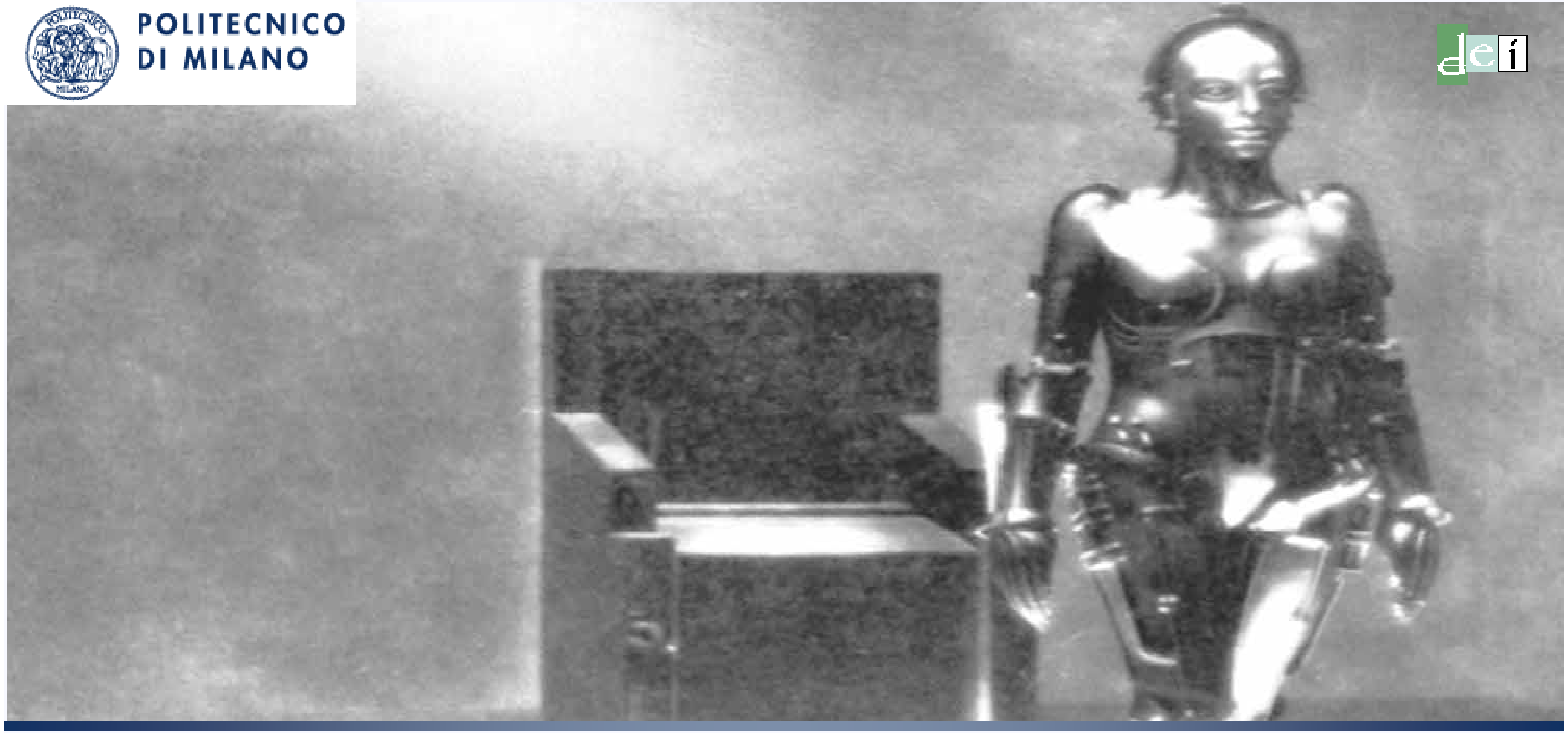
POLITECNICO DI MILANO

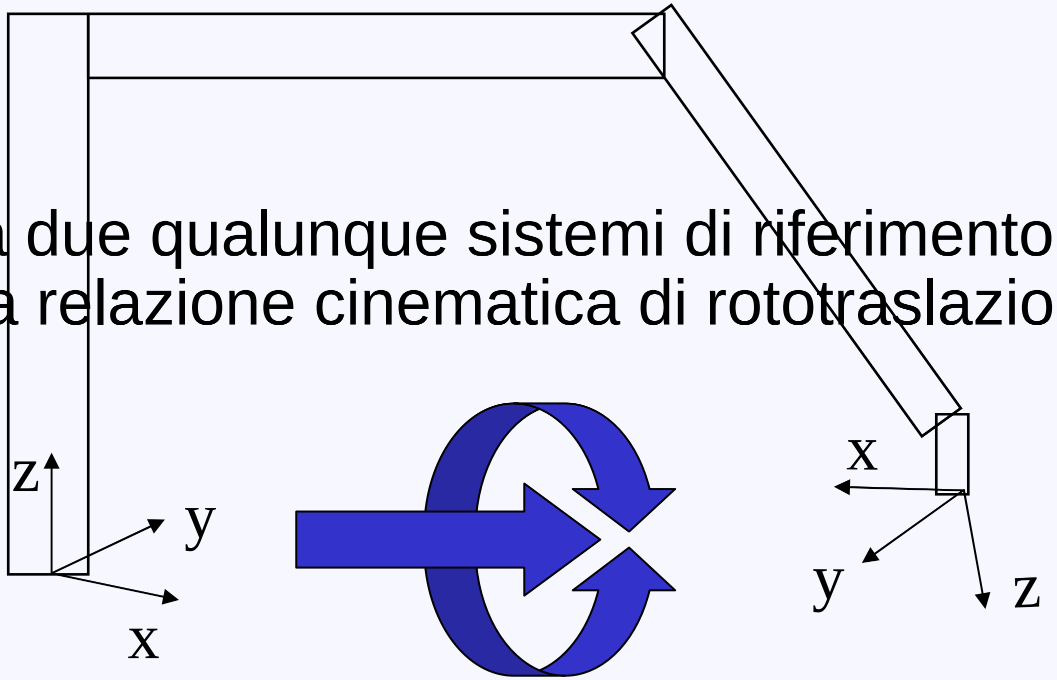


070342 – Robotica

http://home.dei.polimi.it/gini/robot/

Cinematica inversa

Relazioni cinematiche



– Fra due qualunque sistemi di riferimento esiste una relazione cinematica di rototraslazione

– Questa relazione è rappresentata da una matrice di trasformazione in coordinate omogenee.

– Per il manipolatore, la relazione fra il sistema 0 e la mano è espressa dalla matrice T

Cinematica inversa: problema

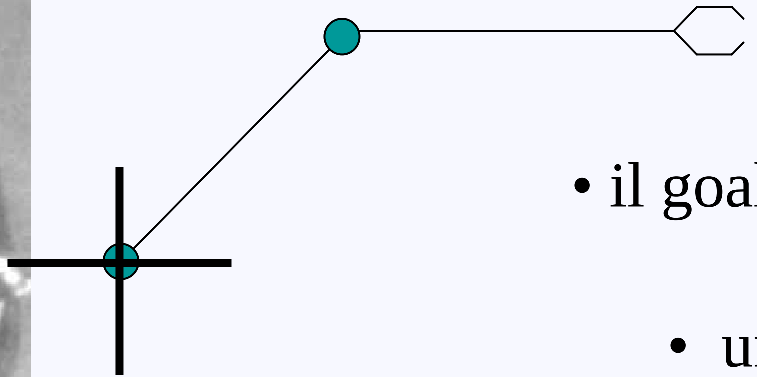
data una posizione ed un orientamento nello spazio cartesiano trovare una configurazione dei giunti che permetta di raggiungerla

- esistenza
- metodi di soluzione
- unicità
- uso real-time

esistenza

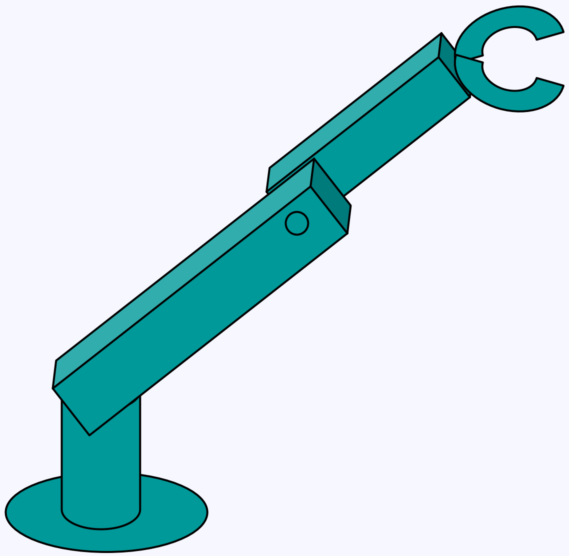
- se il punto da raggiungere è nello spazio di lavoro e il robot ha 6 gradi di libertà allora esiste una soluzione *(il robot è risolubile).*
- Se il robot ha meno di 6 gradi di libertà occorre verificare che il punto sia raggiungibile
  - Non è facile determinare con sicurezza con metodi geometrici se un punto è o meno nello spazio di lavoro raggiungibile, tanto meno in quello di destrezza.

Non esistenza



Goal

• il goal è fuori dallo spazio di lavoro



• un punto non raggiungibile può essere anche interno allo spazio di lavoro - vincoli fisici .

