

# Industrieroboter

Eric Dokulil

## Einleitung

Einen allgemein gültigen Artikel über Industrieroboter zu schreiben ist mittlerweile so schwierig wie allgemein über Computer zu schreiben. Was ich aber dennoch versuchen möchte, ist einen groben Überblick über das Thema zu geben und, dann ein bisschen aus meiner ganz persönlichen Erfahrung zu berichten. Daher muss ich mich gleich dafür entschuldigen, dass diese Erfahrungen natürlich sehr mit den Firmen und Marken zu tun haben bei denen ich arbeite bzw. gearbeitet habe. Tatsache ist, dass das Angebot und die Qualität in diesem Bereich ist in den letzten Jahren stark gestiegen. Oft wird daher die Entscheidung, ob das eine oder andere Robotersystem eingekauft wird, nur dadurch getroffen, welches Gesamtkonzept besser gefällt und nicht welche Marke oder Type von Industrieroboter eingesetzt wird.

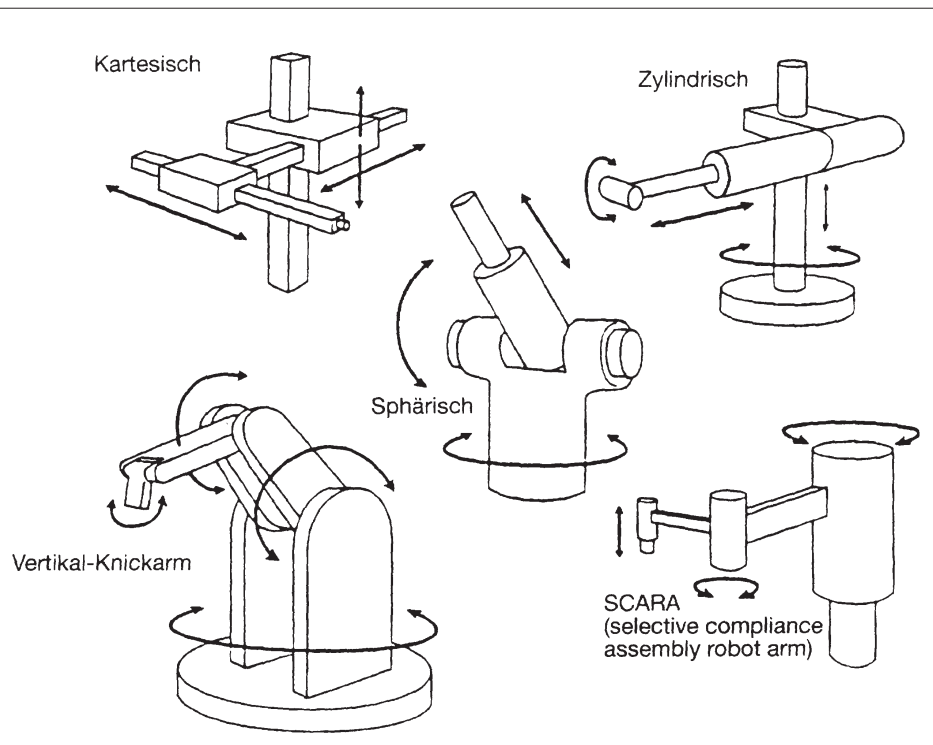
Und damit bin auch schon mitten im Thema, denn die Entwicklung der Roboter ist nur ein Teil, wenn auch ein wesentlicher, in der Entwicklung der flexiblen Automation. Oft begegnet man in den Firmen der Angst, dass Kollege Roboter als „Jobkiller“ eingesetzt wird und dadurch Arbeitsplätze verloren gehen. In den meisten Fällen ist das Gegenteil der Fall. Die wichtigsten Aspekte für den Einsatz von flexibler Automation:

- effiziente Produktion kleiner oder mittlerer Serien;
- schneller Produkt- und Modellwechsel;
- rasch umrüstbare effiziente Fertigung und Montage;
- Produktion größerer Typenvielfalt;
- Erhöhung und Sicherung eines gleichbleibenden Qualitätsstandards;
- Verhinderung der Abwanderung der Produktion in Niedriglohnländer;
- Senkung der Produktionskosten;
- Entkopplung des Menschen vom Maschinentakt und damit der Abbau der Belastungen und Gefahren in der Fertigung.

Wohl ersetzt er einzelne Arbeitsplätze, die aber vielfach monoton, schwer und sogar gesundheitsgefährdend sind. Damit leistet der Industrieroboter einen wesentlichen Beitrag zur Humanisierung der Arbeitswelt. Außerdem entstehen in der stark expandierenden Branche Automatisierung und Datenverarbeitung neue Arbeitsplätze und Berufswege.

## Definition und Beschreibung

„Industrieroboter sind universell einsetzbare Bewegungsautomaten mit mindestens drei Achsen, deren Bewegung hinsichtlich Bewegungsfolgen und Wegen bzw. Winkel frei programmierbar und gegebenenfalls sensorgeführt sind.“



Bauformen von Industrierobotern

Die Tätigkeiten, die ein Industrieroboter abdecken kann, lassen sich grob wie folgt gliedern:

- **Fertigen:** Roboter trägt Werkzeug;
  - **Montieren und Handhaben:** Roboter trägt Greifsystem;
  - **Kontrollieren und Messen:** Roboter trägt Messeinrichtung;
- Ein Industrieroboter besteht aus folgenden Komponenten und Teilsystemen:

- **Kinematik:** Räumliche Zuordnung zwischen Werkstück bzw. Werkzeug und Fertigungseinrichtung.
- **Antrieb und Bremsen:** Übertragung und Umwandlung der Energie bis hin zum Effektor.
- **Steuerung:** Informationseingabe, Programmablaufsteuerung und -überwachung, Informationsspeicherung, Funktionsverknüpfung mit der Industrieroboterumwelt.
- **Messsysteme:** Lage- und Geschwindigkeitsmessungen der Achsen.
- **Effektoren:** Werkstück-, Werkzeug- oder Prüfmittelaufnahme
- **Sensoren:** Erfassung der Industrieroboterumwelt, Lage- und Mustererkennung, Erfassung physikalischer Einsatzparameter.
- **Sicherheitseinrichtungen:** Schutz von Personen, Schutz vor Selbstbeschädigung des Industrieroboters, Kollisionsschutz zum Werkstück bzw. Werkzeug.
- **Peripheriegeräte:** Zuführung des Werkstückes in definierte Position und Lage, Fixierung während der Bearbeitung sowie An- und Abtransport des Werkstückes.
- **Software:** Betriebssystemsoftware und Anwendersoftware.

