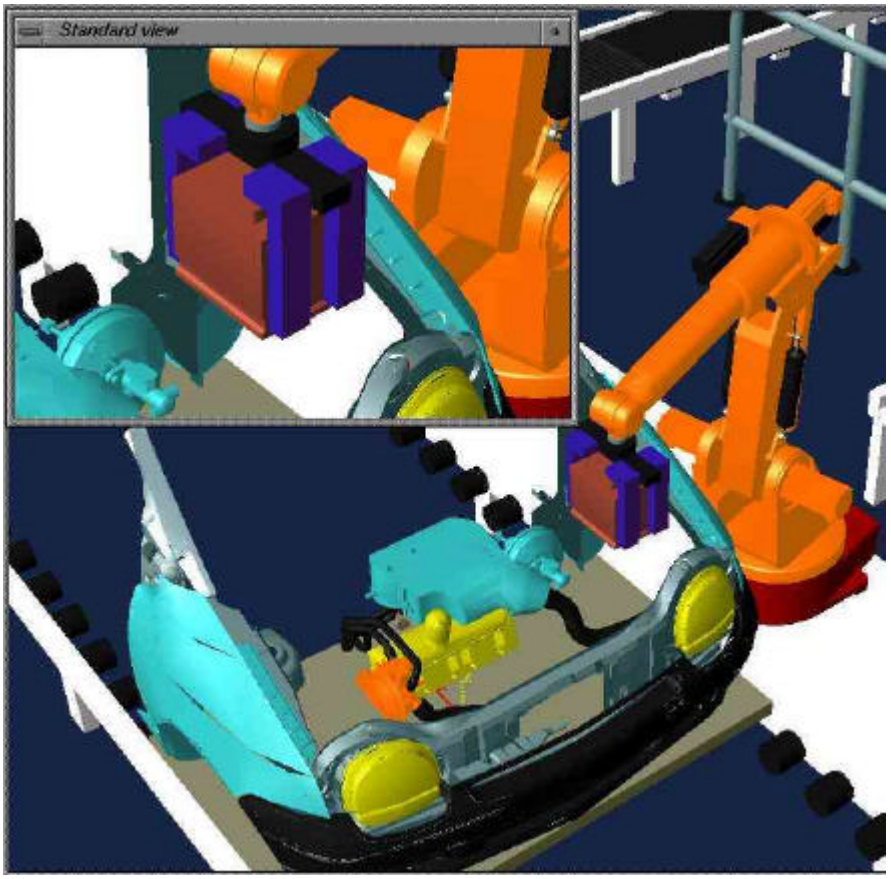
Lackierprozesse



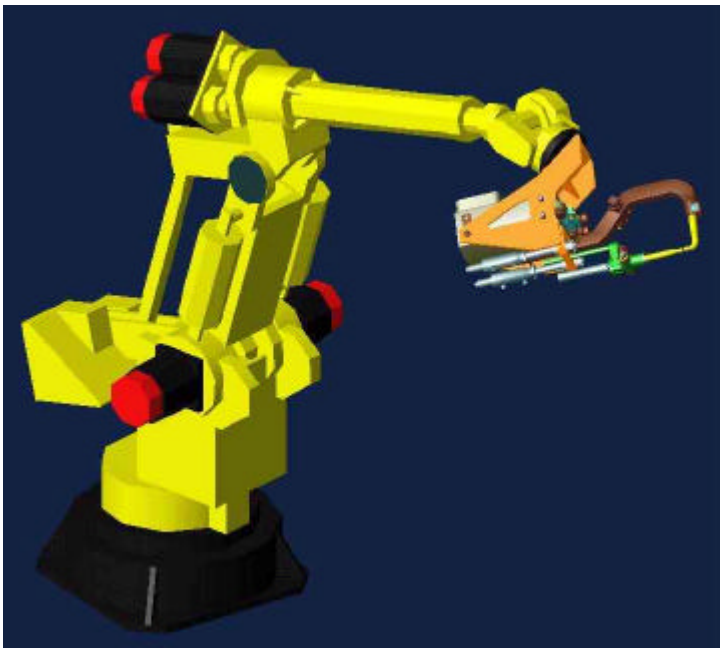
Schutzgasschweißzellen MIG-MAG

- **ROBCAD von Tecnomatix Technologies Ltd.**

Anwendungsbeispiele (siehe http://www.saktec.com/robcad_de.htm .....) :



Montage und Handhabung



Roboter mit Punktschweißzange (Widerstandschweißen)

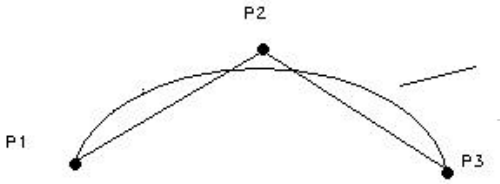
3.5 Verfahrmöglichkeiten:

M2/Tr.Bl.8/S17/58

Drei verschieden Verfahrensmöglichkeiten sind möglich.

- **Punktsteuerung (PTP...Point - To - Point):**
  Dem Roboter wird ein Punkt im Raum programmiert, dem er von seiner Ausgangsposition aus in kürzester Distanz anfährt.

- **Bahnsteuerung (CP...Continous Path):**
  Dem Roboter werden drei Punkte programmiert, die er in einer Bahn abfährt. Dadurch entsteht eine harmonische Bewegung.



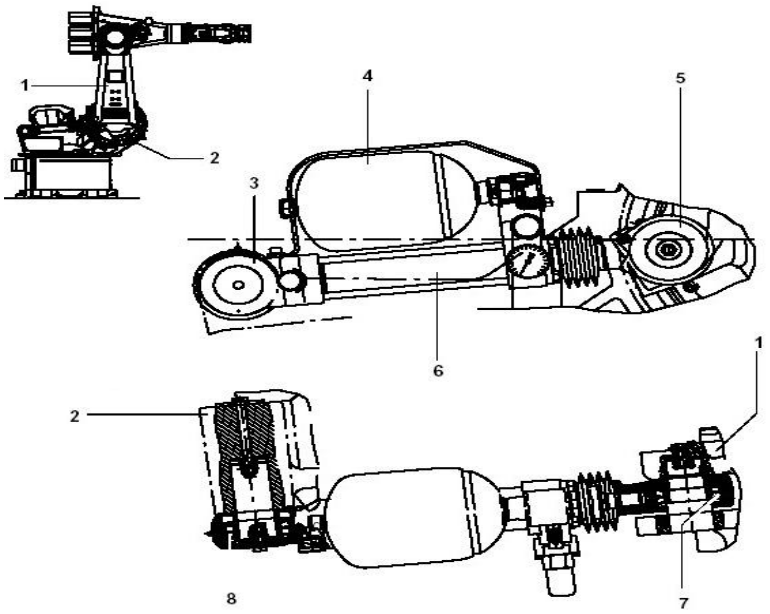
Tatsächlich abgefahrene Bahn

- **Linearsteuerung:**
  Zwischen zwei programmierten Punkten fährt der Roboter eine Gerade.

3.6. Antriebssysteme und Kraftausgleich:

- **Antriebssystem:**
  Alle 6 Achsen werden elektro-mechanisch mit transistorgesteuerten AC-Servomotoren angesteuert. Der Greifer wird allerdings pneumatisch gesteuert.

- **Kraftausgleich:**
  Der Roboter verfügt zum Ausgleich des Lastmoments der Achse 2 über ein geschlossenes hydropneumatisches Ausgleichssystem. Das System ist zwischen Karussell und Schwinge installiert. Es verhindert somit ein Kippen des Roboters bei schnellen Bewegungen bzw. beim Beschleunigen.



1 Schwinge
2 Karussell
3 Gelenkkopf
4 Membranspeicher
5 Gelenkkopf
6 Hydraulikzylinder
7 Schwenklager
8 Schwenklager

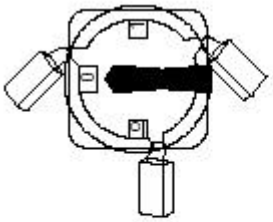
4. Steuerkasten:

M2/Tr.Bl.1,2,3,4,5,7

Der Steuerschrank ist das Hirn des Roboters und muss dementsprechend gesichert werden. Die Magnetkarte verhindert das unbefugte Personen das System hochfahren können.

Durch das Vorhängeschloss wird ein ungewünschtes Einschalten (z.B. bei Wartungsarbeiten am Roboter) verhindert.

Magnetschloss



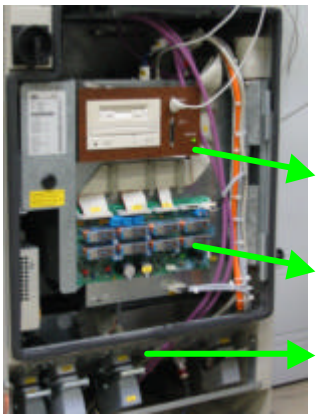
Steuerschrank mit Bildschirm

Innenleben des Steuerschrank

4.1 Schnittstellen-Übersicht

M2/Tr.Bl.2/S8/27

- **Schnittstellen des KRC1**



PC-Teil

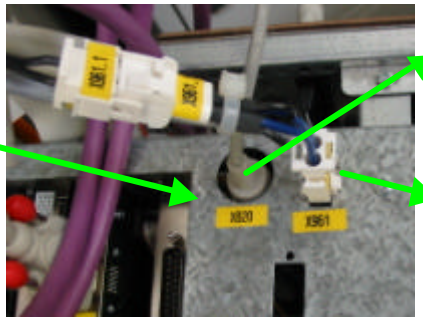
Relais-Karte (A1) mit Sicherheitslogik

externe Schnittstellen (A3)

COM 1: Serielle Schnittstelle (z.B. für serielle Maus)

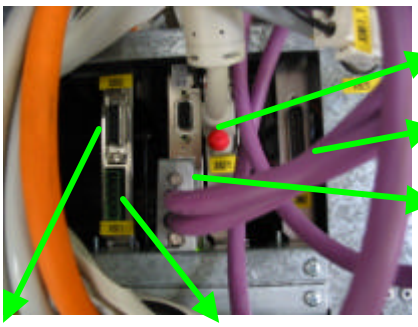
LPT 1: Parallele Schnittstelle (z.B. als Drucker-Anschluss)

Sensor-Schloss



X820:
PS2-Schnittstelle
(für Tastatur)

X961:
Stromversorgung für PC-Teil



X821: Monitor-Anschluss

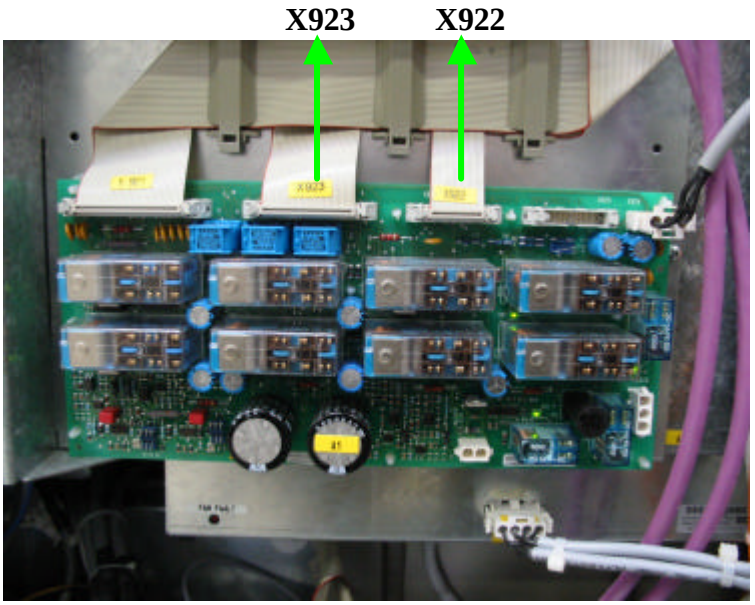
COM 2: 2. Serielle Schnittstelle

Multifunktionskarte MFC (mit System- und Anwender- I/O, Ethernetcontroller und Schnittstelle zwischen KCP und PC) mit **Digital Servo Elektronik DSE-IBS** (mit Digitalem Signal Prozessor; steuert Servos am Roboter an und verarbeitet Fehler- und Situationsinformationen; mit Interbus-S Interface)

X802:
Ethernet
Sub-D-Stecker

X801:
DeviceNet
Combicon-Stecker

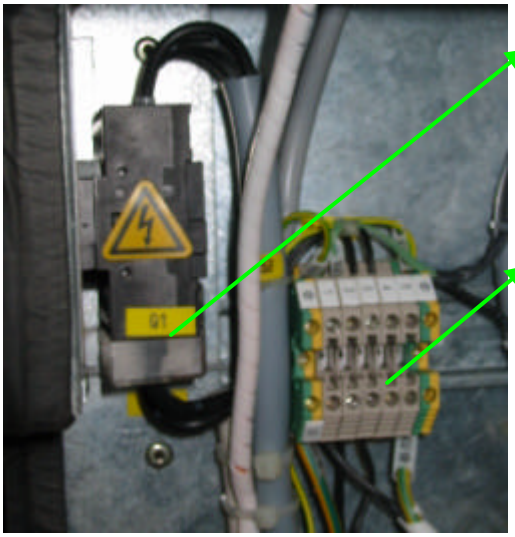
- **Schnittstellen der Relaiskarte (A1)**



X922:
Verbindung zum Powermodul N1 (über Klemme **X105)**

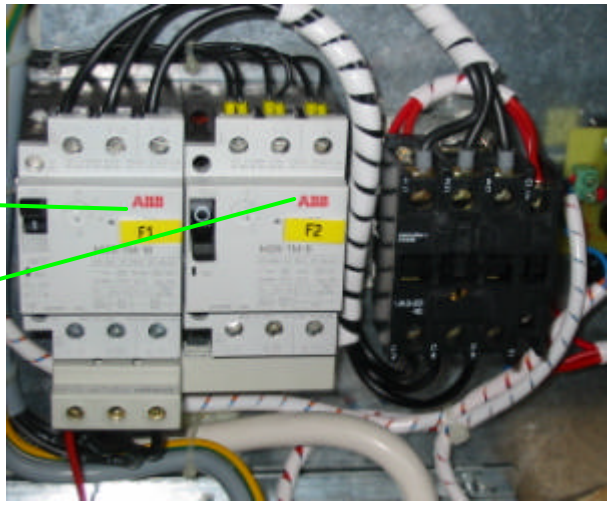
X923: Verbindung zur MF Karte (siehe oben) im PC
(über Klemme **X807)**

- **Hauptschalter und Sicherungen**



Q1: Hauptschalter

X3: Klemmleiste für Netzanschluss

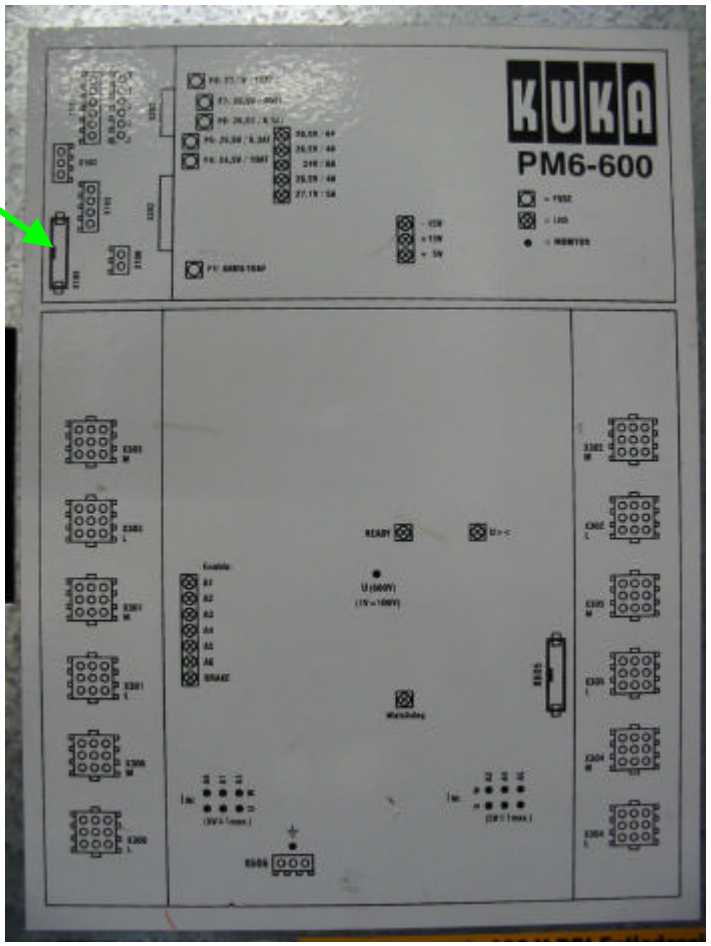


F1:
Sicherung des gesamten KRC

F2:
Sicherung des Gebläses (als Schalter verwendet)

