

1. Introduzione

mercoledì 2 marzo 2022 10:49

2 MARZO

Robotica

Ripensare retroazione
e feedback

Studio di macchine autonome con la possibilità di sostituire l'uomo in compiti specifici.

Dallo dialetto romanesco (= lavoratori)

Le prime macchine si distinguevano in

- macchine a controllo numerico
- telemanipolatori meccanici comandati tramite telecomandi.
→ dalla funzione dei due usciti i robot manipolatori che riescono a saperne la specificità delle operazioni eseguite dalle singole tipologie.

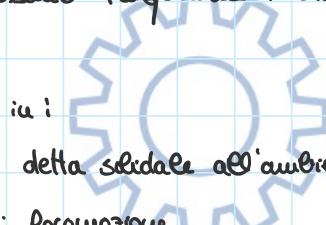
Ciò permetteva di usare una stessa catena di montaggio per diversi modelli o una versatilità dei comportamenti.

Dunque, per ottenere ciò, era necessario trasformare i movimenti di un robot in un software.

I robot possono essere classificati in:

- industriali → hanno una base detta simile all'ambiente
- mobili → sono dotati di sistemi di locomozione

I robot come i droni telecomandati non sono considerati autonomi



Per la ISO/IEC 670 - 2.3, un robot industriale è "Un manipolatore con più gradi di libertà, governato automaticamente, riprogrammabile, multiscopo, che può essere fisso sul posto o mobile per utilizzo in applicazioni di automazioni industriali".

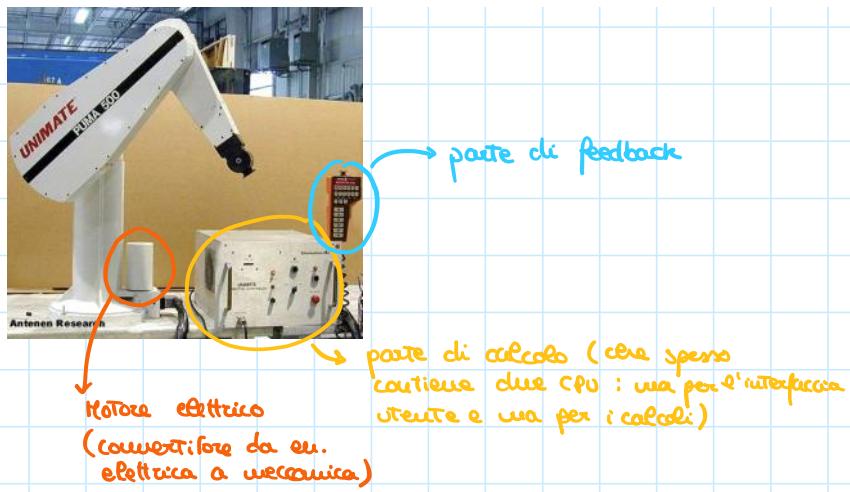
In base alla composizione meccanica si può classificare il robot.

Si parla di:

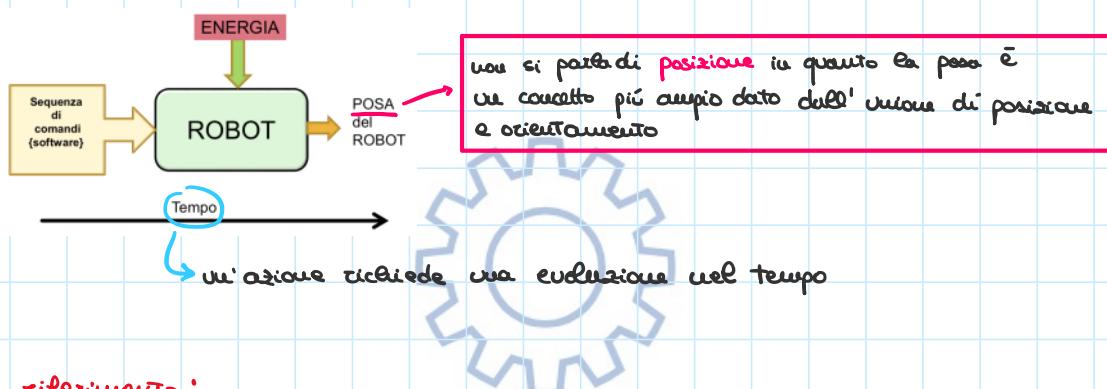
- catena cinemática aperta se le estremità sono libere e dunque sconnese.
- catena cinemática chiusa se le estremità sono bloccate e dunque si crea un anello.

Si parla di interazione forte e debole con l'uomo in base alla necessità interazione richiesta.

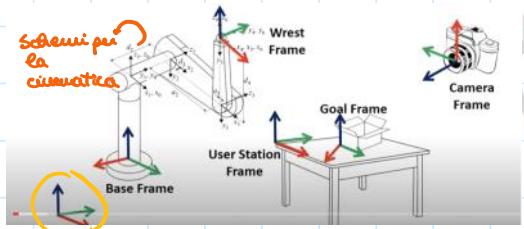
Esempio di robot industriale



È possibile classificare le componenti necessarie di un robot nel seguente schema



Sistemi di riferimento:



**ITI DI INGEGNERIA
INFORMATICA
GAIA BERTOLINO**

→ sistema di riferimento solidale col mondo

Un robot è programmato sul suo sistema di riferimento di base ma fra i vari sistemi deve esserci dialogo.

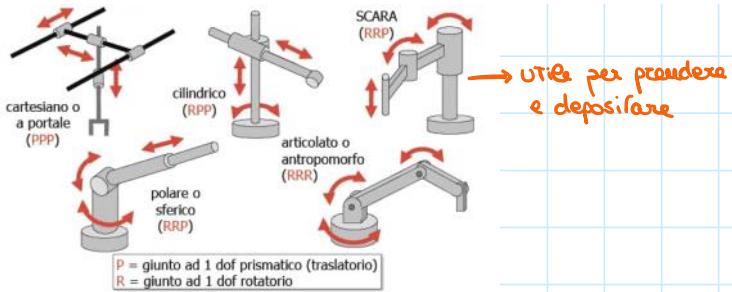
ROBOT INDUSTRIALI

Un robot industriale è costituito da **bracci** e giunti che possono essere di due tipi!

- reticolari per le rotazioni
- prismatici per le traslazioni

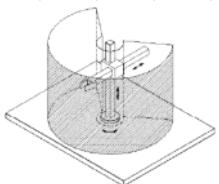
→ detti Link

→ i bracci sono Link Rigidi

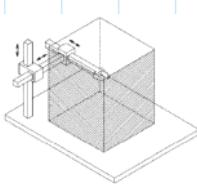


I volumi sono determinati dal volume descritto dai movimenti:

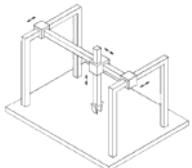
• Cilindrico



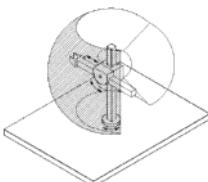
• Cartesiano



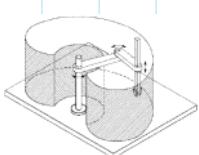
• Polare



• Polare o sferico



• Scara

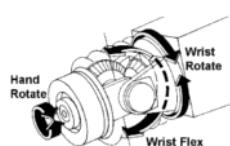
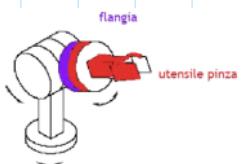


PUNTI DI INERIA INFORMATICA

Questi roboti descritti sono a catena cinemantica aperta, in quanto non si chiudono su loro stessi quando muoiono.

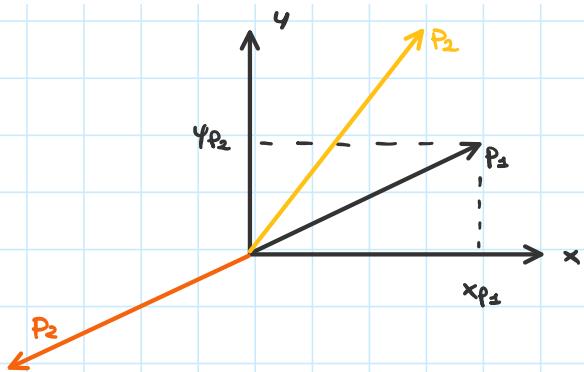
Alle strutture portanti dei roboti è possibile collegare un componente detto **polsi sferici**, il quale garantisce delle rotazioni. La struttura portante è invece responsabile del posizionamento.

Polsi sferici:



ACCENNI DI ALGEBRA

Si assume di lavorare in R^2 per semplicità.



È data la seguente matrice:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Il prodotto $A P_1$ restituisce un vettore P_2 composto da due elementi.

Assumendo successivamente che $A P_1 = P_2 = d P_1$ si definisce d come autovettore della matrice A e in R^2 ne esistono solamente 2 (a meno di considerazioni sul rango). Autovettori e autotuttori si definiscono solo su matrici quadrate ovvero dove spazio di corrispondenza e portamento coincidono.

Una matrice 3×2 mappa elementi da R^2 a R^3 mentre una matrice 2×3 mappa elementi da R^3 a R^2 .

Un problema del tipo $Ax = y$ che è dato da un sistema di equazioni lineari consiste nell'individuare il punto x tale che, moltiplicandolo per A , si ottiene il vettore y .

La posa, come detto, è l'unione di posizione e orientamento.



