

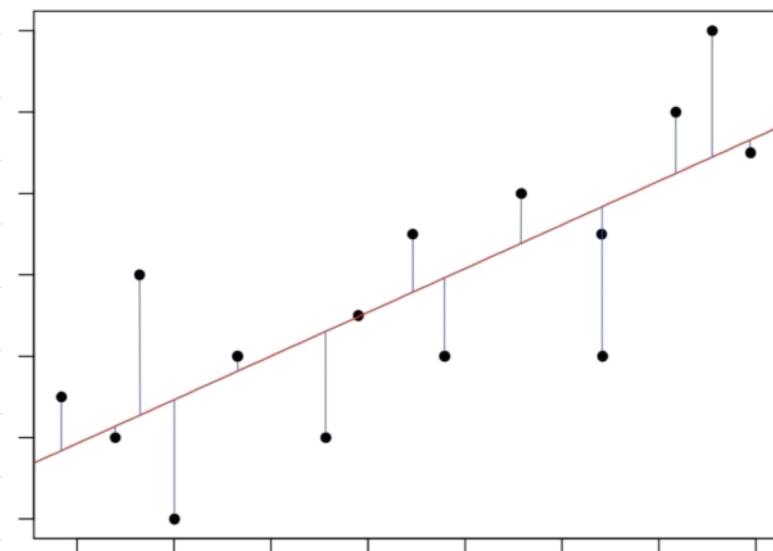
## 5. Vincoli

martedì 31 maggio 2022

15:46

Nella progettazione delle Traiettorie si parla di vincoli **hard** e vincoli **soft** dove i primi vanno rispettati assolutamente, i secondi invece anche solo approssimativamente.

Fra le Tecniche, si utilizzano i LMS o anche detti **Least Mean Squares** che consistono nell'approssimare delle soluzioni di maschi sovradeterminate (numero di sistemi con più equazioni che incognite) minimizzando la somma dei quadrati dei residui fatti in ogni rispetto di ciascuna equazione