• un punto può essere non raggiungibile nell'eseguire una traiettoria - *punto singolare* - occorre accelerazione ∞ per seguire la traiettoria

Equazioni da risolvere

$$\mathbf{T_6} = \mathbf{A}_{0,1}^{0} \ \mathbf{...} \ \mathbf{A}_{5,6}^{5} = T\left(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6\right)$$

si eguaglia la matrice nota T della mano

- che contiene direttamente o indirettamente le 6 coordinate cartesiane

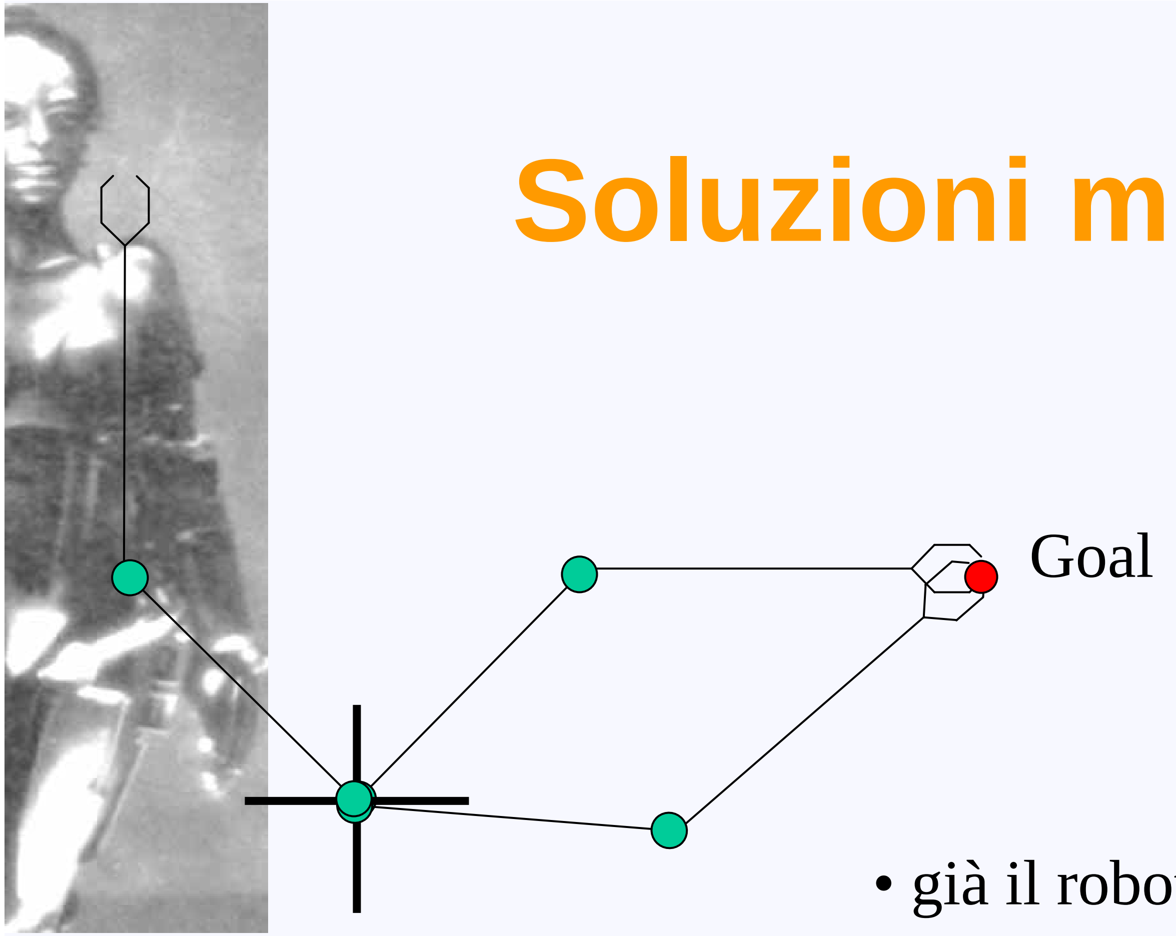
con la sua espressione simbolica

- che contiene il prodotto delle matrici A che dipendono dalle 6 variabili di giunto

si ottengono 12 equazioni in 6 incognite. Le 9 equazioni della rotazione non sono indipendenti quindi si ottengono:

- *6 equazioni trascendenti in 6 incognite (le 6 variabili di giunto)*

Soluzioni multiple



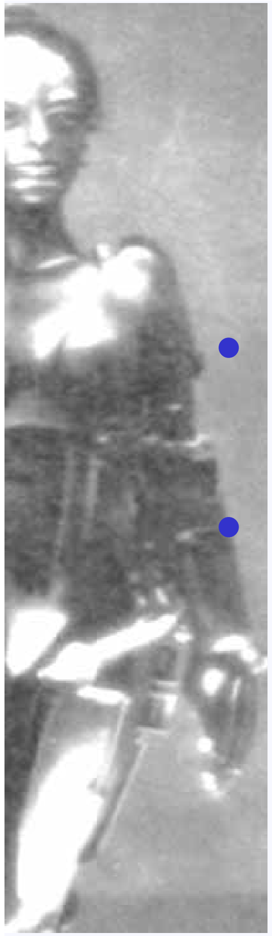
- già il robot planare RR ha due soluzioni
- il problema aumenta col crescere dei gdl
- ridondanza dei movimenti

Soluzioni multiple 2

- Il numero delle soluzioni dipende dal numero di parametri D-H $\neq 0$
  - max = 16 per 6R
- Nel caso ci siano più soluzioni queste interessano **tutte** per scegliere la migliore da attuare.
  - Es: minimizzare i movimenti dei giunti facendo una somma pesata degli spostamenti, assegnando pesi maggiori ai giunti più vicini alla base. La somma minima trovata individua la soluzione da attuare.
- *La molteplicità delle soluzioni talvolta è utile per permettere al robot di evitare ostacoli.*

Metodi di soluzione principali

- Soluzioni in Forma chiusa (espressioni)
  - Geometrici
  - Algebrici
- Soluzioni iterative (procedimenti iterativi)



Metodi per forma chiusa

- ***metodo di Paul***: pre-moltiplicazioni o post-moltiplicazioni con matrici di trasformazione.
- ***metodo di Pieper***: soluzione di un polinomio di 4° grado in una incognita e di un sistema in forma chiusa per le altre. Le due condizioni di Pieper sono CS per avere soluzioni in forma chiusa.
- ***Altri metodi***: geometrici, quaternioni, approcci iterativi, o algebrici basati sulle sostituzioni:
  – $u = \tan \theta/2$, $\cos \theta = (1-u^2)/(1 +u2)$, $\sin\theta = 2u/(1+u^2)$

Soluzioni in forma chiusa

Condizioni per trovare una soluzione in forma chiusa

⇒ *condizioni sufficienti* (Pieper e Ang) che dis-accoppiano gli assi.

⇒ Esse valgono per robot con

- tre giunti di traslazione
- tre di rotazione con gli assi che si intersecano in un punto (polso sferico PUMA)
- 2 di traslazione normali a uno di traslazione
- 1 di traslazione normale a 2 giunti paralleli
- 3 giunti di rotazione con assi paralleli (SCARA)

Soluzioni numeriche

- *Per robot che non presentano geometrie dis-accoppiate, la soluzione in forma chiusa può non esistere => soluzioni iterative*
  - m equazioni in n incognite
  - Si parte con una stima iniziale
  - Si calcola la matrice T con le stime e l'errore rispetto ai valori cartesiani noti
  - Si modifica la stima per ridurre l'errore
  - *tempo di esecuzione indefinito se si vuole un errore definito oppure un errore indefinito se si vuole un tempo di esecuzione definito.*

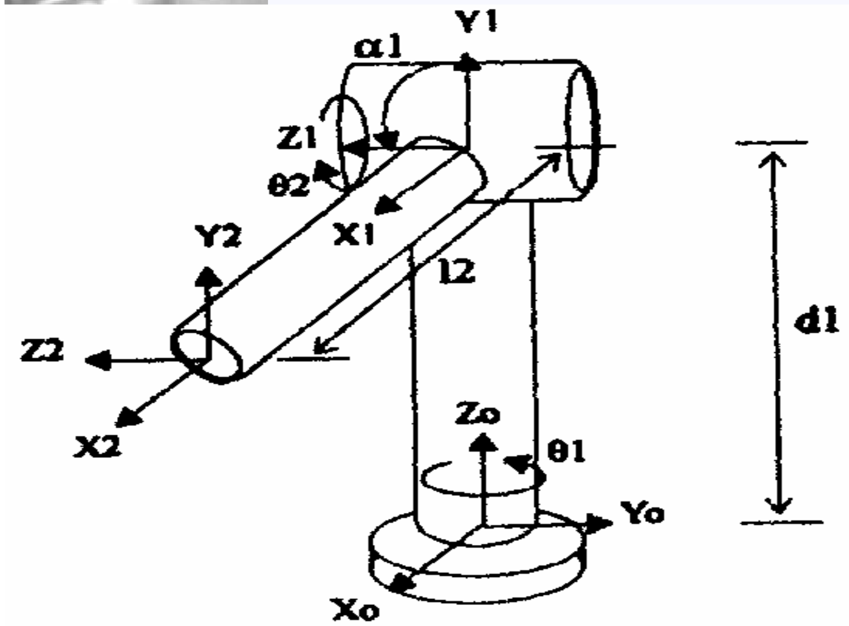
- *Per robot complesso soluzioni possono essere trovate mediante metodi di apprendimento (reti neurali, etc).*
- *Altre soluzioni si trovano usando lo Jacobiano.*

Metodo di Paul

1. Uguagliare la matrice T (cartesiana nota) alla matrice del manipolatore (con variabili di giunto)
2. Cercare nella seconda matrice:
   - elementi che contengono una sola variabile di giunto
   - coppie di elementi che danno un'espressione in una sola variabile di giunto quando vengono divisi fra loro
   - elementi che possono essere semplificati
3. Si uguagliano questi elementi ai corrispondenti ottenendo un'equazione. La soluzione da' un legame fra una variabile di giunto ed elementi noti di T.
4. Se non si identificano elementi al passo 2, si premoltiplica per l'inversa della **A** del primo link. Si ripete per tutti i link

Alternativamente postmoltiplicare per l'inversa della matrice A dell'ultimo link.

Esempio metodo Paul: RR



$$\mathbf{T}=\begin{bmatrix} C1C2 & -C1S2 & S1 & l_2C1C2 \\ S1C2 & -S1S2 & -C1 & l_2S1C2 \\ S2 & C2 & 0 & d_1+l_2S2 \\ 0 & 0 & 0 & 1 \end{bmatrix}=\begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