Z1: Funkestörfilter (unterhalb der Sicherungen; im Bild nicht zu sehen)

- **Schnittstellen des Power Moduls (PM6-600)**



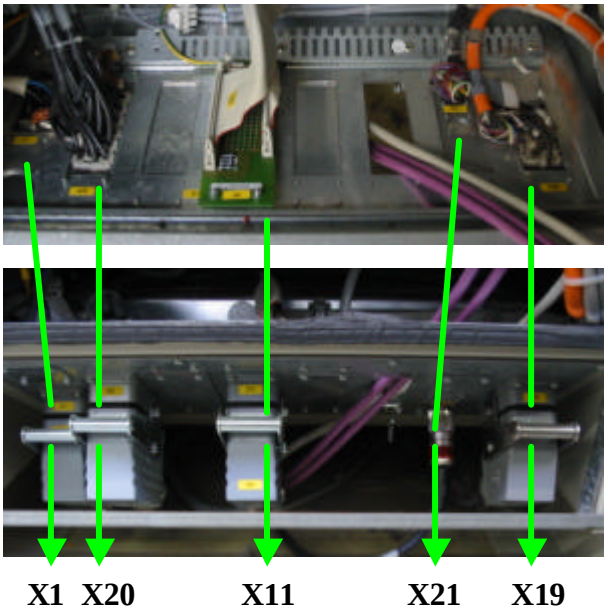
Steuerleitungen:

X101: Anspeisung von F1
X102: Ballastwiderstand
X105: Verbindung zur Relais Karte A1 über Klemme X922
X106: Lüftersteuerung & Ballastwiderstand-Steuerung (für Bremsen)
X202: Verbindung zum Akku
X203: Versorgung des Logikteils von N1

X301M: Achse 1 (X)
X302M: Achse 2 (Y)
X303M: Achse 3 (Z)
X304M: Achse 4 (Drehung um Z)
X305M: Achse 5 (Drehung um Y)
X306M: Achse 6 (Drehung um X)

X605: D-SEAT
X606: Erdungsanschluss
F1: Akku-Sicherung
F4: 24V gepuffert
(**X961** zum PC)
(**X964** zur Relais Karte A1)
F5: FE201 Peripherie Beleuchtung
(**X964**)
F6: Stromversorgung für optionale Peripherie (**X962**)
F7: Stromversorgung für optionale Peripherie (**X963**)
F8: Bremsen (**X608**)
A1-A6: LED-Anzeige für „aktiv“-Status der jeweiligen Achse
Brake: LED-Anzeige für „aktiv“-Status der Bremsen

- **Externe Schnittstellen des Steuerschranks**



X1: Netzanschluss (Han6 HsB)

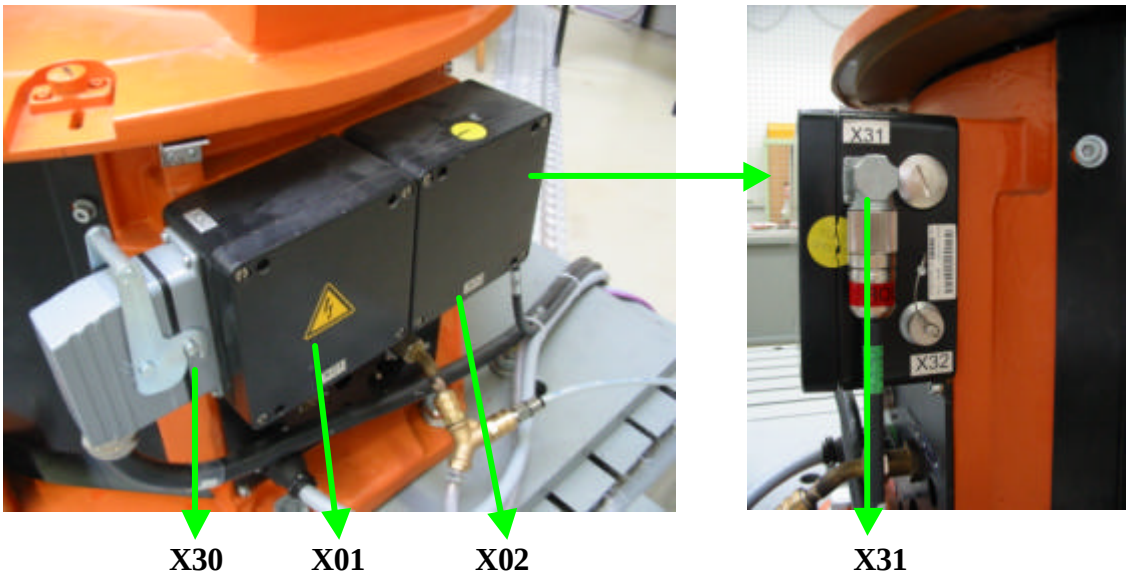
X11: Peripheriestecker (**X76** am Roboter)

X19: KCP-Stecker (KUKA-Control-Panel)

X20: Motorstecker Achse 1 bis 6 (ist die Motorleitung zum Roboter)

X21: Datenleitungsstecker Achse 1 bis 8 (ist die Datenleitung zum Roboter)

- **Schnittstellen am Roboter KR125**

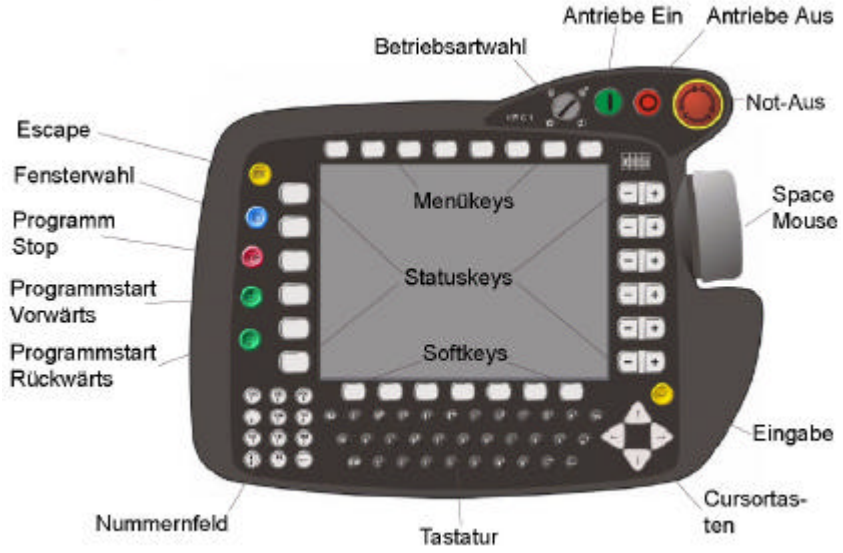


X30: Anschluss (des Moduls **X01**) für Motorleitung vom Steuerschrank
X31: Anschluss (des Moduls **X02**) für Datenleitung vom Steuerschrank

5. KUKA Control Panel (KCP)

M2/Tr.Bl.8/S10-23/66

Durch das KCP kann der Roboter im ONLINE – Betrieb programmiert und gesteuert werden.
Das KCP ist über ein Kabel mit dem Steuerschrank verbunden. Man hat daher eine eingeschränkte Bewegungsfreiheit.
Funktion Siehe Kapitel Bedienung.



6. Inbetriebnahme des KUKA Roboter:

6.1 Hochfahren:

M2/Tr.Bl.8/S10/60

- Entfernung der Vorhangschlösser an der Tür der Manipulationszelle sowie am Steuerschrank und Entriegelung des Sicherheitsschalters am Steuerschrank, mittels Magnetkarte der dazu berechtigten Lehrpersonen.

- Durch Betätigung des Hauptschalters am Steuerschrank werden die Steuerung und die dazu benötigten Daten hochgeladen.
  Dauer ca. 3 min

- Druckluftventil der Pressluft öffnen
  (befindet sich links hinten im eingezäunten Bereich)

- Entriegeln aller NOT-AUS-Taster da der Roboter sonst nicht funktionieren würde

- Betätigung des Quittierungstasters um die Inbetriebnahme abzuschließen

Bemerkung:
Die Inbetriebnahme des Roboters darf nur unter Aufsicht einer Lehrperson erfolgen!

Beim Automatikbetrieb muss die Türe unbedingt verriegelt werden da ansonsten akute Verletzungsgefahr herrscht!

6.2 Herunterfahren:

- Der Roboter sollte sich in seiner HOME – Position (Sicherheitsposition) befinden
- Hauptschalter am Steuerschrank und wenn nötig Bildschirm Ausschalten
- Druckventil der Pressluft schließen
- Hauptschalter am Steuerschrank und Sicherheitstüre versperren
- Schlüssel und Magnetkarte an eine Lehrperson zurückgeben

7. Beschreibung im Hand- und Automatikbetrieb:

M2/Tr.Bl.9/S20/122

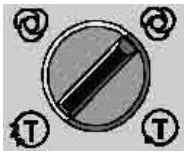
7.1 Beschreibung des KCP:

- **NOT - AUS – Schalter:**
  Bei Betätigung werden die Antriebe des Roboters sofort Abgeschaltet. Die Anlage kann erst nach der Entriegelung dieses Schalters wieder in Betrieb genommen werden. Anschließend muss die NOT-AUS-Meldung dir. im Fehlermeldungsfenster quittiert werden.

- **Antrieb EIN:**
  Durch Betätigung dieser Taste werden alle Antriebe des Roboters eingeschaltet. Vorausgesetzt die NOT – AUS – Taste ist nicht gedrückt.
  In der Betriebsart „HAND“ hat diese Taste keine Funktion.

- **Antrieb AUS:**
  Bei Betätigung dieser Taste werden alle Antriebe des Roboters wieder ausgeschaltet. Die Bremsen des Roboters halten die Achsen in ihrer befindlichen Position. In der Betriebsart „HAND“ hat sie wiederum keine Bedeutung.

- **Betriebsartenwahl:**
  Folgende Betriebsarten können mit diesem Schalter gewählt werden.



  - **Test 1:**
    Bewegung mit reduzierter (im Menü def.) Geschwindigkeit. Der Roboter verfährt erst wenn die Zustimmtaste und mindestens einmal die „Programmstart Vorwärts“ betätigt wurde.

  - **Test 2:**
    Zustimmtaste und eine Programmstart - Taste müssen betätigt werden. Bewegung mit prog. Geschwindigkeit.

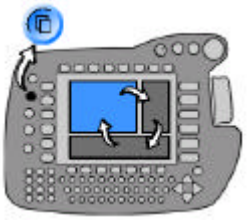
  - **Automatik:**
    Bewegung mit programmierter Geschwindigkeit. Das angewählte Programm wird Programmautomatisch durchgearbeitet und dabei vom KCP kontrolliert.

  - **Extern:**
    Funktion genauso wie bei Automatikbetrieb, nur der Roboter wird entweder von einem Leitrechner oder von einer SPS kontrolliert.

- **ESCAPE – Taste:**
  Mit der Escape-Taste kann eine begonnene Aktion jederzeit abgebrochen werden. Dazu gehören beispielsweise geöffnete In Line- Formulare und Zustandsfenster. Auch versehentlich geöffnete Menüs lassen sich durch Betätigen dieser Taste wieder schrittweise schließen.

- **Programm – STOP:**
  Bei Betätigung wird ein laufendes Programm in der Befindlichen Position angehalten. Um das Programm fortzusetzen muss wiederum die Taste „Programmstart Vorwärts“ betätigt werden.

- **Fensterwahl – Taste:**
  Mithilfe dieser Taste kann zwischen den geöffneten Fenstern gewählt werden. Sie sind durch einen bläulichen Hintergrund hervorgehoben.

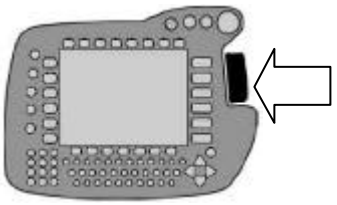


- **Programmstart Vorwärts:**
  Durch Betätigung wird ein gewähltes Programm gestartet. Der Start im T I – T II – Betrieb ist nur dann möglich, wenn die Zustimmtaste betätigt ist. Beim Loslassen dieser Taste erfolgt ein bahngetreuer Stop.

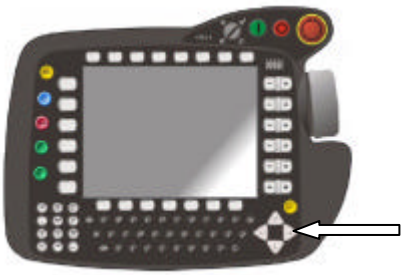
- **Programmstart Rückwärts:**
  Beim Betätigen dieser Taste wird das Programm schrittweise von hinten bis zum Programmanfang abgearbeitet. Diese Betriebsart ist nur in der Programmablaufsart „Einzelschritt“ verwendbar.

- **Eingabe Taste:**
  Diese Taste entspricht der „RETURN – Taste“ am PC. Es können hiermit Befehle abgeschlossen und bestätigt werden.

- **Space – Mouse:**
  Sie dient zur Positionierung des Roboters in allen 6. Achsen oder Freiheitsgraden. Je nach Auslenkung kann die Fahrgeschwindigkeit variiert werden.

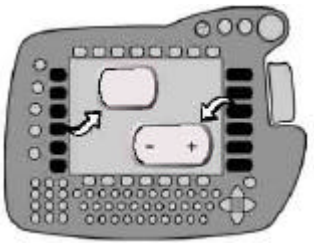


- **Cursortasten:**
  Mithilfe dieser Tasten können verschiedene Menüs in der Menüleiste gewählt werden oder zwischen den einzelnen Sätzen im Programm gewählt werden. Die Pfeile zeigen die Bewegungsrichtung des Cursors an. Funktion ist sonst gleich wie am PC.

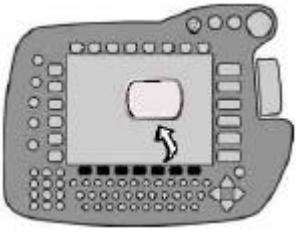


- **Menükeys:**
  Bei Betätigung dieser Tasten können Menüs in der dir. darrunterliegenden Menüleiste geöffnet werden. Die einzelnen Punkte in den Menüs können entweder durch die Cursor Tasten oder durch die links nebenstehenden Zahlen gewählt werden.

- **Statuskeys:**
  Die Statuskeys (links und rechts neben dem Display) dienen zur Auswahl von Betriebsoptionen, zum Schalten einzelner Funktionen sowie zum Einstellen von Werten. Die jeweilige Funktion wird durch entsprechende Symbole in der Statuskeyleiste grafisch dargestellt.



- **Softkeys:**
  Mit diesen Bedienelementen werden die (unten im Display) in der Softkeyleiste angezeigten Funktionen ausgewählt. Die zur Auswahl stehenden Funktionen werden dynamisch angepasst, d.h. die Softkeyleiste verändert ihre Belegung.



- **Nummernfeld:**
  Über diese links unten am KCP liegenden Tasten können Zahlen für ein Programm eingegeben werden. In einer zweiten Belegung ist es mit Cursor – Steuerfunktionen ausgestattet.

- **Zustimmtaste:**
  Auf der Rückseite des KCP befinden sich die Zustimmtasten, die bei der Programmierung eines Programms von großer Bedeutung sind. Ohne diese Tasten kann der Roboter im Test Betrieb nicht bewegt werden.

7.2 Funktion des KCP:

M2/Tr.Bl.10/S49/122

7.2.1 Koordinatensysteme:

Zum Handverfahren des Roboters mit den Verfahrtasten muss ein Koordinatesystem ausgewählt werden, auf das sich die Roboterbewegungen beziehen.

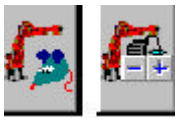
- **Achsspezifisches Koordinatensystem:**
  Jede Roboterachse kann einzeln positiv oder negativ verfahren werden. Dies geschieht mit Verfahrtasten oder der Spacemouse, wobei hier 3 bzw. 6 Achsen simultan bewegt werden können.