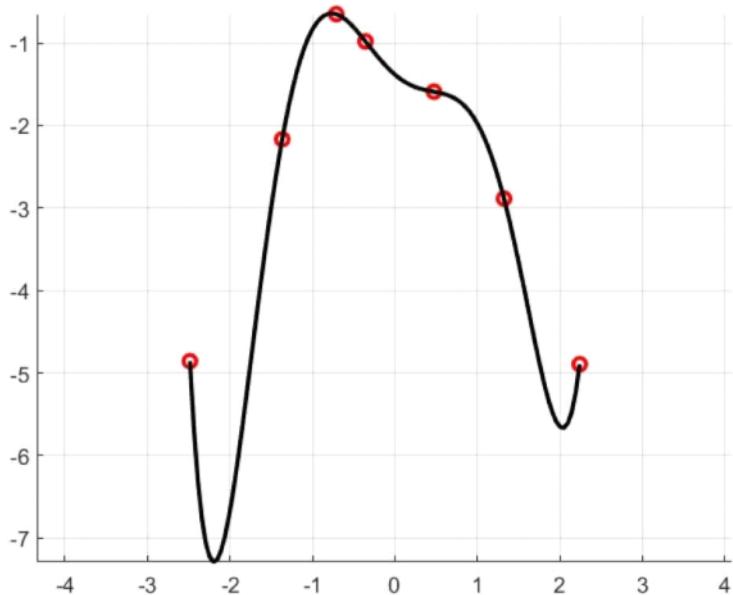


ERIA

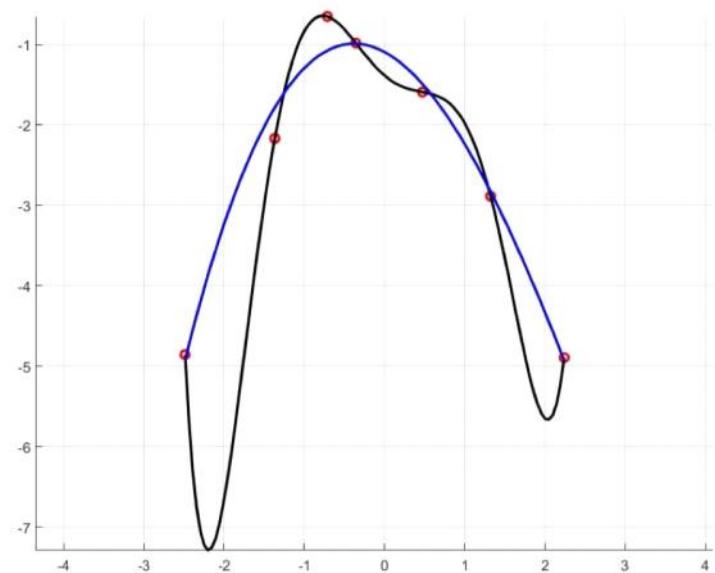
Gli scarsi vengono presi al quadrato per evitare che scarsi positivi e negativi sommati si annullino / compensino.

Un'altra Tecnica è l'interpolazione polinomiale detta **polynomial fit** che richiede di essere di grado  $n-1$  per passare da  $n$  punti.

$N = 7$  points  $\rightarrow$  7 constraints  $\rightarrow$  6-th degree



Tuttavia questo approccio genera "spostamenti" superflui che con altre curve si eviterebbero. Tuttavia vi consente una maggiore precisione.



RIA

Possibile domanda di esame:

Calcolare la cinematica diretta di una catena cinematica per via  
geometrica o analitica

↳ punto per punto calcolando direttamente la matrice

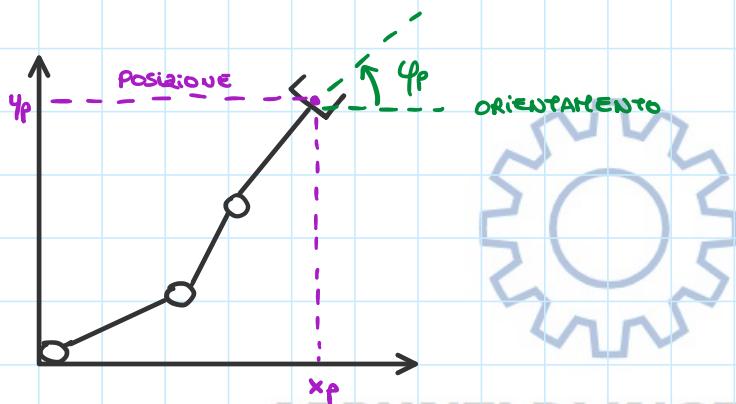
Compito della cinematica inversa è calcolare i valori delle variabili di giunto ma non come i giunti debbano arrivare.

Inoltre, per evitare a delle carenze fisiche / ostacoli, si introduce **RIDONDANZA** nel robot introducendo più giunti (e dunque più gradi di libertà) per compensare le eventuali carenze.

### VINCOLI DI ORIENTAMENTO

Per orientamento si intende l'angolazione dell'end effector rispetto agli assi.

es.



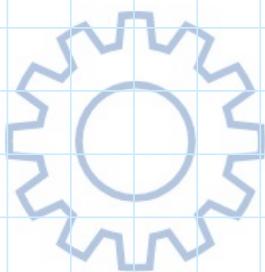
### Tecnica di disaccoppiamento

In questo caso, l'orientamento è determinato dall'ultimo giunto. Quiudi disaccoppio il problema di orientamento e quello di posizionamento e vedo dove dovrebbe essere posto il giunto di orientamento per dare soluzione al problema e verifico se corrisponde ad una delle soluzioni del problema di posizionamento. In caso affermativo, è trovata una soluzione.

La strategia di cinematica inversa viene quindi suddivisa in due case, una di **posizioning** e una di **attitude** e una volta che il problema di posizionamento è risolto con la parte di posizioning, con la parte di attitude si risolve quello di orientamento.