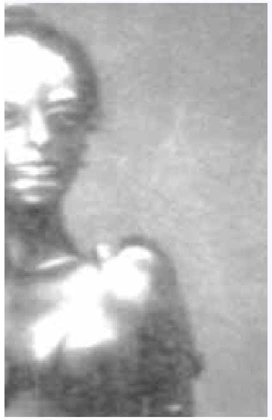
$$\frac{p_x}{p_y}=\frac{\cancel{l_2}\cdot\cos(\theta_1)\cdot\cancel{\cos(\theta_2)}}{\cancel{l_2}\cdot sen(\theta_1)\cdot\cancel{\cos(\theta_2)}}\Rightarrow\theta_1=\tan^{-1}\left(\frac{p_x}{p_y}\right)$$

premoltiplicazione per inversa di A0

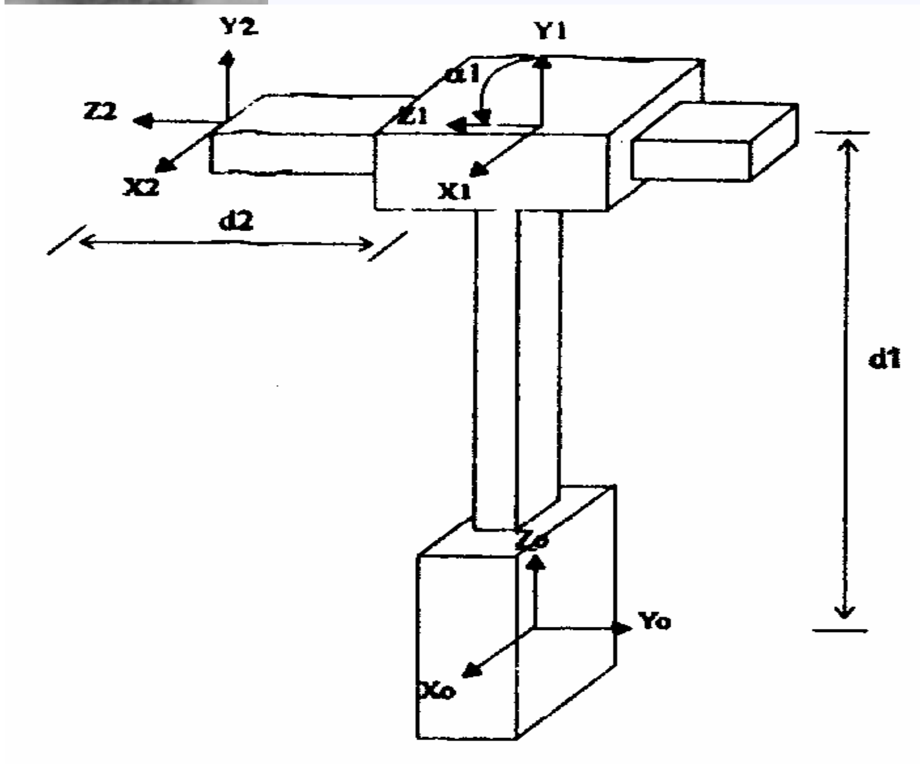
C2/S2 = (pxC1+pyS1)/(pz-d1) =>θ2= tan-1 [(pz-d1)/ (pxC1+pyS1)]

$$\left(\mathbf{A}_{0,1}^{0}\right)^{-1}\cdot\mathbf{T}=\mathbf{A}_{1,2}^{1}$$

$$\begin{bmatrix} n_xC1+n_yS1 & o_xC1+o_yS1 & a_xC1+a_yS1 & \boxed{p_xC1+p_yS1} \\ n_z & o_z & a_z & \boxed{p_z-d_1} \\ n_xS1-n_yC1 & o_xS1-o_yC1 & a_xS1-a_yC1 & p_xS1-p_yC1 \\ 0 & 0 & 0 & 1 \end{bmatrix}=\begin{bmatrix} C2 & -S2 & 0 & \boxed{l_2\cdot C2} \\ S2 & C2 & 0 & \boxed{l_2\cdot S2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Esempio TT



$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -d_2 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$d_2 = -p_y$$

$$d_1 = p_z$$

RRR planare

$$
= \begin{bmatrix} c1c2c3 - c1s2s3 - s1s2c3 - s1c2s3 & -c1c2s3 - c1s2c3 + s1s2s3 - s1c2c3 & 0 & c1(L2c2 + L1) - s1s2L2 \\ s1cc3 - s1s2s3 + c1s2c3 + c1c2s3 & -s1c2s3 - s1s2c3 - c1s2s3 + c1c2c3 & 0 & s1(L2c2 + L1) + c1s2L2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

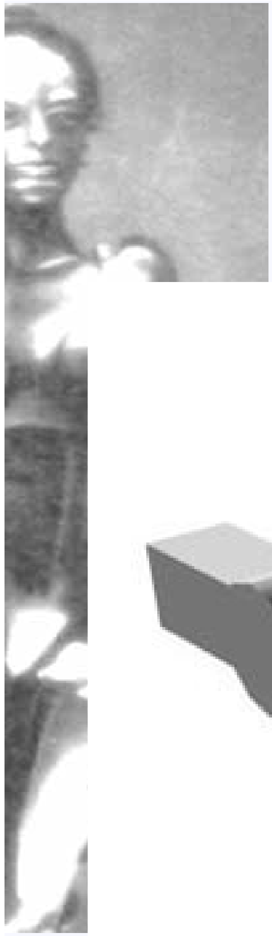
$$
= \begin{bmatrix} c_{123} & -s_{123} & 0 & L_1c_1 + L_2c_{12} \\ s_{123} & c_{123} & 0 & L_1s_1 + L_2s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_\phi & -s_\phi & 0 & x \\ s_\phi & c_\phi & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

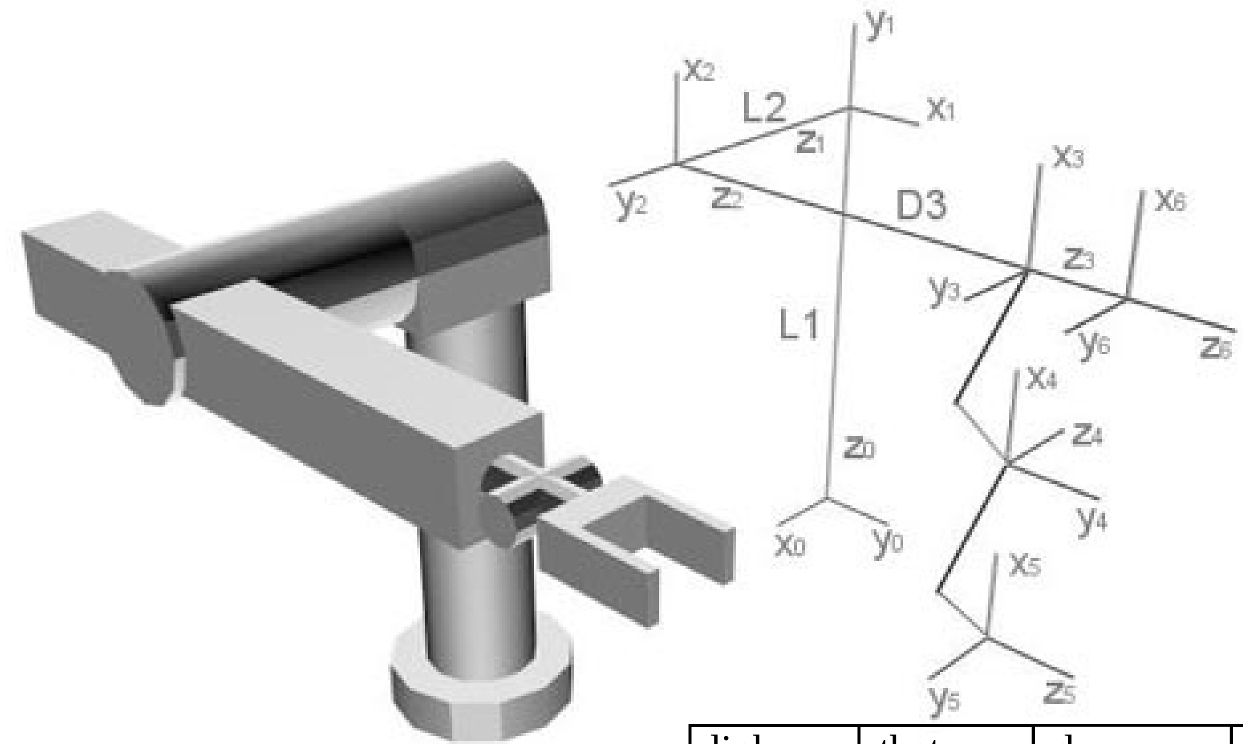
$$
c_\phi = c_{123}
$$
$$
s_\phi = s_{123}
$$
$$
x = l_1c_1 + l_2c_{12}
$$
$$
y = l_1s_1 + l_2s_{12}
$$

è una rotazione attorno
a z ed una traslazione

Variabili cartesiane:
x, y, ϕ

Stanford arm (polare 6gdl)





link	theta	d	a	alpha
1	-	L1	0	90
2	-	L2	0	90
3	0	-	0	0
4	-	0	0	90
5	-	0	0	-90
6	-	L6	0	0

inversa stanford arm

Con metodo di Paul:

$T = A_1A_2A_3A_4A_5A_6$

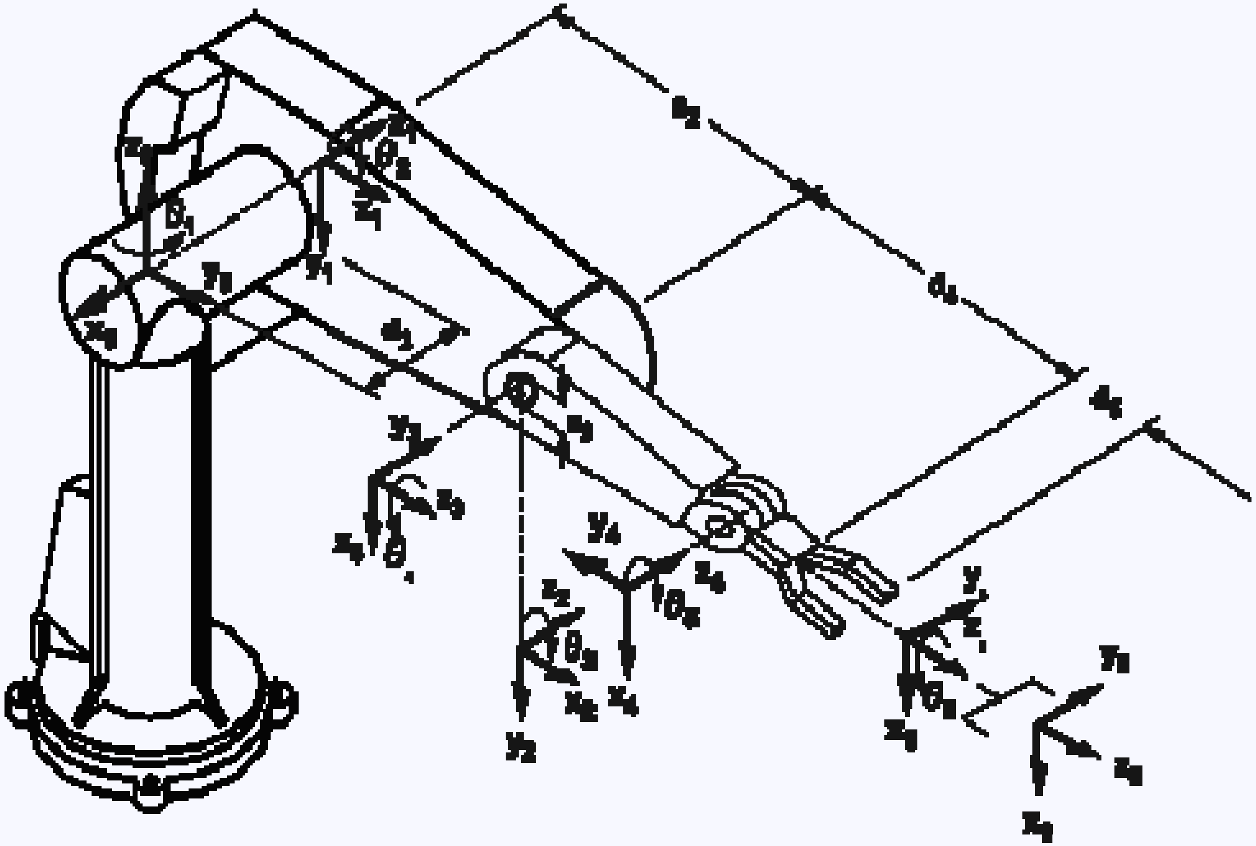
Si premoltiplica per l'inversa

- $T^1{}_6 = A_1{}^{-1}\, T = A_2A_3A_4A_5A_6$
- $T^2{}_6 = A_2{}^{-1}A_1{}^{-1}T = A_3A_4A_5A_6$
- $T^3{}_6 = A_3{}^{-1}\, A_2{}^{-1}A_1{}^{-1}T = A_4A_5A_6$
- $T^4{}_6 = A_4{}^{-1}A_3{}^{-1}\, A_2{}^{-1}A_1{}^{-1}T = A_5A_6$
- $T5_6 = A_5{}^{-1}A_4{}^{-1}A_3{}^{-1}\, A_2{}^{-1}A_1{}^{-1}T = A_6$