Per descrivere un oggetto sarebbe necessario avere infiniti vettori ciascuno individuante uno degli infiniti punti che lo compongono.

Essendo impraticabile, si procede ad aggiungere un sistema di riferimento solidale con l'oggetto.

Quando l'oggetto ruota / trasla, anche il sistema di riferimento si muove ma rimane solidale all'oggetto.

Perciò per identificare l'oggetto si utilizzano due mezzi:

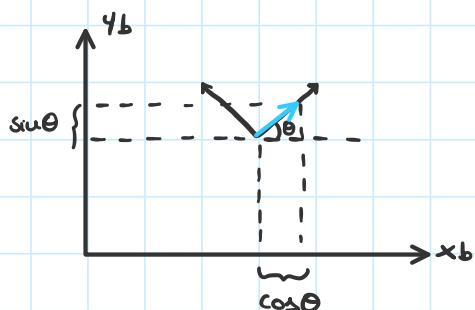
- un vettore che collega le origini del sistema di riferimento del piano con quello solidale all'oggetto
- un angolo che identifica il disallineamento dell'asse x del sistema di riferimento solidale con l'oggetto rispetto a quello del sistema di riferimento del piano.

con l'oggetto rispetto a quello del sistema di riferimento del piano.

Ne consegue che la posa si esprime come composta da $\{P, \Theta\}$ dove P in R^2 è dato da 2 componenti.

Il passaggio in R^3 è invece più complesso a causa della necessità di dover avere più di un angolo per rappresentare il disallineamento dei due sistemi di riferimento. È facile dimostrare che in R^3 si ha bisogno di almeno 3 angoli.

Ancidie gli angoli si preferisce l'uso dei versori del sistema di riferimento dell'oggetto che vengono proiettati sul sistema di riferimento dello spazio. In R^3 ne consegue che si hanno 9 componenti (tuttavia ridandanti in quanto legate fra di loro). Tuttavia, è possibile applicare tale meccanismo anche in R^2 :



Per ruotare una matrice in R^2 si utilizza la seguente matrice:

$$\begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

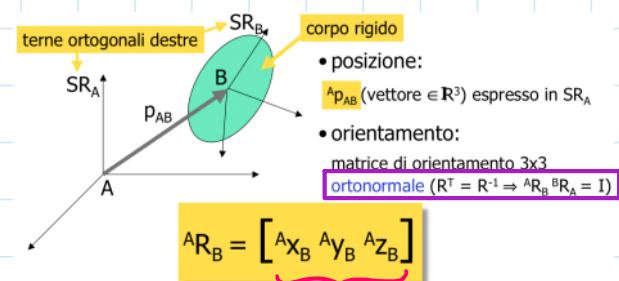
In R^3 una rotazione vi sono più parametri di descrizione che devono seguire un criterio ben definito (es. di eulero).

I robot industriali di solito hanno 6 gradi di libertà:
 tra per la posizione
 tra per l'orientamento

8 HARZO

POSIZIONE E DIRENTAMENTO DEI CORPI

Utilizzando l'appuccio dei versori si ha:

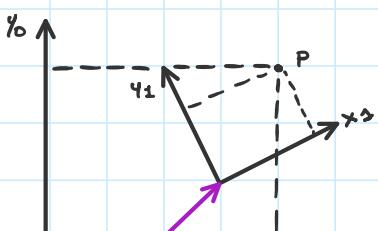


Specifiche di rotazione:

$$\cos(\theta) = c_\theta$$

$$\cos(\theta + \beta) = c_{\theta\beta}$$

Cambiamento di coordinate

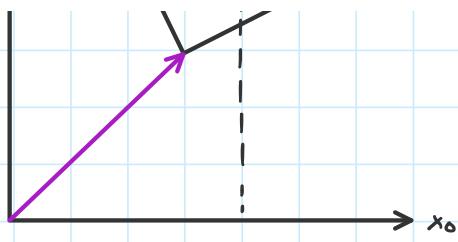


Matrice di rotazione per costruzione geometrica

$$R_1^\circ = \begin{pmatrix} c_\theta & -s_\theta \\ s_\theta & c_\theta \end{pmatrix}$$

dal sistema zero
al sistema 1

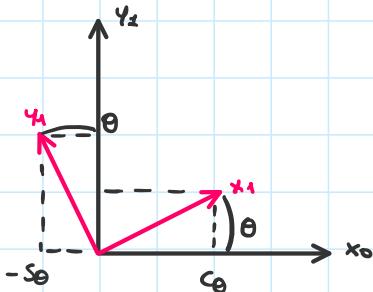
$$p_0 = p_0^\circ + R_1^\circ p_1^\circ$$



\Rightarrow due sistemi sono
al sistema 1

$$P^0 = 'O_1^0 + R_1^0 P^1$$

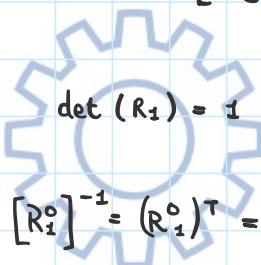
La modellazione più semplice consiste nel far coincidere le due origini:



PROPRIETÀ

$$R_1^0(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

$$[\cos \theta \quad \sin \theta] \begin{bmatrix} -\sin \theta \\ \cos \theta \end{bmatrix} = 0$$



$$\det(R_1) = 1$$

$$[R_1^0]^{-1} = (R_1^0)^T = R_0^1$$

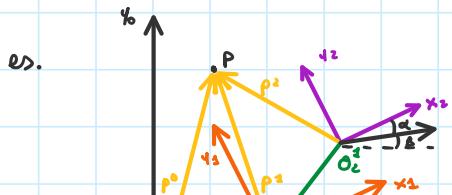
Data una matrice, essa è di rotazione se la norma delle colonne sia pari a 1 ovvero

$$R = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \iff \begin{aligned} &\bullet r_{11}^2 + r_{12}^2 = 1 \\ &\bullet r_{21}^2 + r_{22}^2 = 1 \\ &\bullet [r_{11} \quad r_{21}] \begin{bmatrix} r_{12} \\ r_{22} \end{bmatrix} = r_{11} \cdot r_{12} + r_{21} \cdot r_{22} = 0 \end{aligned}$$

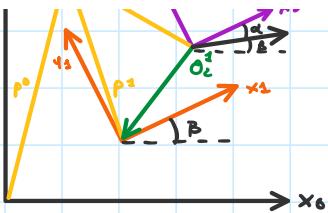
es. $\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ è una matrice di rotazione in quanto rispetta i tre voci.

In R^3 si avrà una matrice 3×3 con 6 voci.

Risolvendo la matrice all' inverso si può ricavare l'angolo.



Il punto P, anche se rimane fermo, può essere descritto rispetto ai vari sistemi di riferimento.



essere descritto rispetto ai vari sistemi di riferimento.

$$P^2 = \underbrace{O_2^1}_{\text{matrice di rotazione dal sistema 1 al 2}} + R_2^1 P^1$$

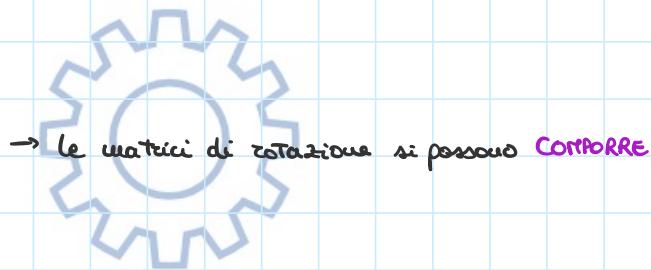
vettore spostamento fra il sistema 2 e il sistema 1

$$R_2^1 = \begin{bmatrix} C_{\alpha} & -S_{\alpha} \\ S_{\alpha} & C_{\alpha} \end{bmatrix}$$

$$R_2^0 \rightarrow "R da 2 a 0"$$

$$P^0 = \underbrace{O_1^0 + R_1^0 P^1}_{O_1^0 + R_1^0 [O_2^1 + R_2^1 P^2]} = \underbrace{O_1^0 + R_1^0 O_2^1}_{O_2^0} + \underbrace{R_1^0 R_2^1 P^2}_{R_2^0 \cdot P^2}$$

Somma Triangolare dei vettori



Le matrici di rotazione in \mathbb{R}^3 presentano delle righe e colonne da matrice identificare se avviene una rotazione elementare intorno ad un asse.

es. $R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$

↳ rotazione intorno a x

APPUNTI DI INGEGNERIA
INFORMATICA
GAIA BERTOLINO

* Si usa una notazione LEONIGRA

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

↳ rotazione intorno a y

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

↳ rotazione intorno a z

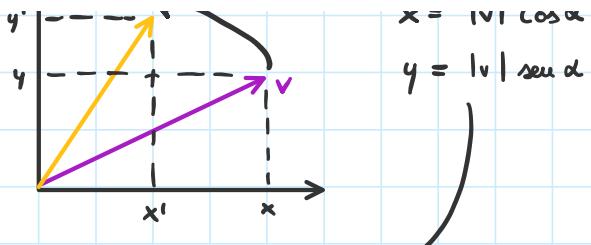
Per ruotare un vettore basta moltiplicarlo per una matrice di rotazione