Nel caso del polso sferico, si prende come punto di partenza è il giunto intermedio del polso sferico e si considerano, solitamente, i giunti del

intermedio del polso sferico e si considerano, solitamente, i gradi del polso per l'orientamento e gli altri tre per il posizionamento. L'orientamento si riesce facilmente a calcolare tramite la cinematica diretta.



# APPUNTI DI INGEGNERIA INFORMATICA

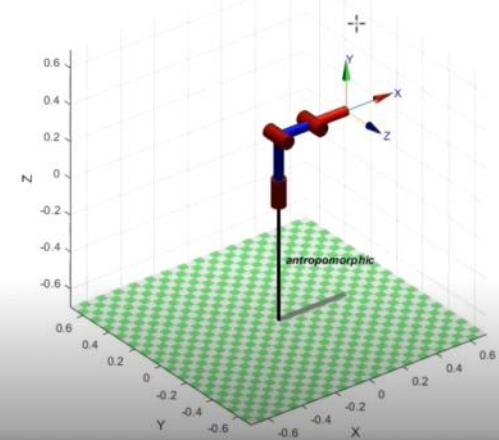
GAIA BERTOLINO

## 6. Robotica mobile

giovedì 2 giugno 2022 20:43

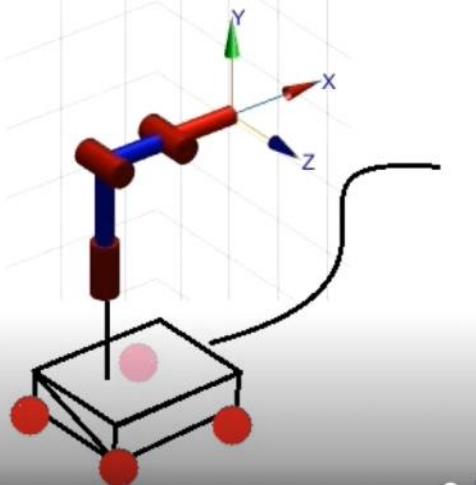
### Robotics

The robot is **FIXED** (more or less):  
fixed reference frames.



### Mobile Robotics

The robot is **ABLE TO MOVE**:  
**moving** reference frame.



Che la robotica mobile non si intende solo sullo spazio  
Tranne ruote ma qualsiasi tipo di **movimento**  
Inoltre, può esservi una coesistenza di parti mobili e di parti  
industriali.

CICLORATI  
UMANOIDI  
CANI

## APPUNTI DI INGEGNERIA INFORMATICA

Per quanto riguarda il movimento su ruote, posizione di affissione  
e tipologia influenzano il moto. GAIA BERTOLINO

Inoltre, le ruote possono essere ad esse collegate o indipendenti.

In base alle ruote si hanno dei veicoli: **autonomi** poiché vi sono  
posizioni che non possono raggiungere.