Puma 560- 6gdl

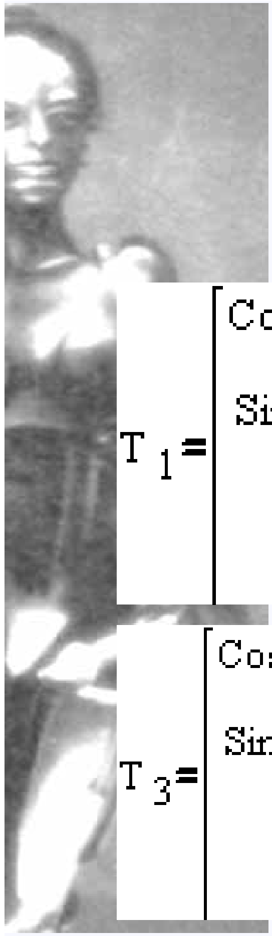
α

θ



Link	a	alpha	d	theta
1	0	90	0.67	-
2	0.4318	0	0	-
3	0.4318	-90	0.15005	-
4	0	90	0	-
5	0	-90	0	-
6	0	0	0	-

Matrici PUMA



$$T_1 = \begin{bmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) & 0 \\ \sin(\theta_1) & 0 & -\cos(\theta_1) & 0 \\ 0 & 1 & 0 & 0.67 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2 = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 0.4318\cdot\cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 0.4318\cdot\sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

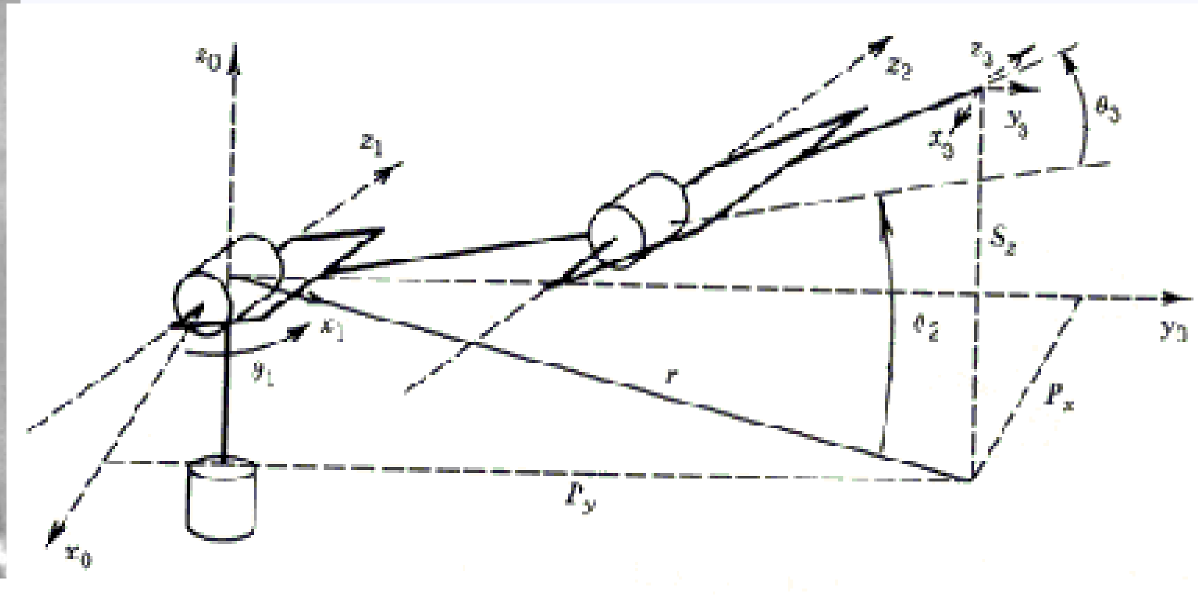
$$T_3 = \begin{bmatrix} \cos(\theta_3) & 0 & -\sin(\theta_3) & 0.4318\cdot\cos(\theta_3) \\ \sin(\theta_3) & 0 & \cos(\theta_3) & 0.4318\cdot\sin(\theta_3) \\ 0 & -1 & 0 & 0.15005 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4 = \begin{bmatrix} \cos(\theta_4) & 0 & \sin(\theta_4) & 0 \\ \sin(\theta_4) & 0 & -\cos(\theta_4) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

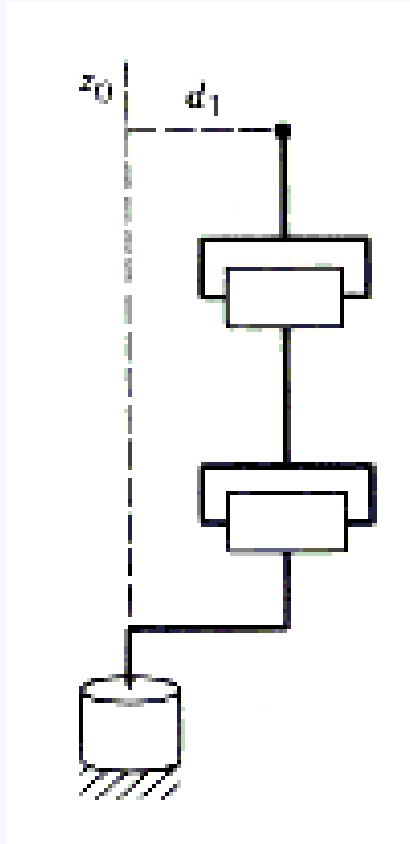
$$T_5 = \begin{bmatrix} \cos(\theta_5) & 0 & -\sin(\theta_5) & 0 \\ \sin(\theta_5) & 0 & \cos(\theta_5) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

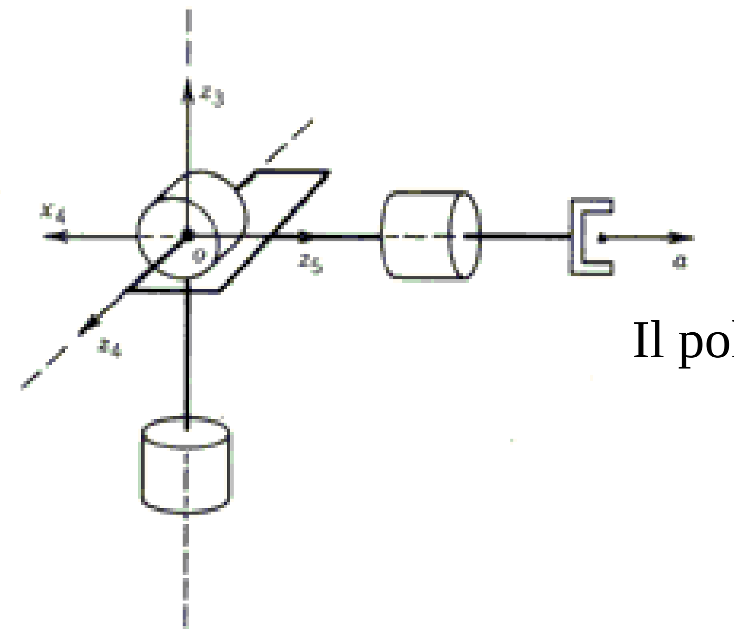
$$T_6 = \begin{bmatrix} \cos(\theta_6) & -\sin(\theta_6) & 0 & 0 \\ \sin(\theta_6) & \cos(\theta_6) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T = T_1 \cdot T_2 \cdot T_3 \cdot T_4 \cdot T_5 \cdot T_6$$



I primi tre link





Il polso sferico

Non allineamento su z0

Cinematica diretta Puma

- In T, in quarta colonna, abbiamo la posizione px, py, pz della frame 6 nella frame 0.

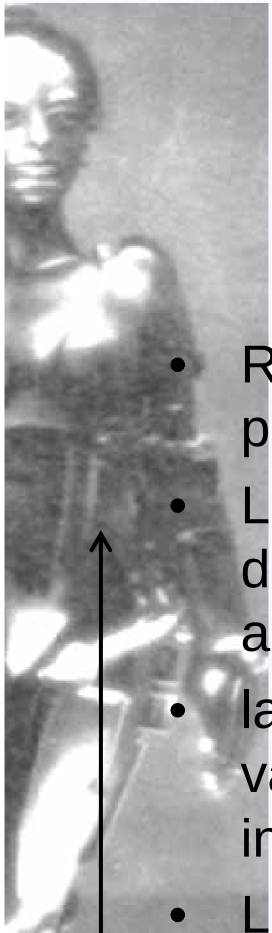
$$p_x = \mathrm{Cos}(\theta_1)\cdot\left[a_2\cdot \mathrm{Cos}(\theta_2) - a_3\cdot\left(\mathrm{Cos}(\theta_2)\cdot \mathrm{Sin}(\theta_3) + \mathrm{Sin}(\theta_2)\cdot \mathrm{Cos}(\theta_3)\right)\right] - d_3\cdot \mathrm{Sin}(\theta_1)$$

$$p_y = \mathrm{Sin}(\theta_1)\cdot\left[a_2\cdot \mathrm{Cos}(\theta_2) - a_3\cdot\left(\mathrm{Cos}(\theta_2)\cdot \mathrm{Sin}(\theta_3) + \mathrm{Sin}(\theta_2)\cdot \mathrm{Cos}(\theta_3)\right)\right] + d_3\cdot \mathrm{Cos}(\theta_1)$$

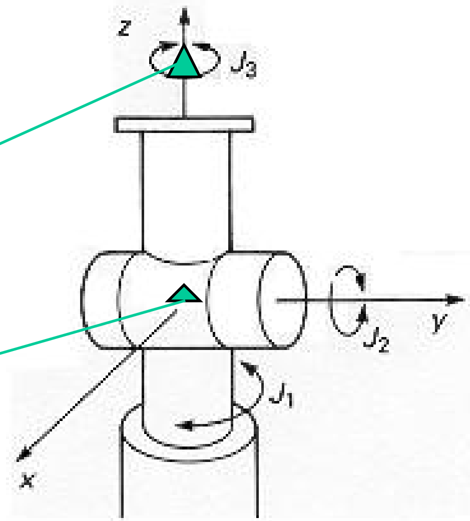
$$p_z = - a_2\cdot \mathrm{Sin}(\theta_2) - a_3\cdot\left(\mathrm{Cos}(\theta_2)\cdot \mathrm{Cos}(\theta_3) - \mathrm{Sin}(\theta_2)\cdot \mathrm{Sin}(\theta_3)\right) + d_1$$

- Per l’orientamento occorre estrarre angoli di Eulero.