Esempio: rotazione intorno



$$x = |v| \cos\alpha$$

$$y = |v| \sin\alpha$$



$$x = |v| \cos \alpha$$

$$y = |v| \sin \alpha$$

$$x' = |v| \cos(\alpha + \theta) = |v| (\cos \alpha \cos \theta - \sin \alpha \sin \theta) = x \cos \theta - y \sin \theta$$

$$y' = |v| \sin(\alpha + \theta) = |v| (\sin \alpha \cos \theta + \cos \alpha \sin \theta) = x \sin \theta + y \cos \theta$$

ovvero

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = R_z(\theta) \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Funzione arco tangente

La funzione arco tangente (ovvero \tan^{-1}) non è univoca ovvero non ha un risultato univoco. Si ricorre dunque, nei software matematici, delle funzioni a due argomenti come ad esempio atan2:

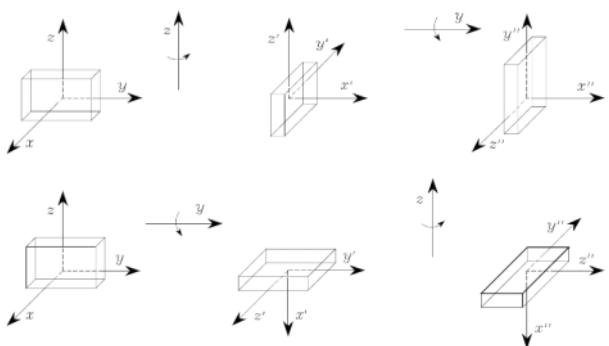
$$\text{atan2}(y, x) = \begin{cases} \arctan\left(\frac{y}{x}\right) & x > 0 \\ \pi + \arctan\left(\frac{y}{x}\right) & y \geq 0, x < 0 \\ -\pi + \arctan\left(\frac{y}{x}\right) & y < 0, x < 0 \\ \frac{\pi}{2} & y > 0, x = 0 \\ -\frac{\pi}{2} & y < 0, x = 0 \\ \text{undefined} & y = 0, x = 0 \end{cases}$$

In poche parole la MATRICE DI ROTAZIONE rappresenta una trasformazione di coordinate mettendo in relazione due tempi differenti.

Nel caso di composizioni di rotazioni rispetto ad una Terra fissa, si moltiplicano da destra verso sinistra le singole matrici di rotazione in base all'ordine delle rotazioni.

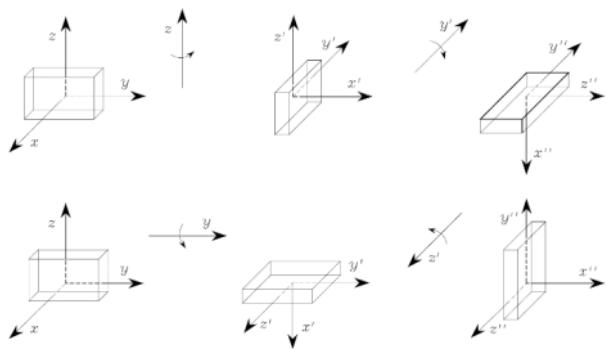
$$\text{es. } R_2^0 = R_1^0 R_0^{-1} R_2^{-1} R_1^0 = I R_2^{-1} R_1^0 = R_2^{-1} R_1^0$$

Rotazioni rispetto ad una terna fissa



Per **Terna mobile** si intende invece la Terna solidale col corpo e si distingue da quella di riferimento (o fissa).

Rotazioni rispetto alla terna corrente



Nel caso di una Terna mobile si moltiplicano invece le matrici di rotazione da sinistra verso destra.

$$\text{es. } R_2^{\circ} = R_1^{\circ} \cdot R_2^{-1}$$

**APPUNTI DI INGEGNERIA
INFORMATICA**
GAIA BERTOLINO

2. Angoli di rotazione

martedì 24 maggio 2022 13:13

9 Koro

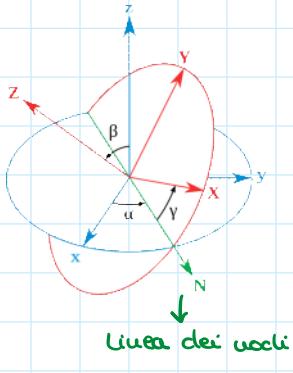
- è di rotazione se
- le colonne hanno norma 1
- i prodotti scalari tra le colonne danno sempre 0 tranne quando pette con loro stesse

Data una matrice di **rotazione**, per ricavare gli angoli di rotazione si utilizzano

Angoli in Terra mobile!

Gli **ANGOLI DI EULER** vengono utilizzati per descrivere l'orientamento di un corpo rigido nello spazio.

Dato il piano di riferimento $x_0y_0z_0$ e uno rotolato $x_1y_1z_1$ (coincidenti nell'origine), si definisce **linea dei nodi** la retta di intersezione fra



Questa è tuttavia una delle 12 possibili combinazioni e prende il nome di 2×2 .

Dal punto di vista algebrico passare da $x_0y_0z_0$ a $x_1y_1z_1$ significa operare un cambiamento di base in tre step che corrispondono a tre rotazioni secondo i tre angoli di Euler che sono:

Angolo di precessione (α)

$\alpha \rightarrow$ angolo fra l'asse x e la linea dei nodi

Angolo di nutazione (β)

$\beta \rightarrow$ angolo fra gli assi z e z'

Angolo di rotazione propria (γ)

$\gamma \rightarrow$ angolo fra la linea dei nodi e l'asse X

dunque le tre rotazioni saranno

1) rotazione di α intorno a z , facendo così coincidere x con N . La matrice sarà:

$$R_\alpha = \begin{pmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

2) rotazione di β intorno a N . La matrice sarà:

$$R_\beta = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & \sin\beta \\ 0 & -\sin\beta & \cos\beta \end{pmatrix}$$

3) rotazione di γ intorno a z . La matrice sarà:

$$R_\gamma = \begin{pmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Data la matrice inversa si cerca di ricavare i valori degli angoli e, come quasi tutti i problemi inversi, la soluzione non è univoca. In questo caso si ottiene sempre una coppia di soluzioni.

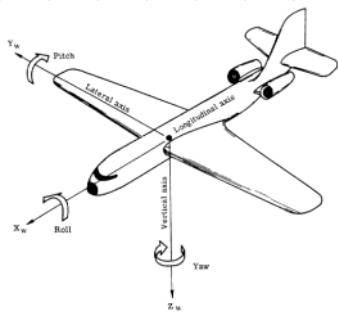
Angoli in Terra mobile!

Gli **ANGOLI ROLL - PITCH - YAW** possono essere facilmente rappresentati tramite il **decello di un aereo**.

Angoli in Terna mobile:

gli ANGOLI ROLL - PITCH - YAW possono essere facilmente rappresentati tramite il decollo di un aereo.

In questo caso, rispetto ad una rotazione finita, bisogna fare una rotazione inversa cioè moltiplicando R_y per R_x.



Queste due rotazioni sono minimale (a tre angoli) ma ne esistono anche non minimale a 6 parametri.

Quale sceglierà? È importante sapere saltare da una terna all'altra in base al movimento compiuto.

COORDINATE QUATERNIONE

Un punto nello spazio è individuato da tre coordinate dette coordinate che sono x, y e z.

Si definiscono analoghe invece quelle coordinate $[x \ y \ z \ u]$ tali che

$$x = \frac{x}{u}, \quad y = \frac{y}{u}, \quad z = \frac{z}{u}.$$

detto **FATTORE DI SCALA**

Dunque si avrà che il punto $P = aix + biy + ciz$ sarà esprimibile attraverso tutte quelle quaterni di numeri $[x \ y \ z \ u]$ tali che $a = \frac{x}{u}, \ b = \frac{y}{u}, \ c = \frac{z}{u}$

L'origine avrà coordinate $[0 \ 0 \ 0 \ u]$

GAIA BERTOLINO

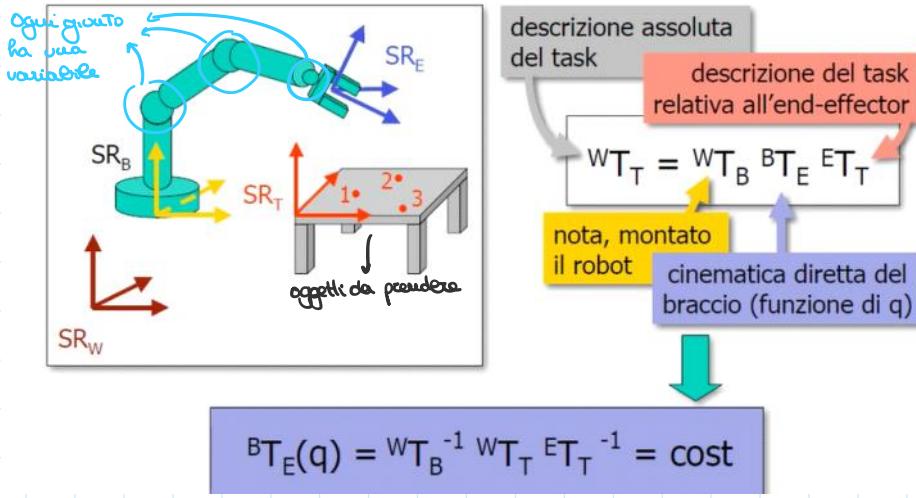
es. $P = ix + jy + kz$ può essere rappresentato come