Industrieroboter weisen unterschiedliche Bauformen in bezug auf die Kinematik

auf. Diese Bauformen haben großen Einfluss auf:

- Arbeitsbereich
- Tragfähigkeit
- Geschwindigkeit
- Wiederholgenauigkeit

Mit der Kinematik ist die Anzahl der Freiheitsgrade, und damit die Anzahl der Bewegungsachsen, die entweder eine translatorische oder eine rotatorische Bewegung ermöglichen, festgelegt. Die Schließbewegung eines Greifers wird dabei nicht als eigene Achse gezählt.

Um einen beliebigen Punkt des Arbeitsraumes eines Industrieroboters anfahren zu können, sind mindestens drei Freiheitsgrade erforderlich. Um dem Effektor (z.B. Greifer, Schweißpistole, Bohrer usw.) an diesem Punkt eine beliebige Orientierung geben zu können, sind drei weitere Freiheitsgrade notwendig. Für einige Applikationen kann auch mit weniger als sechs Achsen das Auslangen gefunden werden.

Folgende Bauformen von Industrierobotern werden in bezug auf die Kinematik unterschieden:

- **Kartesische Koordinatenbauweise** (Linear-Achsen)
- **Zylinderkoordinatenbauweise**
- **Kugelkoordinatenbauweise** (sphärisch)
- **Gelenkarmbauweise** (Vertikal-Knickarmbauweise).
- **SCARA-Bauweise** (Horizontal-Knickarmbauweise).

Industrieroboter mit kartesischer Koordinatenbauweise besitzen nur translatorische Freiheitsgrade. Die Anforderungen an die Steuerung sind minimal und ihr quaderförmiger Arbeitsraum ist im bezug auf die Eigenabmessungen eher klein.

Sie sind gut geeignet für Be- und Entlade-tätigkeiten.

Der Selective Compliance Assembly Robot Arm (SCARA), meist mit drei rotatorischen und einer translatorischen Achse, wurde speziell für Montage- und Bestückungsaufgaben in der Kfz- und Elektronikindustrie (Leiterplattenbestückung) entwickelt. Roboter dieser Bauart zeichnen sich durch sehr hohe Bewegungsgeschwindigkeit bei großer Wiederholgenauigkeit aus. Die Tragfähigkeit ist dagegen in der Regel gering.

Typisches Merkmal der Gelenkarmbauweise ist die Beschränkung auf rotatorische Freiheitsgrade. Die heute meist verwendete Ausführung weist 6 Achsen auf. Roboter dieser Bauart sind äußerst universell einsetzbar und arbeiten in der Regel mit einer Wiederholgenauigkeit von ±0,1mm. Ihr Arbeitsraum ist in bezug auf die Eigenabmessungen groß, die Anforderungen an die Steuerung sind jedoch ebenfalls erheblich. Trotzdem bewährt sich diese Bauweise seit langem sehr und man findet Sie heute am weitesten verbreitet.

Die heutigen Industrieroboter werden in den Applikationen als Bausteine eines Baukastensystems verwendet und oft mit weiteren „externen“ Achsen wie z.B. Fahrbahnen, Drehtisch, Manipulator und Portale erweitert. Je nach Steuerung können eine bestimmte Anzahl von Zusatzachsen Verwendung finden.

Generell kann man feststellen, dass bestimmte Industrieroboter mit ihrer Bauform sich auf eine bestimmte Anwendung spezialisieren und andere wiederum versuchen, Ihr System möglichst offen und universell zu halten.

Die Firma Wittmann etwa hat sich mit ihren kartesischen Robotern darauf spezialisiert, Spritzgussmaschinen aus der Kunststoffindustrie zu be- und entladen.

Die Firma igm-Robotersysteme wiederum hat durch eine patentierte Bauform der sechsten Achse ihres 6-Achs-Knickarmroboters für Schweißaufgaben optimiert.

Die speziell ausgelegten Bauformen er-

rer Bedienung speziell an diese Aufgaben anzupassen. Damit kann Bedienung und Umgang mit den Robotern einfach gehalten werden. Firmen wie ABB und Kuka



Bild 4: Neue igm K5 Konsole mit Joystick. Gut sieht man hier die Optimierung der Bedienkonsole auf den Schweiß-einsatz.

gehen den Weg durch ein möglichst vielfältiges Angebot an Traglasten und Abmessungen ihrer Knickarmroboter viele Anwendungen abdecken zu können. Daraus resultiert natürlich, dass die Steuerung universell ausgelegt werden muss und somit die Handhabung und Programmierung komplexer wird.

### Steuerungstechnik für Industrieroboter

Als Beispiel für Robotersteuerungen möchte ich die Kuka Steuerung KRC1 anführen, mit der Kuka 1996 als erster

Roboter-Hersteller den PC zur Steuerung eingesetzt hat.

Die Anforderungen an moderne Industrieroboter sind neben den schon fast selbstverständlichen Eigenschaften wie hohe Genauigkeit (Bahn- und Wiederholgenauigkeit), Geschwindigkeit, großer Arbeitsraum etc. Kriterien wie die Integrationsmöglichkeit der Robotersteuerung in übergeordnete Steuerungsstrukturen, aber auch die Integration in unterlagerte Steuerungen mit Sensoren und Aktoren sowie einfache Bedienung und Programmierung.

Mit dem Einsatz der PC-Technologie zur Steuerung von Anlagen und Maschinen ergeben sich für den Anwender folgende Vorteile:

- **Offenheit und Netzwerkfähigkeit** (z.B. Ethernet, Busanschlungen) mit vereinfachter Systemintegration durch am Markt verfügbare Einsteckkarten und Treiber. Offenheit heißt dabei nicht nur den Zugang zum System selbst über definierte Schnittstellen sondern darüber hinaus auch die Veränderung und Erweiterung bestehender Funktionalitäten in der Steuerung.
- **Nutzung von Quasistandards** in Software und Hardware (Betriebssystem, interne Buskommunikation). Daher ist es auch möglich, verfügbare und gebräuchliche Entwicklungswerkzeuge einzusetzen. Damit ist nicht nur der Spezialist des Steuerungsherstellers in der Lage, Steuerungserweiterungen vorzunehmen (eine