Cinematica inversa
2 passi



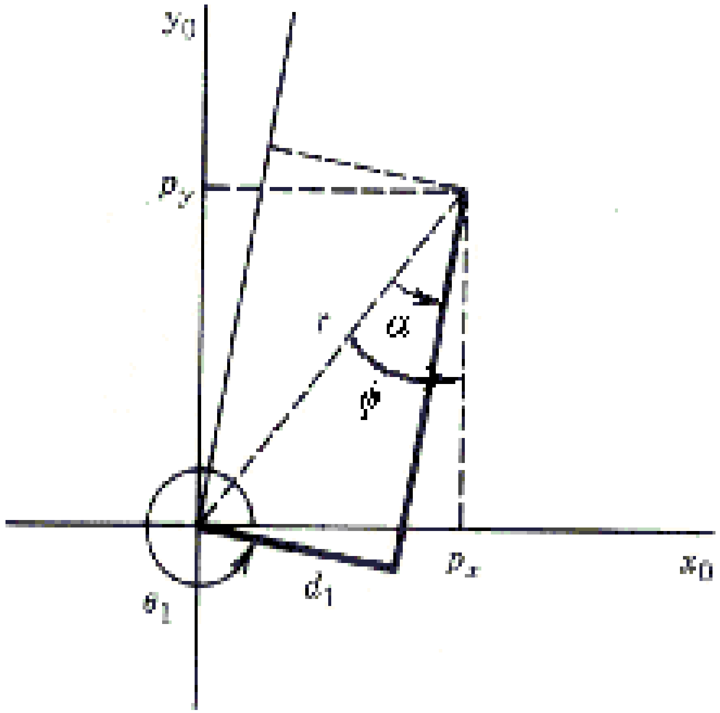
- Risolvere separatamente su primi 3 (posizione polso) ed ultimi 3 (orientamento polso) gdl
- La posizione del centro del polso, p_c si trova dalla posizione della mano e dalla direzione di approach z_6.
- la posizione dipende solo dalle prime tre variabili di giunto: ci sono 3 equazioni in tre incognite che risolviamo per θ_1 , θ_2. , θ_3.
- L'orientamento è funzione solo delle ultime tre variabili θ_4 , θ_5, θ_6



0

@ G. Gini 2009

Soluzione primo giunto



Braccio destro

guardando dall'alto il robot e allineando l2 e l3 come unico link, per raggiungere p_x e p_y del polso, dalla trigonometria risolviamo **θ1**

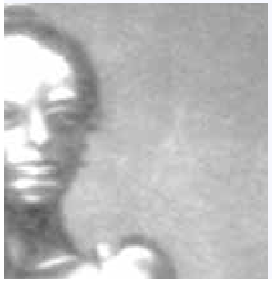
(due soluzioni - braccio destro o sinistro)

$$\theta_1 = \phi - \alpha$$

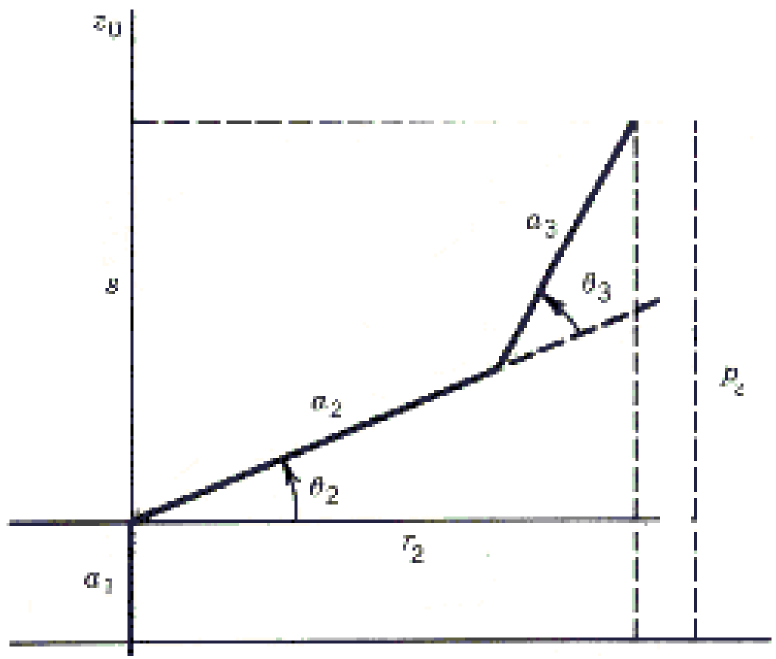
$$\phi = \mathrm{Atan}\left(\frac{p_y}{p_x}\right)$$

$$\alpha = \mathrm{Atan}\left(\frac{-\sqrt{r^2 - d_1^2}}{d_1}\right) = \mathrm{Atan}\left(\frac{\sqrt{p_x^2 + p_y^2 - d_1^2}}{d_1}\right)$$

$$\theta_1 = \mathrm{Atan}\left(\frac{p_y}{p_x}\right) - \mathrm{Atan}\left(\frac{\sqrt{p_x^2 + p_y^2 - d_1^2}}{d_1}\right)$$



Soluzione terzo giunto



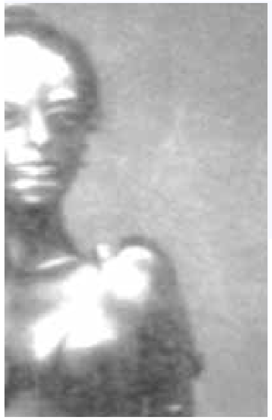
Consideriamo il piano in cui stanno il secondo e terzo link, risolviamo **θ3** (due soluzioni – gomito alto e basso - come RR).

$$\mathrm{Cos}(\theta_3) = \frac{r_2^2 + s^2 - a_2^2 - a_3^2}{2 \cdot a_2 \cdot a_3}$$

$$\mathrm{Cos}(\theta_3) = \frac{p_x^2 + p_y^2 - d_1^2 + p_z^2 - a_2^2 - a_3^2}{2 \cdot a_2} = D$$

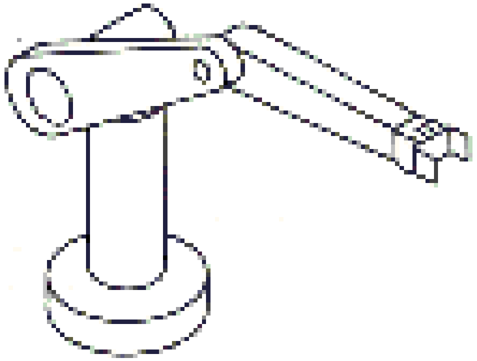
$$\theta_3 = \mathrm{Atan}\left[\frac{\sqrt{a_3^2 - \left(\frac{p_x^2 + p_y^2 - d_1^2 + p_z^2 - a_2^2 - a_3^2}{2 \cdot a_2}\right)^2}}{\frac{p_x^2 + p_y^2 - d_1^2 + p_z^2 - a_2^2 - a_3^2}{2 \cdot a_2}}\right]$$

$$\theta_3 = \mathrm{Atan}\left(\frac{\sqrt{a_3^2 - D^2}}{D}\right)$$



Soluzione secondo giunto

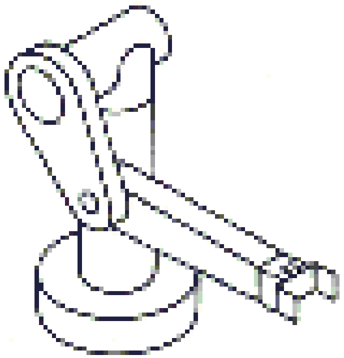
$$\theta_{23} = \mathrm{Atan}\left[\frac{p_z \cdot \left(-a_2 \cdot \mathrm{Cos}(\theta_3)\right) - \left(\mathrm{Cos}(\theta_1) \cdot p_x + \mathrm{Sin}(\theta_1) \cdot p_y\right) \cdot \left(a_3 - a_2 \cdot \mathrm{Sin}(\theta_3)\right)}{p_z \cdot \left(a_2 \cdot \mathrm{Sin}(\theta_3) - a_3\right) - a_2 \cdot \mathrm{Cos}(\theta_3) \cdot \left(\mathrm{Cos}(\theta_1) \cdot p_x + \mathrm{Sin}(\theta_1) \cdot p_y\right)}\right]$$



Right arm-Elbow up

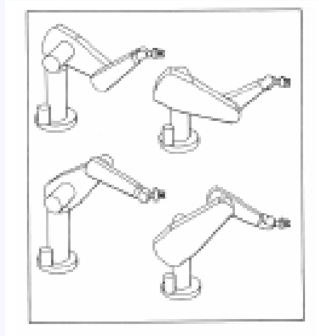
$$\theta_2 = \theta_{23} - \theta_3$$

Abbiamo radici in θ1 e θ3, con due possibili soluzioni; 4 possibili soluzioni per portare il polso in una certa posizione



Right arm-Elbow down

Algoritmo Puma



1. Trovare il centro del polso p_c :

   p_6 *(quarta colonna di T)* – d_6 *(tool offset length)* * z_6 *(terza colonna di T)*