$$P = [1 \ 1 \ 1 \ 1]$$

$$P = [2 \ 2 \ 18 \ 2]$$

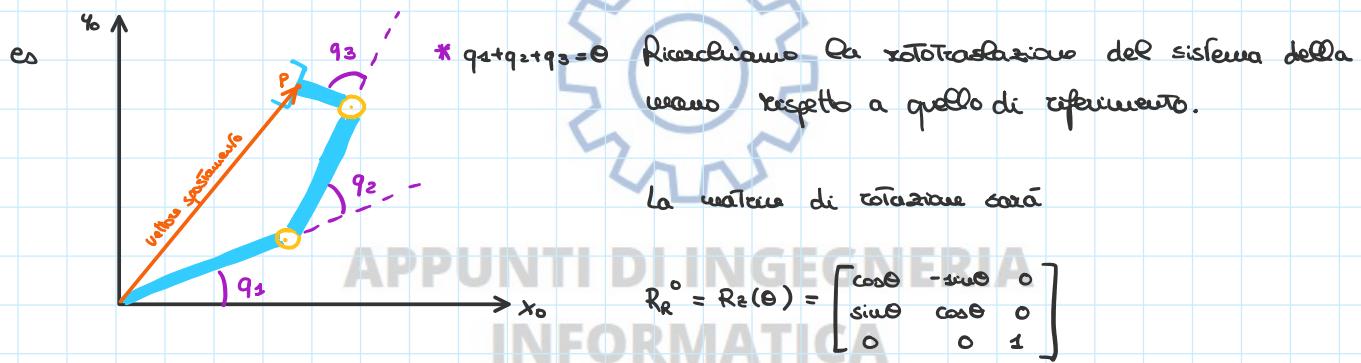
$$P = [-9 \ -9 \ -81 \ -9]$$

ESEMPIO CINETICA DIRETTA DI UN ROBOT



Nel caso di un robot mobile, per capire il punto di partenza rispetto alla terra fissa si deve integrare le equazioni del moto al rovescio.

Per **cinematica diretta** si intende il calcolo della posizione di un oggetto conoscendo le coordinate dei suoi giunti. Per **cinematica inversa** si intende la determinazione dei parametri dei giunti per ottenere la posa cercata.



Il vettore di spostamento che individua P sarà del tipo

$$P_x = L_1 \cdot \underline{\underline{c_{11}}} + L_2 \cdot \underline{\underline{c_{21}}} + L_3 \cdot \underline{\underline{c_{31}}}$$

$$P_y = L_1 \cdot s_1 + L_2 \cdot s_{21} + L_3 \cdot s_{31}$$

$$P_z = 0$$

Si avrà dunque che

$$\hat{R}_R^\theta = \hat{R}_1^\theta \cdot \hat{R}_2^\theta \cdot \hat{R}_3^\theta$$

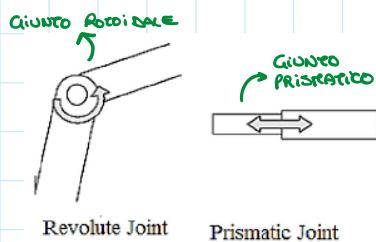
$$\hat{R}_1^\theta = \begin{bmatrix} c_1 & -s_1 & 0 & | & l_1 \cdot c_1 \\ s_1 & c_1 & 0 & | & l_1 \cdot s_1 \\ 0 & 0 & 1 & | & 0 \\ 0 & 0 & 0 & | & 1 \end{bmatrix}$$

$$\hat{R}_2^\theta = \begin{bmatrix} c_2 & -s_2 & 0 & | & l_2 \cdot c_2 \\ s_2 & c_2 & 0 & | & l_2 \cdot s_2 \\ 0 & 0 & 1 & | & 0 \\ 0 & 0 & 0 & | & 1 \end{bmatrix}$$

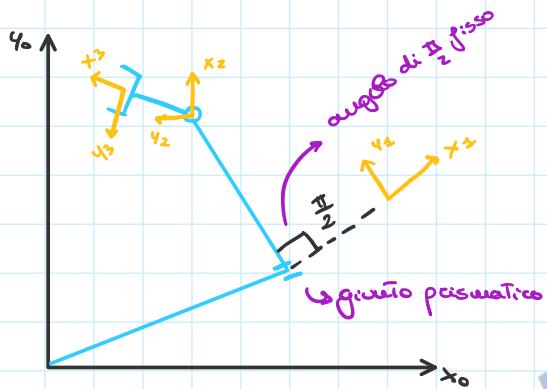
$$\hat{R}_3^\theta = \begin{bmatrix} c_3 & -s_3 & 0 & | & l_3 \cdot c_3 \\ s_3 & c_3 & 0 & | & l_3 \cdot s_3 \\ 0 & 0 & 1 & | & 0 \\ 0 & 0 & 0 & | & 1 \end{bmatrix}$$

$$\rightarrow \hat{R}_R^\theta = \hat{R}_1^\theta (q_1) \cdot \hat{R}_2^\theta (q_2) \cdot \hat{R}_3^\theta (q_3)$$

15.2



es. Braccio ruotato con giunto prismatico

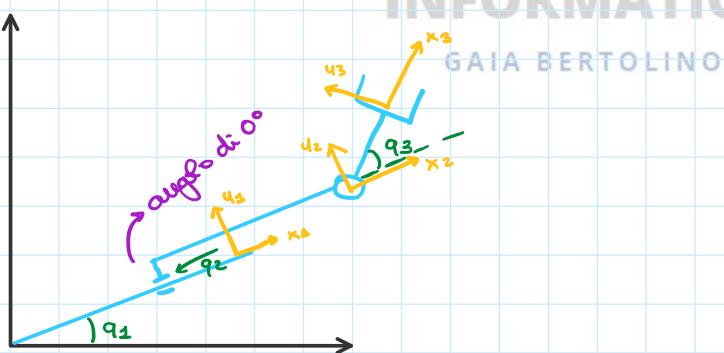


$$Q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} \text{ rad}$$

$$\hat{R}_3^o(Q) = \hat{R}_1^o(q_1) \cdot \hat{R}_2^a(q_2) \cdot \hat{R}_3^a(q_3)$$

$$\hat{R}_2^a(q_2) = \begin{bmatrix} C_{\pi/2} & -S_{\pi/2} & 0 & | & 0 \\ S_{\pi/2} & C_{\pi/2} & 0 & | & 0 \\ 0 & 0 & 1 & | & 0 \\ 0 & 0 & 0 & | & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & | & q_2 \\ 1 & 0 & 0 & | & L_2 \\ 0 & 0 & 1 & | & 0 \\ 0 & 0 & 0 & | & 1 \end{bmatrix}$$

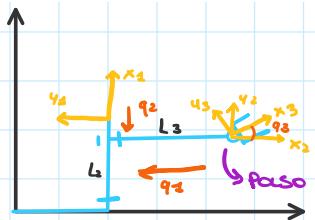
es.



$$\hat{R}_3^o(\Theta) = \hat{R}_1^o \cdot \hat{R}_2^a \cdot \hat{R}_3^a$$

$$\hat{R}_2^a(q_2) = \begin{bmatrix} 1 & 0 & 0 & | & L_2 - q_2 \\ 0 & 1 & 0 & | & 0 \\ 0 & 0 & 1 & | & 0 \\ 0 & 0 & 0 & | & 1 \end{bmatrix}$$

es. Braccio con giunto rotoidale e origini sovrapposte



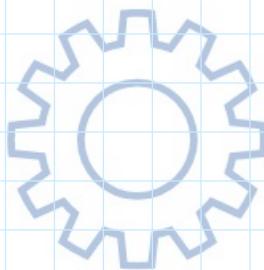
$$\hat{R}_3^o(Q) = \hat{R}_z^o \cdot \hat{R}_x^o \cdot \hat{R}_y^o$$

$$\hat{R}_y^o = \begin{bmatrix} c_3 & -s_3 & 0 & 1 & 0 \\ s_3 & c_3 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\hat{R}_x^o = \begin{bmatrix} 0 & 1 & 0 & 1-q_2 \\ -1 & 0 & 0 & 1-L_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\hat{R}_z^o = \begin{bmatrix} 0 & -1 & 0 & 1 & L_3-q_2 \\ 1 & 0 & 0 & 0 & L_2 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\hat{R}_3^o = \begin{bmatrix} c_3 & -s_3 & 0 & 1 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



APPUNTI DI INGEGNERIA INFORMATICA

GAIA BERTOLINO

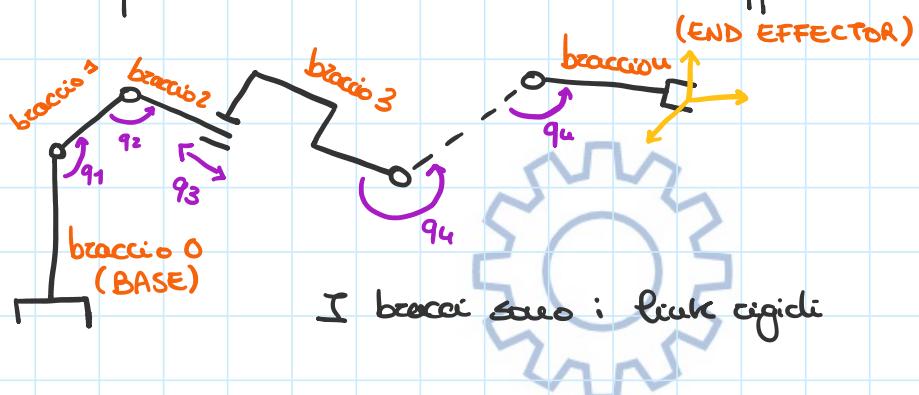
3. Cinematica dei robot

martedì 24 maggio 2022 19:31

Che CINEMATICA si intende la relazione, espressa tramite una matrice di rototraslazione, fra due tempi di sistemi di riferimento, uno fisso e l'altro pari all'estremità del robot. La matrice dipenderà dai bracci e dai giunti che sono presenti.