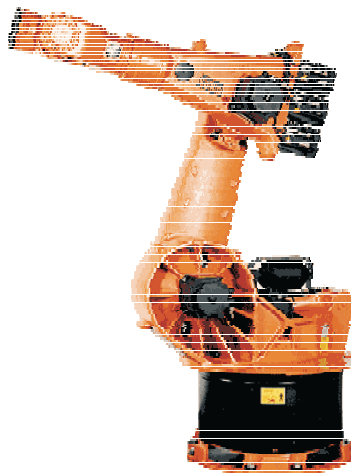


Bild 5: Kuka KR125 Roboter

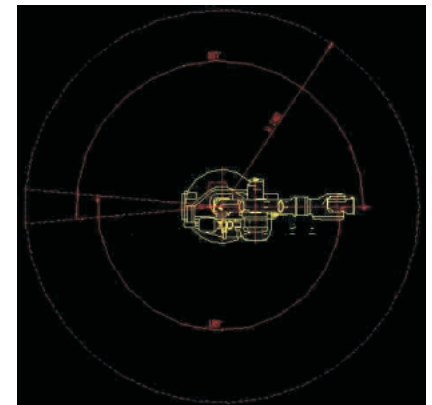


Bild 7: Grundriß des Kuka KR125



Bild 3: Neuer igm-schweißroboter Rti2000 auf Kuka Basis mit eigener patentierter Roboterhand

möglichen es auch, die Steuerung mit ih-

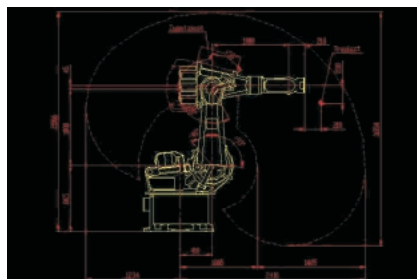


Bild 6: Aufriß des Kuka KR125



Bild 8: KCP ( Kuka Control Panel) Neben den windows Standarttasten sind viele Funktionstasten vorhanden die sich je nach Einstellung ändern.

Grundvoraussetzung, um die Offenheit der Steuerung zu gewährleisten).

- **Nutzung der Entwicklung in der PC-Technik:** Performancesteigerung durch neue Prozessorgenerationen, interne Bussysteme (z.B. PCI), Einsatz markt-gängiger Speichermedien (Festplatte, CD-ROM, Floppy etc.) und Bedienelemente (MF2-Tastatur, Mouse etc.), weiterentwickelte Betriebssysteme.
- **Grafisch orientierte Bedienoberfläche** auf Basis von WINDOWS (95, NT).
- **Preisvorteil** gegenüber Sonderentwicklungen.

**Systemaufbau der PC-basierten Robotersteuerung KR CI**

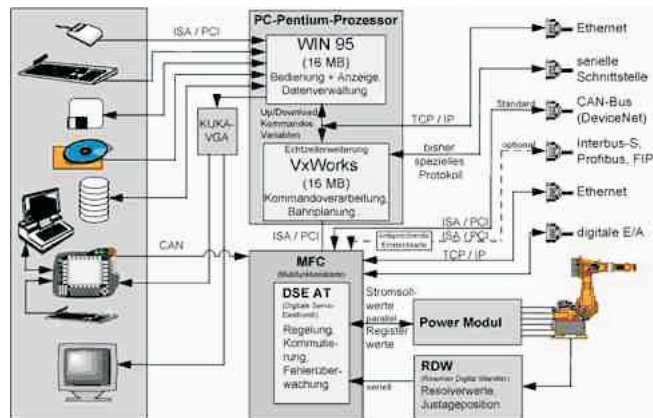


Bild 1: Steuerungskonzept KRC1

**Bild 1** zeigt die gesamte Systemarchitektur. Kern der Steuerung ist aus Sicht der Hardware ein PC-Motherboard mit Intel Pentium Prozessor und mindestens 32MB RAM. Auf diesem Prozessor laufen die beiden Betriebssysteme WINDOWS 95 und VxWorks, die über das TCP/IP-Protokoll Daten (z.B. Variablenwerte, Kommandos, Up/Download von Roboterprogrammen) austauschen. Dabei wird VxWorks beim Hochlauf von Win95 gestartet. Beiden Systemen wird exklusiv ein Teil des Hauptspeichers zugewiesen. Beide Systeme haben über eigene Treiber Zugriff auf die Hardware und kommunizieren miteinander nur über einen virtuellen Netzwerkkontroller.

Bedienung, Anzeige und Datenverwaltung sind unter WINDOWS 95 implementiert (eine vollständige Bedienoberfläche ist vorhanden und nicht nur eine Bibliothek zu deren Generierung).

Unter VxWorks erfolgen alle echtzeitrelevanten Aufgaben wie Bahnplanung, Kommandoverarbeitung und Verarbeitung der Informationen peripherer Anschaltungen (z.B. Sensordatenbearbeitung).

Zur Regelung der Antriebe (über die DSE AT) sowie zur Busanpassung und der Ankopplung digitaler E/As wird auf das PC-Motherboard die MFC (Multifunktionskarte) aufgesteckt.

Auf dieser MFC befindet sich auch der passive LP-Real Time Accelerator Chip als Logikbaustein. Seine Aufgabe besteht in der Koordination der beiden Betriebssysteme WINDOWS 95 und VxWorks,

indem er alle 4ms durch Auslösung des NMI (non maskable interrupt) eine nicht sperrbare Unterbrechung generiert und damit gleichzeitig VxWorks aktiviert. Da WINDOWS 95 diese Art der Unterbrechung nie sperren kann, ist die Echtzeitfähigkeit auf Basis von VxWorks sowie die höhere Priorität von VxWorks immer gewährleistet. Damit ist sichergestellt, dass die eigentlichen Steuerungsaufgaben als Task unter VxWorks immer Vorrang haben und unabhängig von WINDOWS 95 bearbeitet werden.

Auf der MFC befindet sich eine DSE-AT Karte die als Schnittstelle zu den Lagereglern und Powermodulen dient. Innerhalb des Interpolationstaktes von 12ms müs-

sen in ein shared-memory der DSE-AT die neuen Reglervorgaben geladen werden. Dazwischen werden die Positionen mittels eigenem DSP-Prozessor der DSE-Karte interpoliert. Der Interpolationstakt bestimmt somit die maximale Auflösung der Bahnplanung. Über die RDW-Box gelangen die Resolverdaten der einzelnen Achsen in die DSE-AT Karte. Da die Resolver Absolutdrehgeber sind entfällt nach dem Einschalten des Roboters das Referenzieren der einzelnen Achsen.