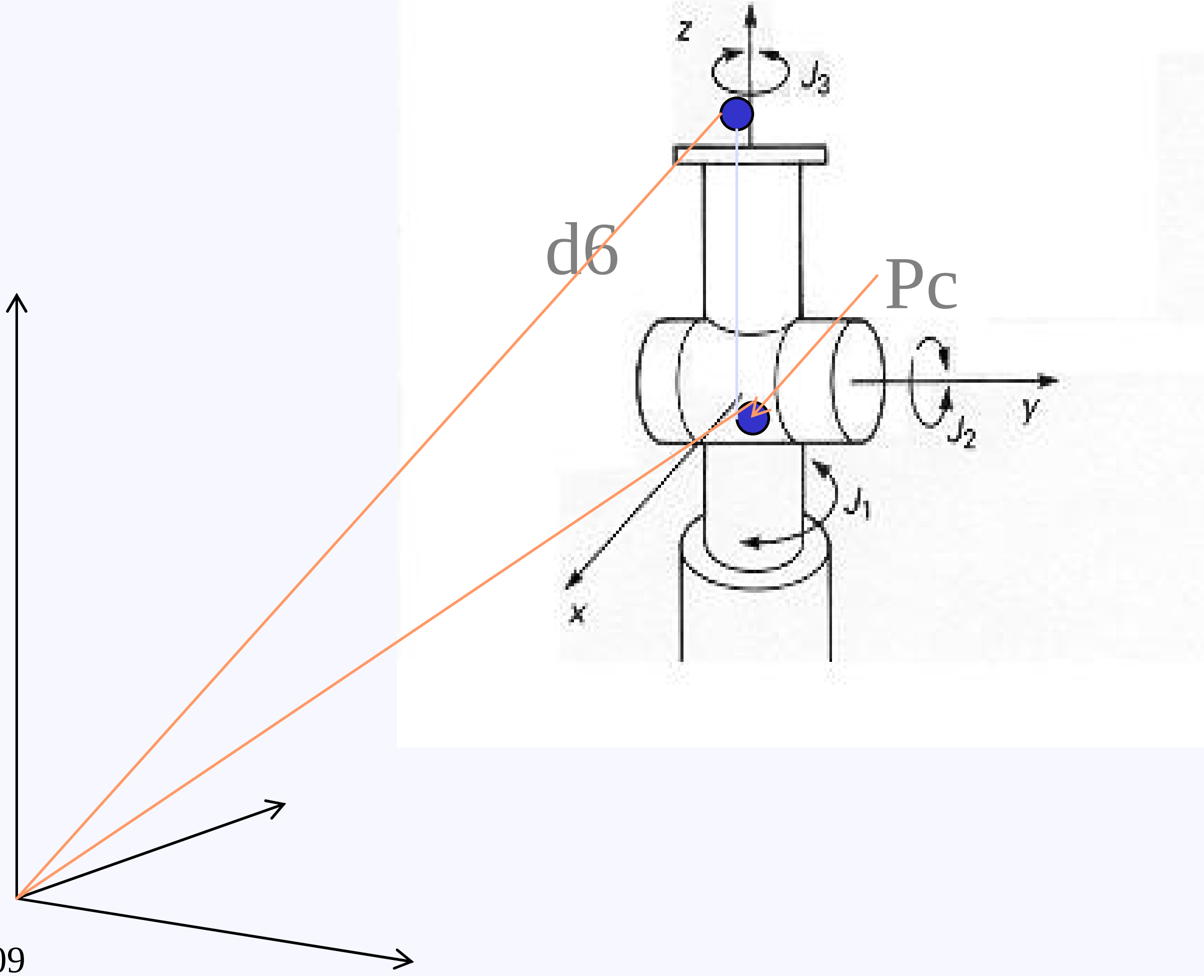
2. p_c = quarta colonna ${}^0A_3(\theta_1, \theta_2, \theta_3)$ e risolvere i primi tre giunti $\theta_1, \theta_2, \theta_3$

3. Usando le soluzioni calcolare ${}^0R_3$; Calcolare ${}^3R_6$ come ${}^0R_3^{-1}\ {}^0R_6$

Uguagliare ${}^3R_6$ così calcolato a ${}^3R_6(\theta_4, \theta_5, \theta_6)$ e risolvere θ_4, θ_5, q_6.

Sistema polso

Per trovare gli angoli del polso, considerare gli angoli di Eulero della rotazione fra frame 3 e frame 6



Stiamo risolvendo

$T^1_6 = A_1A_2A_3A_4A_5A_6$

La arrangiamo come

$A_3^{-1} A_2^{-1} A_1^{-1} T = A_4A_5A_6$

Calcoliamo il lato sinistro, e sostituiamo al lato destro l'equazione cinematica del polso 3R, scritta usando gli angoli di Eulero- .

Per una matrice di rotazione 3x3 abbiamo due soluzioni degli angoli di Eulero:

La prima rotazione è $\theta 4$ o $\pi + \theta 4$

La seconda rotazione è $\theta 5$ o $2\pi - \theta 5$

La terza rotazione è $\theta 6$ o $\pi - \theta 6$

Flip del polso porta quindi dalle 4 soluzioni di prima a 8

– alcune violeranno i limiti dei giunti

Metodo geometrico per PUMA

$$ARM = \begin{cases} +1 \; braccio\ destro \\ -1 \; braccio\ sinistro \end{cases}$$

$$ELBOW = \begin{cases} +1 \; braccio\ alzato \\ -1 \; braccio\ abbassato \end{cases}$$

$$WRIST = \begin{cases} +1 \; polso\ verso\ il\ basso \\ -1 \; polso\ verso\ l'\ alto \end{cases}$$

$$FLIP = \begin{cases} +1 \; inverte\ l'\ orientamento\ del\ \ polso \\ -1 \; non\ inverte\ l'\ orientamento\ del\ \ polso \end{cases}$$

- Equazioni decisionali che usano 4 variabili presettate

Formule importanti per l’inversa

$\sin\theta = a,\ a \in [-1, 1]$

$\cos\theta = b,\ b \in [-1, 1]$

- $\theta = \mathrm{atan2}(a,b)$ soluzione unica

$\sin\theta = a,\ a \in [-1, 1]$

$\cos\theta = \pm\sqrt{(1-a^2)}$

- $\theta = \mathrm{atan2}(a, \pm\sqrt{(1-a^2)})$ 2 soluzioni θ, $180°-\theta$, con singolarità per $a= \pm1$ e $\theta=\pm90°$

$\cos\theta = b,\ b \in [-1, 1]$

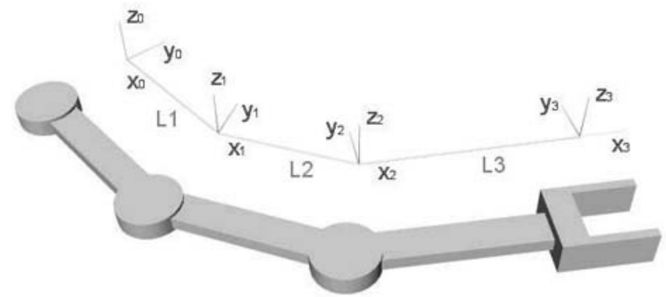
$\sin\theta = \pm\sqrt{(1-b^2)}$

- $\theta = \mathrm{atan2}(\pm\sqrt{(1-b^2)}, b)$, 2 soluzioni θ, $-\theta$, con singolarità per $b= \pm1$ e $\theta=0°$

$b\sin\theta + a\cos\theta = 0$

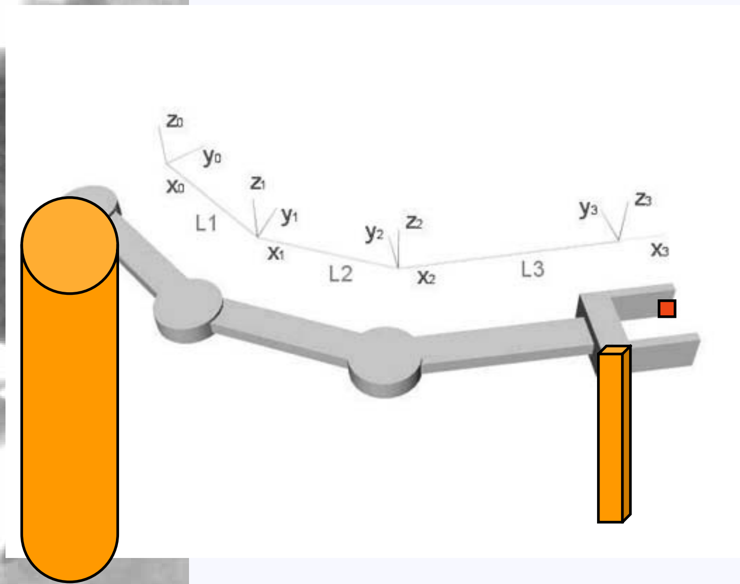
- $\theta = \mathrm{atan2}(a,-b)$ o $\mathrm{atan2}(-a,b)$ 2 soluzioni che distano180°con singolarità per $a=b=0$

Analisi del sottospazio di lavoro



- Per robot con meno di 6gdl, occorre verificare se la struttura del vettore x cartesiano è compatibile
  - In questo caso, deve essere una matrice Rot(z, ϕ), e pz=0
- Altrimenti si calcola la frame più vicina possibile
- Si calcola la cinematica inversa