Catene cinematiche aperte:

sono fati poiché esiste un membro con un solo accoppiamento.



I bracci sono i link rigidi

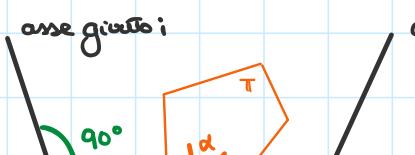
Assegnare le variabili di girezione non è unico.

* Catenza aperta o catena chiusa

INFORMATICA
GAIA BERTOLINO

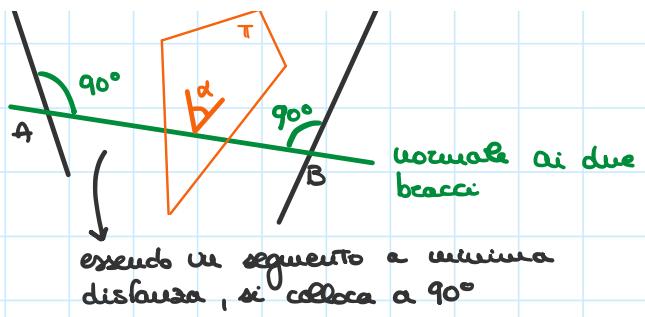
The diagram compares kinematic chains to human arms. On the left, a person's arm is shown with the text 'Open chain'. On the right, a person's arm resting on a chair is shown with the text 'Closed chain'. To the right of these, two robotic arm diagrams are shown: one labeled 'Cadena aperta' (open chain) and another labeled 'Cadena chiusa' (closed chain). The closed chain diagram shows a more compact, integrated structure.

Dati: dei giunti, ognuno ha un asse ed esiste una relazione fra giunti che definisce il disallineamento degli assi attraverso un angolo detto di TWIST

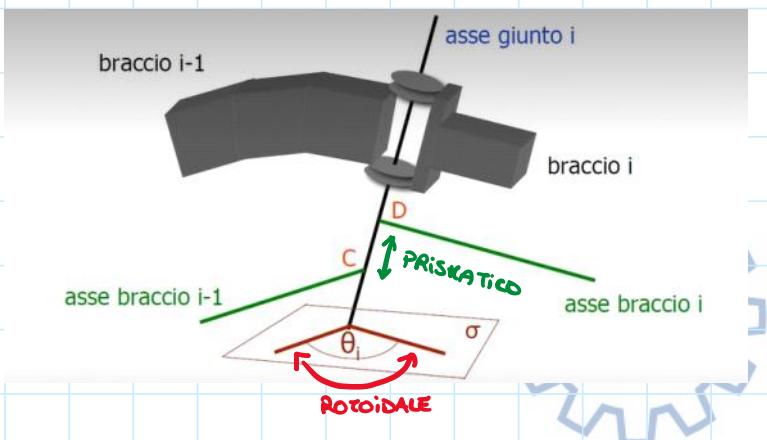


asse giunto 1

asse giunto 2



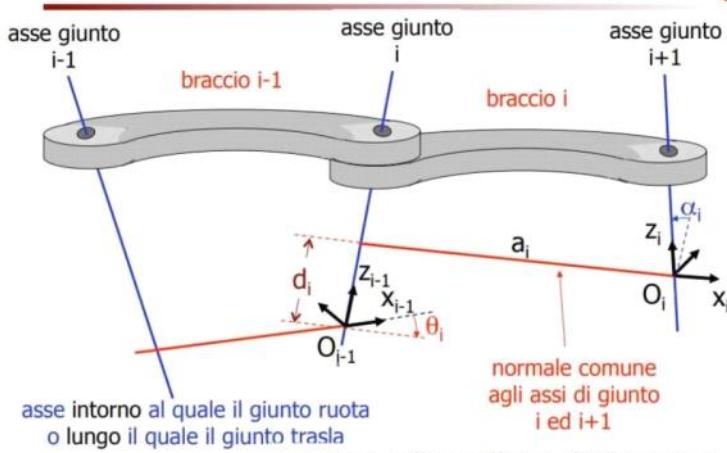
Allo stesso modo è possibile definire una relazione fra gli assi dei bracci. L'angolo in questo caso è detto **VARIABILE** se il giunto è rotoidale mentre variabile è la distanza fra gli assi se il giunto è prismatico.



da questo approccio nasce la
CONVENZIONE DI DENAVIT - HARTENBERG

Tale convenzione viene utilizzata per scrivere i sistemi di riferimento in modo che ogni trasformazione (es. rototraslazione) sia rappresentabile col minimo numero di parametri (che è quattro).

In questa affermazione non è rilevante la meccanica bensì la disposizione degli assi.



PARAMETRI DI
DENAVIT-HARTENBERG

- asse z_i lungo l'asse di giunto $i+1$
- asse x_i lungo la normale comune agli assi di giunto i e $i+1$ (verso: $i \rightarrow i+1$)
- a_i = distanza O_i -D orientata con x_i (costante = "lunghezza" braccio i)
- d_i = distanza O_{i-1} -D orientata con z_{i-1} (variabile se giunto i PRISMATICO)
- α_i = angolo di twist tra z_{i-1} e z_i intorno a x_i (costante)
- θ_i = angolo tra x_{i-1} e x_i intorno a z_{i-1} (variabile se giunto i ROTATORIO)

(distanza zera causa)
La matrice di rototraslazione sarà del tipo:

$$\hat{R}_i^{i-1}(q_i) = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cdot \cos \alpha_i & \sin \theta_i \cdot \sin \alpha_i & | a_i \cdot \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cdot \cos \alpha_i & -\cos \theta_i \cdot \sin \alpha_i & | a_i \cdot \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & | d_i \\ 0 & 0 & 0 & | 1 \end{bmatrix}$$

INFORMATICA

GAIA BERTOLINO

Vi possono essere ambiguità quando

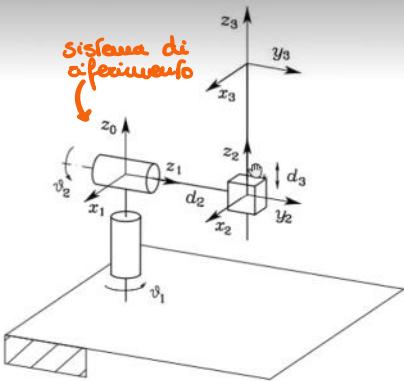
- $a_i = 0$. Si cerca di posare quindi $\theta_i = 0$
- z_{i-1} e z sono paralleli quindi a_i può essere posta anche. Si cerca quindi di posare in modo da avere $d_i = 0$

Inoltre, le condizioni iniziali sono arbitrarie mentre le finali dipendono da quelle immediatamente precedenti.

ESEMPIO TABELLA DH

sferico

Braccio	a_i	α_i	d_i	ϑ_i
1	0	$-\pi/2$	0	ϑ_1
2	0	$\pi/2$	d_2	ϑ_2
3	0	0	d_3	0



Matrici di rototraslazioni:

$$A_1^0 = \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_2^1 = \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^0 = A_1^0 A_2^1 A_3^2$$

$$= \begin{bmatrix} c_1 c_2 & -s_1 & c_1 s_2 & c_1 s_2 d_3 - s_1 d_2 \\ s_1 c_2 & c_1 & s_1 s_2 & s_1 s_2 d_3 + c_1 d_2 \\ -s_2 & 0 & c_2 & c_2 d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

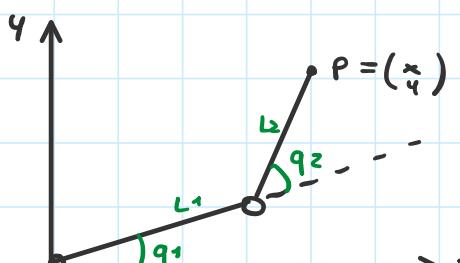
NERIA

GAIA BERTOLINO

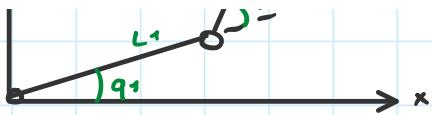
CINEMATICA INVERSA

Per cinematica inversa si intende il calcolo delle variabili di giunto conoscendo la posa del robot, ovvero la Tabella di D-H.
→ dunque il problema consiste nell' individuare dati di posizionamento di bracci e girarli per ottenere una posa ricercata.

Ese. cinematica diretta



$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} L_1 \cdot c_{q1} + L_2 \cdot c_{q2} \\ L_1 \cdot s_{q1} + L_2 \cdot s_{q2} \end{pmatrix}$$



* nuova la dimostrazione della cinematica inversa

I metodi di risoluzione sono di due tipologie:

- Soluzione ANALITICA

È un tipo di soluzione NON numerica ma detta in **forma chiusa** ovvero attraverso una formula matematica.

- Soluzione di ottimizzazione che va a cercare la soluzione migliore attraverso step successivi di approssimazione numerica

CINERATICA DIFFERENZIALE

Negli spostamenti di un robot, Tramite Trasformazioni e rotazioni, si può parlare di velocità e, dunque, con cinematica differenziale si intende lo studio dei legami tra il moto (cioè la loro velocità) dei giunti e il moto (cioè la velocità lineare e angolare) nello spazio cartesiano attraverso lo **jacobiano del manipolatore**.

