

Modalità di programmazione

- On-line programming
 - skill
 - down-time
 - accurate
- Off-line programming (OLP)
 - graphic tools
 - less down-time
 - less accurate



M. Strano, La programmazione, Lez. 8

p.1

Modalità di programmazione

- Off-line programming
 - vantaggi
 - Effective programming of program logics
 - calculations with state-of-the-art debugging facilities
 - Process support tools for instance selection of welding parameters
 - Verification of program through simulation and visualization
 - Well documented through simulation model with appropriate programs
 - Reuse of existing CAD data
 - Cost independent of production
 - Production can continue while programming
 - svantaggi
 - Investimento iniziale nel software e per costi/tempi di apprendimento
 - Necessità di calibrazione

M. Strano, La programmazione, Lez. 8

p.2

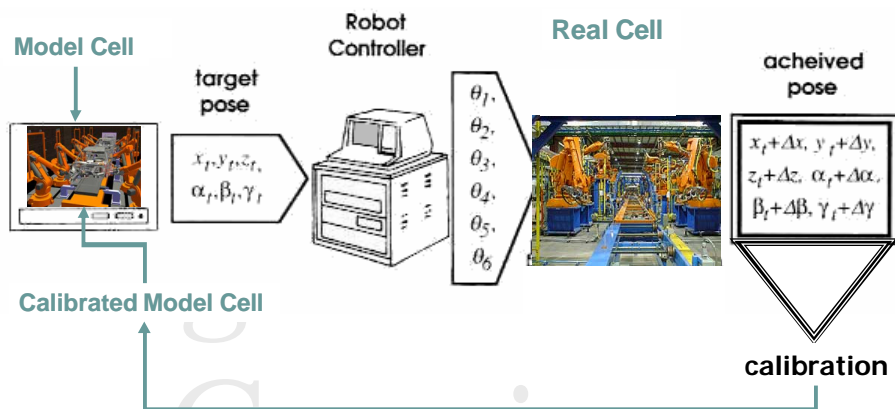
Modalità di programmazione

- A robot program consists mainly of two parts
 - Locations
 - (position and alignment)
 - program logics
 - (controller structures, communication, calculations)
- **hybrid programming**
 - OFF LINE
 - The program logics
 - effective debugging and simulation facilities are available
 - The major part of movement commands
 - reuse of CAD data and interaction of the programmer
 - ON LINE
 - Movement commands to locating the placement of the piece in the robot's workcell

M. Strano, La programmazione, Lez. 8

p.3

Programmazione OFF LINE dei robot

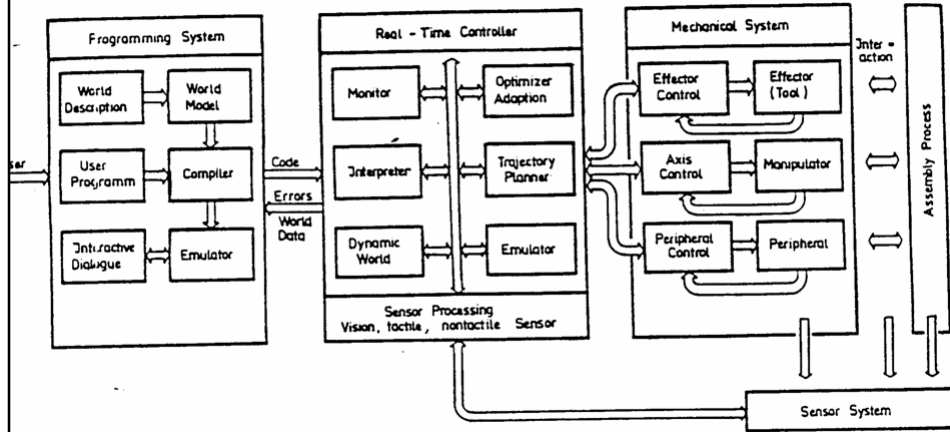


Off-line programming

M. Strano, La programmazione, Lez. 8

p.4

Programmazione OFF LINE dei robot

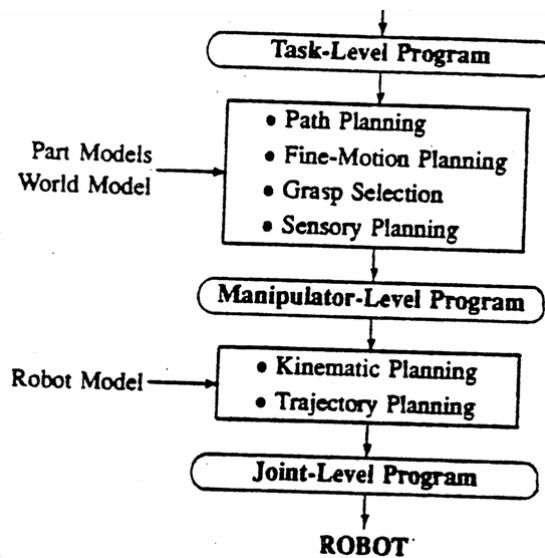


M. Strano, La programmazione, Lez. 8

p.5

Programmazione OFF LINE dei robot

- Schema del processo di programmazione



M. Strano, La programmazione, Lez. 8

p.6

Programmazione OFF LINE dei robot

Passi da compiere

1. Definire il prodotto
2. Definire il ciclo (e verificare il carico)
3. Definire lo spazio di lavoro
4. Definire i punti di riferimento per il robot (keypoints)
5. Scrivere il programma
6. Definire i punti di riferimento effettivi
7. Verificare il programma

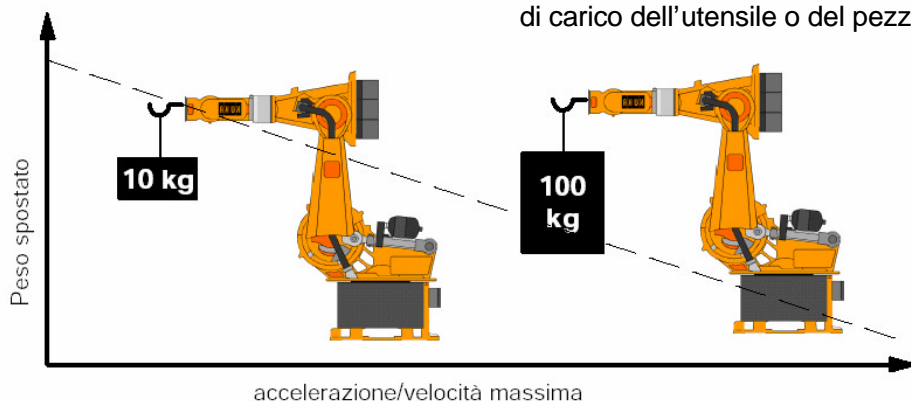


M. Strano, La programmazione, Lez. 8

p.7

Programmazione OFF LINE dei robot

Dati di carico



Il controllo deve disporre del peso, del baricentro e del momento di inerzia della massa dell'utensile o del pezzo.