Auf der MFC ist zur Kommunikation mit der Peripherie standardmäßig ein CAN-Bus Modul integriert, dass das Protokoll DeviceNet unterstützt. Die Ankopplung von weiteren Bussystemen (Interbus-S, Profibus Master oder Slave, FIP etc.) erfolgt über zusätzliche verfügbare Einsteckkarten. Zur Kommunikation über Ethernet (Protokoll TCP/IP) bestehen zwei Anschlußmöglichkeiten: zum einen an der MFC mit direktem Zugang zum Echtzeitbetriebssystem VxWorks, zum anderen am Motherboard mit Zugang zum Datenaustausch zwischen WINDOWS 95 und VxWorks. Ebenfalls unterstützt wird die PC-typische serielle Schnittstelle (z.B. zur Integration von Sensorik).

Zur Bedienung und Programmierung wird das Programmierhandgerät KCP (KUKA Control Panel) mit 8" Farbdisplay, Verfahrstasten sowie zusätzlich der 6D-Mouse zum gleichzeitigen, intuitiven Teachen in allen sechs Freiheitsgraden genutzt. Bis auf den Hauptschalter befinden sich alle Bedienelemente am Pro-

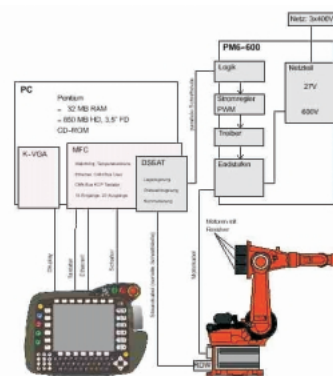


Bild 2: vereinfachtes Steuerungskonzept.

grammierhandgerät, was den Bedienkomfort deutlich erhöht. Um Entfernungen zwischen Steuerung und Programmierhandgerät von bis zu 100m zu realisieren, war die Entwicklung einer speziellen VGA-Karte (KUKA-VGA) notwendig. Diese Karte erlaubt ebenfalls den Anschluß eines zusätzlichen Monitors. Da die Rückmeldung der Tastenbetätigungen an die Steuerung echtzeitfähig sein muss (um z.B. den Nachlauf des Roboters bei Loslassen der Bedienelemente zum Verfahren zu verhindern), erfolgt sie nicht über ISA/PCI sondern mittels CAN-Bus direkt an die MFC. Als weitere Bedienelemente können PC-typisch eine MF2-Tastatur und eine Mouse angeschlossen werden. In Verbindung mit dem Monitor kann daher die Bedienung auch ohne Programmierhandgerät durchgeführt werden.

**Handhabung und Programmierung von Knickarm-Industrierobotern**

Die Programmierung der Kuka KRC1 Steuerung erfolgt mittels KRL (Kuka Robot Language). Diese Anwendungsprogrammiersprache ist hochsprachenorientiert (Mischung aus C und Basic) mit steuerungstechnischen Erweiterungen (Ein- u. Ausgabeanweisungen, Bewegungsbefehle). Der besondere Umstand, dass jede Bewegung mittels sechs rotatorischen Achsen ausgeführt wird, hat zur Folge, dass es keine einfache Zuordnung einer bestimmten Achse zu einer Raumkoordinate gibt. Eine Position ist daher immer durch sechs Winkel bestimmt (A1 bis A6). Um den Umgang und die Programmierung des Roboters zu vereinfachen werden von der Steuerung zusätzlich errechnete 3 fixe und 16 frei definierbare Koordinatensysteme angeboten.

Die fixen Koordinatensysteme sind \$WORLD, \$ROBROOT und \$TOOL=\$NULLFRAME (entspricht dem Roboterflansch).

Bei einem Knickarmroboter mit 6 Achsen wird eine Position durch 6 Freiheitsgrade bestimmt.

Als Frame bezeichnet man die Information, die durch Angabe von Koordinaten X, Y, Z und der Drehwinkel A, B, C einen Punkt im Raum darstellt. Die Winkel A,

B, C geben die Orientierung des Werkzeuges in einem Punkt an. Der Winkel A ist die Drehung um die Z-Achse. Der Winkel B ist die Drehung um die Y-Achse. Der Winkel C ist die Drehung um die X-Achse.

Nullframe bedeutet, dass alle Koordinaten und Winkel 0 sind.

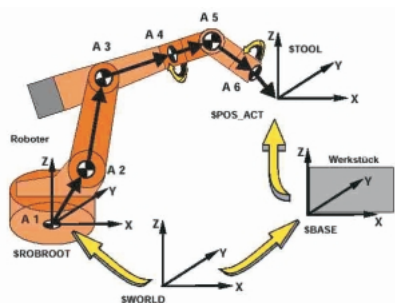


Bild 1: Vektorenkette der Robotermechanik, Achsen 1 bis 6

Die folgenden Bezeichnungen stellen Systemvariablen der Steuerung dar. Alle Systemvariablen beginnen mit „\$“.

**\$WORLD** ... Ortsfestes rechtwinkliges (kartesisches) Koordinatensystem in der Arbeitszelle. Bei einer Bewegung des Roboters ändert bzw. bewegt sich dieses Koordinatensystem nicht mit.

**\$ROBROOT** ... Sitzt immer im FuÙe des Roboters. Mit „\$ROBROOT“ kann somit eine Verschiebung des Roboters zu „\$WORLD“ definiert werden. Bei Anlagen mit nur einem Roboter wird \$WORLD mit \$ROBROOT zusammengelegt.

**\$BASE** Frei definierbares rechtwinkliges (kartesisches) Koordinatensystem, das seinen Ursprung am zu bearbeitenden Werkstück hat.

**\$TOOL** Frei definierbares rechtwinkliges (kartesisches) Koordinatensystem, dessen Ursprung in der Werkzeugspitze liegt. Diesen Ursprung nennt man auch TCP (Tool Center Point).

**\$POS\_ACT** Aktuelle IST-Position des Roboters. Dieser Wert stellt die Position und Orientierung des TCP im eingestellten \$BASE Koordinatensystem dar. In diesem Frame ist die Verschiebung zwischen „\$BASE“ und „\$TOOL“ dargestellt (Koordinaten des geteachten Punktes in der Datenliste).