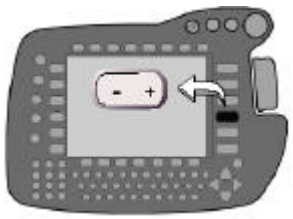
- **World – Koordinatensystem:**
  Globales (absolutes) Koordinatensystem, dessen Ursprung im Fuß des Roboters liegt;

- **Base – Koordinatensystem:**
  Lokales Koordinatensystem, das seinen Ursprung in oder am zu bearbeitenden Werkstück bzw. einer Vorrichtung hat;

- **Tool – Koordinatensystem:**
  Koordinatensystem, dessen Ursprung im Werkzeug liegt.

Das Bezugskoordinatensystem ist nur in der Betriebsart „Handverfahren“ umschaltbar Statuskey „Verfahrart“ zeigt „Spacemouse“ oder „Verfahrenstasten“ an. Zur Auswahl des Koordinatensystems wird nun die entsprechende Statustaste (+/-) gedrückt.





7.2.2 Handverfahren des Roboters:

Das Handverfahren dient zum manuell gesteuerten Bewegen des Roboters, z.B.: beim Teach – In von Zielpunkten oder beim Freifahren. Zum Handverfahren muss der Betriebsartenschalter auf „Tippbetrieb“ – T I oder T II eingestellt sein. In der Stellung „Automatik“ bzw. „Automatik Extern“ ist Handverfahren nicht möglich. Die aktuelle Einstellung des Betriebsartschalter wird in der Statuszeile angezeigt.

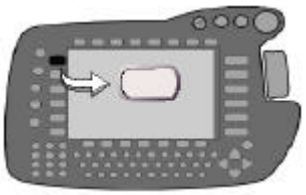
- **Verfahrart auswählen:** M2/Tr.Bl.10 /S50/122

Verfahren mit der Spacemouse:
Abhängig von der Einstellung der Freiheitsgrade gleichzeitig in 3 bzw. 6 Achsen

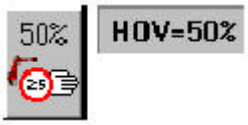
Verfahren mit Verfahrenstasten:
Jede Achse einzeln

Handverfahren ausgeschaltet:

Zur Auswahl betätigen Sie die Statustaste „Verfahrart“ so oft, bis das gewünschte Symbol erscheint.



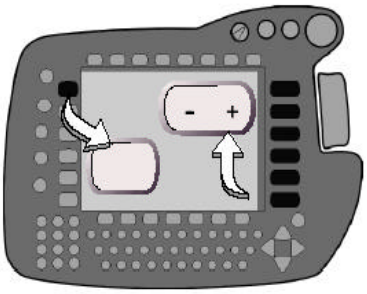
- **Geschwindigkeit zum Handverfahren „Handoverride“:**
  In vielen Fällen ist es nötig, die Verfahrgeschwindigkeit des Roboters zu reduzieren. Nur so können Punkte genau angefahren oder Kollisionen vermieden werden.



Zu diesem Zweck dient die Funktion „Handoverride“ welche nur in der Betriebsart „Handverfahren zur Verfügung steht. Der Wert des Handoverride kann mit der +/- Statustaste rechts neben dem Symbol „Override ändern“ vertellt werden.

- **Verfahren mit den Verfahrtasten:**

Der Roboter kann durch betätigen der +/- Statuskeys in Abhängigkeit vom eingestellten Bezugskoordinatensystem verfahren.



- **Achsspezifisches Koordinatensystem:** M2/Tr.Bl.10/S55/122

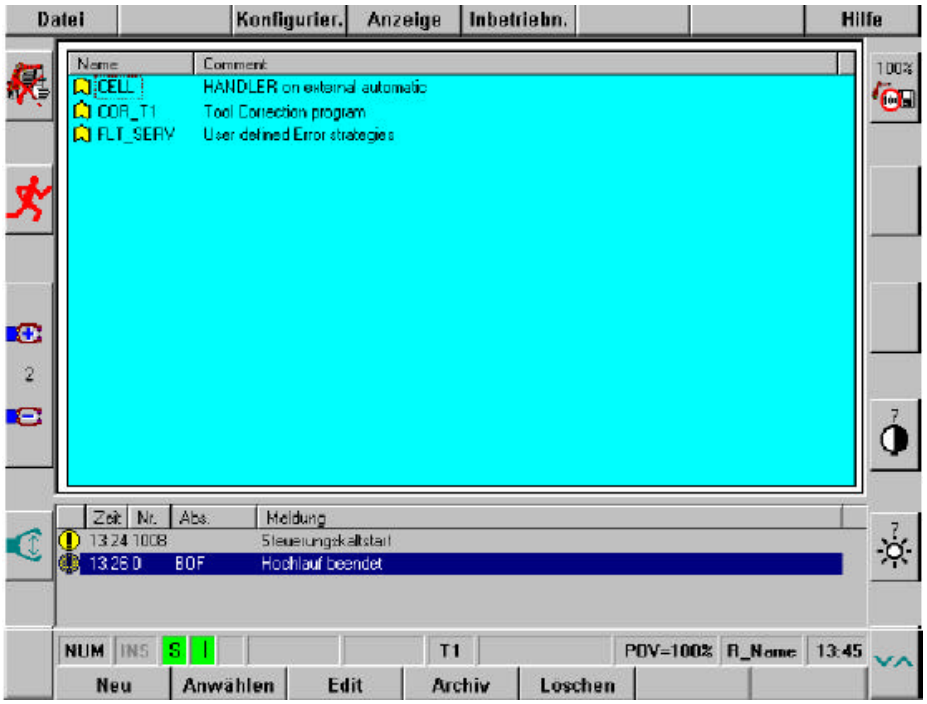
In diesem Modus kann jede Roboterachse einzeln durch Betätigung des entsprechenden +/- Statuskeys (A1 – A6) bewegt werden. Die Pfeilrichtung steht für die positive Bewegungsrichtung.

- **Koordinatensysteme Tool, Base, World:**
  Befinden Sie sich in einem der Koordinatensysteme Tool, Base, World werden die Grundachsen X,Y und Z, sowie die Drehachsen A,B und C angezeigt. Beim Verfahren werden in der Regel mehrere Achsen synchron bewegt.

7.2.3 Programm ausführen, stoppen und zurücksetzen:

M2/Tr.Bl.10/S59/122

- **Programm Aus – und Anwählen:**
  Beim Hochfahren der Steuerung werden alle Programme von der Festplatte geladen. Mit den Cursor – Tasten wird das gewünschte Programm angewählt.



8. Beispiele für Übungsprogramme

8.1 Stapeln von Würfeln

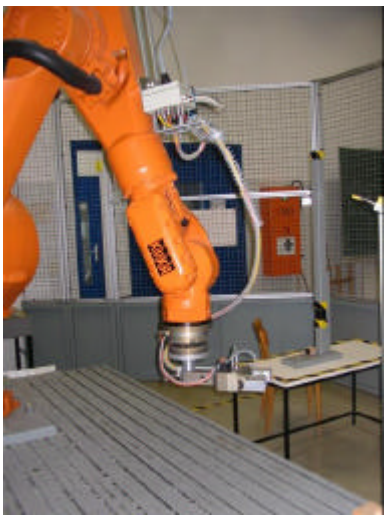
➢ **Programmbeschreibung:**

Der Roboter soll mit dem Greifer einen Würfel aus dem Speicher holen und auf dem vorgesehenen Ablagepunkt ablegen. Der nächste Würfel wird dann auf den bereits abgelegten positioniert. So werden insgesamt drei Würfel gestapelt.
Anschließend sollen sie wieder einzeln in den Speicher zurückgebracht werden.

➢ **Erklärung der einzelnen Programmschritte:**

- Am Beginn des Programms werden die Initialisierungsdateien aufgerufen.
- Home; der Roboter befindet sich in der Ausgangsstellung

- Der Punkt P1 wird angefahren. Dies geschieht mit dem Befehl PTP (point to point), dabei sind die Endpunkte definiert. Der Punkt P1 liegt ungefähr zwischen Abhol- und Ablagepunkt und ist ein Hilfspunkt zu weiteren Bahnbeschreibungen.



Die einzelnen Punkte werden im Teach in - Touch up Verfahren eingegeben. Werden sie an anderer Stelle im Programm wieder benötigt, können sie einfach wiederverwendet werden.

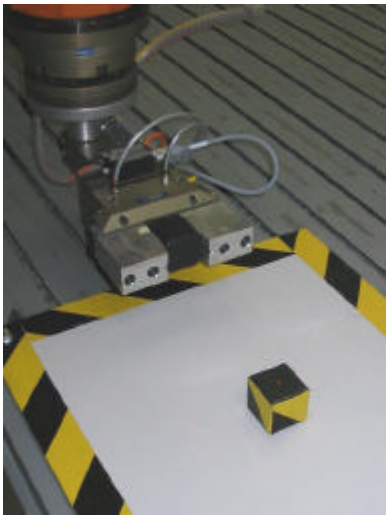
In den folgenden Programmschritten wird, wenn von Achsbewegungen auf bestimmte Punkte die Rede ist immer der Befehl PTP verwendet.

- Nun bewegt sich der Greifer weiter auf P2. Dieser Punkt ist ca. 20 cm vom eigentlichen Greifpunkt entfernt. Die X und Z- Koordinaten sind schon die gleichen wie am Greifpunkt an der Speicheröffnung.

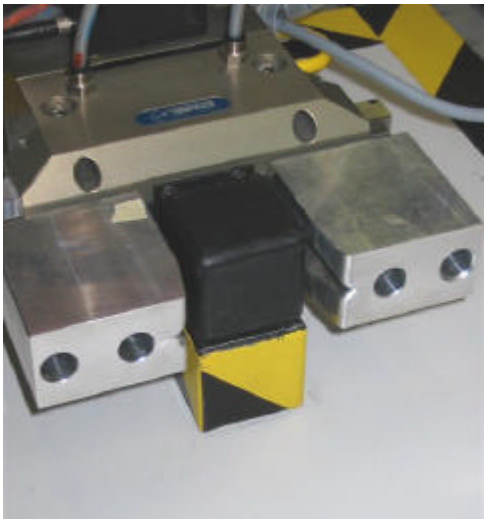
- Der Roboter fährt nun mit verminderter Geschwindigkeit (30%) P3 an. Ist dieser erreicht, wird der Greifer geschlossen.

- P4 besitzt die gleichen Koordinatenwerte wie P3, nur der Z- Wert ist um ca. 4mm größer. Dadurch kann der Würfel "sauber", und ohne auf der Grundfläche zu streifen, herausgezogen werden.
- Beim Herausziehen wird zunächst langsam auf P2 zurückgefahren.

- Beim Transport zum Ablagepunkt wird zuerst P1 angefahren, dann weiter auf P5. Das ist nun ein Punkt in ca. 20 cm Abstand vom richtigen Ablagepunkt.

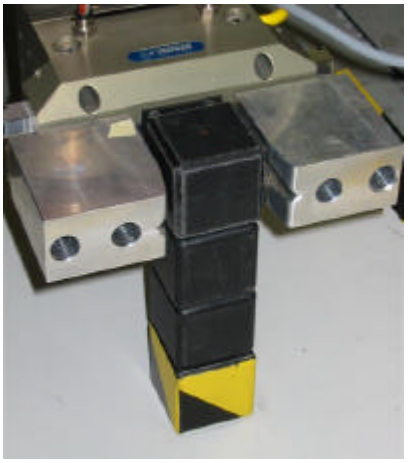


- Mit ca. 30% der Maximalgeschwindigkeit wird nun der "Vorpositionierpunkt" P6 angefahren. Dieser liegt ca. 5mm über der eigentlichen Ablagefläche und soll ein Schleifen verhindern. Die Koordinatenwerte in X und Y stimmen auch hier schon mit dem "Loslasspunkt" überein.
- Die Z Koordinate wird nun verkleinert, bis nur ein Lichtspalt übrigbleibt. Dieser Punkt ist P7 und beim Erreichen wird der Greifer geöffnet und der Würfel "fällt" ohne zu verrutschen auf die vorgesehene Stelle.

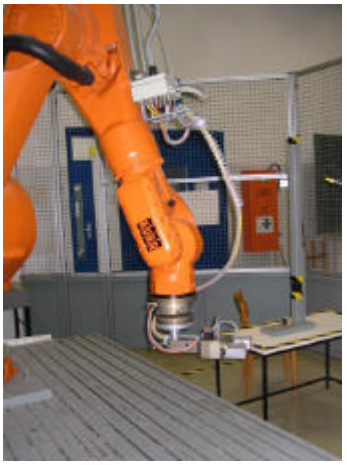


- Ist der Würfel nun abgelegt, muss der Roboter wieder auf P5 gerade zurückfahren, um den Würfel nicht umzuwerfen.
- Nun bewegt sich der Roboter wieder auf P1 und dann weiter auf P2, um den nächsten Würfel zu holen.
- Die folgenden Punkte werden nicht neu definiert (Teach in - Touch up nicht erforderlich!), da sie sowieso gleich bleiben. Die Schritte vom Anfahren über P1, Greifen des Würfels, Herausziehen, bis zur Rückkehr zum Punkt P1 sind bereites am Programmbeginn erklärt.
- Nach P1 wird nun nicht mehr P5 angefahren, sondern P8. Dieser liegt in Z ca. eine Würfelhöhe (5cm) über P5.

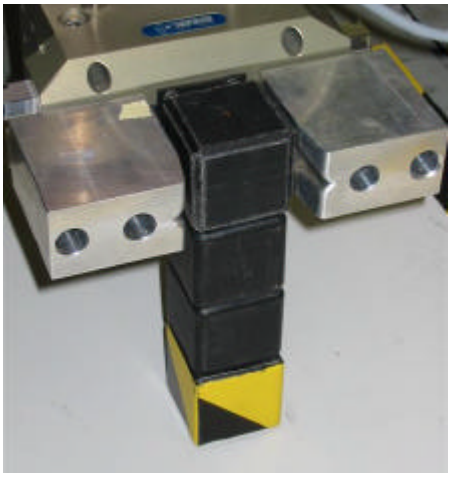
- Von hier wird nun über den bereits liegenden Würfel, auf P9 positioniert. P9 hat einen Sicherheitsabstand um den ersten Würfel nicht zu verschieben.
- Beim Bewegen auf P10 wird dann die "Resthöhe" wieder auf einen Lichtspalt verkleinert. Beim Erreichen von P10 wird auch der Greifer geöffnet und der Würfel abgelegt.
- Nun wird auf P8 gerade zurückgefahren und von dort weiter auf P1. Hier beginnt das oben beschriebene "Szenario" zum Würfelholen wieder.
- Kommt der Greifer nun mit dem dritten Würfel über P1 zurück, fährt er auf P11. Dieser ist ca. eine Würfelhöhe höher als P8.
- Nun wird der "Vorpositionierpunkt" P13 angefahren.
- Der Punkt mit der Lichtspalthöhe ist P12 und auch der Greiferöffnungspunkt.



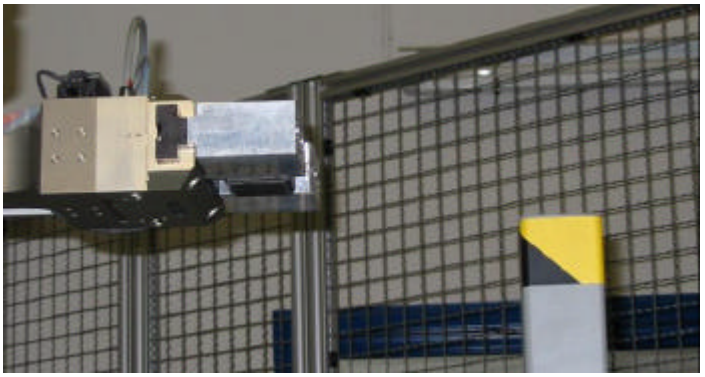
- Nun befinden sich 3 Würfel gestapelt am Ablagepunkt. Sie sollen nun wieder in den Speicher zurückbefördert werden. Dazu wird zuerst auf P11 zurückgefahren. Dann wird als "Abschluss" P1 angefahren. Der Stapelvorgang ist beendet.



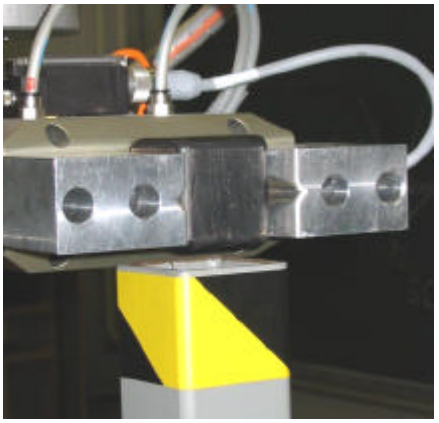
- Zum Abholen wird nun als "Vorpunkt" P11 angefahren.