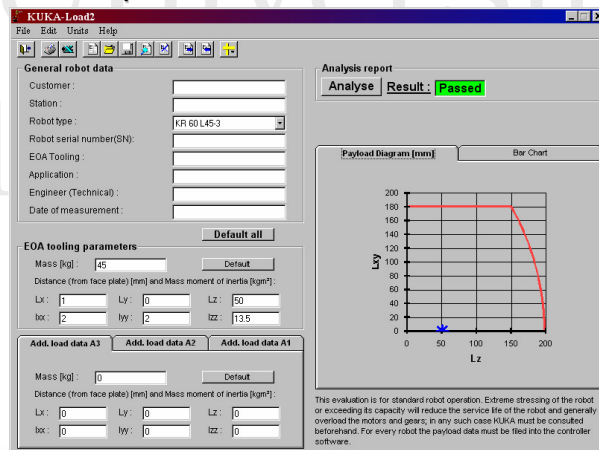
M. Strano, La programmazione, Lez. 8

p.8

Programmazione OFF LINE dei robot

verificare il carico

- payload is specified as the maximum value before serious performance loss



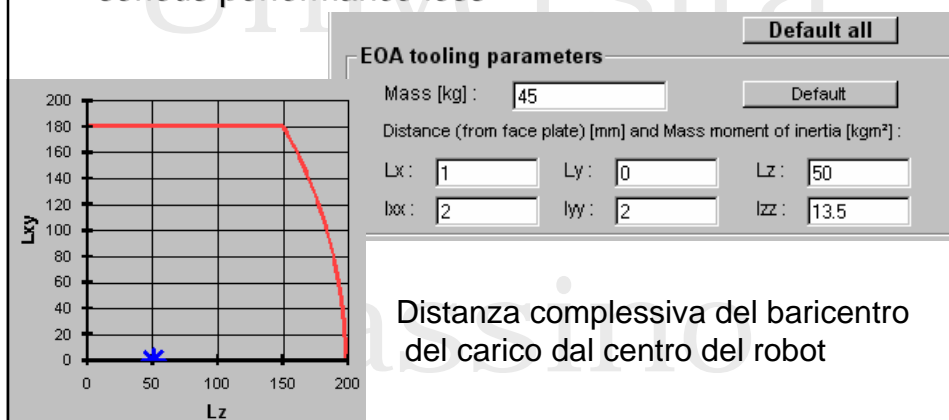
M. Strano, La programmazione, Lez. 8

p.9

Programmazione OFF LINE dei robot

verificare il carico

- payload is specified as the maximum value before serious performance loss



M. Strano, La programmazione, Lez. 8

p.10

Programmazione OFF LINE dei robot

verificare il carico

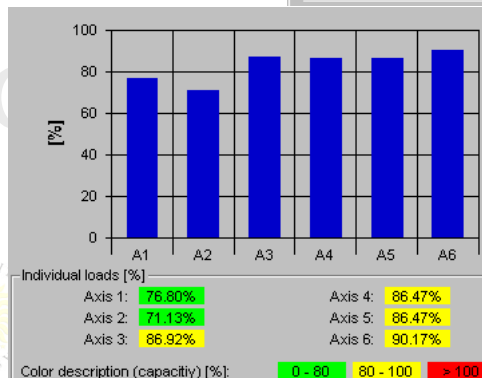
EOA tooling parameters

Mass [kg] :

Distance (from face plate) [mm] and Mass moment of inertia [kgm²] :

Lx : Ly : Lz :

Ixx : Iyy : Izz :



Carico per ogni asse

M. Strano, La programmazione, Lez. 8

p.11

Programmazione OFF LINE dei robot

verificare il carico

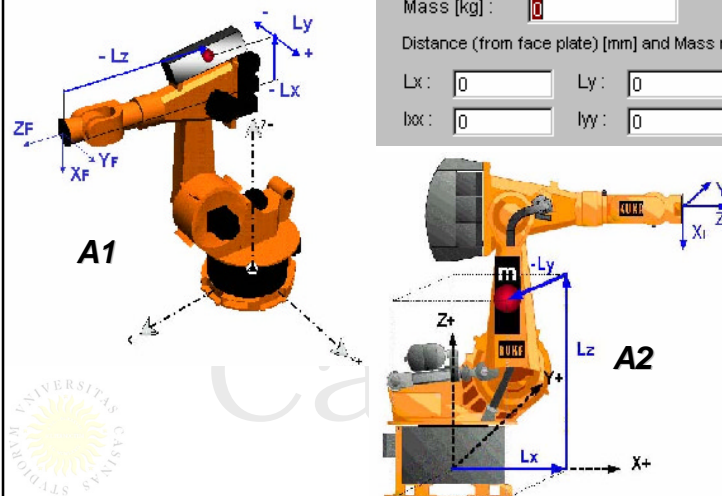
Add. load data A3 Add. load data A2 Add. load data A1

Mass [kg] :

Distance (from face plate) [mm] and Mass moment of inertia [kgm²] :

Lx : Ly : Lz :

Ixx : Iyy : Izz :