**\$NULLFRAME** Als Frame bezeichnet man die Information, die durch Angabe der Koordinaten X,Y,Z und der Drehwinkel A,B,C einen Punkt im Raum darstellt. Nullframe bedeutet zusätzlich, dass alle Koordinaten und Winkel 0 sind.

**Beschreibung**

Das BASE-Koordinatensystem wird als Bezugssystem für die Beschreibung der Lage des Werkstücks herangezogen. Die Programmierung des Roboters erfolgt im BASE-Koordinatensystem, welches das WORLD-Koordinatensystem als Bezugssystem hat.

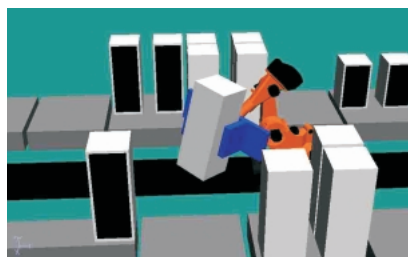


Bild 5: Simulation einer Handlingszelle für Kühlschränke

Bei der Interpolation der Bewegungsbahn berechnet die Robotersteuerung im Normalfall (feststehendes Werkstück, Werkzeug am Roboterflansch) die aktuelle Position (\$POS\_ACT) bezogen auf das \$BASE-Koordinatensystem.

Bei der Programmierung einer Bewegung können die Positionen durch mehrere Methoden ermittelt werden:

- **Geteachte Positionen:** Beim „teachen“ werden die gewünschten Positionen in Handfahrt unter Zuhilfenahme der einzelnen Koordinatensysteme angefahren und dann auf Knopfdruck als Punkt in eine Positionenliste aufgenommen (Fixpunkte).
- **Errechnete Positionen:** (relativ zu einem geteachten Fixpunkt). Solche Positionen werden z.B. aus einer

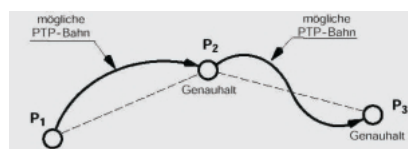


Bild 2: PTP Bewegung mit Genauhalt

tionen werden z.B. für Stapelaufgaben verwendet, wo von einer Startposition ausgehend Werkstücke in einem bestimmten Stapelmuster abgelegt werden.

- **Absolut berechnete Positionen:** Die Positionen werden z.B. aus einer

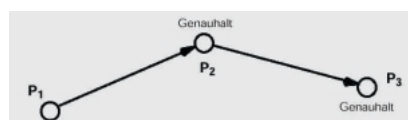


Bild 3: LIN Bewegung mit Genauhalt.

CAD-Datei berechnet und dem Roboter übergeben.

Bei der Bewegung zwischen zwei Punkten kann man prinzipiell zwischen zwei Arten unterscheiden:

**PTP Bewegung (Point To Point)**

Die Positionierung des Robotersystems erfolgt hier auf dem schnellsten Weg zwischen zwei Punkten. Die Bewegung wird in allen Achsen gleichzeitig gestartet und beendet. Das heißt, die Achse die den längsten Weg hat bestimmt die Dauer der Bewegung. Alle anderen Achsen werden auf diese Achse synchronisiert. Die Bahn des Roboters ist deshalb nicht exakt vorherzusehen.

**Lineare Bewegungen (LIN)**

Bei einer linearen Bewegung werden die Roboterachsen so aufeinander abge-

stimmt, dass der Werkzeug- bzw. Werkstück-Bezugspunkt entlang einer Gera-

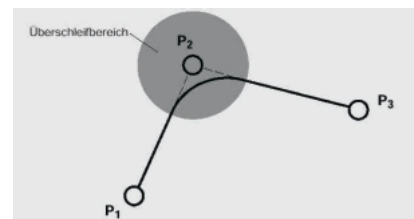


Bild 4: LIN Bewegungen die überschleifen werden.

den zum Zielpunkt bewegt wird. Das Werkzeug bzw. Werkstück selbst kann seine Orientierung während der Bewegung ändern. Die Bahn des Roboters ist exakt vorherzusehen.

Um einen Bewegungsstopp an einem Punkt zu verhindern, können gleichartige aufeinanderfolgende Bewegungen überschleifen werden.

Beim Überschleifen überwacht die Steuerung einen so genannten Überschleifbereich um den Zielpunkt. Tritt der Bezugspunkt in diesen Bereich ein, wird die Roboterbewegung auf den Zielpunkt des nächsten Bewegungsbefehls überführt und es kommt zu keinem Bewegungsstopp.

Damit der Roboter Bewegungen verschleifen kann und Bewegungsstops minimiert werden, muss die Bahnplanung für einige Bewegungen im voraus erfolgen. Das wird dadurch erreicht, dass die Bewegung dem Programm um einige Bewegungsbefehle hinterher hinkt. In diesem Zusammenhang spricht man von Vorlauf- und Hauptlaufzeiger.

Der Vorlaufzeiger zeigt auf die Position im Programm an der gerade der Code interpretiert wird und Befehle die keine Bewegungen sind ausgeführt werden.

Der Hauptlaufzeiger zeigt auf jene Position im Programm an der man sich gerade aufgrund der tatsächlichen Bewegung befindet.

**Programmbeispiele siehe nächste Seite.**

Programmbeispiel 1 zeigt einige Bewegungsbefehle. Im Programmbeispiel 2 wird durch das Setzen der Eingänge 1, 2 oder 3 der Roboter in drei verschiedene Stellungen verfahren. Ist der Roboter in der gewünschten Position angekommen, so wird dies durch Setzen der entsprechenden Ausgänge 13, 14 oder 15 angezeigt. Durch Setzen von Ausgang 16 wird das Programm abgebrochen.

Zur Untersuchung von Zugänglichkeiten und zur Offline-Programmierung bieten die meisten Roboterhersteller Simulationsstationen an. Der Vorteil solcher Systeme ist, dass die originale Steuerungssoftware Verwendung findet und nur die Robotermechanik mit der ihn umgebenden Zelle durch spezielle Software simuliert wird. Damit ist es möglich Programme im Vorfeld zu schreiben und so Projekte auf ihre Durchführbarkeit und ihre Taktzeit zu untersuchen.