### Sistema di controllo:

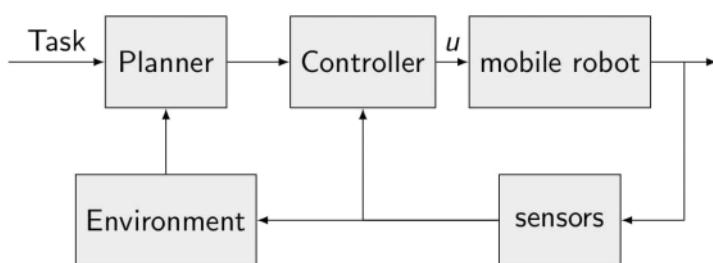


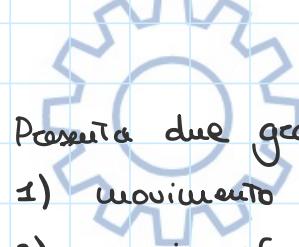
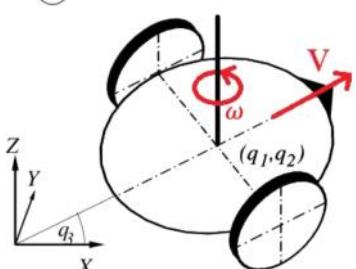
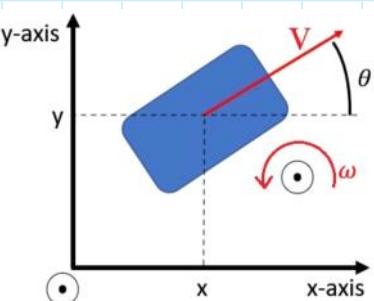
Figure: Mobile robot control scheme

Parlando solamente di cinematica, non ci si interessa se il piano è obliquo o meno in quanto il sistema di riferimento è legato con il piano e non ha una angolazione. In particolare, ci si cura dell'aspetto dinamico solo nel caso in cui risultino rilevanti le componenti fisiche dell'oggetto (es. la massa).

Inoltre, bisogna riconoscere che ci possa essere un **errore di trasferimento** ovvero un errore dato dalla dinamica del robot che causa un ritardo nella "partenza" stessa (es. attrito) o nei movimenti. Tuttavia, è un errore facilmente ignorabile poiché iniziale ovvero esiste solo fino a quando il sistema non arriva a regime. Insieme a questo, anche altri fenomeni vengono trascurati.

A seconda dello sfondato del robot, esso avrà delle limitazioni intrinseche.

### Robot uniciclo



Presenta due gradi di libertà:

- 1) movimento lineare frontale
- 2) movimento di rotazione

### UNTI DI INGEGNERIA INFORMATICA GAIA BERIOLINO

Indicando con  $v(t)$  la velocità di guida e con  $\omega(t)$  quella di rotazione, si ha che

$$\begin{cases} \dot{x}(t) = v(t) \cdot \cos(\theta(t)) \\ \dot{y}(t) = v(t) \cdot \sin(\theta(t)) \\ \dot{\theta}(t) = \omega(t) \end{cases}$$

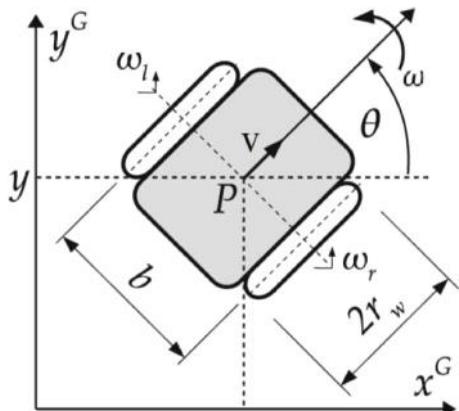
Si definisce **viacolo olomorfo** un viacolo di forma Pfaffiana che è integrabile ovvero si riesce ad eliminare la dipendenza dalla velocità. I viacoli non integrabili sono detti **anolomorfi**.

I viacoli olomorfi prevedono la presenza di limitazioni nelle configurazioni mentre gli anolomorfi sulle velocità.

### Modello differential drive

Il **differential drive** è un meccanismo di guida di un robot che presenta un

Il **differential drive** è un meccanismo di guida di un robot che presenta una differenziale fra le ruote, il quale permette alle due ruote di ruotare con velocità diverse.



$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{r}{2} & -\frac{r}{2} \\ \frac{r}{b} & -\frac{r}{b} \end{bmatrix} \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix}$$

\* Essendo invertibile, si ha il seguente per il differential drive e viceversa

Il modello di differential drive permette che avvenga una traslazione insieme ad una rotazione.

Il modello unicycle e il differential drive sono matematicamente identici perché la relazione che lega la velocità  $v$  e  $\omega$  dell'unicycle alle velocità  $\omega_R$  e  $\omega_L$  del differential drive è sempre invertibile.