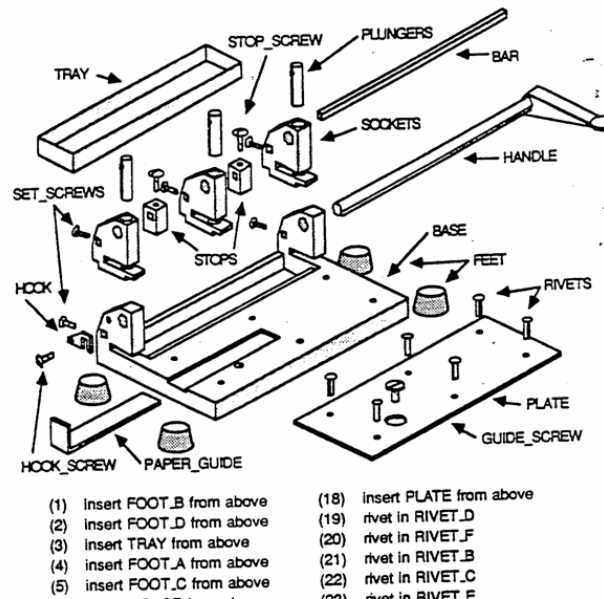


Carichi
addizionali
per asse

M. Strano, La programmazione, Lez. 8

p.12

Programmazione TASK LEVEL



M. Strano, La programmazione, Lez. 8

p.13

Programmazione MANIPULATOR LEVEL

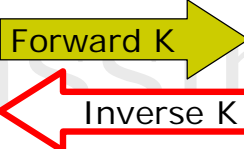
Need to find the Inverse Kinematics

Tipi di controllo

- Point-to-point
- Linear interpolation
- Circular interpolation
- Complex curve interpolation
 - B-spline
 - Cubic spline

Link Space

n variables
($q_1 \dots q_n$)



Tool Space

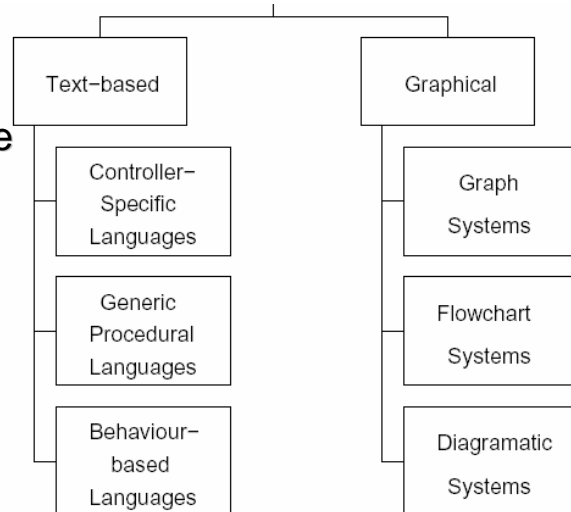
6 variables
(x, y, z, q_x, q_y, q_z)

M. Strano, La programmazione, Lez. 8

p.14

Programmazione MANIPULATOR LEVEL

- Metodi di programmazione off-line manuale



M. Strano, La programmazione, Lez. 8

p.15

Programmazione MANIPULATOR LEVEL

- Esempio: AML, della IBM

```
MOVE (JZ, 3.0);
MOVE (<JR,JP,JW>, <-45.0, 0.0, 0.0>)
```

Comandi di movimento

```
M=MONITOR (<SLP,SRP>, 1, 0, F);
```

Gestione sensori

```
IF ... THEN... ELSE
WHILE... DO...
```

Controllo flusso sequenziale

```
=
+ - * /
AND OR NOT XOR
```

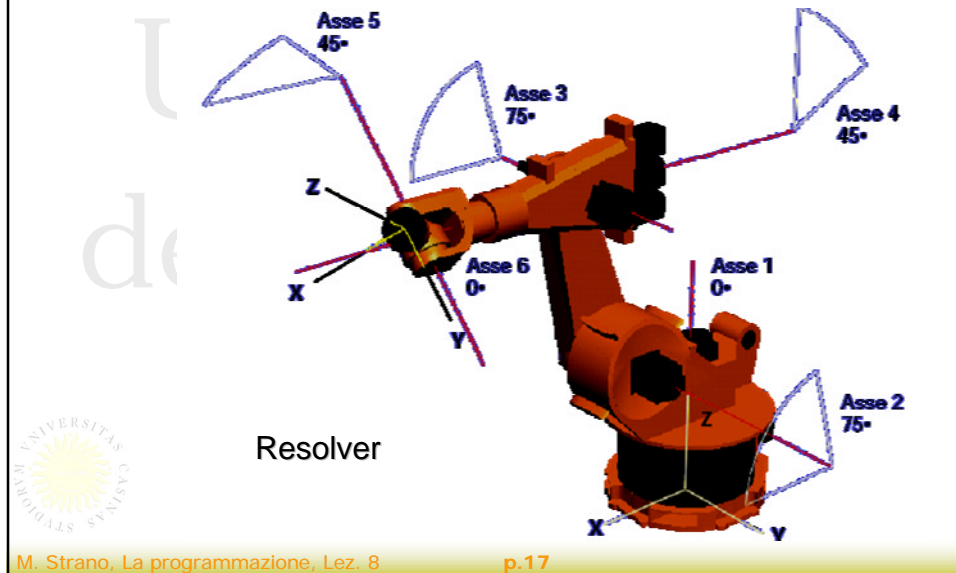
Operatori logici, aritmetici, relazionali

M. Strano, La programmazione, Lez. 8

p.16

Il robot KUKA

- Sistemi di coordinate e assi di rotazione



M. Strano, La programmazione, Lez. 8

p.17

Sistemi di coordinate e assi di rotazione

- Calibrazione del robot

- gli assi vengono portati in una posizione meccanica definita, la cosiddetta posizione **zero meccanica**.
- correlazione con l'angolo di azionamento asse e viene definita tramite una tacca di misura o un contrassegno
- viene memorizzato il valore assoluto del trasduttore per ogni asse

Una calibrazione dev'essere effettuata...