**Programmbeispiel 1**

```

DEF LIN BEW ( )
:----- Deklarationsteil -----
EXT BAS (BAS COMMAND: IN, REAL: IN)
DECL AXIS HOME ;Variable HOME vom Typ AXIS
:----- Initialisierung -----
BAS (#INITMOV, 0) ;Initialisierung von Geschwindigkeiten,
;Beschleunigungen, $BASE, $TOOL, etc.
;Definieren einer HOME-Position
HOME = {AXIS: A1 0,A2 -90,A3 90,A4 0,A5 0,A6 0}
:----- Hauptteil -----
PTP HOME ; SAK-Fahrt
PTP {A5 30} ;A5 auf absolut 30 Grad drehen.
; Linearbewegung zur angegebenen Position, die Orientierung
; wird kontinuierlich auf die Zielorientierung verändert
LIN {X 1030,Y 350,Z 1300,A 160,B 45,C 130}
; Linearbewegung zur angegebenen Position, die Orientierung
; wird dabei nicht verändert
$ORI_TYPE=#CONST
LIN {FRAME: X 700,Y -300,Z 1000,A 23,B 230,C -90}
; die Orientierung wird weiterhin nicht geändert
LIN {FRAME: Z 1200,A 90,B 0,C 0}
; Relativbewegung entlang der X-Achse
LIN REL {FRAME: X 300}
PTP HOME
END
    
```

```

SIGNAL MITTE $OUT[14]
SIGNAL RECHTS $OUT[15]
:----- Initialisierung -----
BAS (#INITMOV,0) ;Initialisierung von Geschwindigkeiten,
;Beschleunigungen, $BASE, $TOOL, etc.
HOME={AXIS: A1 0,A2 -90,A3 90,A4 0,A5 0,A6 0}
:----- Hauptteil -----
PTP HOME ;SAK-Fahrt
LINKS=FALSE
MITTE=TRUE ;in mittlerer Stellung
RECHTS=FALSE
WHILE ABRUCH==FALSE ;Abbruch, wenn Eingang 16 gesetzt
  IF $IN[1] AND NOT LINKS THEN ;Eingang 1 gesetzt
    PTP {A1 45}
    LINKS=TRUE ;in linker Stellung
    MITTE=FALSE
    RECHTS=FALSE
  ELSE
    IF $IN[2] AND NOT MITTE THEN ;Eingang 2 gesetzt
      PTP {A1 0}
      LINKS=FALSE
      MITTE=TRUE ;in mittlerer Stellung
      RECHTS=FALSE
    ELSE
      IF $IN[3] AND NOT RECHTS THEN ;Eingang 3 gesetzt
        PTP {A1 -45}
        LINKS=FALSE
        MITTE=FALSE
        RECHTS=TRUE ;in rechter Stellung
      ENDIF
    ENDIF
  ENDIF
ENDWHILE
PTP HOME
END
    
```

**Programmbeispiel 2**

```

DEF BINSIG ( )
:----- Deklarationsteil -----
EXT BAS (BAS_COMMAND:IN,REAL:IN) ;Das externe Unterprogramm BAS
;wird deklariert.
DECL AXIS HOME
SIGNAL ABRUCH $IN[16] ;ABBRUCH wird dem Eingang 16 zugewiesen
SIGNAL LINKS $OUT[13] ;LINKS wird dem Ausgang 13 zugewiesen
    
```

**Anwendungsbeispiel 1: Verpacken und Abstapeln von Fensterbänken**

Diese Anlage wird von der Fa. Isovolta in Wr. Neudorf eingesetzt.

Ich möchte mich an dieser Stelle für die freundliche Unterstützung und für die Genehmigung bedanken, diese Anlage als Beispiel anführen zu dürfen.



Bild 1: Totalansicht der Anlage

**Beschreibung**

Die fertigen Fensterbänke gelangen von den Fertigungsmaschinen über eine Rollbahn in einen Pufferspeicher.

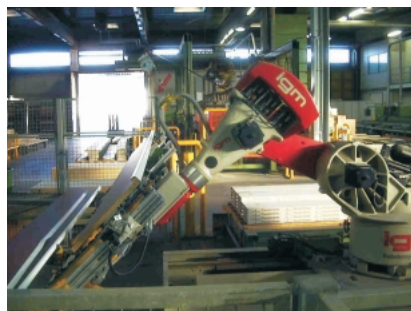


Bild 2: Roboter greift ein Brett vom Wender

Die letzte Position ist so ausgeführt, dass sie um 45° aufgeklappt werden kann. Somit kann der Roboter mit seinem Sauggreifer diese Fensterbank von unten nehmen und sie umgedreht auf die vorletzte Bank legen.



Bild 3: Roboter hat das Brett vom Wender geholt und wartet auf das nächste Brett um es drauflegen zu können.

Nun holt der Roboter mit dem Sauggreifer vom Kartonstapel einen neuen Karton und legt ihn in den Verpacker.



Bild 4: Roboter holt einen Karton

Jetzt werden von der vorletzten Pufferposition die zusammgelegten Fensterbänke mit dem Sauggreifer, der jetzt zusätzlich mit Greifhaken unterstützt wird, abgeholt und in den Verpacker eingelegt.



Bild 5: Roboter greift die zusammgelegten Fensterbänke



Bild 6: Roboter legt Fensterbänke in den Verpacker

Im Verpacker werden die Fensterbänke verpackt. Danach werden die fertigen Pakete auf Paletten abgestapelt.



Bild 7: Roboter legt Paket auf einer Palette ab

Die Fensterbänke haben eine Breite von 150mm bis 805mm und eine Länge von 2000mm bis 4100mm. Daraus ergeben sich extreme Größen- und Gewichtsunterschiede. Eingesetzt ist ein Kuka Industrieroboter KR200 mit 6-Achsen auf einer Fahrbahn mit einer Traglast von 200kg. Da der Greifer 80kg wiegt ergibt das eine maximale Teilelast von 120kg. Der Greifer besteht aus einem Grundgestell auf dem mehrere Sauger und Greifhaken angebracht sind. Der Greifer lässt sich in seinen Abmessungen so verstellen, dass alle Plattentypen gegriffen werden können. Die Fahrbahn ist notwendig um alle Stationen der Anlage erreichen zu können. Die Robotersteuerung muss neben dem Roboter mit einer externen Achse und Greifer auch die gesamte Peripherie mit insgesamt etwa 300 Ein- und Ausgangssignalen steuern. Daher ist das Anwendungsprogramm bestehend aus Bahnprogramm und Softsps der komplexeste Teil der Anlage. Es wurde so gestaltet, dass die komplexe Funktionalität mit der großen Typenvielfalt hinter einer einfachen Bedienung verborgen wird. Grundvoraussetzung für jede erfolgreiche Anlage ist natürlich eine reibungslos funktionierende Mechanik. Generell kommt natürlich bei Anlagen mit Industrierobotern in der Planungsphase den Bewegungsstudien eine große Bedeutung zu. Nur so ist es möglich die Erreichbarkeit der einzelnen Position zu kontrollieren.