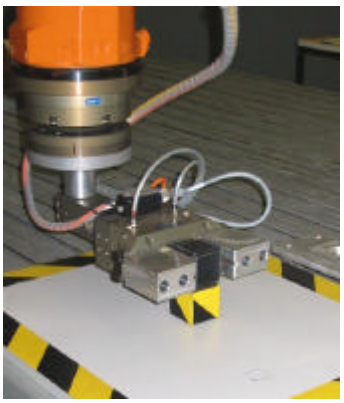
- Dann bewegt sich der Greifer auf P12, dem ehemaligen "Absenken auf Lichtspalt", wo er den Würfel greift.
- Dann fährt er auf den ebenfalls bereits in die Steuerung eingegebenen Punkt P13. Dieser liegt nur wenige mm über P12. So kann beim Wegfahren ein Umwerfen des Stapels ausgeschlossen werden.
- Nun kann aber direkt auf P1 gefahren werden. Von dort geht es weiter zum "Anfahrpunkt", zum Einwerfen in den Speicher.



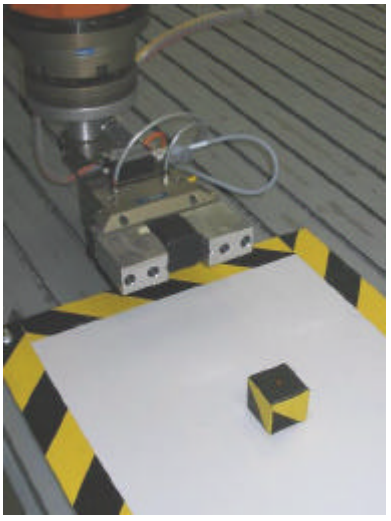
- P15 dient zur Vorpositionierung in den Koordinaten X und Y knapp über dem Speicherrohr. Dabei werden nur sehr kleine Geschwindigkeiten gefahren. (3 - 30%)



- Die Absenkung wenige mm in das Rohr soll ein Verkanten verhindern. Beim Erreichen von P16 wird der Würfel dann eingeworfen.
- Das Zurückfahren auf P14 ist notwendig, da dem Greifer sonst bei direktem Bewegen auf P1 das Speicherrohr im Weg steht.
- Nun wird eben über P1 auf P8 gefahren.
- Dann auf P10, wo der zweite Würfel gegriffen wird.
- Um ein Verschieben des ersten Würfels zu vermeiden wird auf P9, der einige mm über P10 liegt gefahren.
- Nun wird über P1 wieder zur Speichereinwurföffnung gefahren. Dabei bleiben die nötigen Punkte gleich und sind bereits oben erklärt.
- Nach dem Ende des Einwurfvorganges und dem Zurückfahren über P1 wird P5 angefahren.
- Dann wird der ehemalige "Absenkpunkt" P7 angefahren, wo der letzte Würfel gegriffen wird.



- Der Zwischenpunkt P6 verhindert ein mögliches Streifen auf der Unterlage.
- Hier muss der Greifer jedoch auf P5, den Anfahrpunkt zurückfahren, da sonst durch die geringe Höhe Kollisionsgefahr mit anderen Versuchsaufbauten besteht.



- Über P1 wird nun der oben beschriebene "Speichervorgang" durchgeführt.
- Nach dem Punkt 14 fährt der Roboter wieder in die Ausgangsstellung (home) zurück.

- **Kuka - Programm: Stapeln von Würfeln**

```
DEF  CHTA5_1 ( )
 INI
 BAS INI
 A20 INI
 A10 INI
 GRIPPER INI
 SPOT INI
 TOUCHSENSE INI
 USER INI
 PTP HOME  Vel= 100 % DEFAULT
 PTP P1 CONT Vel= 100 % PDAT1   // Zwischenpunkt
 PTP P2  Vel= 100 % PDAT2       // "Vorspeicherpkt."
 PTP P3  Vel= 30 % PDAT3        // Greifpunkt
 SET GRP 2  State= Greifen  GDAT2
 PTP P4  Vel= 10 % PDAT5        // Würfel anheben
 PTP P2  Vel= 60 % PDAT4
 PTP P1 CONT Vel= 100 % PDAT10
 PTP P5  Vel= 100 % PDAT7       // 1."Vorablagepkt."
 PTP P6  Vel= 30 % PDAT8        // Positionieren
 PTP P7  Vel= 10 % PDAT11       // Würfel absenken
 SET GRP 2  State= Loesen  GDAT3
 PTP P5  Vel= 60 % PDAT9        // Zurückfahren
 PTP P1 CONT Vel= 100 % PDAT6
 PTP P2  Vel= 100 % PDAT12      // Würfel 2 holen
 PTP P3  Vel= 30 % PDAT13
 SET GRP 2  State= Greifen  GDAT4
 PTP P4  Vel= 10 % PDAT14
 PTP P2  Vel= 60 % PDAT15
 PTP P1 CONT Vel= 100 % PDAT16
 PTP P8  Vel= 100 % PDAT17      // 2."Vorablagepkt."
 PTP P9  Vel= 30 % PDAT18       // Positionieren
 PTP P10  Vel= 10 % PDAT19      //2. Würfel absenken
 SET GRP 2  State= Loesen  GDAT5
 PTP P8  Vel= 60 % PDAT20       // Zurückfahren
 PTP P1 CONT Vel= 100 % PDAT21
 PTP P2  Vel= 100 % PDAT22      // Würfel 3 holen
 PTP P3  Vel= 30 % PDAT23
 SET GRP 2  State= Greifen  GDAT6
 PTP P4  Vel= 10 % PDAT24
 PTP p2  Vel= 60 % PDAT25
 PTP p1 CONT Vel= 100 % PDAT26
```

```
PTP p11  Vel= 100 % PDAT27     // 3."Vorablagepkt."
PTP p13  Vel= 30 % PDAT32      // Positionieren
PTP p12  Vel= 10 % PDAT28      //3. Würfel absenken
SET GRP 2  State= Loesen  GDAT7
PTP p11  Vel= 60 % PDAT29
PTP p1  Vel= 100 % PDAT30      // Ende der Stapel-
PTP p11  Vel= 100 % PDAT31     // sequenz
PTP p12  Vel= 30 % PDAT33
SET GRP 2  State= Greifen  GDAT13
PTP p13  Vel= 30 % PDAT34      //3.Würfel anheben
PTP p1 CONT Vel= 100 % PDAT35
PTP p14  Vel= 100 % PDAT36     //"Vorspeicherpkt."
PTP p15  Vel= 30 % PDAT37      // Positionieren
PTP p16  Vel= 10 % PDAT38      // Absenken
SET GRP 2  State= Loesen  GDAT8
PTP p14  Vel= 60 % PDAT39      //Zurückfahren
PTP p1 CONT Vel= 100 % PDAT40
PTP p8  Vel= 100 % PDAT41      // "Vorabholpkt.2"
PTP p10  Vel= 30 % PDAT42      // Greifpunkt
SET GRP 2  State= Greifen  GDAT9
PTP p9  Vel= 30 % PDAT43       // Anheben
PTP p1 CONT Vel= 100 % PDAT44
PTP p14  Vel= 100 % PDAT45     //"Speicher-"
PTP p15  Vel= 30 % PDAT46      // Vorgang
PTP p16  Vel= 10 % PDAT47
SET GRP 2  State= Loesen  GDAT10
PTP p14  Vel= 60 % PDAT48
PTP p1 CONT Vel= 100 % PDAT49
PTP p5  Vel= 100 % PDAT50      //"Vorabholpkt.1Wfl"
PTP p7  Vel= 30 % PDAT51       //Greifen
SET GRP 2  State= Greifen  GDAT11
PTP p6  Vel= 30 % PDAT52       //Anheben
PTP p5  Vel= 60 % PDAT58       // Zurückfahren
PTP p1 CONT Vel= 100 % PDAT53
PTP p14  Vel= 100 % PDAT54     // "Einwerfen"
PTP p15  Vel= 30 % PDAT55
PTP p16  Vel= 10 % PDAT56
SET GRP 2  State= Loesen  GDAT12
PTP p14  Vel= 60 % PDAT57
PTP HOME  Vel= 100 % DEFAULT

END
```

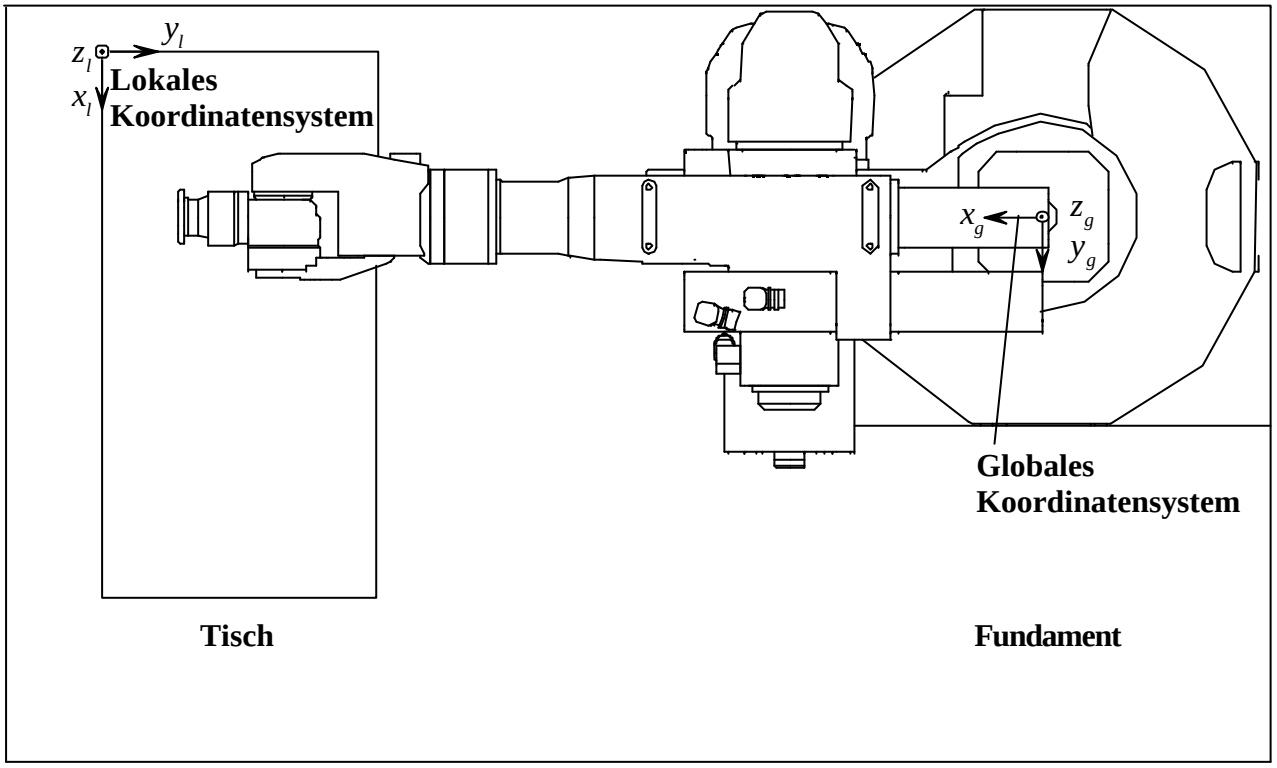
8.2 Programmieren eines Linienzuges; Rennwagen

➢ Programmbeschreibung:

Der Roboter sollte mit einem im Greifer eingespannten Stift auf einem Tisch einen Linienzug zeichnen. In unserem Fall zeichneten wir ein Rennauto. Die Punkte des Fahrzeuges wurden durch lineare Relativkoordinaten (LIN_REL) definiert und von einem externen Unterprogramm aufgerufen.
Insgesamt sollte das Auto dreimal gezeichnet werden. Dafür verwendeten wir eine FOR-Schleife.

➢ Werkzeug- und Basisvermessung

Bei der Werkzeugvermessung wird der Bezugspunkt, auf den sich die Steuerung bezieht genau vermessen (ansonsten würde der Greifer als Werkzeugnullpunkt angesehen werden). Es wird die 4 Punkt Methode verwendet (Siehe 9.1.1).
Zur Werkstückvermessung (in unserem Fall der Tisch) wird die 3-Punkt Basisvermessung verwendet (Siehe 10.1).



➢ Erklärung der einzelnen Programmschritte:

- Definition des Hauptprogramms *CHTA5_G1*

- Aufrufen des externen Unterprogramms *GUSTAVF2*, welches den Liniezug für das Rennauto enthält

- Definition der 3 Koordinatenvariablen *ABSTAND [ ]* für die Anfangspunkte der 3 Autos
- Definition der Integervariable *I*

- Festlegung der 3 Startpunkte *ABSTAND [1 bis 3]* für die Rennautos

- Nun werden die Initialisierungsdateien aufgerufen
- *HOME*; der Roboter befindet sich in der Ausgangsstellung

- Der Punkt *P1* wird angefahren. Dies geschieht mit dem Befehl PTP (point to point), Mit P1 wird der Eckpunkt des Tisches (Ausgangspunkt) definiert.

- Aufruf der FOR- Schleife die dreimal (für 3 Rennwagen) durchlaufen werden soll
- Inder Schleife wird jeweils das Unterprogramm *GUSTAVF 2* mit dem Linienzug für das Fahrzeug aufgerufen
- Mit *ENDFOR* wird die Schleife beendet
- Nun fährt der Roboter wieder in seine Ausgangsstellung *HOME*

- *END*; Ende des Programms

Hauptprogramm:

```
FRAME ABSTAND[3]
DECL INT I

; ----------------------  Initialisierung

ABSTAND[1]={X 50,Y 50,Z 20}
ABSTAND[2]={X 450,Y 50,Z 20}
ABSTAND[3]={X 850,Y 50,Z 20}
```

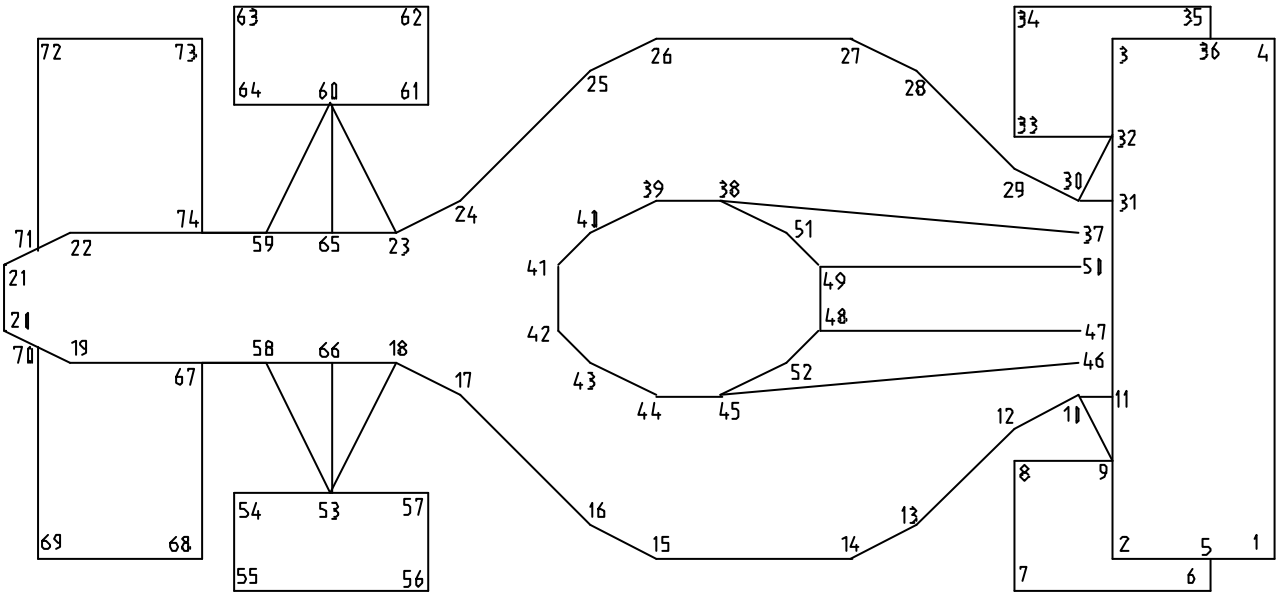
```
LIN  {X 0,Y 0,Z 20,A -88.74,B 89.82,C 1.26}

; ----------------------  Schleifen Beginn
FOR I=1 TO  3

LIN  ABSTAND[I]
GUSTAVF2 ( )
;                          Aufruf Unterprogramm
ENDFOR
; ----------------------  Schleifen Ende

END
```