...dopo riparazioni
(ad es. sostituzione del motore di azionamento o RDW)

...se il robot è stato spostato senza controllo (ad es. con la manovella)

...dopo il tamponamento di un arresto di finecorsa meccanico con una velocità superiore a quella manuale (20 cm/s)

...dopo la collisione dell'utensile o robot con un pezzo in lavorazione

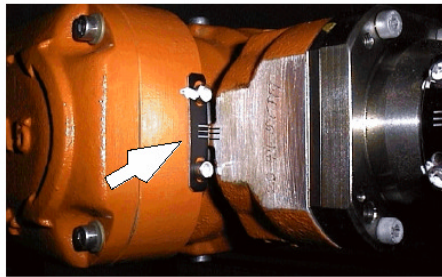
M. Strano, La programmazione, Lez. 8

p.18

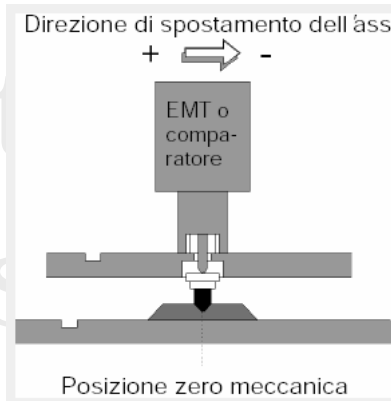
Sistemi di coordinate e assi di rotazione

• Calibrazione del robot

- Per il preciso raggiungimento della posizione zero meccanica viene impiegato un nonio, un comparatore o un tastatore elettronico (EMT)



Nonio sull'asse 5



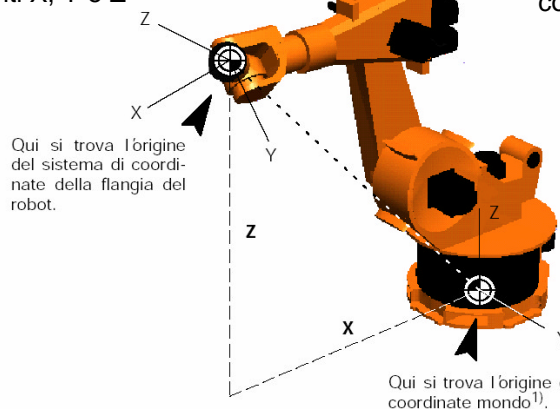
M. Strano, La programmazione, Lez. 8

p.19

Sistemi di coordinate

La **posizione** del centro della flangia viene descritta tramite la sua distanza dall'origine del sistema di coordinate mondo, suddivisa nelle componenti X, Y e Z

L'**orientamento** del sistema di coordinate della flangia del robot la cui origine si trova nel centro della flangia viene descritto tramite la rotazione rispetto al sistema di coordinate mondo



Qui si trova l'origine del sistema di coordinate della flangia del robot.

Qui si trova l'origine del sistema di coordinate mondo¹⁾.

¹⁾ Nell'impostazione base il sistema di coordinate mondo e il sistema di coordinate robot si trovano uno sopra l'altro.

M. Strano, La programmazione, Lez. 8

p.20

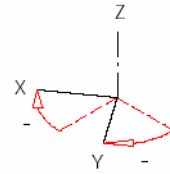
Sistemi di coordinate e assi

L'informazione che descrive un punto nello spazio tramite l'indicazione delle coordinate X, Y e Z e degli angoli di rotazione A, B e C viene denominata **FRAME**.



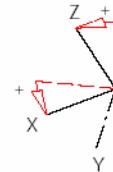
1. Rotazione intorno all'asse Z

Valore dell'angolo **A**



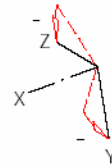
2. Rotazione intorno all'asse Y

Valore dell'angolo **B**



3. Rotazione intorno all'asse X

Valore dell'angolo **C**



M. Strano, La programmazione, Lez. 8

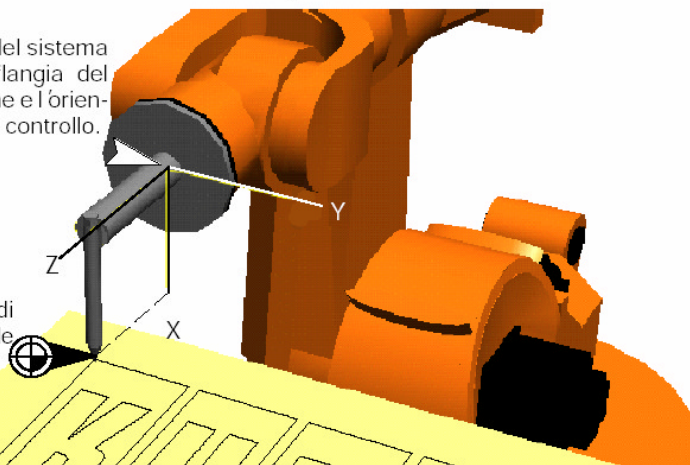
p.21

Sistemi di coordinate e assi di rotazione

Per poter calcolare la posizione del punto di riferimento di un utensile applicato alla flangia del robot oppure di un pezzo, il controllo del robot deve conoscere la sua posizione e l'orientamento rispetto al sistema di coordinate della flangia del robot.

Qui si trova l'origine del sistema di coordinate della flangia del robot. La sua posizione e l'orientamento sono note al controllo.

Questo è il punto di riferimento dell'utensile (Tool Center Point).

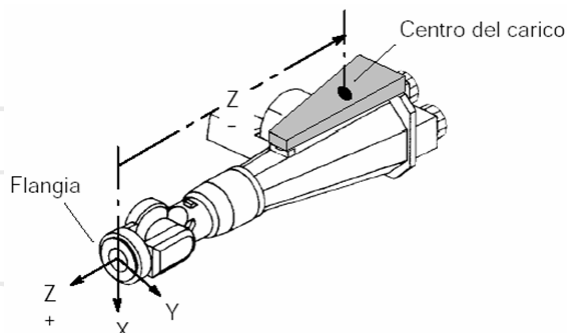


M. Strano, La programmazione, Lez. 8

p.22

Sistemi di coordinate e assi di rotazione

- Dati di carico



M	Peso del carico aggiuntivo.
X, Y, Z	Distanza tra il baricentro del carico aggiuntivo e l'origine del sistema di coordinate della flangia del robot (si trova <u>sul</u> centro della flangia) con riferimento al sistema di coordinate della flangia del robot.
A, B, C	Rotazione degli assi di inerzia principali del carico aggiuntivo (secondo gli angoli di Eulero <u>Z-Y-X</u>) rispetto al sistema di coordinate della flangia del robot.