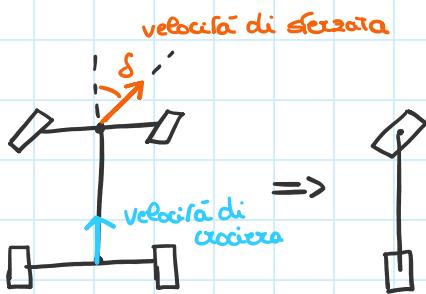
Il modello differential drive è molto usato nei percorsi meno stabili.

GAIA BERTOLINO

### Modello car-like

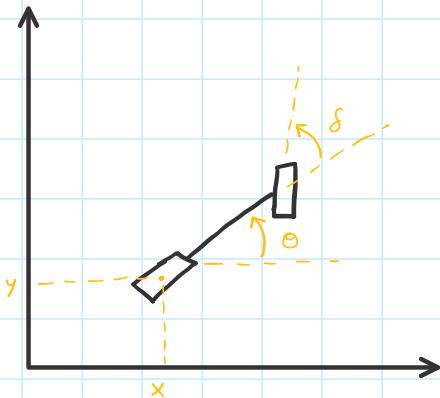
Il modello del **car-like** presenta invece degli assi fissi lungo i quali avvengono delle rotazioni studiate in dinamica che sono di tipo roll and pitch.

Tale modello, avendo le ruote accoppiate a due a due con gli stessi movimenti, viene spesso rappresentato con sole due ruote (Trascurando i fenomeni di roll and pitch).



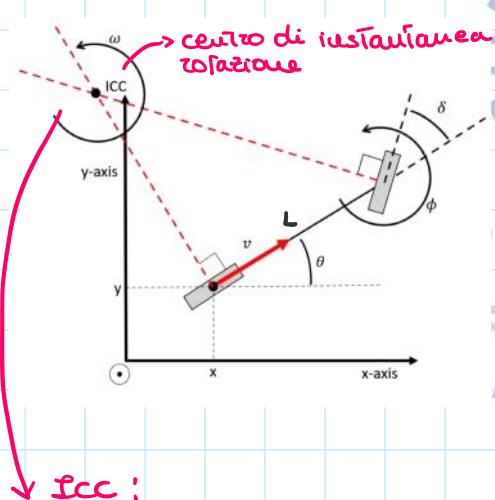
Le coordinate del modello sono

- $x, y \Rightarrow$  posizione del centro di massa
- $\theta \Rightarrow$  orientamento
- $\dot{\gamma} \Rightarrow$  angolo di sfera



Nel caso della trazione posteriore, la velocità è diretta verso l'asse della macchina, nel caso della trazione anteriore è diretta verso l'asse della ruota

#### • TRAZIONE POSTERIORE



ICC:

Ottogonal alle ruote

Modello complessivo

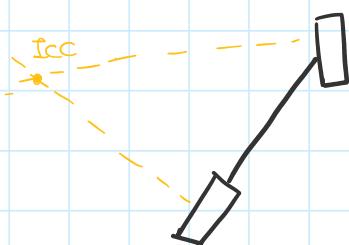
$$\begin{cases} \dot{x} = v \cdot \cos \theta \\ \dot{y} = v \cdot \sin \theta \\ \dot{\theta} = \omega = \frac{v \cdot \tan \delta}{L} \\ \dot{\phi} = \dot{\phi} \end{cases}$$

I DI INGEGNERIA

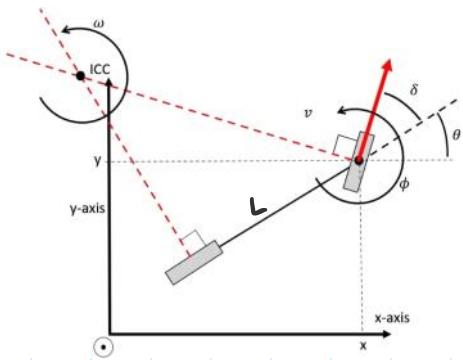
FORMATICA  
AIA BERGONI

$\rightarrow$  il CAR-LIKE

In questo modello NON è possibile muovere senza trazione (in entrambe le trazioni)