Unterprogramm:

```
DEF  GUSTAVF2 ( )


LIN_REL  {X 0,Y 0,Z 0,A 0,B 0,C 0}


LIN_REL  {X 20,Y 10,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z -30,A 0,B 0,C 0}
LIN_REL  {X 0,Y 50,Z 0,A 0,B 0,C 0}
LIN_REL  {X 160,Y 0,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y -50,Z 0,A 0,B 0,C 0}
LIN_REL  {X -160,Y 0,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z 30,A 0,B 0,C 0}


LIN_REL  {X 0,Y 20,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z -30,A 0,B 0,C 0}
LIN_REL  {X -10,Y 0,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 60,Z 0,A 0,B 0,C 0}
LIN_REL  {X 40,Y 0,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y -30,Z 0,A 0,B 0,C 0}
LIN_REL  {X 20,Y 10,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z 30,A 0,B 0,C 0}


LIN_REL  {X 0,Y -10,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z -30,A 0,B 0,C 0}
LIN_REL  {X 0,Y 10,Z 0,A 0,B 0,C 0}
LIN_REL  {X -10,Y 20,Z 0,A 0,B 0,C 0}
LIN_REL  {X -30,Y 30,Z 0,A 0,B 0,C 0}
LIN_REL  {X -10,Y 20,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 60,Z 0,A 0,B 0,C 0}
LIN_REL  {X 10,Y 20,Z 0,A 0,B 0,C 0}
LIN_REL  {X 40,Y 40,Z 0,A 0,B 0,C 0}
LIN_REL  {X 10,Y 20,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 100,Z 0,A 0,B 0,C 0}
LIN_REL  {X 10,Y 20,Z 0,A 0,B 0,C 0}
LIN_REL  {X 20,Y 0,Z 0,A 0,B 0,C 0}
LIN_REL  {X 10,Y -20,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y -100,Z 0,A 0,B 0,C 0}
LIN_REL  {X 10,Y -20,Z 0,A 0,B 0,C 0}
LIN_REL  {X 40,Y -40,Z 0,A 0,B 0,C 0}
```

```
LIN_REL  {X 10,Y -20,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y -60,Z 0,A 0,B 0,C 0}
LIN_REL  {X -10,Y -20,Z 0,A 0,B 0,C 0}
LIN_REL  {X -30,Y -30,Z 0,A 0,B 0,C 0}
LIN_REL  {X -10,Y -20,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y -10,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z 30,A 0,B 0,C 0}


LIN_REL  {X 0,Y 10,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z -30,A 0,B 0,C 0}
LIN_REL  {X 20,Y -10,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 30,Z 0,A 0,B 0,C 0}
LIN_REL  {X 40,Y 0,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y -60,Z 0,A 0,B 0,C 0}
LIN_REL  {X -10,Y 0,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z 30,A 0,B 0,C 0}


LIN_REL  {X -60,Y 40,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z -30,A 0,B 0,C 0}
LIN_REL  {X 10,Y 110,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 20,Z 0,A 0,B 0,C 0}
LIN_REL  {X -10,Y 20,Z 0,A 0,B 0,C 0}
LIN_REL  {X -10,Y 10,Z 0,A 0,B 0,C 0}
LIN_REL  {X -20,Y 0,Z 0,A 0,B 0,C 0}
LIN_REL  {X -10,Y -10,Z 0,A 0,B 0,C 0}
LIN_REL  {X -10,Y -20,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y -20,Z 0,A 0,B 0,C 0}
LIN_REL  {X 10,Y -110,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z 30,A 0,B 0,C 0}

LIN_REL  {X 10,Y 0,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z -30,A 0,B 0,C 0}
LIN_REL  {X 0,Y 80,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z 30,A 0,B 0,C 0}
LIN_REL  {X 20,Y 0,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z -30,A 0,B 0,C 0}
LIN_REL  {X 0,Y -80,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z 30,A 0,B 0,C 0}

LIN_REL  {X 20,Y 110,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z -30,A 0,B 0,C 0}
LIN_REL  {X -10,Y -20,Z 0,A 0,B 0,C 0}
LIN_REL  {X -10,Y -10,Z 0,A 0,B 0,C 0}
LIN_REL  {X -20,Y 0,Z 0,A 0,B 0,C 0}
LIN_REL  {X -10,Y 10,Z 0,A 0,B 0,C 0}
LIN_REL  {X -10,Y 20,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z 30,A 0,B 0,C 0}


LIN_REL  {X 10,Y 100,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z -30,A 0,B 0,C 0}
LIN_REL  {X -40,Y 20,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 30,Z 0,A 0,B 0,C 0}
LIN_REL  {X -30,Y 0,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y -60,Z 0,A 0,B 0,C 0}
LIN_REL  {X 30,Y 0,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 30,Z 0,A 0,B 0,C 0}
LIN_REL  {X 40,Y 20,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z 30,A 0,B 0,C 0}
```

```
LIN_REL  {X 40,Y 0,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z -30,A 0,B 0,C 0}
LIN_REL  {X 40,Y -20,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y -30,Z 0,A 0,B 0,C 0}
LIN_REL  {X 30,Y 0,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 60,Z 0,A 0,B 0,C 0}
LIN_REL  {X -30,Y 0,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y -30,Z 0,A 0,B 0,C 0}
LIN_REL  {X -40,Y -20,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z 30,A 0,B 0,C 0}

LIN_REL  {X 40,Y 20,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z -30,A 0,B 0,C 0}
LIN_REL  {X -40,Y 0,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z 30,A 0,B 0,C 0}
LIN_REL  {X -40,Y 0,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z -30,A 0,B 0,C 0}
LIN_REL  {X -40,Y 0,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z 30,A 0,B 0,C 0}


LIN_REL  {X 40,Y 40,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z -30,A 0,B 0,C 0}
LIN_REL  {X -60,Y 0,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 50,Z 0,A 0,B 0,C 0}
LIN_REL  {X 65,Y 0,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z 30,A 0,B 0,C 0}
LIN_REL  {X 30,Y 0,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z -30,A 0,B 0,C 0}
LIN_REL  {X 65,Y 0,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y -50,Z 0,A 0,B 0,C 0}
LIN_REL  {X -60,Y 0,Z 0,A 0,B 0,C 0}
LIN_REL  {X 0,Y 0,Z 30,A 0,B 0,C 0}

END
```



1: X 20, Y 10
2: X 20, Y 50
3: X 180, Y 50
4: X 180, Y 10
5: X 20, Y 30
6: X 10, Y 30
7: X 10, Y 90
8: X 50, Y 90
9: X 50, Y 60
10: X 70, Y 50
11: X 70, Y 40
12: X 60, Y 70
13: X 30, Y 100
14: X 20, Y 120
15: X 20, Y 180
16: X 30, Y 200
17: X 70, Y 240
18: X 80, Y 260
19: X 80, Y 360
20: X 90, Y 380
21: X 110, Y 380
22: X 120, Y 360
23: X 120, Y 260
24: X 130, Y 240
25: X 170, Y 200
26: X 180, Y 180
27: X 180, Y 120
28: X 170, Y 100
29: X 140, Y 70
30: X 130, Y 50
31: X 130, Y 40
32: X 150, Y 40
33: X 150, Y 70
34: X 190, Y 70
35: X 190, Y 130
36: X 180, Y 130
37: X 120, Y 50
38: X 130, Y 160
39: X 130, Y 180
40: X 120, Y 200
41: X 110, Y 210
42: X 90, Y 210
43: X 80, Y 200
44: X 70, Y 180
45: X 70, Y 160
46: X 80, Y 50
47: X 90, Y 50
48: X 90, Y 130
49: X 110, Y 130
50: X 110, Y 50
51: X 120, Y 140
52: X 80, Y 140
53: X 40, Y 280
54: X 40, Y 310
55: X 10, Y 310
56: X 10, Y 250
57: X 40, Y 250
58: X 80, Y 300
59: X 120, Y 300
60: X 160, Y 280
61: X 160, Y 250
62: X 190, Y 250
63: X 190, Y 310
64: X 160, Y 310
65: X 120, Y 280
66: X 80, Y 280
67: X 80, Y 320
68: X 20, Y 320
69: X 20, Y 370
70: X 85, Y 370
71: X 115, Y 370
72: X 180, Y 370
73: X 180, Y 320
74: X 120, Y 320

8.2.1 Programmierbefehle; Schleifen

➢ **Allgemein**:

Mit Schleifen ist die wiederholte Abarbeitung von Anweisungen gewährleistet. Es gibt verschiedene Schleifentypen die nun näher erklärt werden.
(M3/Tr.Bl.0/S124-129/194)

- **Zählschleife FOR:**

Strukturaufbau:

FOR *Zähler-Start* **TO** *Ende* **STEP** *Schrittweite*
Anweisung
ENDFOR

Bei der Zählschleife wird der Startwert, Endwert und die Schrittweite vorgegeben. Die Schleife führt die nachstehenden Anweisungen solange aus, bis der Endwert erreicht ist. Am Ende der FOR-Schleife muss es immer die Anweisung ENDFOR geben, damit die Anweisungen hinter ENDFOR nach dem Schleifendurchlauf fortgesetzt werden.

- **Abweisende Schleife WHILE:**

Strukturaufbau:

WHILE *Ausführbedingung*
Anweisung
ENDWHILE

Bei der WHILE- Schleife erfolgt die Abfrage am Anfang, d.h. wenn die Ausführbedingung nicht erfüllt ist erfolgt kein einziger Schleifendurchgang (abweisende Schleife). Wenn die Ausführbedingung erfüllt ist, werden die nachstehenden Anweisungen ausgeführt. Am Ende der WHILE- Schleife muss immer die Anweisung ENDWHILE stehen, damit die nachfolgenden Anweisungen bearbeitet werden.

- **Nicht abweisende Schleife REPEAT:**

Strukturaufbau:

REPEAT
Anweisung
UNTIL *Abbruchbedingung*

Da die Abfrage erst am Ende erfolgt, wird die REPEAT- Schleife mindestens einmal durchlaufen, auch wenn die Abbruchbedingung schon vor Schleifenanfang erfüllt ist.

- **Endlosschleife LOOP:**

Strukturaufbau:

LOOP
Anweisung
ENDLOOP

Die Schleife wird solange durchlaufen bis eine EXIT- Anweisung auftritt und den Schleifendurchlauf beendet.

Realisierung einer EXIT- Bedingung:

LOOP
Anweisung
IF Anweisung erfüllt
EXIT
ENDIF
ENDLOOP

Die Endlosschleife führt eine Anweisung solange aus, bis eine IF- Anweisung erfüllt ist, und somit eine EXIT- Anweisung ausgeführt wird. Dies führt zum Abbruch der Endlosschleife.

➢ Programmierbefehle einer FOR- Schleife anhand des Beispiels Rennwagen

...

FRAME:
Definierung einer Frame-Variable,die die angegebenen Koordinaten X,Y,Z und A,B,C enthält. Ein Punkt im Raum ist daher nach Lage und Orientierung eindeutig definiert

DECL INT I:
Deklarieren von Variablen. Die Deklaration beginnt mit dem Schlüsselwort DECL, gefolgt vom Datentyp (hier INT für Integer, heißt ganzzahlig) und der Variable (in unserem Fall I),die diesen Datentyp enthalten soll

Definieren der Startpunkte für die Wiederholung der Rennautos

...

Zählschleife **FOR**. Von I=1 bis I=3

Ausführen der Linearbewegungen aus dem Unterprogramm ausgehend von 3 verschiedenen Startpunkten.
Beim ersten Durchlauf wird I in **ABSTAND[I]** auf 1 gesetzt und dadurch der erste Startpunkt verwendet.
Beim zweiten Durchlauf der Schleife wird i in **ABSTAND[I]** auf 2 gesetzt und daher zweite Startpunkt verwendet. ...
Durch diese **FOR**- Schleife wird praktisch der Rennwagen 3mal dupliziert.

Ende der **FOR**- Schleife

9. Werkzeugvermessung

9.1 Methoden zur Positionsbestimmung:

Position des TCP gegenüber dem Roboterflansch:

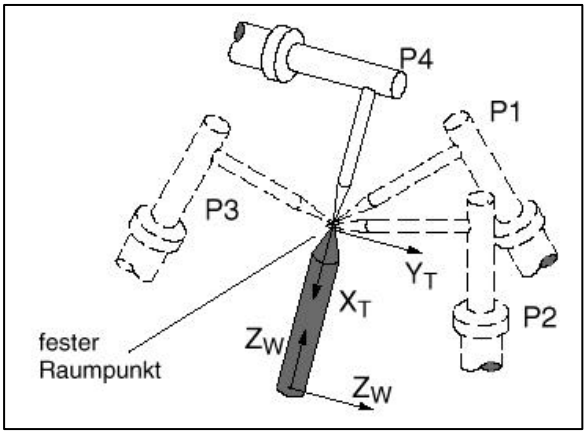
9.1.1 XYZ-4 Punkt

Dabei wird das Werkzeug mit seinem Bezugspunkt(TCP) aus 4 verschiedenen Richtungen auf einen Referenzpunkt gefahren.

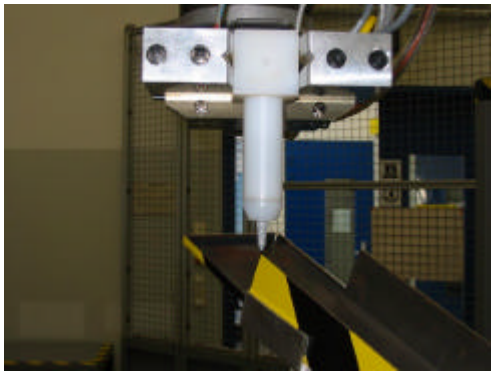
Ausführung:

- Werkzeug montieren
- Referenzpunkt wählen
- Menükey Inbetriebn. , „Vermessen“, „Werkzeug“, Untermenü „XYZ-4Punkt“
- Die gewünschte Werkzeugnummer wählen (In unserem Fall Nr. 15)
- Werkz. Ok drücken
- Nun das Werkzeug aus 4 Richtungen mit dem Bezugspunkt auf den (Bild1.1) Referenzpunkt anfahren. Nach jedem Anfahren Punkt Ok drücken.
  (Bei unserer Übung „Schriftzug“ wurde als Bezugspunkt die Rutsche der Übung „Lagermontage“ gewählt)
- Am Ende der Vermessung Sichern drücken.

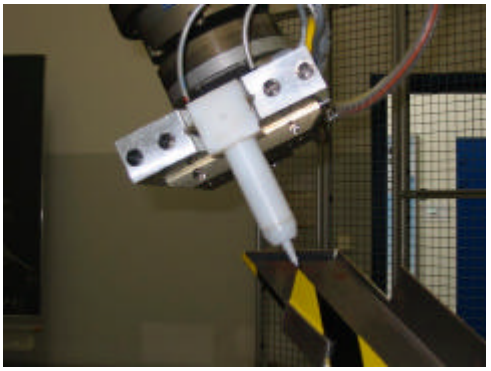
Bild 1.1



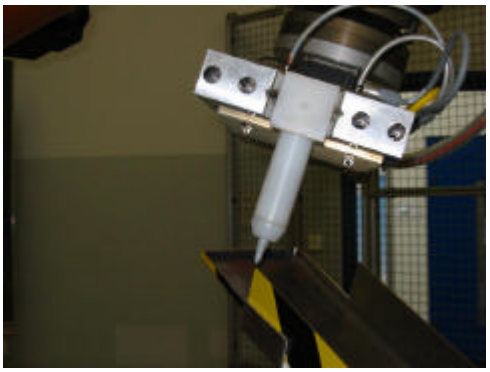
P1



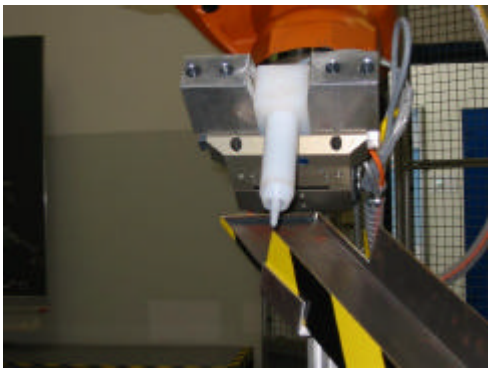
P2



P3



P4



9.1.2 XYZ-Referenz

Bei dieser Methode werden die Daten des zu vermessenden Werkzeugs durch Vergleich mit einem bekannten Werkzeug ermittelt.
Es wird mit einem bekannten Werkzeug aus einer beliebigen Richtung ein Referenzpunkt angefahren und danach wird derselbe Punkt mit dem unbekannten Werkzeug aus einer anderen Richtung angefahren.