M. Strano, La programmazione, Lez. 8

p.23

Sistemi di coordinate e assi di rotazione

- Misurazione dell'utensile

Programma	Misurazione tramite...
XYZ - 4-punti	Spostamento su un punto di riferimento fisso
XYZ - Riferimento	Spostamento con un utensile di riferimento noto
ABC - 2_punti	Spostamento su 2 punti con dati di orientamento
ABC - mondiale	Posizionamento verticale sul sistema di coordinate mondo
Registrazione numerica	Immissione dei dati dell'utensile
Dati di carico dell'utensile	Immissione della massa, del centro della massa, del momento di inerzia della massa



M. Strano, La programmazione, Lez. 8

p.24

Sistemi di coordinate e assi di rotazione

- Misurazione dell'utensile
 - tolleranze

TitleErrBounds	
TextTool	
TextMinToolDist	8
TextBase	
TextMinBaseDist	50
TextMinBaseAlpha	25
TextMaxToolErr	5

Distanza minima [TOOL]
Distanza minima per la misurazione dell'utensile. Gamma di valori da 0...200 mm.

Distanza minima [BASE]
Distanza minima per la misurazione della base. Gamma di valori da 0...200 mm.

Angolo minimo
Angolo minimo tra le rette attraverso i tre punti di misura per la misurazione della base. Gamma di valori da 0...360°.

Errore di misura massimo
Errore massimo del calcolo. Gamma di valori da 0...200 mm.

M. Strano, La programmazione, Lez. 8 p.25

Sistemi di coordinate e assi di rotazione

- Misurazione dell'utensile
 - Posizione: XYZ -- 4—punti
 - l'utensile col suo TCP (punto di riferimento dell'utensile) viene spostato da quattro direzioni (metodo "4 punti") su un punto di riferimento.



Cassino

Sistemi di coordinate e assi di rotazione

- Misurazione dell'utensile

- **Posizione: XYZ --riferimento**

- i dati di un utensile da misurare vengono rilevati tramite il confronto con un utensile noto spostandosi su un punto di riferimento



M. Strano, La programmazione, Lez. 8

p.27

Sistemi di coordinate e assi di rotazione

- Misurazione dell'utensile: rotazione

- **ABC -2_punti**

- se per il posizionamento e la guida è necessario un orientamento esatto dei tre assi dell'utensile. Esso richiede la presenza di punti contrassegnati sul lato positivo del piano XY e sul lato negativo dell'asse X dell'utensile

- **ABC – mondiale (5D)**

- se è necessaria soltanto la direzione di lavoro dell'utensile per il suo posizionamento e la sua guida (saldatura MIG/MAG, taglio con laser o con getto d'acqua).

- **ABC – mondiale (6D)**

- se è necessario l'orientamento di tutti i tre assi dell'utensile per il posizionamento e la guida (ad es. pinze di saldatura, gripper, ugelli di adesivo, ecc.)



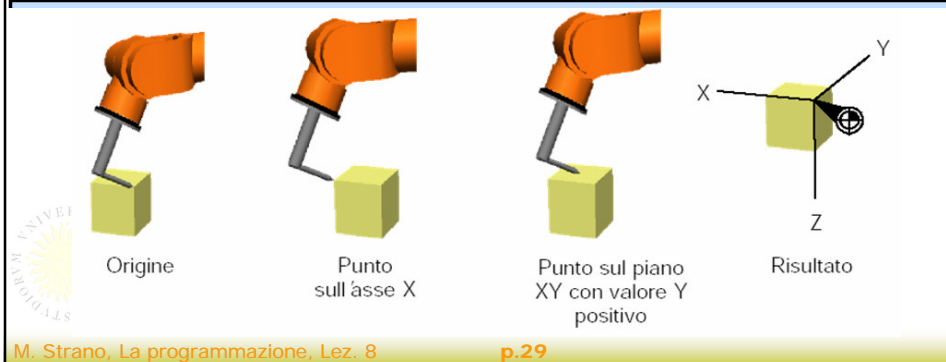
M. Strano, La programmazione, Lez. 8

p.28

Sistemi di coordinate e assi di rotazione

- Base (riferimento pezzo)

Programma	Misurazione tramite...
3-punti	Spostamento sul punto di riferimento di un pezzo in lavorazione
Indiretto	Indicazione del punto di riferimento non raggiungibile di un pezzo in lavorazione
Immissione numerica	Immissione manuale di un punto di riferimento



Sistemi di coordinate e assi di rotazione

- I sistemi di coordinate

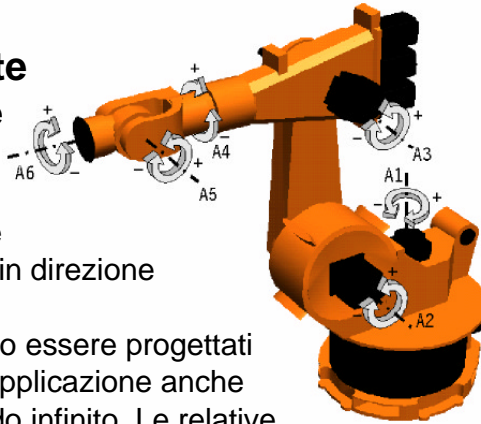
- **specifico per asse**
 - Ogni asse del robot da **A1** ad **A6** può essere spostato singolarmente in direzione positiva o negativa
- **TOOL**
 - Un sistema di coordinate ortogonale che ha la sua origine nella punta dell'utensile.
- **BASE**
 - Sistema di coordinate ortogonale che ha la sua origine sul pezzo in lavorazione
- **WORLD**
 - Sistema di coordinate ortogonale fisso che ha la sua origine nel piede del robot

Sistemi di coordinate e assi di rotazione

• I sistemi di coordinate

• Sistema di coordinate specifico per asse

- Ogni asse del robot da A1 ad A6 può essere spostato singolarmente in direzione positiva o negativa
- Gli assi A4 e A6 possono essere progettati per determinati casi di applicazione anche come assi rotanti in modo infinito. Le relative impostazioni vengono effettuate nel file "\$MACHINE.DAT"
- Gli assi A1 e A3 sono assi rotanti in modo finito.