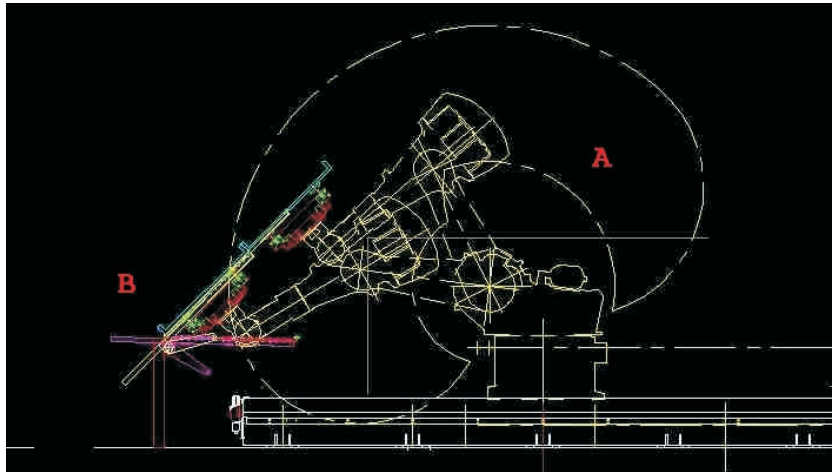


Bild 9: Untersuchung zur Abnahme der Platten vom Wender. Ersichtlich ist die Hüllkurve der maximalen Reichweite des Roboters. A: Hüllkurve der maximalen Reichweite, B: Wender

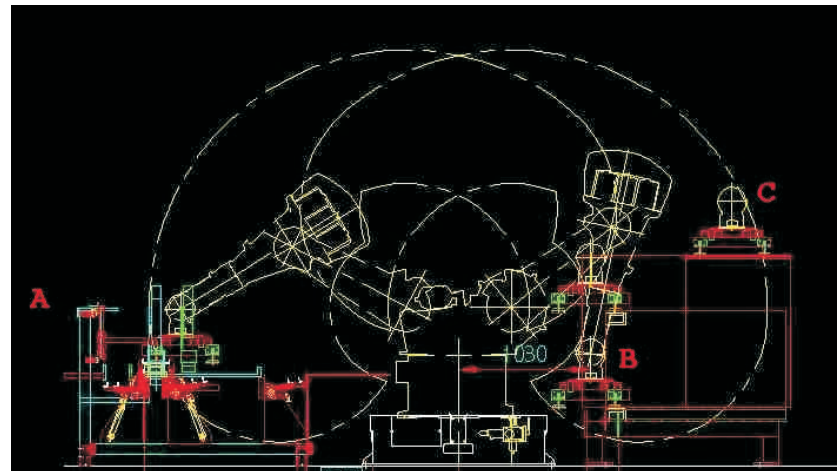


Bild 10: Untersuchung zum Einlegen in den Verpacker und zum Ablegen auf den Palettenplätzen mit den maximalen Ablegepositionen. A: Verpacker, B: engste Stapelposition am Palettenplatz, C: weiteste Stapelposition am Palettenplatz

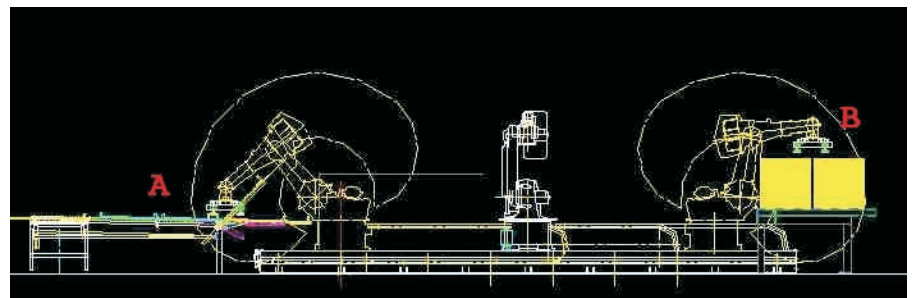


Bild 11: Untersuchung zur Entnahme der zusammengelegten Platten vom Pufferspeicher und zur Abnahme des Kartons vom Kartonplatz. A: Pufferspeicher, B: Kartonstapel

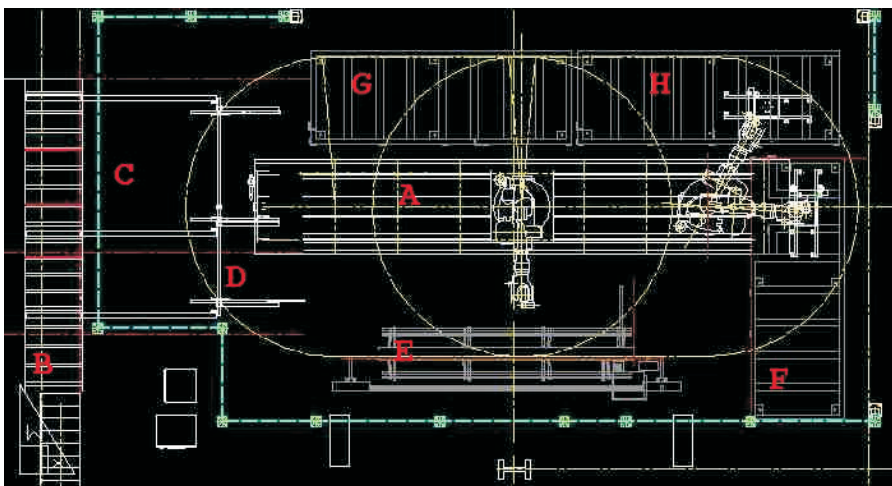


Bild 8: Layout: A: Fahrbahn, E: Verpacker, B: Zuförderrollenbahn, F: Kartonplatz, C: Pufferspeicher, G: Palettenplatz 1, D: Wender, H: Palettenplatz 2

http://www.excon.at/

**Software @ PC-Systeme @ Netzwerklösungen @ Beratung**

**Faktura, Lager, Fibu e-business**

1090 Wien, Rögergasse 6-8  
 Tel: +43/1/3109974-0  
 Fax: +43/1/3109974-14  
 EMail: office@excon.at  
 http: www.excon.at

## Anwendungsbeispiel 2: Schweißen von Auspuffanlagen

Diese Unterlagen wurden freundlicherweise von Firma igm-Robotersysteme zur Verfügung gestellt.