Ausführung:

- Bekanntes Werkzeug montieren
- Menükey „Inbetriebn.“, „Vermessen“, „Werkzeug“, Untermenü „XYZ-Refrenz“
- Werkzeugnummer wählen (bei der Übung „Schriftzug“ Nr. 15)
- Werkz. Ok drücken
- XYZ-Koordinaten des bekannten Werkzeugs eingeben
- Daten Ok drücken
- Referenzpunkt anfahren (Bild 2.1)
- Punkt Ok drücken
- Freifahren und bekanntes durch unbekanntes Werkzeug wechseln
- Referenzpunkt aus beliebiger Richtung anfahren (Bild 2.2)
- Punkt Ok drücken

- Sichern drücken

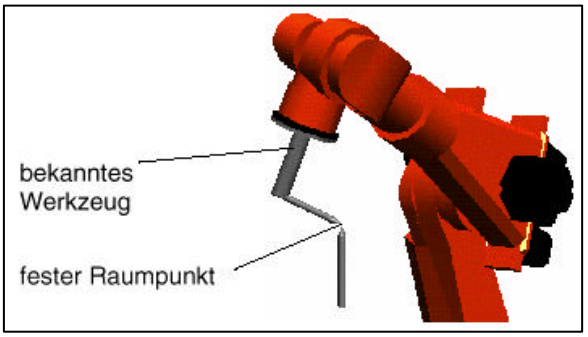


Bild 2.1

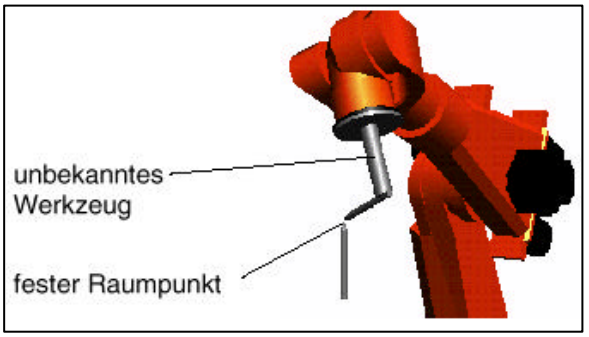


Bild 2.2

9.2 Methoden zur Orientierungsbestimmung:

9.2.1 ABC-2-Punkt

Dieses Verfahren wird zur Positionierung und Führung verwendet, wenn eine genaue Orientierung der 3 Werkzeugachsen benötigt wird.
Zuerst wird ein beliebiger Referenzpunkt mit dem TCP des Werkzeugs angefahren. Dann wird der Referenzpunkt mit einem Punkt am Werkzeug angefahren der dem TCP des Werkzeugs gegen der Arbeitsrichtung gegenüber liegt. Um die YZ-Ebene zu definieren wird das Werkzeug so verfahren, dass es mit pos. Y-Wert in der zukünftigen XY-Ebene liegt.

Ausführung:

- Werkzeug montieren
- Menükey „Inbetriebn.“, „Vermessen“, „Werkzeug“, Untermenü „ABC-2-Punkt“
- Werkzeugnummer wählen
- Werkz. Ok drücken
- Mit TCP Referenzpunkt anfahren (Bild 3.1)
- Punkt Ok drücken
- Werkzeug freifahren
- Mit einem Punkt am Werkzeug der dem TCP entgegen der Arbeitsrichtung gegenüber liegt anfahren (Bild 3.2)
- Punkt Ok drücken
- Werkzeug so verfahren dass der Referenzpunkt mit pos. Y-Wert in der XY-Ebene liegt (Bild 3.3)
- Punkt Ok drücken
- Sichern drücken

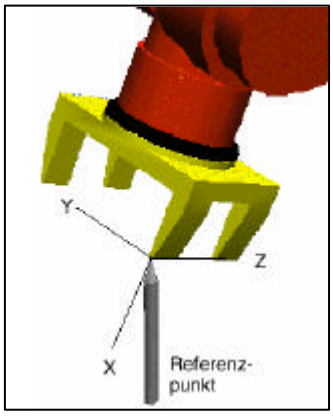


Bild 3.1

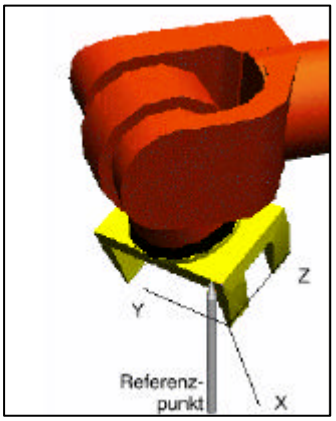


Bild 3.2

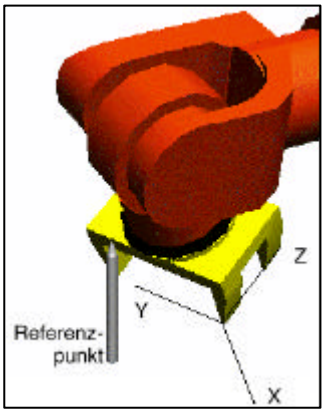


Bild 3.3

9.2.2 ABC-World (5D)

Wird verwendet wenn nur die Arbeitsrichtung zur Positionierung und Führung benötigt wird.

Ausführung:

- Menükey „Inbetriebn.“, „Vermessen“, „Werkzeug“, Untermenü „ABC-2-Punkt“
-
- Werkzeug montieren
- Werkzeugnummer wählen
- Werkz. Ok drücken
- 5D wählen
- Arbeitsrichtung einstellen
- Daten Ok drücken
- Werkzeug nach Abbildung 4.1 einstellen
- Punkt Ok drücken
- Sichern drücken

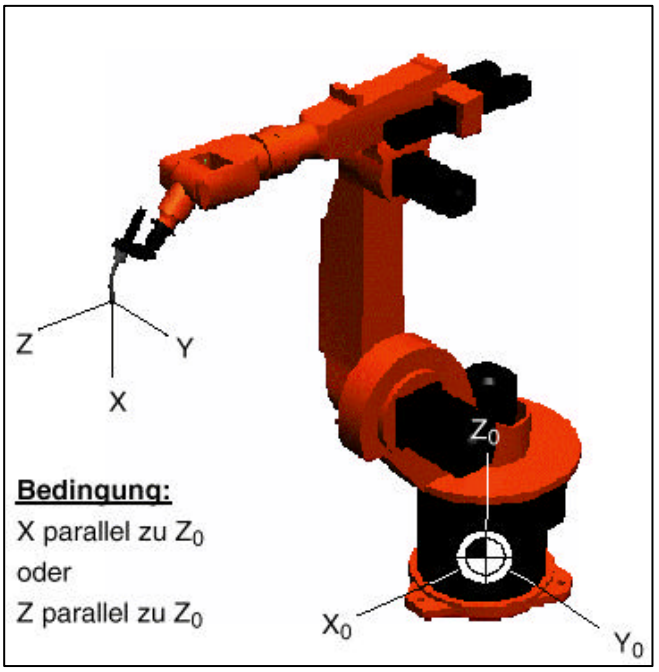


Bild 4.1

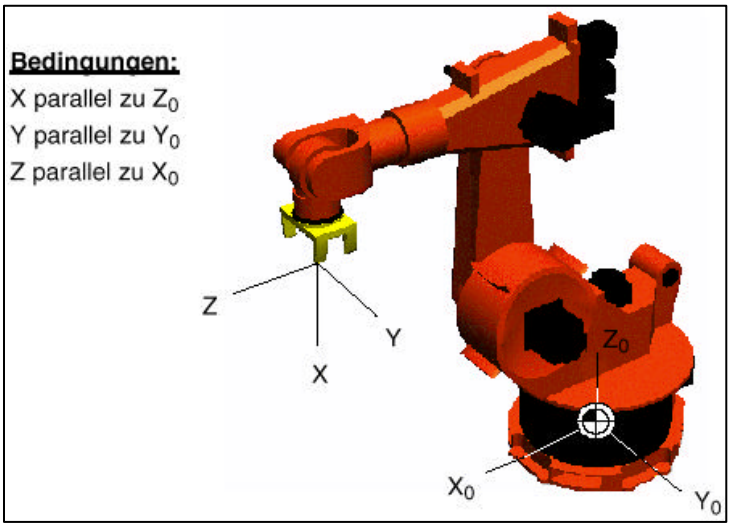
9.2.3. ABC-World(6D)

Wird verwendet wenn zur Führung und Orientierung eines Werkzeugs alle 3 Koordinaten benötigt werden.

Ausführung:

- Analog zur 5D-Methode bis Schritt 5
- 6D wählen
- Arbeitsrichtung eingeben
- Daten Ok drücken
- Werkzeug nach Abbildung 5.1 orientieren
- Punkt Ok drücken
- Sichern drücken

Bild 5.1



9.2.3 Numerische Eingabe

Ist ein Werkzeug bekannt so werden dessen Abmessungen sowie Winkelstellung eingegeben. (Werteingabe nur für feststehende Werkzeuge).

Ausführung:

- Menükey „Inbetriebn.“, „Vermessen“, „Werkzeug“, Untermenü „Num. Eingabe“
- Werkzeug wählen
- Werkz. Ok
- Daten des Werkzeugs bearbeiten (siehe Bilder 6.1 und 6.2)
- Daten Ok drücken
- Sichern drücken

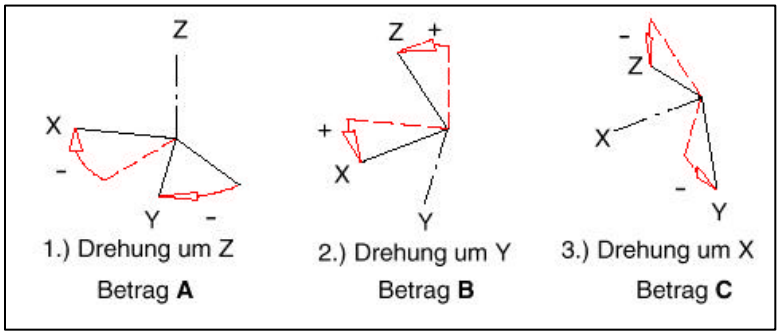


Bild 6.1

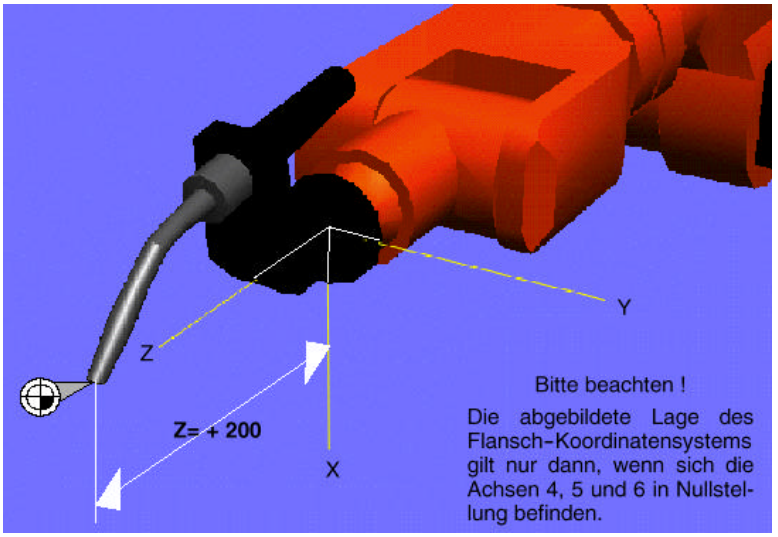


Bild 6.2

9.3 Werkzeuglastdaten:

Die jeweiligen Lastdaten der einzelnen Werkzeuge werden verwendet, um das maximal mögliche Anfahrmoment bei der Achsenbeschleunigung optimal nutzen zu können.

Ausführung:

- Menükey „Inbetriebn.“, „Vermessen“, „Werkzeug“, Untermenü „Werkzeuglastdaten“
- Werkzeugnummer wählen
- Werkz. Ok drücken
- Daten eingeben

XYZ= Entfernung des Werkzeugsschwerpunktes zum Flansch-Ursprung (Bild 7.1)
ABC= Verdrehung der Hauptträgheitsachsen des Werkzeugs gegenüber dem Roboterflansch-Koordinatensystems (Bild 7.2)
JX, JY, JZ= Massenträgheitsmoment um die Hauptträgheitsachsen des Werkstücks (Bild 7

- Default drücken
- Daten Ok drücken
- Sichern drücken

Bild 7.1

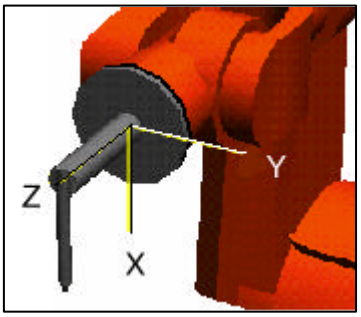
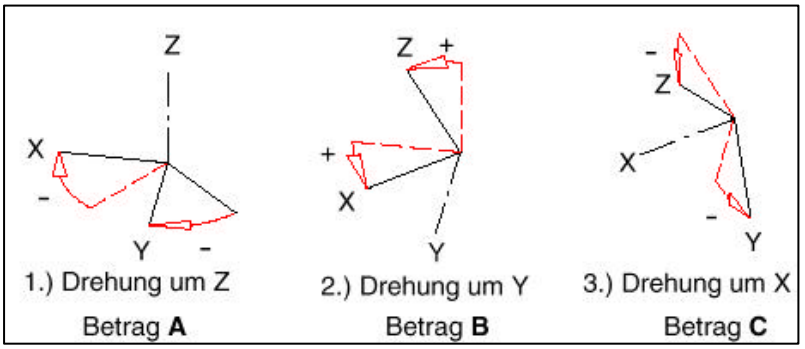


Bild 7.2



10. Basisvermessung

Soll ein neues Werkstück mit dem Kukaroboter bearbeitet werden, so muss dessen Basispunkt sowie Abmessungen der Steuerung bekannt gegeben werden.