M. Strano, La programmazione, Lez. 8

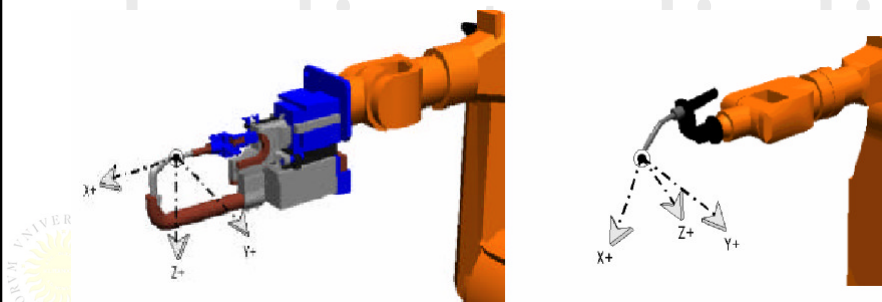
p.31

Sistemi di coordinate e assi di rotazione

• I sistemi di coordinate

• Sistema di coordinate TOOL

- Un sistema di coordinate ortogonale che ha la sua origine nella punta dell'utensile (TCP)
 - Il sistema TOOL segue sempre il movimento dell'utensile.



M. Strano, La programmazione, Lez. 8

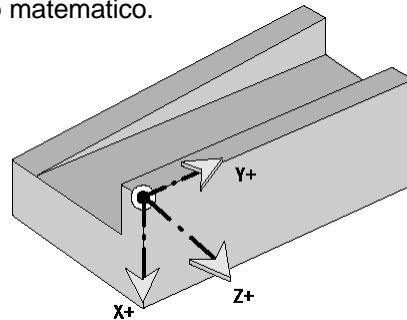
p.32

Sistemi di coordinate e assi di rotazione

- I sistemi di coordinate

- Sistema di coordinate BASE

- Sistema di coordinate ortogonale che ha la sua origine sul pezzo in lavorazione
 - Il sistema di coordinate BASE si sposta soltanto insieme ad un pezzo se questo è applicato su una cinematica esterna, accoppiata in modo matematico.



M. Strano, La programmazione, Lez. 8

p.33

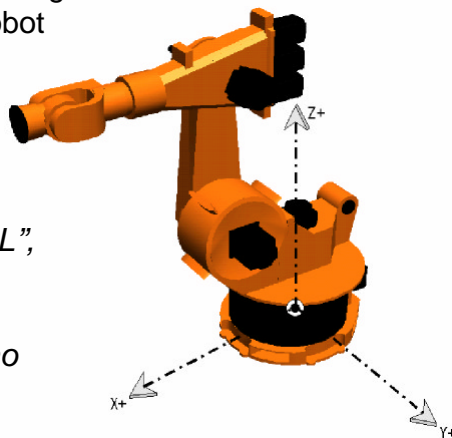
Sistemi di coordinate e assi di rotazione

- I sistemi di coordinate

- Sistema di coordinate WORLD

- Sistema di coordinate ortogonale fisso che ha la sua origine nel piede del robot

In caso di selezione del sistema di coordinate "TOOL", "BASE" o "WORLD", normalmente vengono spostati più assi in sincrono



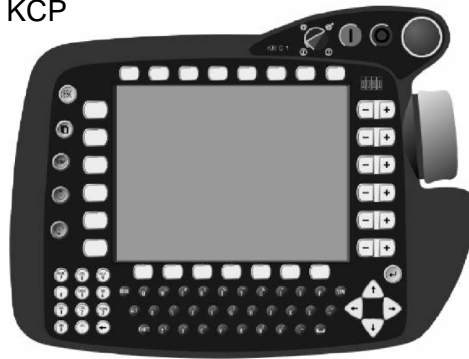
M. Strano, La programmazione, Lez. 8

p.34

Il linguaggio KUKA KRL

- **Il control panel**

- Il KUKA Control Panel, denominato "KCP", costituisce l'interfaccia tra uomo e macchina per il comando semplice del controllo robot KR C1.
- Tutti gli elementi per la programmazione e il comando del sistema robot ad eccezione dell'interruttore principale sono alloggiati direttamente sul KCP



M. Strano, La programmazione, Lez. 8

p.35

Il linguaggio KUKA KRL

- **Spostamento manuale del robot**

- Lo spostamento manuale serve per il movimento del robot sotto comando manuale, ad esempio per l'insegnamento di punti di destinazione o per portare il robot fuori ingombro dopo che un asse del robot ha urtato contro uno dei fine corsa software



- **Spostamento con lo "space--mouse"**

- in base all'impostazione dei gradi di libertà contemporaneamente tre oppure sei assi



- **Spostamento con i tasti di spostamento**

- ogni asse singolarmente



- **Spostamento manuale disattivato**

M. Strano, La programmazione, Lez. 8

p.36

Il linguaggio KUKA KRL

Visualizzazione cartesiana

Posizione: [mm]	
X	<input type="text" value="0"/>
Y	<input type="text" value="0"/>
Z	<input type="text" value="0"/>

Orientamento: [deg]	
A	<input type="text" value="0"/>
B	<input type="text" value="0"/>
C	<input type="text" value="0"/>

Nel caso di questo tipo di visualizzazione viene indicata la posizione del punto di riferimento dell'utensile (TCP) rispetto al punto di riferimento del pezzo in lavorazione (BASE) e la rotazione tra i due sistemi di coordinate.

Visualizzazione riferita agli assi

Angolo [deg]	
A1	<input type="text" value="0.000000"/>
A2	<input type="text" value="0.000000"/>
A3	<input type="text" value="0.000000"/>

[deg]	
A4	<input type="text" value="0.000000"/>
A5	<input type="text" value="0.000000"/>
A6	<input type="text" value="0.000000"/>

In questo caso viene visualizzata la rotazione di ogni asse del robot rispetto al suo punto zero meccanico rilevato durante la calibrazione.

Visualizzazione incrementale

Incrementi: [Incr]	
I1	<input type="text" value="0"/>
I2	<input type="text" value="0"/>
I3	<input type="text" value="0"/>

[Incr]	
I4	<input type="text" value="0"/>
I5	<input type="text" value="0"/>
I6	<input type="text" value="0"/>

In questo caso vengono visualizzati i momenti cinetici forniti dagli azionamenti degli assi.