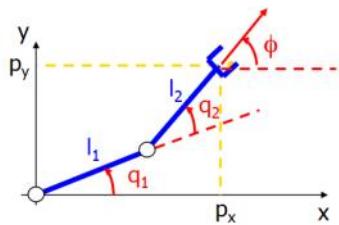
Inoltre, la velocità è esprimibile come la derivata dello spazio sul tempo.

Inoltre, si parla di **JACOBIANO** in quanto, dato un sistema di funzioni che esprime il cambiamento di coordinate, è possibile definire una matrice di derivate parziali di tali trasformazioni il cui determinante è proprio chiamato **jacobiano associato al cambiamento di coordinate**.

Lo jacobiano è ottenibile tramite diverse modalità ovvero

- per differenziazione e si ottiene lo jacobiano **analitico**
- senza derivazione e si ottiene lo jacobiano **geometrico**

Esempio jacobiano analitico robot 2R **planare** → movimento in 2D (su un piano)



cinematica diretta

$$\begin{aligned} \mathbf{r} &= \begin{cases} p_x = l_1 c_1 + l_2 c_{12} \\ p_y = l_1 s_1 + l_2 s_{12} \\ \phi = q_1 + q_2 \end{cases} \end{aligned}$$

$$\dot{p}_x = -l_1 s_1 \dot{q}_1 - l_2 s_{12} (\dot{q}_1 + \dot{q}_2)$$

$$\dot{p}_y = l_1 c_1 \dot{q}_1 + l_2 c_{12} (\dot{q}_1 + \dot{q}_2)$$

$$\dot{\phi} = \omega_z = \dot{q}_1 + \dot{q}_2$$

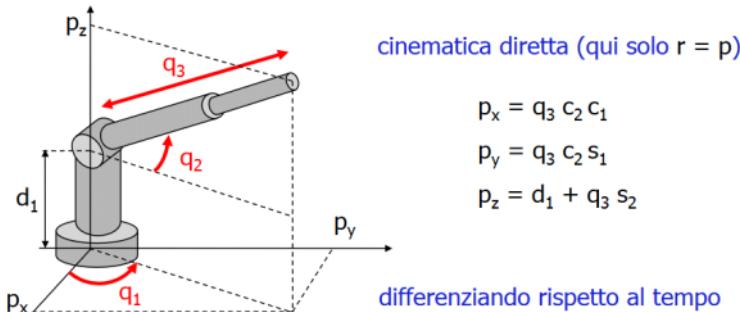
$$\mathbf{J}_r(\mathbf{q}) = \begin{pmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \\ 1 & 1 \end{pmatrix} \begin{cases} \mathbf{J}_L(\mathbf{q}) \\ \mathbf{J}_A(\mathbf{q}) \end{cases}$$

data \mathbf{r} , qui è una matrice 3×2

Jacobiano analitico = geometrico

Esempio Jacobiano analitico robot pilare

movimento in 3D
(nello spazio)



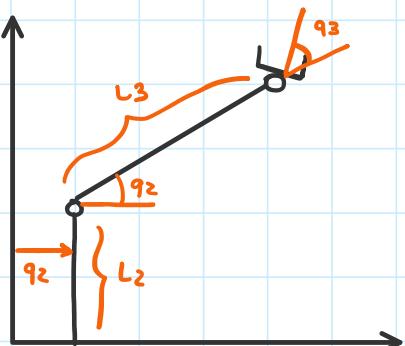
$$\mathbf{v} = \dot{\mathbf{p}} = \begin{pmatrix} -q_3 c_2 s_1 & -q_3 s_2 c_1 & c_2 c_1 \\ q_3 c_2 c_1 & -q_3 s_2 s_1 & c_2 s_1 \\ 0 & q_3 c_2 & s_2 \end{pmatrix} \dot{\mathbf{q}} = \mathbf{J}_r(\mathbf{q}) \dot{\mathbf{q}}$$

APPUNTI DI INGEGNERIA

INFORMATICA

GAIA BERTOLINO

$$\begin{cases} x = q_1 + L_3 \cdot c_2 \\ y = L_2 + L_3 \cdot s_2 \\ \theta = q_2 + q_3 \end{cases}$$



Ne deriva che

$$v_x = \dot{x} = q_1 - L_3 \cdot s_2 \cdot q_2$$

$$v_y = \dot{y} = L_3 \cdot c_2 \cdot q_2$$

$$\omega = \dot{\theta} = q_2 + q_3$$

$$\Rightarrow \mathbf{J} = \begin{bmatrix} 1 & -L_3 \cdot s_2 & 0 \\ 0 & L_3 \cdot c_2 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

configurazione singolare! La matrice perde di rango

$$\det(\mathbf{J}) = L_3 \cdot c_2 \text{ se e solo se } q_2 = \frac{\pi}{2}$$

$$\det(\mathbf{J}) = l_3 \cdot c_2 \text{ che è pari a } 0 \text{ quando} \quad \begin{cases} q_2 = \frac{\pi}{2} \\ q_2 = \frac{\pi}{2} + \pi \end{cases}$$

→ configuration singolare! La matrice perde di rango

In generale, l'angolo più facile da calcolare è q_2 (secondo la convenzione).

La matrice di rotazione è detta OPERATORE UNITARIO ovvero si ha che se $\mathbf{U}^T \cdot \mathbf{U} = \mathbf{U} \cdot \mathbf{U}^T = \mathbf{I}$ allora $\mathbf{U}^{-1} = \mathbf{U}^T$ e dunque \mathbf{U} è un operatore unitario. Inoltre, il determinante di tale matrice è 1, nel caso in cui fosse pari a -1 si parla di rotazione immaginaria ovvero non conforme allo schema che segue la regola della mano destra.

In aggiunta, gli operatori unitari sono invarianti rispetto alla norma ovvero anche se si applica tale operatore, la norma non cambia. Ciò significa che ruotando un corpo, la sua dimensione NON si modifica, ovvero non viene sfacciato.

Per riassumere:

in \mathbb{R}^3 si ha che una matrice di rotazione sarà

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

Dunque per ruotare P da A a B bisogna fare $P_B = R_{A \rightarrow B} \cdot P_A$ da A a B

mentre nella rototraslazione sarà $P_B = R_{A \rightarrow B} \cdot P_A + \underline{t}$ traslazione

Tuttavia, anziché avere una matrice e una traslazione, si scrive tutto come una singola matrice che sarà

$$\begin{bmatrix} P_B \\ 1 \end{bmatrix} = \left[\begin{array}{c|c} R_{A \rightarrow B} & t \\ \hline 0 & 0 & 1 \end{array} \right] \begin{bmatrix} P_A \\ 1 \end{bmatrix}$$

$R_{A \rightarrow B}$

Attenzione! Anche se R (matrice di rotazione) è un operatore unifero, ciò non vale per R_t (matrice di rototraslazione), a meno che, ad esempio, $t = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ ovvero la traslazione sia pari a 0 e la matrice sia di rotazione.

Procedimento per ottenere l'inversa: ↗ possibile domanda d'esame

R_t^{-1} computation, given $P \in \mathbb{R}^n$, $n = 2, 3$.

$$\begin{bmatrix} \tilde{P} \\ 1 \end{bmatrix} = R_t \begin{bmatrix} P \\ 1 \end{bmatrix} \longleftrightarrow \tilde{P} = R_t P \longleftrightarrow \begin{cases} \tilde{P} = RP + t \\ 1 = 1 \end{cases}$$

$$\tilde{P} = RP + t \rightarrow P = R^{-1}(\tilde{P} - t) \rightarrow P = R^T(\tilde{P} - t)$$

$$P = R^T \tilde{P} + (-R^T t) \rightarrow \begin{bmatrix} P \\ 1 \end{bmatrix} = \left[\begin{array}{c|c} R^T & -R^T t \\ \hline 0^T & 1 \end{array} \right] \begin{bmatrix} \tilde{P} \\ 1 \end{bmatrix} \longleftrightarrow P = R_t^{-1} \tilde{P}$$

$$R_t^{-1} = \left[\begin{array}{c|c} R^T & -R^T t \\ \hline 0^T & 1 \end{array} \right]$$



Le rotazioni di $\frac{\pi}{2}$ di tutto un sistema di riferimento sono dette **rotazioni elementari**.