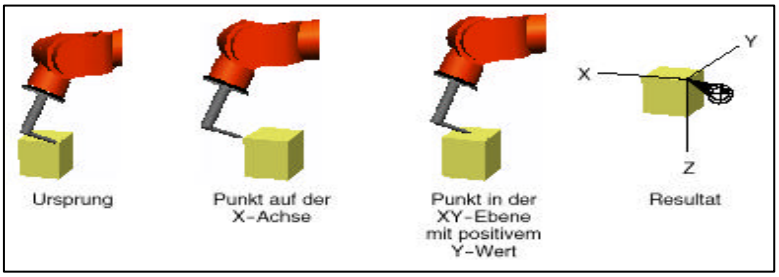
10.1 3.Punkt

Mit dieser Methode wird der Bezugspunkt eines Werkstückes ermittelt.
Dies geschieht durch Anfahren und Speichern dreier spezifischer Punkte mit einem bekannten Werkzeug. (Bild 8.1)

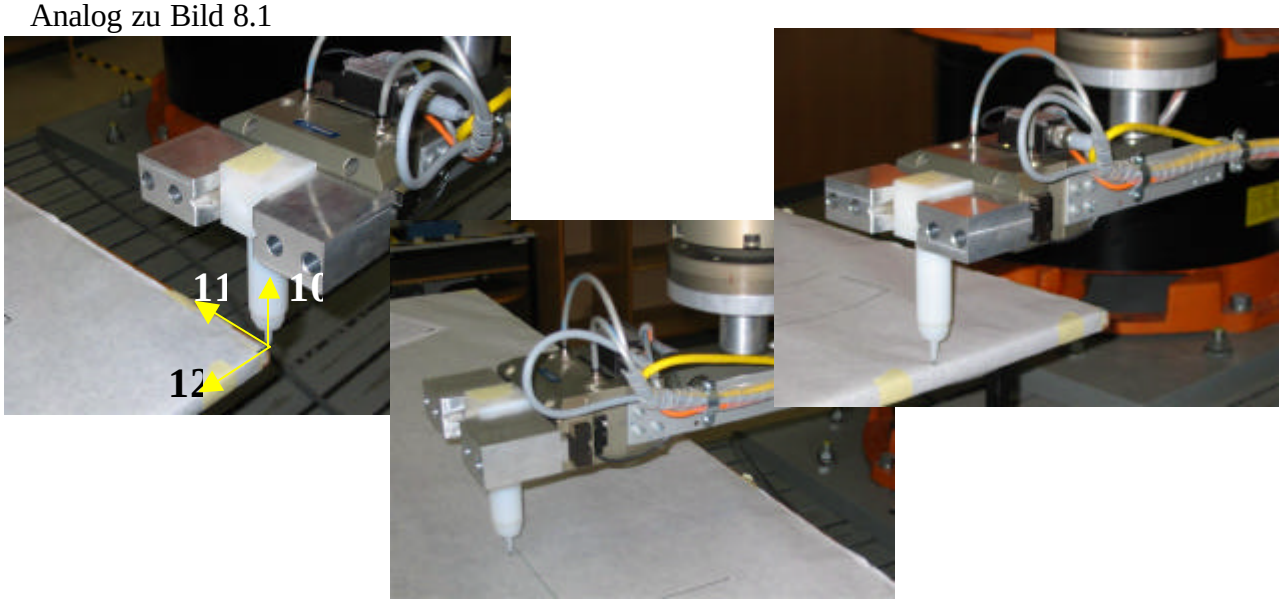
Ausführung:

- Bekanntes Werkzeug montieren (Bei Beispiel „Schriftzug“ Nr. 15)
- Menükey „ Inbetriebn. “, „Vermessen“, „Basis“, Untermenü „3.Punkt“
- Basiskoordinatensystem auswählen
- Basis Ok drücken
- Referenzwerkzeug auswählen (Nr. 15)
- Werkz. Ok drücken
- Anfahren des Ursprungs des zukünftigen Koordinatensystems (Bild 8.1)
- Punkt Ok drücken
- Der Steuerung durch Anfahren eines Punktes die X-Achse anzeigen (Bild 8.1)
- Punkt Ok drücken.
  Achtung: Punkt muss mindestens 30 mm vom Ursprung entfernt sein
- Der Steuerung die Orientierung der XY-Achse anzeigen (Bild 8.1)
- Punkt Ok drücken
- Sichern drücken

Bild 8.1



Analog zu Bild 8.1



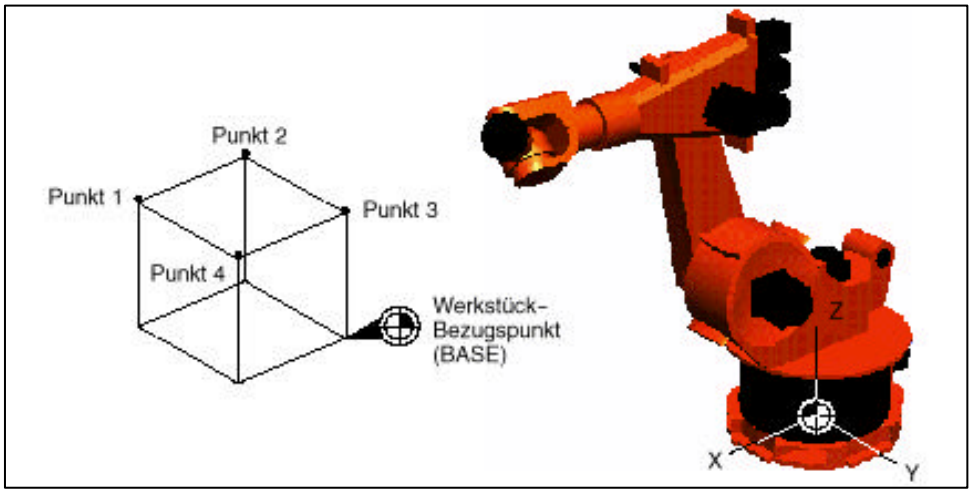
10.2 Indirekt

Wird dann verwendet wenn der Werkstücksbezugspunkt nicht im Arbeitsbereich des Roboters liegt.

<u>Ausführung:</u>

- bekanntes Werkzeug montieren
- Menükey „Inbetriebn.“, „Vermessen“, „Basis“, Untermenü „Punkt 3“
- Werkstücknummer wählen
- Base Ok drücken
- Werkzeug wählen
- Werkz. Ok drücken
- Geben sie die XY und Z-Abstände eines bekannten Punktes vom Basispunkt ein (Bild 9.1)
- Angegebenen Punkt anfahren (Bild 9.1)
- Punkt Ok drücken
- Die letzten 3 Punkte 3mal wiederholen
- Sichern drücken

Bild 9.1



10.3 Numerisch

Manuelle Eingabe eines Bezugspunktes

Ausführung:

- Menükey „Inbetriebnahme“, „Vermessen“, „Basis“,
Untermenü „Numerisch“
- Koordinatensystem wählen
- Base Ok drücken
- Werte Eingeben
X, Y, Z Entfernung zw. Ursprung des Weltkoordinatensystem und dem Werkstück-Bezugspunkt in Bezug auf das Weltkoordinatensystems (Bild 10.1)
A, B, C Verdrehung des Werkstückskoordinatensystems gegenüber dem Weltkoordinatensystems (Bild 10.2)
- Daten Ok drücken
- Sichern drücken

Bild 10.1

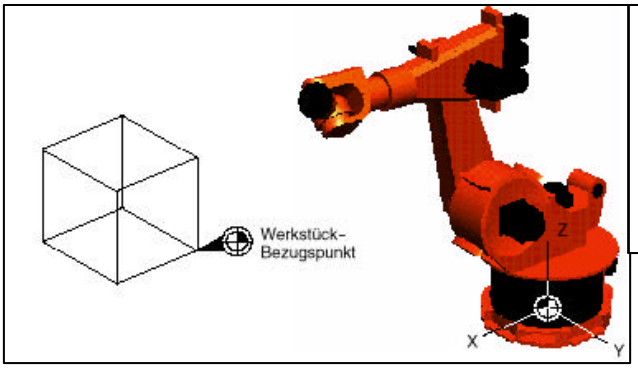
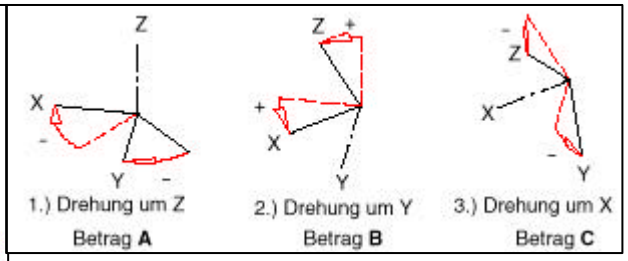


Bild 10.2



11. Unterprogramme und Funktionen

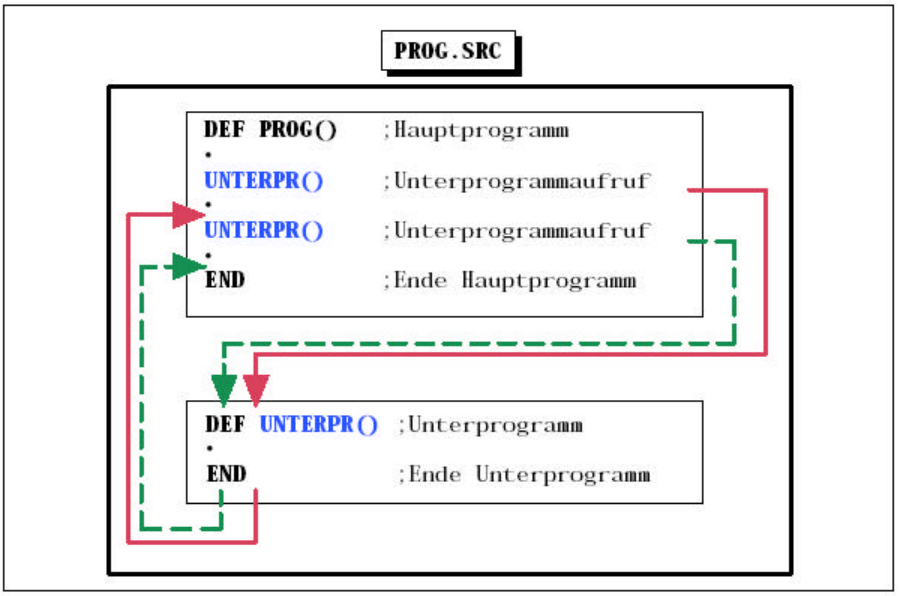
Damit für gleichartige, häufiger sich wiederholende Programmabschnitte die Schreibarbeit beim Programmieren sowie die Programmlänge reduziert werden, wurden Unterprogramme und Funktionen als Sprachkonstrukte eingeführt.
Ein bei größeren Programmen nicht zu unterschätzender Effekt von Unterprogrammen und Funktionen ist die Wiederverwendbarkeit einmal aufgeschriebener Algorithmen in anderen Programmen und besonders der Einsatz von Unterprogrammen zur Strukturierung des Programms.
Diese Strukturierung kann zu einem hierarchischen Aufbau führen, so dass einzelne Unterprogramme, von einem übergeordneten Programm aufgerufen, Teilaufgaben vollständig bearbeiten und die Ergebnisse abliefern.

11.1 Vereinbarung

Ein Unterprogramm oder eine Funktion ist ein separater Programmteil mit Programmkopf, Vereinbarungsteil und Anweisungsteil, der von beliebigen Stellen im Hauptprogramm aufgerufen werden kann. Nach Abarbeitung des Unterprogramms oder der Funktion erfolgt ein Rücksprung an den nächsten Befehl nach dem Aufruf des Unterprogramms. Von einem Unterprogramm bzw. einer Funktion aus können weitere Unterprogramm und/oder Funktionen aufgerufen werden. Die hierbei zulässige Schachtelungstiefe ist 19. Darüber hinaus erfolgt die Fehlermeldung “ÜBERLAUF PROGRAMMSCHACHTELUNG”. Der rekursive Aufruf von Unterprogrammen oder Funktionen ist nicht erlaubt. Das heißt, ein Unterprogramm oder eine Funktion kann sich nicht selbst wieder aufrufen.

DEF Alle Unterprogramme werden genau wie die Hauptprogramme mit der DEF—Vereinbarung plus Namen deklariert und mit End abgeschlossen, z.B.:

```
DEF UNTERPR()
.
END
```



Unterprogrammaufruf und Rücksprung

DEFFCT Eine Funktion ist eine Art Unterprogramm, jedoch ist der Programmname gleichzeitig eine Variable eines bestimmten Datentyps. Damit lässt sich das Ergebnis der Funktion durch einfache Wertzuweisung an eine Variable übergeben. Bei der Vereinbarung von Funktionen mit dem speziellen Schlüsselwort DEFFCT muss daher neben dem Namen der Funktion auch der Datentyp der Funktion angegeben werden. Abgeschlossen wird eine Funktion mit ENDFCT.
Da eine Funktion einen Wert übergeben soll, muss dieser Wert vor der ENDFCT—Anweisung mit der RETURN--Anweisung spezifiziert werden. Beispiel:

```
DEFFCT INT FUNKTION()
.
RETURN(X)
ENDFCT
```

lokal Grundsätzlich unterscheidet man zwischen lokalen und globalen Unterprogrammen bzw. Funktionen. Bei lokalen Unterprogrammen oder Funktionen befinden sich das Hauptprogramm und die Unterprogramme/Funktionen in der selben SRC--Datei. Die Datei trägt den Namen des Hauptprogramms. Dabei steht das Hauptprogramm im Quelltext immer an erster Stelle, während die Unterprogramme und Funktionen in beliebiger Reihenfolge und Anzahl nach dem Hauptprogramm folgen.

global Lokale Unterprogramme/Funktionen können nur innerhalb des SRC--Files, indem sie programmiert wurden, aufgerufen werden. Sollen Unterprogramm--/Funktionsaufrufe auch von anderen Programmen möglich sein, so müssen sie global sein. Globale Unterprogramme oder Funktionen werden in einem eigenen

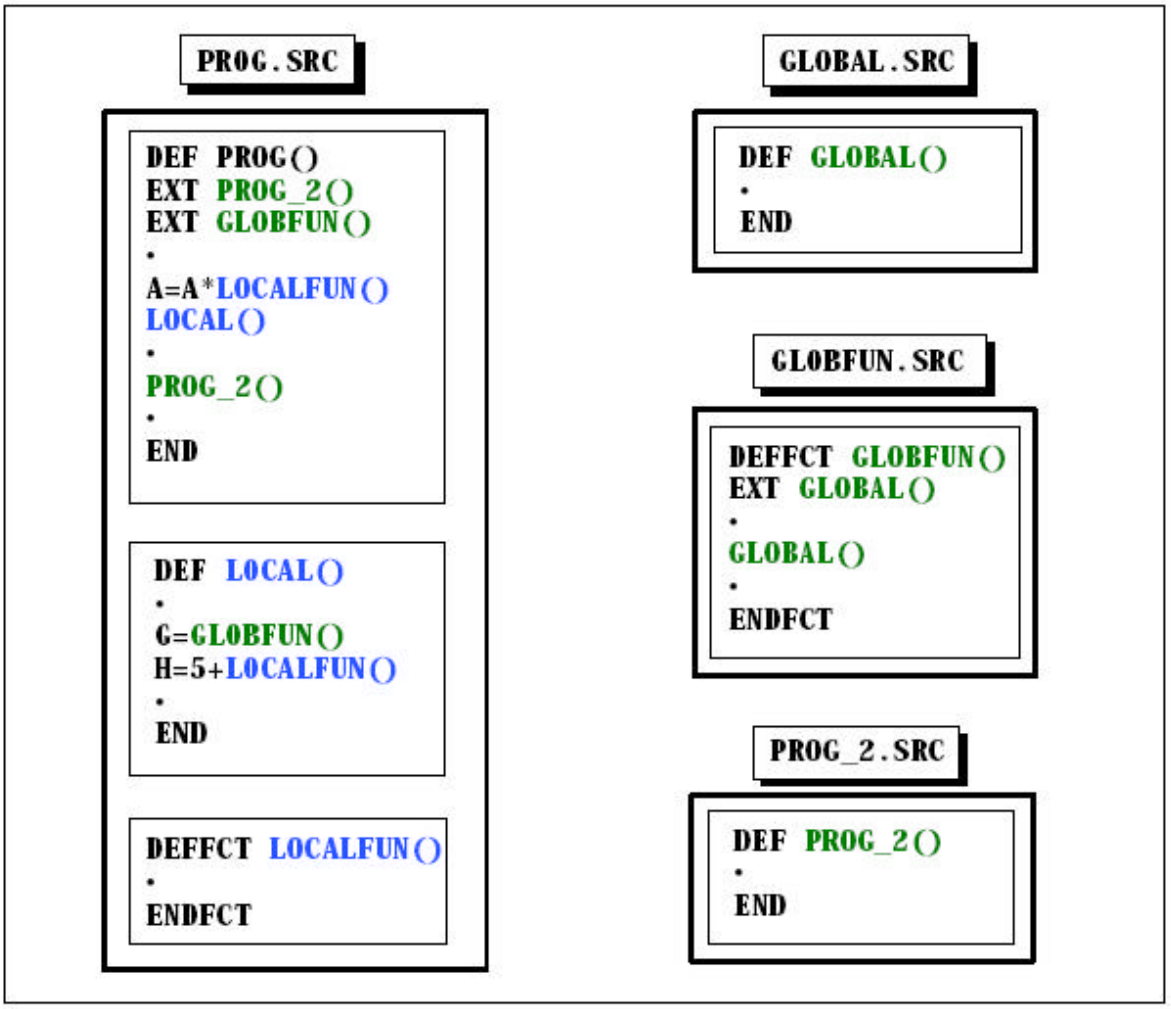
SRC--File abgespeichert. Somit ist jedes Programm ein Unterprogramm, wenn es von einem anderen Programm (Hauptprogramm, Unterprogramm oder Funktion) aufgerufen wird.

EXT
EXTFCT

Namen und Pfade aller aufzurufenden externen Unterprogramme und Funktionen sowie die verwendeten Parameter müssen dem Compiler mit der EXT-- bzw. EXTFCT—Vereinbarung kenntlich gemacht werden. Mit der Angabe der Parameterliste ist auch der benötigte Speicherplatz eindeutig festgelegt. Beispiele:

```
EXT PROG_3()
EXTFCT FUNCTION(REAL:IN)
```

In der folgenden Abbildung ist der Unterschied zwischen globalen und lokalen Unterprogrammen bzw. Funktionen dargestellt:
LOCAL ist ein lokales Unterprogramm und LOCALFUN eine lokale Funktion des Programms PROG, GLOBAL und PROG_2 sind globale Unterprogramme, GLOBFUN ist eine globale Funktion.