M. Strano, La programmazione, Lez. 8

p.37

Il linguaggio KUKA KRL

• Esecuzione, arresto e reset di un programma

• Scelta e selezione di un programma

- Scegliere col tasto cursore “~” oder “-” il programma desiderato ed azionare il softkey “Selezionare”(in basso a sinistra del display). Appare il testo del programma

• Esecuzione manuale del programma



disinserire lo spostamento manuale



passo per passo (una frase di movimento dopo l'altra)



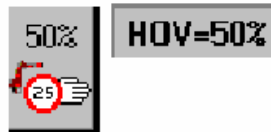
eseguito completamente (impostazione “Go”)

M. Strano, La programmazione, Lez. 8

p.38

Il linguaggio KUKA KRL

- **Esecuzione, arresto e reset di un programma**
 - Ridurre la velocità di spostamento nel funzionamento in manuale (Override manuale)



M. Strano, La programmazione, Lez. 8

p.39

Il linguaggio KUKA KRL

- **Istruzioni di programma**
 0. **Ultima istruzione**
 - Comando per l'inserimento dell'istruzione eseguita per ultima;
 1. **Movimento**
 - Consente la programmazione di movimenti PTP, LIN e CIRC;
 2. **Logica**
 - Programmazione di istruzioni logiche e di tempi di attesa, di funzioni di comando e di impulso in base al percorso, settaggio e interrogazione di uscite ed ingressi;
 3. **CmdAnalog (Uscita analogica)**
 - Settaggio delle uscite analogiche sotto controllo del programma;
 4. **Commento**
 - Inserimenti di commenti nei moduli di programma;
 5. **KRL assistant**
 - Programmazione KRL di funzioni speciali supportata da sintassi

M. Strano, La programmazione, Lez. 8

p.40

Il linguaggio KUKA KRL

- Istruzioni di programma

1. Movimento

Movimenti standard	
PTP (da punto a punto)	L'utensile viene spostato lungo la traiettoria più veloce su un punto di destinazione
LIN (lineare)	Guida dell'utensile con velocità definita lungo una retta
CIRC (circolare)	Movimento dell'utensile con velocità definita lungo una curva
In caso di più comandi di movimento successivi esistono due possibilità per definire come deve avvenire il movimento tra i singoli punti:	
Movimento tra i singoli punti	
Arresto preciso	Il punto programmato viene raggiunto esattamente.
Approssimazione (Cont)	Un movimento può essere raccordato in modo morbido con un altro, il punto di destinazione non viene raggiunto esattamente.

M. Strano, La programmazione, Lez. 8

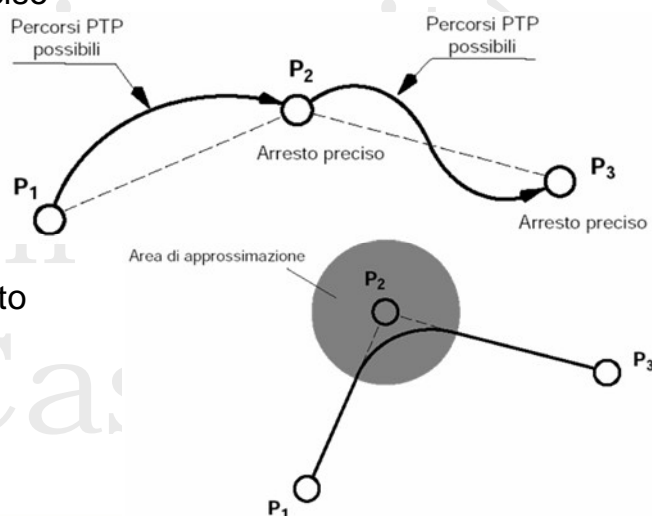
p.41

Il linguaggio KUKA KRL

1. Movimento PTP

- arresto preciso

- arresto approssimato



M. Strano, La programmazione, Lez. 8

p.42

Il linguaggio KUKA KRL

1. Movimento PTP

Campo	Funzione	Gamma valori
PTP	Tipo di movimento	PTP, LIN, CIRC
P1	Denominazione punto	
Tool	N° utensile	Nullframe, Tool_Data[1]...[16]
Base	N° pezzo	Nullframe, Base_Data[1]...[16]
external TCP	Il robot guida l'utensile/il pezzo	True, False
CONT	Approssimazione inserita	" ", Cont
Vel=100%	Velocità	da 1 a 100% del valore massimo (Valore default 100%)
PDAT1	Parametri di movimento	
Acceleration	Accelerazione	0 ... 100
Approximation Distance *1	Area di approssimazione	0 ... 100
*1 Viene proposto soltanto se è stato inserito "CONT"		

M. Strano, La programmazione, Lez. 8

p.43

Il linguaggio KUKA KRL

1. Movimento lineare (LIN)

Campo	Funzione	Gamma valori
LIN	Tipo di movimento	PTP, LIN, CIRC
P1	Denominazione punto	
Tool	N° utensile	Nullframe, Tool_Data[1]...[16]
Base	N° pezzo	Nullframe, Base_Data[1]...[16]
external TCP	Il robot guida l'utensile/il pezzo	True, False
CONT	Approssimazione inserita	" ", Cont
Vel=2m/s	Velocità	0,001 ... 2m/s (Valore predefinito 2m/s)
CPDAT1	Parametri di movimento	
Acceleration	Accelerazione	0 ... 100
Approximation Distance *1	Area di approssimazione	0 ... 100
*1 Viene proposto soltanto se è stato inserito "CONT"		

M. Strano, La programmazione, Lez. 8

p.44

Il linguaggio KUKA KRL

1. Movimento circolare (CIRC)

- La traiettoria viene determinata tramite il punto iniziale, il punto ausiliario e il punto di destinazione
- I 3 punti individuano un piano. Affinché il controllo possa determinare questo piano in modo più preciso possibile, questi tre punti dovrebbero trovarsi distanti uno dall'altro.