### Beschreibung

Ein hängender igm Schweißroboter RT330 schweißt in einer Station eines Taktisches (Drehtisch) Auspuffanlagen. In der zweiten Station des Taktisches kann währenddessen die Anlage ent- und beladen werden. Damit wird die Produktivität der Anlage wesentlich erhöht. Die Werkstücke werden je nach Länge auf eine Vorrichtung zwischen zwei Planscheiben aufgenommen oder an der angetriebenen Planscheibe befestigt. Der Manipulator der einen Seite besteht aus einer für den Roboter zusätzlichen externen rotatorischen Achse, die zweite Seite ist freilaufend. Somit ist es möglich, das Werkstück in die jeweils für den Schweißprozess optimale Position zu drehen.

**Legende für Bild 1 und 2:** A: Hängender igm Schweißroboter RT330., B: Taktisch oder auch Drehtisch, C: angetriebene Wendevorrichtung, D: freilaufende Wendevorrichtung

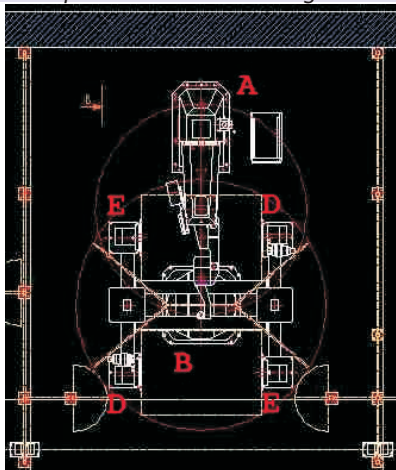


Bild 1: Layout der Anlage im Grundriss

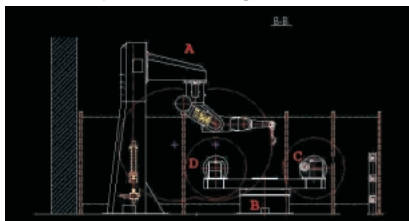


Bild 2: Layout im Aufriss

### Mit freundlicher Unterstützung von

- **igm-Robotersysteme AG**  
A-2355 Wr. Neudorf  
<http://www.igm.at/>
- **Isovolta**  
A-2355 Wr. Neudorf  
<http://www.isovolta.com/>
- **Kuka-Roboter GmbH**  
D-86165 Augsburg  
<http://www.kuka-roboter.de/>
- **sps-technik**  
A-4300 St. Valentin  
<http://www.sps.at/>
- **Rob-An-Tec**  
A-4050 Traun  
<http://www.rob-an-tec.com/>
- **Robotix**  
A-3400 Klosterneuburg  
<http://www.robotix.at/>

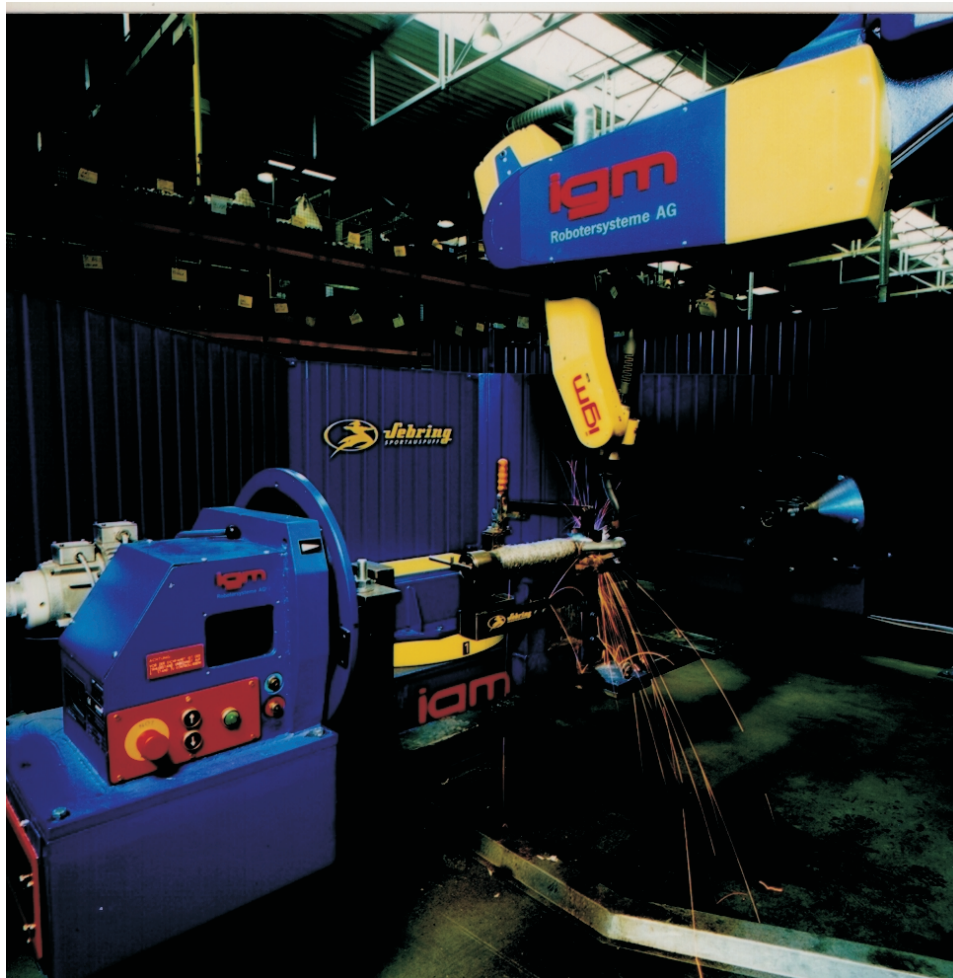


Bild 3: Schweißen einer Auspuffanlage

### Literatur- und Quellverweis

Industrieroboter für Klein- und Mittelbetriebe; Band 222 der Schriftenreihe des WIFI

Steuerungstechnik für Industrieroboter, Dr. Ing. Gerd Schneider, KUKA Roboter GmbH, Augsburg

Bedienungs- und Programmieranleitung der Kuka KRC1 Steuerung, KUKA Roboter GmbH, Augsburg