Unterschied zwischen lokalen und globalen Unterprogrammen

11.2 Aufruf und Parameterübergabe

Der Aufruf eines Unterprogramms erfolgt einfach durch die Angabe des Unterprogramm-Namens und runden Klammern. Er sieht somit aus wie eine Anweisung, z.B.:

UNTERPROG1()

Ein Funktionsaufruf ist eine besondere Form einer Wertzuweisung. Eine Funktion kann daher nie alleine stehen, sondern der Funktionswert muss stets im Rahmen eines Ausdrucks einer Variablen vom gleichen Datentyp zugewiesen werden, z.B.:

INTVAR = 5 * INTFUNKTION() + 1
REALVAR = REALFUNKTION()

Parameterliste

In lokalen Unterprogrammen und Funktionen sind die im Hauptprogrammdeklarierten Variablen bekannt, in globalen dagegen nicht. Über eine Parameterliste können aber auch Werte an globale Unterprogramme und Funktionen übergeben werden.

Die Übergabe mit Parameterlisten ist oft auch in lokalen Unterprogrammen und Funktionen sinnvoll, da so eine klare Trennung zwischen Hauptprogramm und Unterprogramm/Funktion vorgenommen werden kann: Im Hauptprogramm deklarierte Variablen werden nur dort verwendet, alle Übergaben in Unterprogramme und Funktionen erfolgen mittels Parameterlisten.
Durch diese strukturierte Programmierung reduzieren sich Programmierfehler deutlich.

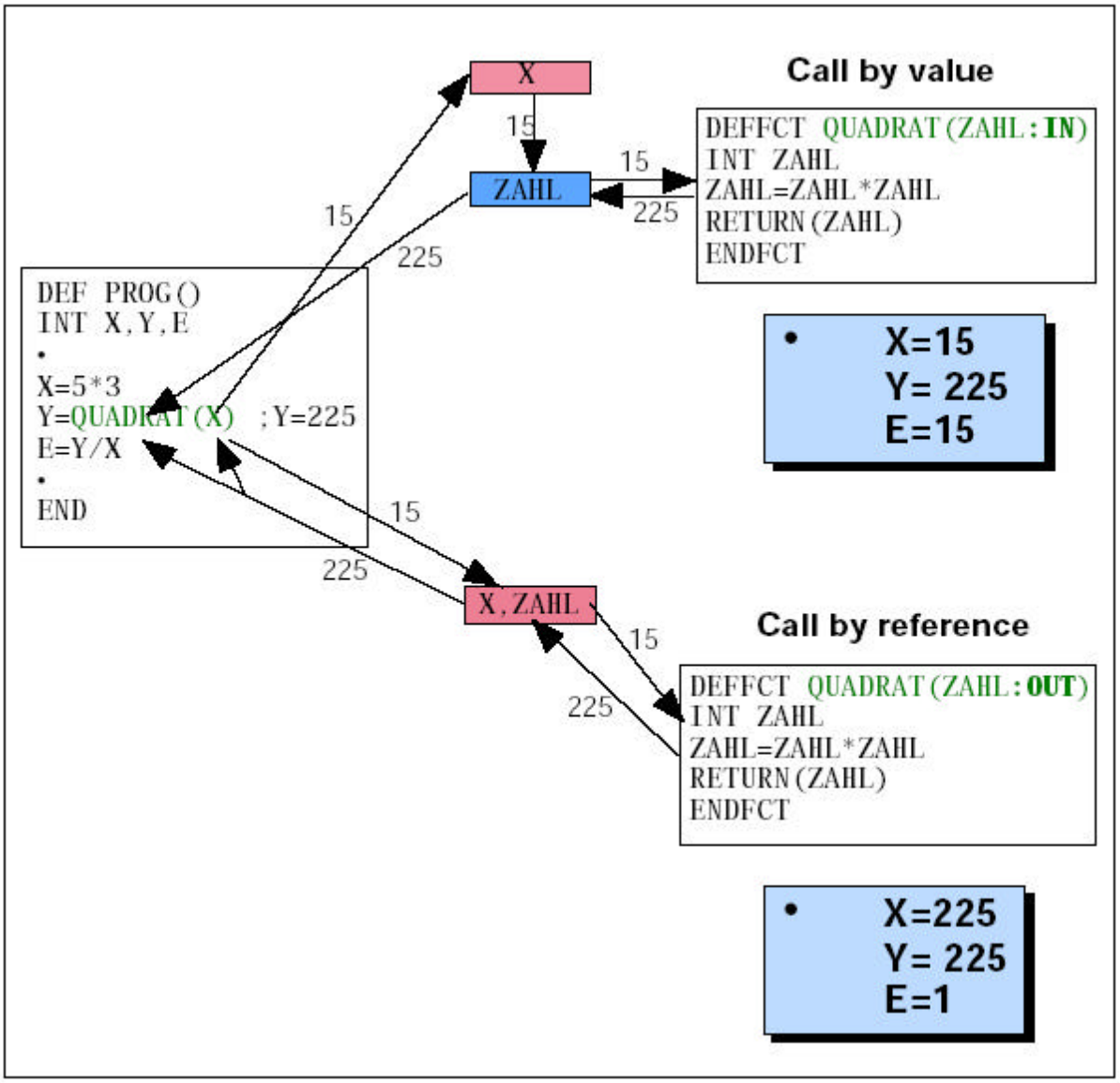
Zur Parameterübergabe gibt es zwei unterschiedliche Mechanismen:

Call by value (IN)
Bei dieser Übergabeart wird ein **Wert** aus dem Hauptprogramm an eine Variable des Unterprogramms oder der Funktion übergeben. Der übergebene Wert kann eine Konstante, eine Variable, ein Funktionsaufruf oder ein Ausdruck sein. Bei unterschiedlichen Datentypen wird, wenn möglich, eine Typanpassung durchgeführt.

Call by reference (OUT)
Durch “Call by reference” wird nur die **Adresse** einer Variablen des Hauptprogramms an das Unterprogramm bzw. die Funktion übergeben. Das aufgerufene Unterprogramm bzw. die Funktion kann nun über einen eigenen Variablennamen den Speicherbereich überschreiben und somit auch den Wert der Variablen im Hauptprogrammverändern.
Die Datentypenmüssen daher identisch sein, eine Typanpassung ist in diesem Fall nicht möglich.
In der Abbildung ist der Unterschied zwischen den beiden Methoden aufgezeigt. Während die Variable X bei “Call by value”im Hauptprogramm aufgrund der getrennten Speicherbereiche unverändert bleibt, wird sie bei “Call

by reference" durch die Variable ZAHL in der Funktion überschrieben.

Unterschied zwischen „call by vallue" und „call by reference"



"Call by value" wird im Unterprogramm-- oder Funktionskopf durch das Schlüsselwort IN hinter jeder Variablen in der Parameterliste angegeben. "Call by reference" erhält man durch Angabe von OUT. OUT ist auch die Default--Einstellung. Beispiel:

DEF RECHNE(X:OUT,Y:IN,Z:IN,B)

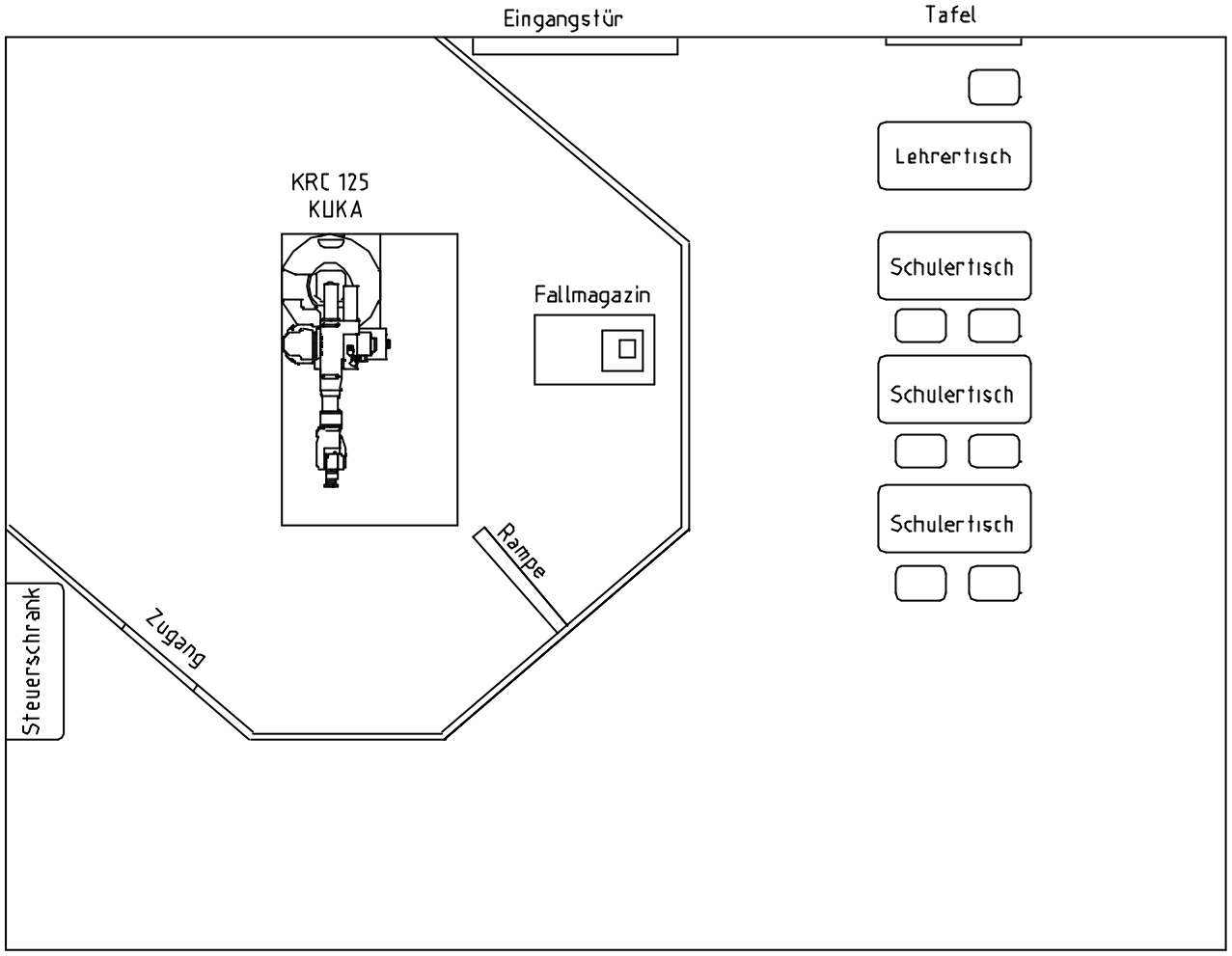
Ist das aufzurufende Unterprogramm oder die Funktion global, so muss bei der Extern—Vereinbarung im Hauptprogramm angegeben werden, welchen Datentyp die jeweiligen Variablen haben und welcher Übergabemechanismus verwendet werden soll. OUT ist wieder die Default-Einstellung. Beispiel:

EXTFCT REAL FUNKT1(REAL:IN,BOOL:OUT,REAL,CHAR:IN)

12. Anhang

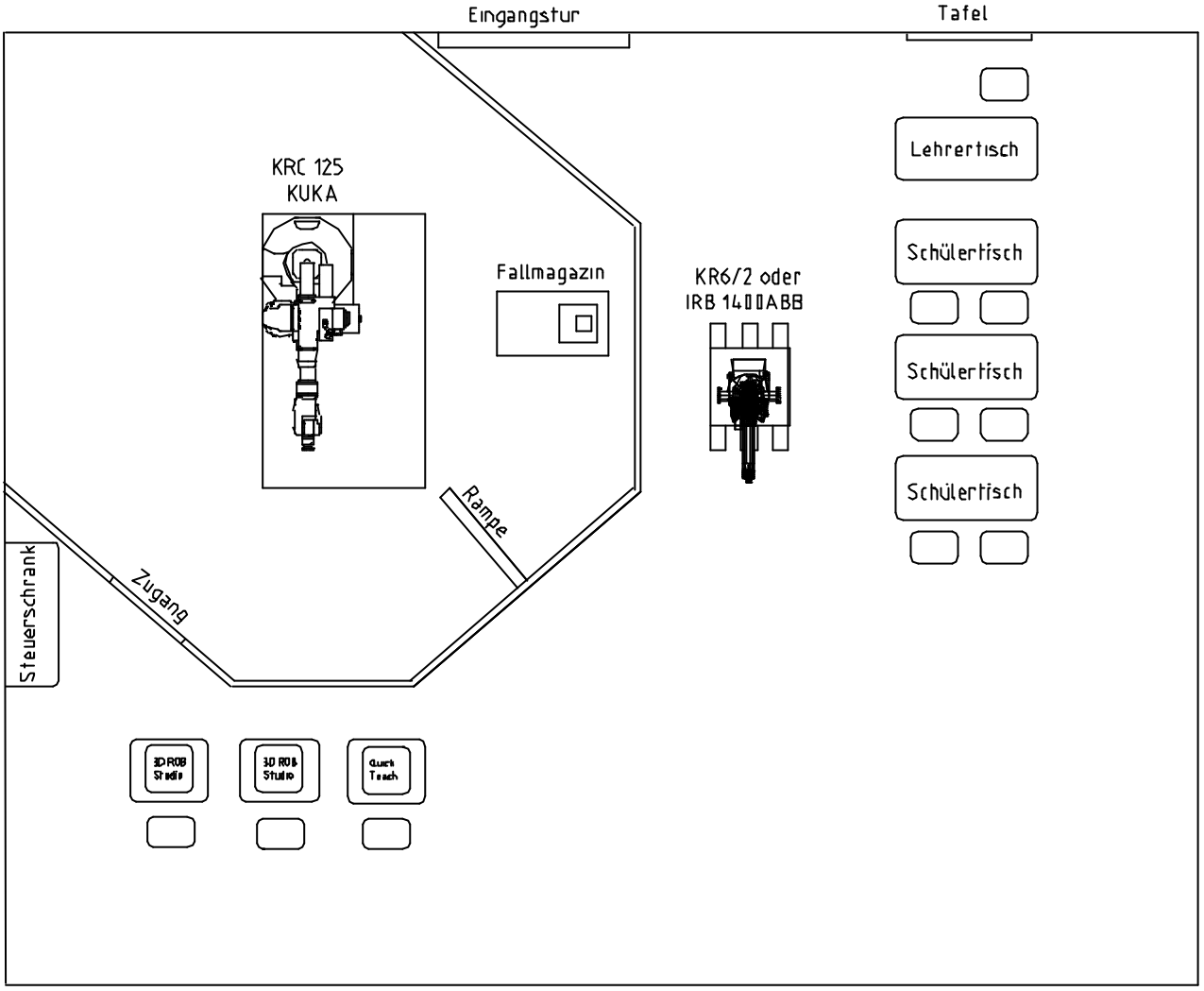
12.1 Layout Roboter Labor – Ist / Soll Situation

Ist Situation:



- Kuka Roboter KRC 125 mit KCP
  - Traglast 125 kg
  - Zusatzlast 120 kg
  - Wiederholgenauigkeit <0.2mm
  - maximale Reichweite: 2410 mm
  - Gewicht: 975kg
  - Einbaulage: Boden, Decke
- Fallmagazin
- Verkettete Automatisierungsaufbauten mit SPS und Profibus

Soll Situation:



- Kuka Roboter mit KCP
- ABB IRB 1400 montiert auf einer Staplerpalette
  - Traglast 5 kg
  - Zusatzlast 10 kg
  - Wiederholgenauigkeit <0.05mm
  - maximale Reichweite: 1440 mm
  - Gewicht: 225 kg
  - Einbaulage: Boden
- oder Kuka KR6/2 montiert auf einer Staplerpalette
  - Traglast 6 kg
  - Zusatzlast 10 kg
  - Wiederholgenauigkeit <0.1mm
  - maximale Reichweite: 1570 mm
  - Gewicht: 205 kg
  - Einbaulage: variabel
- PC Arbeitsplätze mit Simulationssoftware 3D ROB Studio oder ähnliche Software von Kuka
- PC Arbeitsplätze mit QuickTeach oder ähnliche Software von Kuka

13. Literaturverzeichnis

Erster Kasten in Arbeitszelle

- Mappe 1, KR C1, Realease 2.2.8 & 2.3.24, Augsburg
- Mappe 2, KR C1, Realease 2.2.8 & 2.3.24, Augsburg
- Mappe 3, KR C1, Realease 2.3.24, Augsburg
- www.kuka.de
- www.abb.com
- www.delmia.com
- www.robcad.de