Robot RRR - SCARA

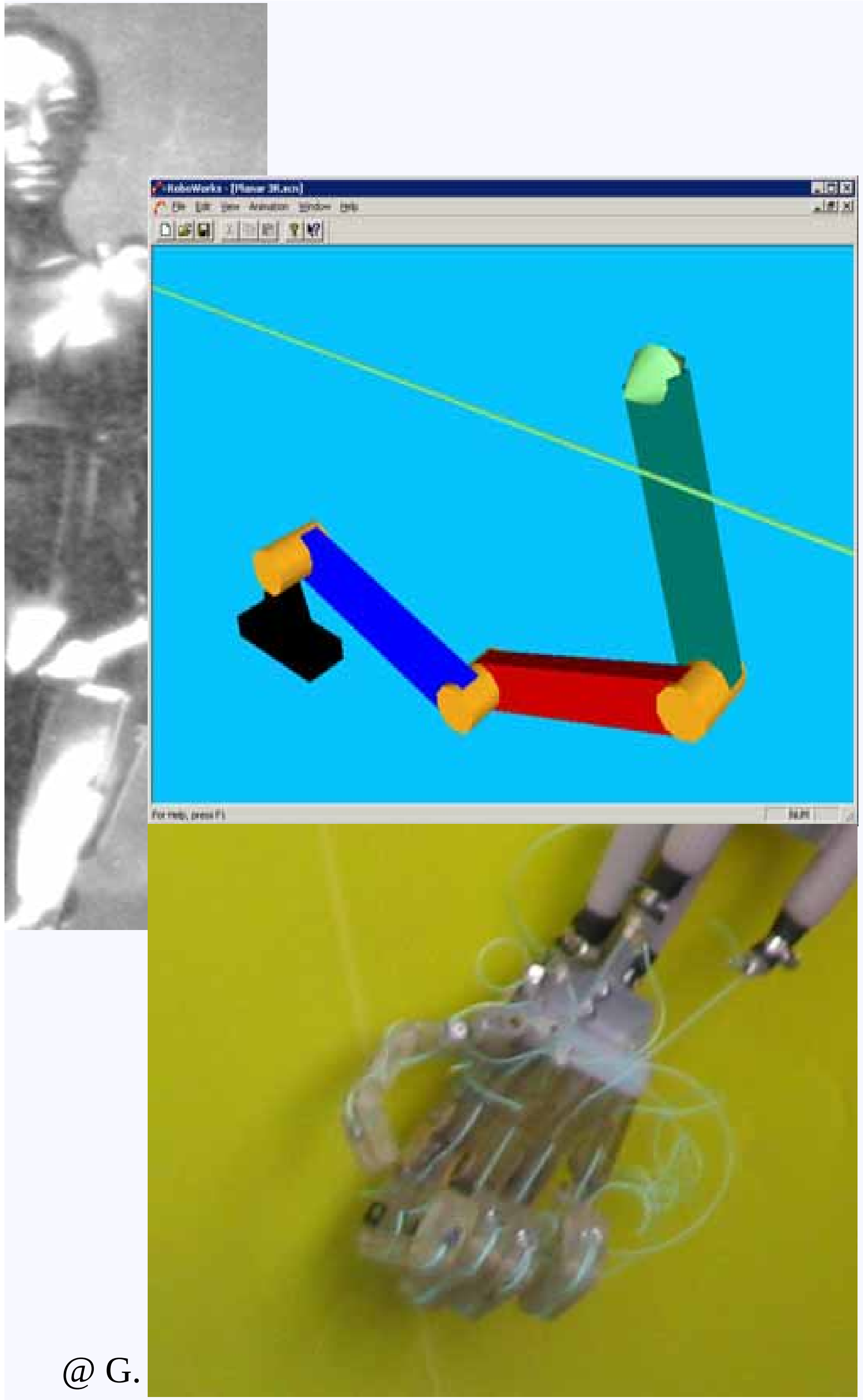


Vettore θ: θ1, θ2, θ3

Vettore **x**: x, y, z, ψ

Se visto come SCARA, si aggiunge il sistema 3 con origine in O2 ed asse z3 ribaltato per gestire la traslazione su z. Si pone il link3=0

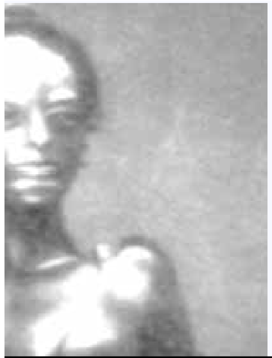
Il sistema 4 ha lo stesso orientamento di 3 ed è posto fra le dita.



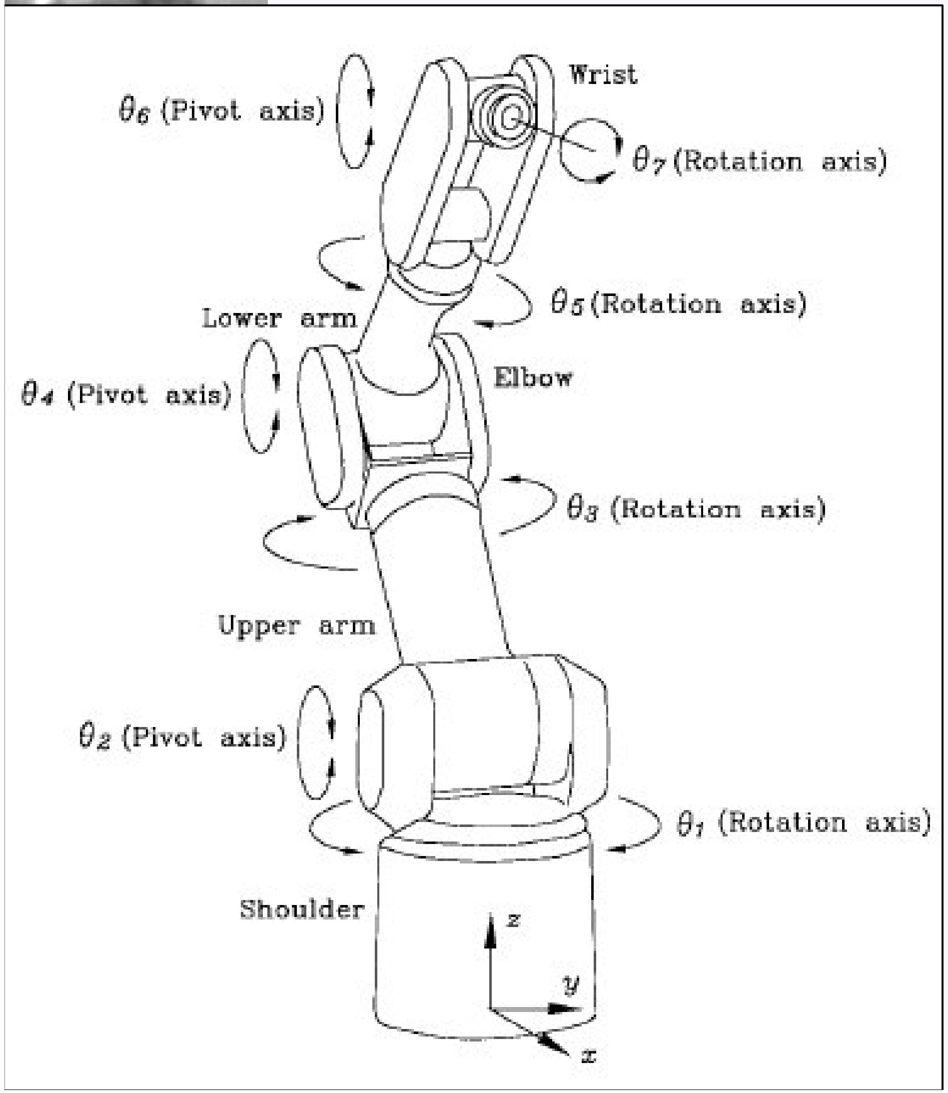
dito

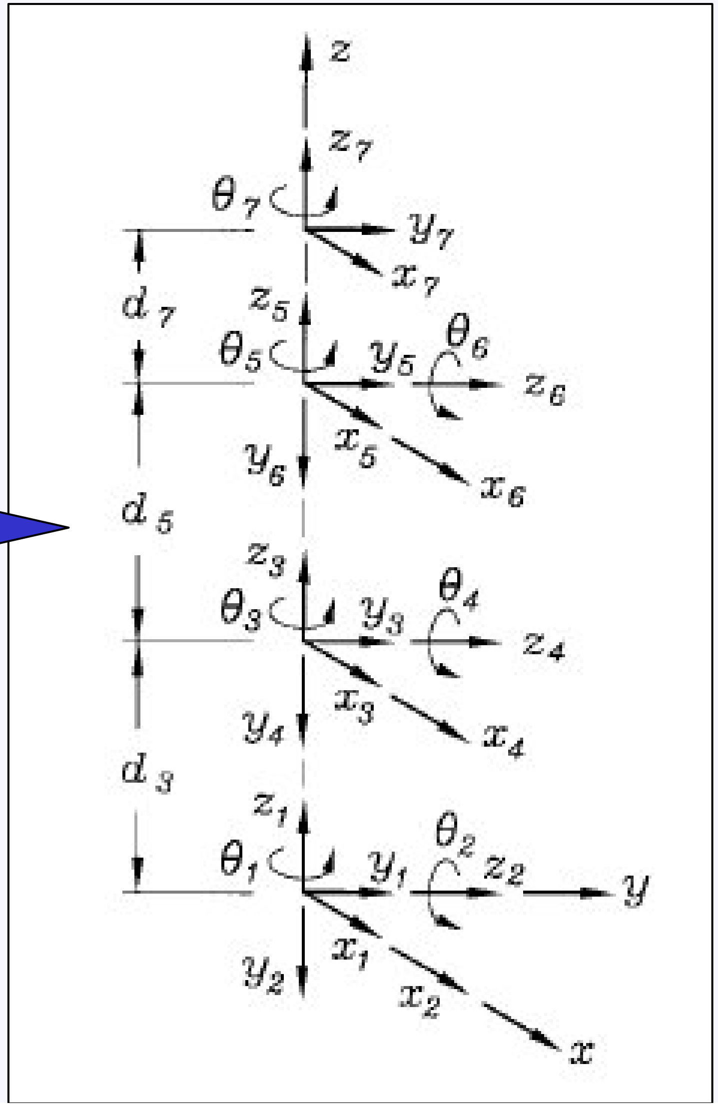
- Un dito ha tre falangi.
- Occorre aggiungere una rotazione attorno a un asse ortogonale al primo giunto per ottenere la adduzione/abduzione
- I range dei valori angolari dei giunti sono limitati

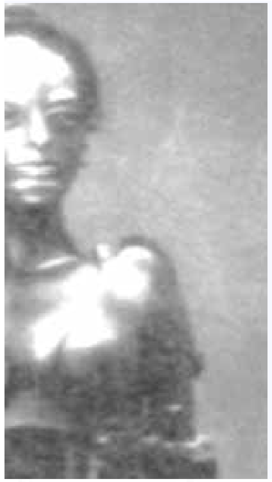
@ G.