**APPUNTI DI INGEGNERIA
INFORMATICA**

GAIA BERTOLINO

4. Toolbox

domenica 29 maggio 2022 14:59

TOOLBOX 2D

Il toolbox di Peter Corke permette di eseguire routine in modo veloce. Le funzioni principali sono:

- rot2(θ) \Rightarrow Calcola la matrice di rotazione

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

- trot2(θ) \Rightarrow calcola la matrice di rototraslazione

con Traslazione nulla

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



- transl2(x, y) \Rightarrow calcola la matrice di Traslazione di x e y

$$\begin{bmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix}$$

Per ottenere una matrice di rototraslazione si deve fare la moltiplicazione

$$\text{transl2}(x, y) \cdot \text{trot2}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & x \\ \sin\theta & \cos\theta & y \\ 0 & 0 & 1 \end{bmatrix}$$

\neq

$$\text{trot2}(\theta) \cdot \text{transl}(x, y)$$

ed.

```
%% 2d rotation
theta=pi/3;
R1=rot2(theta);
R2=[
    cos(theta) -sin(theta);
    sin(theta) cos(theta)
];

theta=pi/3;
TR=trot2(theta);
Tt=transl2(2,1);
TRt=TR*Tt;
plotvol([0 4 0 4]);
trplot2(TRt,'frame','1','color','b');
```

>> TR=trot2(theta)

TR = Matrice di rotazione

0.5000	-0.8660	0
0.8660	0.5000	0
0	0	1.0000

>> Tt=transl2(2,1)

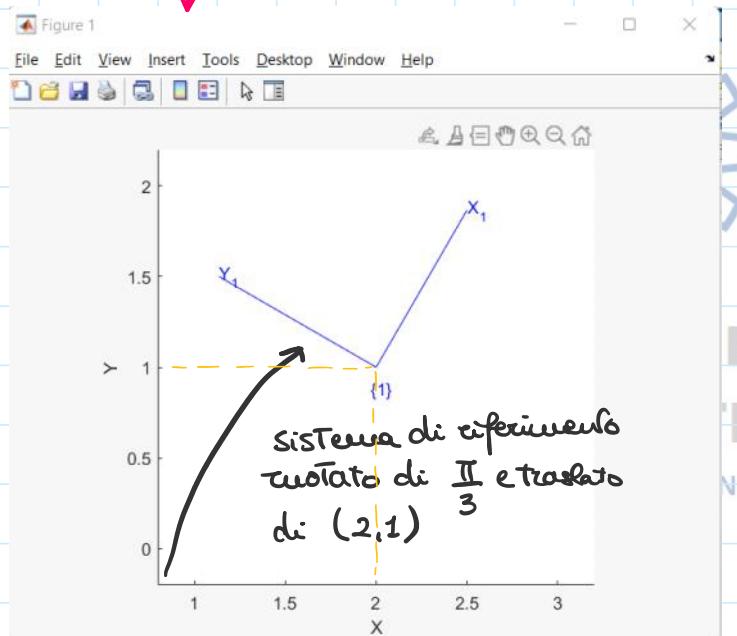
Tt = Matrice di Trasfazione

1	0	2
0	1	1
0	0	1

>> TRt=Tt*TR

TRt = Matrice di rototrasfazione

0.5000	-0.8660	2.0000
0.8660	0.5000	1.0000
0	0	1.0000



TOOLBOX 3D

Nel caso 3D, bisogna realizzare la rotazione del sistema rispetto a tre assi: ciascuna relativa ad uno degli assi.

- rotx(θ) $\Rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$

rotazione intorno all'asse x

- $\text{rot}_y(\theta) \Rightarrow \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$

Rotazione intorno all'asse y

- $\text{rot}_z(\theta) \Rightarrow \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Rotazione intorno all'asse z

- $\text{trplot}(R) \Rightarrow$ crea il grafico 3D di rotazione
dove R è una $\text{rot}_x/\text{rot}_y/\text{rot}_z$

- $\text{tranimate}(R) \Rightarrow$ crea una animazione della rotazione
dove R è una $\text{rot}_x/\text{rot}_y/\text{rot}_z$

Oltre alle matrici di rotazioni, due altri modi di esprimere le rototraslazioni sono gli angoli di Euler e i quaternioni (o asse-angolo).

Nel caso degli angoli di Euler, si hanno 27 possibili combinazioni delle rotazioni intorno ai 3 assi, di cui 12 in cui non ci sono rotazioni consecutive intorno allo stesso asse. Tuttavia, per una stessa soluzione vi sono due di queste combinazioni.

Un esempio di combinazione degli angoli di Euler è la roll-pitch-yaw detta anche 24x0x42.

es.

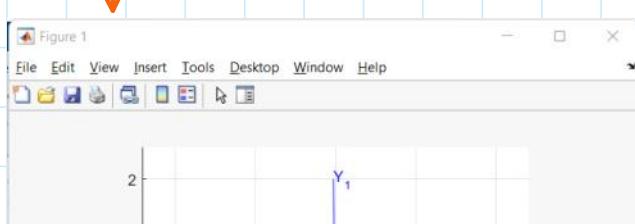
```
roll=-pi/3; → x
pitch=pi/3; → y
yaw=pi/4; → z
TR=trotz(yaw)*trot(y/pitch)*trotx(roll);
Tt=transl(2,1,-1);
TRt=Tt*TR;
trplot(TRt,'frame','1','color','b');
```

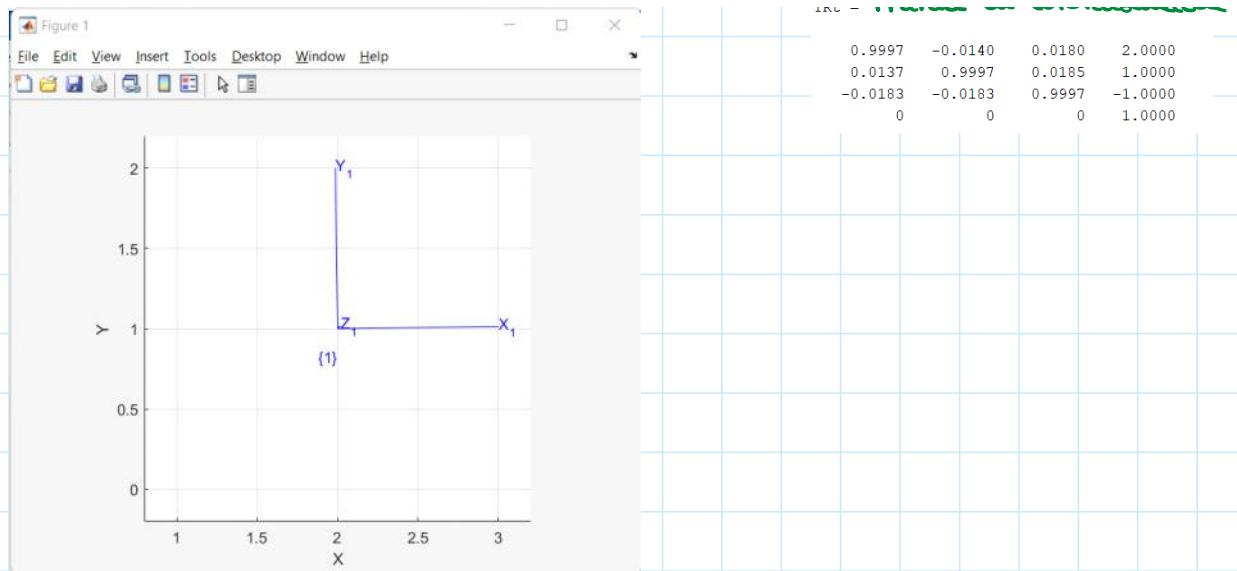
$\Rightarrow \text{TR} = \text{Matrice di rotazione } 2 \times 4$

$\Rightarrow \text{Tt} = \text{Matrice di Traslazione}$

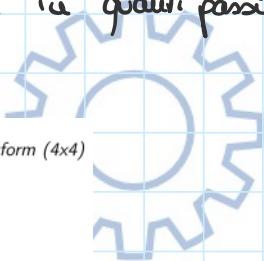
$\Rightarrow \text{TRt} = \text{Matrice di rototraslazione}$

$$\begin{bmatrix} 0.9997 & -0.0140 & 0.0180 & 0 \\ 0.0137 & 0.9997 & 0.0185 & 0 \\ -0.0183 & -0.0183 & 0.9997 & 0 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$





Interpolazione: dati dei punti distanti, ma cerca di interpolazione lineare e va a definire quali sono gli altri punti in mezzo. Ad esempio, con questo codice è possibile definire in quanti passi arrivare da T_0 a T_1 attraverso una interpolazione.



I INGEGNERIA
RMATICA

GAIÀ BERTOLINO

- ▶ **trinterp:** $T = \text{trinterp}(T_0, T_1, S)$ is a homogeneous transform (4x4) interpolated between T_0 when $S=0$ and T_1 when $S=1$

```

roll0=0;
pitch0=0;
yaw0=0;
roll1=pi/3;
pitch1=pi/5;
yaw1=4*pi/5;
T0=transl(0.4, 0.2, 0)*rpy2tr(roll0,pitch0,yaw0);
T1=transl(-0.4, -0.2, 0.3)*rpy2tr(roll1,pitch1,yaw1);
N=50;
Ts=trinterp(T0, T1,[0:N-1]/(N-1));
tranimate(Ts);

```

Nel caso di catene cinematiche aperte è più facile la cinematica diretta che l'inversa in quanto quest'ultima richiede l'inversione di equazioni non lineari. Nel caso di catene cinematiche chiuse invece è più semplice la cinematica inversa che quella diretta.