M. Strano, La programmazione, Lez. 8

p.45

Il linguaggio KUKA KRL

1. Movimento circolare (CIRC)

Campo	Funzione	Gamma valori
CIRC	Tipo di movimento	PTP, LIN, CIRC
P1	Denominazione del punto ausiliario	
P2	Denominazione punto	
Tool	N° utensile	Nullframe, Tool_Data[1]...[16]
Base	N° pezzo	Nullframe, Base_Data[1]...[16]
external TCP	Il robot guida l'utensile/il pezzo	True, False
CONT	Approssimazione inserita	"", Cont
Vel=2m/s	Velocità	0,001 ... 2m/s (Valore predefinito 2m/s)
PDAT1	Parametri di movimento	
Acceleration	Accelerazione	0 ... 100
Approximation Distance *1	Area di approssimazione	0 ... 100

*1 Viene proposto soltanto se è stato inserito "CONT"

M. Strano, La programmazione, Lez. 8

p.46

Il linguaggio KUKA KRL

2. Logica

- Funzioni di attesa
 - Funzione di attesa in base al tempo (WAIT)
 - In secondi
 - Funzione di attesa in base al segnale (WAITFOR)
 - Wait for in
 - Wait for out
- Funzioni di commutazione
 - Funzione di commutazione semplice (OUT)
 - Funzione impulso semplice (PULSE)
 - Funzione di commutazione in base al percorso (SYN OUT)
 - Funzione impulso in base al percorso (SYN PULSE)
- Accoppiamento e disaccoppiamento del segmento INTERBUS (IBUS--Seg on/off)

M. Strano, La programmazione, Lez. 8

p.47

Il linguaggio KUKA KRL

4. Commento

- Per rendere i programmi più chiari si consiglia di inserire delle righe di commento che contengono dei testi illustrativi sui moduli del programma
 - La riga di commento viene preceduta automaticamente da un punto e virgola (;).



M. Strano, La programmazione, Lez. 8

p.48

Esempi di programma

3 modi di esecuzione di un programma

- ISTEP (Incremental Step, frase singola)
 - Il programma si arresta dopo ogni riga di programma
- MSTEP (Motion Step, passo di programma)
 - Il programma si arresta dopo ogni riga di movimento
- GO
 - Il programma si arresta solo alla fine se non ci sono istruzioni WAIT

Questa indicazione viene data in un file di configurazione del robot vale per tutti i programmi

Il linguaggio KUKA KRL

I files

- Struttura del programma *.src
 - DEF NAME()
 - Dichiarazioni
 - Sono lette dal robot tutte insieme all'inizio del programma
 - Si devono sempre trovare in testa al file
 - Istruzioni
 - Sono lette dinamicamente riga per riga
 - Sottoprogrammi
 - END
- 2 tipi di file
 - *.src (file programma vero e proprio)
 - *.dat (eventuale lista dati, contiene solo dichiarazioni)

Esempi di programma

Possibili dichiarazioni

- INT, REAL, CHAR, BOOL
 - Intero, reale, carattere, booleano
- **Strutture dati predefinite**
 - **POS**
 - Posizione + correttori Status S e Turn T (posizione assiale univoca)
 - REAL X, Y, Z, A, B, C, INT S, T
 - **FRAME**
 - Sistema locale di riferimento
 - REAL X, Y, Z, A, B, C
 - **AXIS**
 - Set di valori per gli assi
 - REAL A1, A2, A3, A4, A5, A6

Esempi di programma

Un programma molto semplice (1/2)

```
DEF PROG1()
;..... sezione delle dichiarazioni .....
INT J ; dichiaraz. di integer per la variabile J
;..... sezione delle istruzioni .....
$ VEL_AXIS[1]=100 ; determinazione velocità per asse
$ VEL_AXIS[2]=100
...
$ VEL_AXIS[6]=100

$ ACC_AXIS[1]=100 ; determinazione accelerazioni per asse
$ ACC_AXIS[2]=100
...
$ ACC_AXIS[6]=100
```

Esempi di programma

Un programma molto semplice (2/2)

;..... continua la sezione delle istruzioni

PTP {A1 0, A2 -90, A3 90, A4 0, A5 0, A6 0}

FOR J=1 TO 5

PTP {A1 45}

PTP {A2 -70, A3 50}

PTP {A1 0, A2 -90, A3 90}

ENDFOR

PTP_REL {A1 0, A2 -90, A3 90, A4 0, A5 0, A6 0} ; movimento
incrementale

END

Esempi di programma

Un programma più complesso (1/2)

DEF PROG2()

;..... sezione delle dichiarazioni

DECL AXIS HOME ; variabile HOME del tipo AXIS

DECL FRAME MYBASE[2] ; campo del tipo FRAME

;..... sezione dell'inizializzazione

;... inizial. veloc. e acceler.....

HOME= {AXIS: A1 0, A2 -90, A3 90, A4 0, A5 30, A6 0}

\$BASE={X 1000, Y 0, Z 1000, A0, B0, C0} ; sist. di coord. base

REF_POS={X 100, Y 0, Z0, A0, B0, C0} ; posizione di riferimento

MYBASE[2]={X0, Y200, Z250, A0, B90, C0}; sist. di coordinate locale

Esempi di programma

Un programma più complesso (2/2)

```
;..... sezione delle istruzioni .....  
PTP HOME; movimento fino a "casa" nel sistema di coordinate BASE  
{A1 0, A2 -90, A3 90, A4 0, A5 0, A6 0}  
PTP REF_POS      ; movimento nel sistema BASE fino a REF_POS  
PTP MYBASE[2]    ; movimento nel sistema BASE fino a MYBASE[2]  
PTP MYBASE[2]: REF_POS ; movimento, nel sistema MYBASE[2],  
                    fino a REF_POS
```



Esempi di programma

Un altro programma

```
DEF PROG3()  
;..... sezione delle dichiarazioni .....  
$BASE = $WORLD      ; il sistema base è al piede del robot  
$TOOL= $NULLFRAME   ; il TCP è al centro della flangia  
EXT BAS(BAS_COMMAND :IN, REAL :IN) ; programma standard  
                                esterno BAS.src  
  
;..... sezione dell'inizializzazione .....  
BAS(#INITMOV,0) ; ... inicial. veloc. e acceler.....  
;..... sezione delle istruzioni .....  
PTP {POS: X 1025, Y 0, Z 1480, A0, B90, C0, S 'B010', T 'B000010'} ;
```