Robot a più di 6 gdl
PA - 10







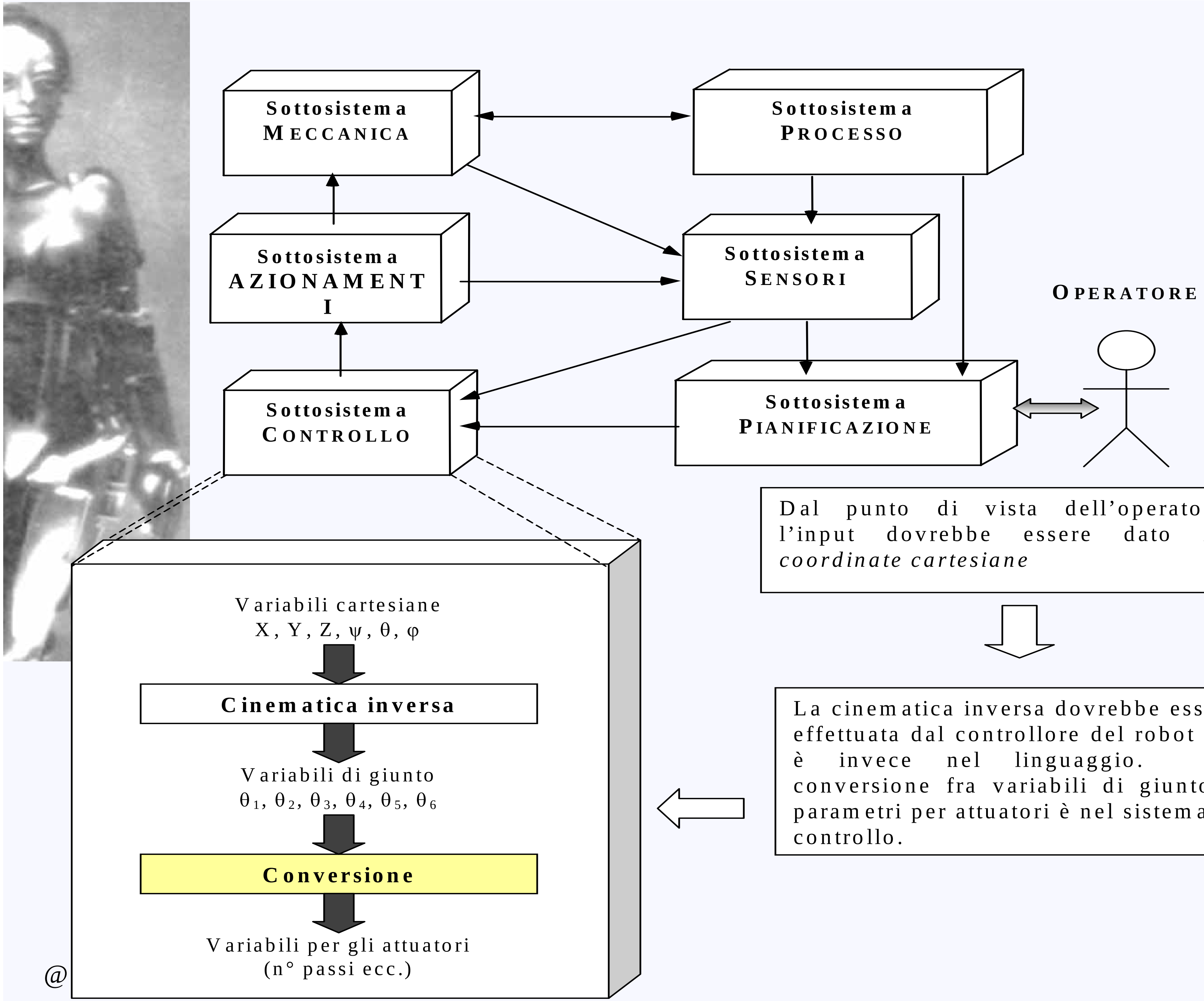
axis	d	a	α
1	0.315	0	0
2	0	0	$-\pi/2$
3	0.45	0	$\pi/2$
4	0	0	$-\pi/2$
5	0.5	0	$\pi/2$
6	0	0	$-\pi/2$
7	0.08	0	$\pi/2$

$$T_{07} = T_{01} \cdot T_{12} \cdot T_{23} \cdot T_{34} \cdot T_{45} \cdot T_{56} \cdot T_{67} = \begin{bmatrix} n & s & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

L'inversa si calcola con lo Jacobiano

Infinite soluzioni – occorrono vincoli

Aggiungere come equazione di vincolo l'energia cinetica del manipolatore e determinare con una rete neurale i θ *che minimizzano l'energia cinetica durante i movimenti.*



Dal punto di vista dell'operatore l'input dovrebbe essere dato in *coordinate cartesiane*

La cinematica inversa dovrebbe essere effettuata dal controllore del robot ma è invece nel linguaggio. La conversione fra variabili di giunto e parametri per attuatori è nel sistema di controllo.

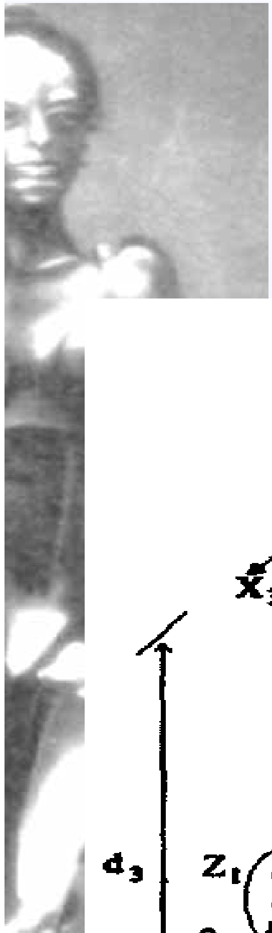
@

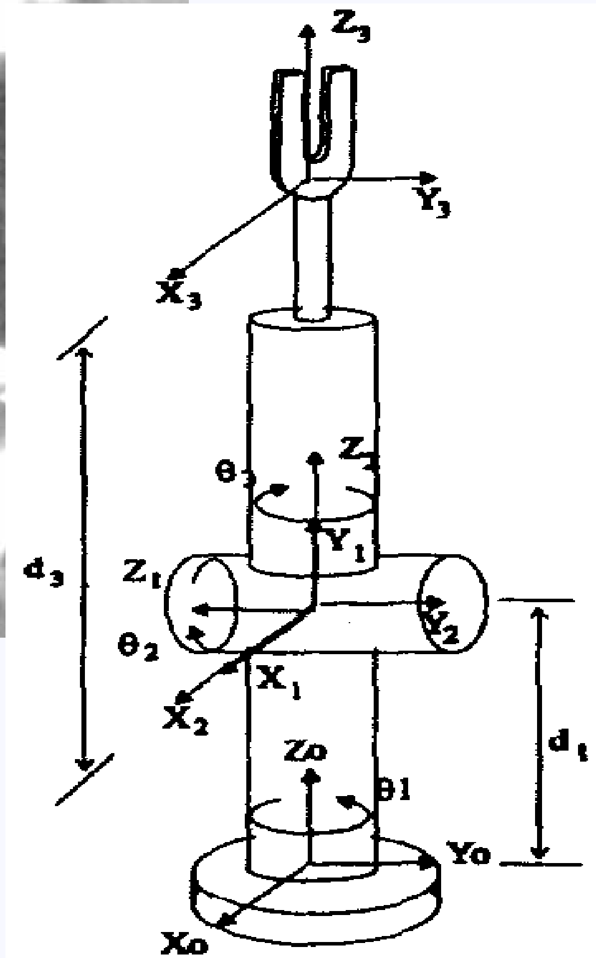
Ridondanze e degenerazioni

robot ridondante = può raggiungere una posizione con più di una configurazione.

- Si hanno sempre ridondanze per manipolatori con più di 6 gdl. Tali manipolatori sono ***infinitamente ridondanti***
  - E' complicato costruire e operare robot infinitamente ridondanti.
  - Però permettono di raggiungere posizioni vincolate.
- Se esiste un numero infinito di configurazioni per raggiungere una posizione, la posizione si dice ***punto di degenerazione*** (in essa la cinematica inversa ha infinite soluzioni).
    - Questo concetto si affronta anche con lo Jacobiano e le sue singolarità.

Esempio: polso PUMA - 1

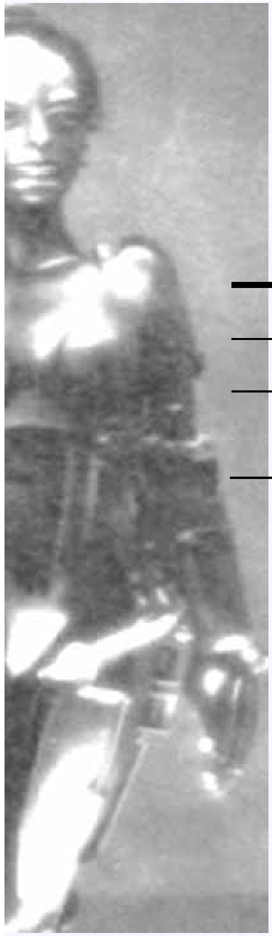




Questo robot ¸ sostanzialmente il polso di un robot PUMA visto come robot.

Ci sono due assi che possono essere collineari per questo il manipolatore dovrebbe essere degenere.

- Analizzando la matrice della mano si vede che esso è degenere.



Polso PUMA - 2

	θ	α	a	d
1	θ_1	90°	0	d_1
2	θ_2	-90°	0	0
3	θ_3	0°	0	d_3

$$\mathbf{T}=\mathbf{A}^{0}_{0,1}\cdot\mathbf{A}^{1}_{12}\cdot\mathbf{A}^{2}_{23}=\begin{bmatrix} C1 & 0 & S1 & 0 \\ S1 & 0 & -C1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}\cdot\begin{bmatrix} C2 & 0 & -S2 & 0 \\ S2 & 0 & C2 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\cdot\begin{bmatrix} C3 & -S3 & 0 & 0 \\ S3 & C3 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Se θ2=0, abbiamo:

$$\mathbf{T}=\begin{bmatrix} C1C3 & -S1S3 & 0 & 0 \\ S1S3 & C1C3 & 0 & 0 \\ 0 & 0 & 1 & d_1+d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}=Trans(p_x,p_y,p_z)\cdot\mathbf{R}(\Phi,\Theta,\Psi)$$

$$p_x=0 \quad p_y=0 \quad p_z=d_1+d_3 \quad \Phi=0 \quad \Theta=0$$

$$\mathbf{T}=\begin{bmatrix} C\Phi & -S\Phi & 0 & 0 \\ S\Phi & C\Phi & 0 & 0 \\ 0 & 0 & 1 & d_1+d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\tan(\Phi)=\tan(\theta_1+\theta_3) \Rightarrow \Phi=\theta_1+\theta_3 \;\; (+k\pi)$$

PUMA - 3 singolarità

- ``alignment'' singularity (il polso è più vicino possibile all'asse del giunto 1)

  $$d4*S_{23} + a2*C_2 - a3*C_{23} == 0$$

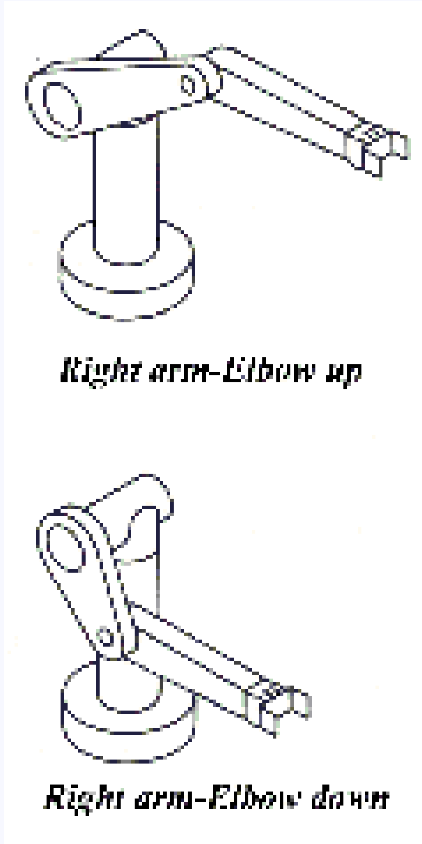
- ``elbow'' singularity (gomito completamente esteso o ripiegato al massimo)

  $$\sin(\theta 3 - \mathrm{atan2}(a3,d4)) == 0$$

- wrist singularity (assi dei giunti 3 e 4 allineati)

  $$S_5 == 0$$

- Dipendono dai parametri (DH) a2, a3, d3, d4



Frames e programmazione

- ❖riferimento di presa sull'oggetto
- ❖far coincidere mano con presa:
  - ❖ si ottiene T, si risolve CI