CREAZIONE DI UN ROBOT

Si data una catena cinematica

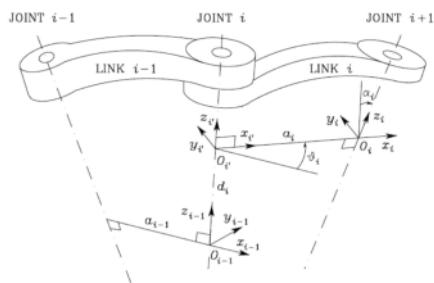


Fig. 2.16. Denavit-Hartenberg kinematic parameters

<i>parametro che varia se il giunto è PRISMATICO</i>	<i>parametri che NON variano poiché sono legati alla geometria del secondo braccio rispetto al primo</i>
θ_1	d_1

Nel toolbox è presente una funzione chiamata **Link** che riceve cinque parametri:

$$L = \text{Link} ([\theta, d, a, \alpha], \text{'standard'})$$

$$\text{dove } \alpha = \begin{cases} 0 & \text{se il giunto è rotoidale} \\ 1 & \text{se il giunto è prismatico} \end{cases}$$

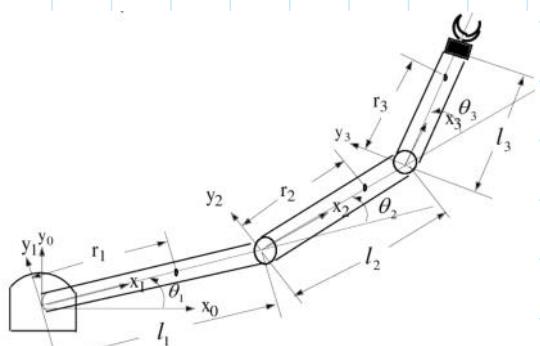
ogni Link crea un braccio e tramite la funzione **SerialLink** posso unirli in un robot. Tale funzione riceve tre parametri:

$$\text{robot} = \text{SerialLink} ([L_1 \ L_2 \ \dots \ L_n], \text{'name'}, \text{'myRobot'})$$

APPUNTI DI INGEGNERIA

I robot non sono **destri** in determinati zone dello spazio di azione ovvero non gli si riesce ad imprimere, in quelle zone, la forza necessaria a farlo agire.

Esempio Robot planare a Tre bracci



link	θ	d	a	α	σ
L_1	q_1	0	l_1	0	0
L_2	q_2	0	l_2	0	0
L_3	q_3	0	l_3	0	0

```

l1=0.25;%m
l2=0.25;%m
l3=0.2;%m
sigma1=0;%rot
sigma2=0;%rot
sigma3=0;%rot
q1_min=0;q1_max=pi/2;
q2_min=0;q2_max=pi/2;
q3_min=-pi/2;q3_max=pi/2;

%Link([theta d a alpha])
L1=Link([0,0,l1,0,sigma1],'standard');
L2=Link([0,0,l2,0,sigma2],'standard');
L3=Link([0,0,l3,0,sigma3],'standard');

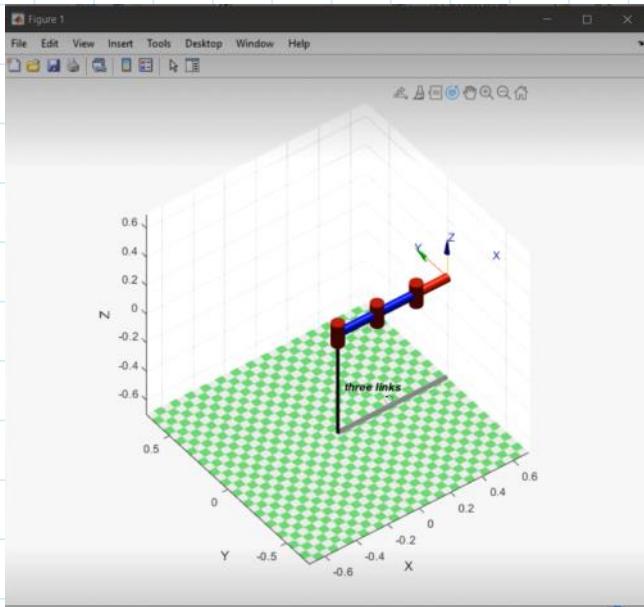
L=[L1 L2 L3];
threeLinks=SerialLink(L, 'name', 'three_links');

```

utilizza la tabella di DH e funzione ad oggetti

Attenzione! Non conviene utilizzare i pulsanti di giunto del toolbox ma conviene costruire la traiettoria affinché non ci possano.

es. richiamare `threeLink.plot([0,0,0])` genera per il robot precedente tale plot:

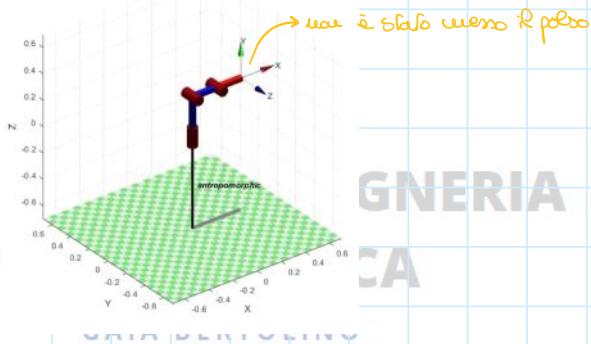


Esempio Robot autropomorfo



link	θ	d	a	α	σ
L_1	q_1	l_1	0	$\frac{\pi}{2}$	0
L_2	q_2	0	l_2	0	0
L_3	q_3	0	l_3	0	0

```
L1=Link([0,l1,0,pi/2,sigma1]);
L2=Link([0,0,l2,0,sigma2]);
L3=Link([0,0,l3,0,sigma3]);
L=[L1 L2 L3];
antro=SerialLink(L, 'name', 'antro');
```



**GENERIA
CA**

La funzione `gtay` opera la join fra delle pose del braccio in modo da creare una sorta di traiettoria. Attraverso questa funzione è possibile, ad esempio, realizzare il movimento di pick and place.

Come scegliere una soluzione?

Fra le multiple soluzioni, si sceglie una soluzione in base a due caratteristiche:

- 1) quella che rispetta i vincoli ambientali
- 2) quella che è più vicina alla configurazione attuale e, in caso,
- 3) quello più desira

La funzione `fkine` genera la soluzione di cinematica diretta ovvero la matrice di robotrasformazione necessaria a realizzare lo spostamento del robot su cui viene chiamato fino al punto passato come argomento

$$T1 = \text{antro.fkine}([0, 0, 0]) = \begin{bmatrix} 1 & 0 & 0 & l_2 + l_3 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La funzione `ikine` genera invece la soluzione della cinematica inversa

 **matrice di robotrasformazione**

$$\text{antro.ikine}(T1, 'mask', [1 1 1 0 0 0]) = \begin{bmatrix} \text{rot}^x & \text{rot}^z \\ \text{x} & \text{y} & \text{z} \end{bmatrix} \rightarrow \text{posizione di arrivo}$$

La maschera guida l'inversione.
Metendo '1' all'indice (diminuendo i gradi di libertà) potrebbe non esercitare selezione.

Il problema di cinematica inversa può essere risolto anche in altri due modi.

lez. 16 aprile, cinematica differenziale

ne deriva che

Conoscendo il valore attuale di una variabile e la sua derivata è possibile ricavare una stima del suo valore successivo.

Il concetto alla base della prima metodicità consiste nel risolvere il problema una sola volta con le condizioni iniziali e, successivamente, di risolvere il problema come di cinematica differenziale. Tale struttura nella teoria del controllo è detta ad arnoldo opero. Tuttavia, con questo procedimento si rischia a lungo andare di accumulare errori. Comunque, comunque, periodicamente "azzerare" il sistema risolvendo nuovamente il problema di cinematica inversa.

ESEMPI:

Cinematica differenziale

Ikine

```

%% manipolatore planare a due bracci

l1=0.2;%m
l2=0.2;%m

sigma1=0;%%rot
sigma2=0;%%rot

%Link([theta d a alpha])
L1=Link([0,0,l1,0,sigma1], 'standard');
L2=Link([0,0,l2,0,sigma2], 'standard');

L=[L1 L2];
twolinks=SerialLink(L, 'name', 'two links');

R=0.4;

% theta=0:deg2rad(10):2*pi;
theta_dot=deg2rad(10);%%rad/sec

y_dot=@(theta) (R*cos(theta)*theta_dot);
x_dot=@(theta) (-R*sin(theta)*theta_dot);

theta0=0;
Q0=[0.01;0.01];

Q=Q0';

T=0.1;%sec
Qk=Q0;

for k=1:2*pi/(theta_dot*T)
J=twolinks.jacob0(Qk);
J=J(1:2,1:2);
theta=theta0+theta_dot*k*T;
v=[

x_dot(theta);
y_dot(theta);
];
Qk1=Qk+pinv(J)*v*T;
Q=[

Q;
Qk1';
];
Qk=Qk1;
end

twolinks.plot(Q);

```

↳ casella ad anello aperto

```

%% manipolatore planare a due bracci

l1=0.2;%m
l2=0.2;%m

sigma1=0;%%rot
sigma2=0;%%rot

%Link([theta d a alpha])
L1=Link([0,0,l1,0,sigma1], 'standard');
L2=Link([0,0,l2,0,sigma2], 'standard');

L=[L1 L2];
twolinks=SerialLink(L, 'name', 'two links');

R=0.4;%0.3;

% theta=0:deg2rad(10):2*pi;
theta_dot=deg2rad(10);%%rad/sec

yt=@(theta) (R*sin(theta));
xt=@(theta) (R*cos(theta));

% % % y_dot=@(t) (R*cos(t)*theta_dot);
% % % x_dot=@(t) (-R*sin(t)*theta_dot);

theta0=0;
Q0=[0.01;0.01];

Q=Q0';

T=0.1;%sec
Qk=Q0;

for k=1:2*pi/(theta_dot*T)
theta=theta0+theta_dot*k*T;
Rt=transl(xt(theta),yt(theta),0)*trotz(theta);

Qk1=twolinks.ikine(Rt,Qk,'mask',[1 1 0 0 0 0]);
Q=[

Q;
Qk1(:)';
];
Qk=Qk1;
k
end

twolinks.plot(Q);

```

↳ sempre corretto