#### • TRAZIONE ANTERIORE



$$\begin{cases} \dot{x} = v \cdot \cos(\theta + \delta) \\ \dot{y} = v \cdot \sin(\theta + \delta) \\ \dot{\theta} = \frac{v}{R} = \frac{v \cdot \sin \delta}{L} \\ \dot{\delta} = \dot{\phi} \end{cases}$$

## codici

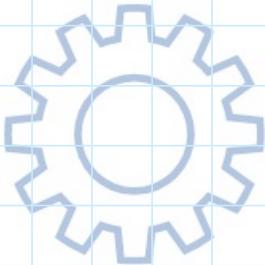
- **Unicicle**

```

1 - clear;
2 - close all;
3 - clc;
4
5
6 - [t,X] = ode23(@robot, [0 10], [0,0,pi/2]');
7 - x = X(:,1);
8 - y = X(:,2);
9 - theta = X(:,3);
10
11 % Rappresentazione come punto
12 plot(x,y, 'xk');
13 axis('equal');
14 plot(theta);
15
16 % Rappresentazione come triangolo orientato
17 xM = max(x); xm = min(x);
18 yM = max(y); ym = min(y);
19
20 h = figure();
21 hold on;
22 grid on;
23 axis([xm xM ym yM]);
24 for k=1 : length(t)
25   cla;
26   Xk = X(k,:);
27   [h] = plot_robot(Xk, 1, h);
28
29   axis('equal');
30   pause(0.1);
31 end
32

```

```
33  [ ] function[Xdot] = robot(t,x)
34 - | v = 0;
35 - | w = 1;
36 - | [Xdot] = unicycle(X,v,w);
37 - | end
38
39  [ ] function[Xdot] = unicycle(X,v,w)
40 - | x = X(1);
41 - | y = X(2);
42 - | theta = X(3);
43
44 - | xdot = x*cos(theta);
45 - | ydot = v*sin(theta);
46 - | thetadot = w;
47
48 - | Xdot = [xdot; ydot; thetadot];
49 - | end
```



# APPUNTI DI INGEGNERIA INFORMATICA

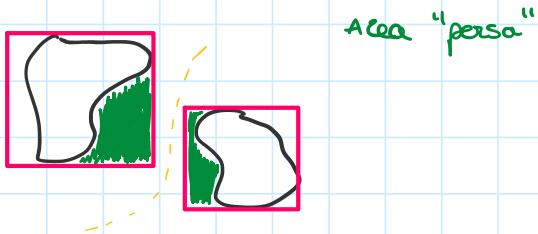
GAIA BERTOLINO

## 7. Traiettoria

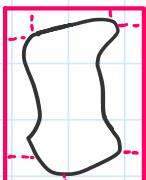
sabato 4 giugno 2022 19:10

Nella scelta di una Traiettoria, non sempre quello minimo è il migliore, soprattutto se bisogna passare in mezzo a degli ostacoli.

Inoltre, ogni ostacolo che ha una forma particolare viene approssimata per eccesso ad una forma uota. Ciò ovviamente esclude delle zone immediatamente intorno alla forma.



In particolare, si parla di **ingranare gli ostacoli** ovvero considerarli più massicci di quanto in realtà sono. Ciò causa la perdita di spazio di azione ma crea una certa sicurezza nel movimento del robot in quanto se anche esso dovesse Toccare il "bordo" dell'oggetto in realtà starebbe facendo l'ingranatura dello stesso senza alcuna conseguenza



APPUNTI DI INGEGNERIA  
INFORMATICA

GAIA BERTOLINO

## PATH PLANNING

Le tre domande della robotica mobile sono

- dove devo andare  $\Rightarrow$  path planning
- come ci arrivo
- dove sono

Queste sono le domande a cui rispondere per risolvere un problema di robotica mobile.

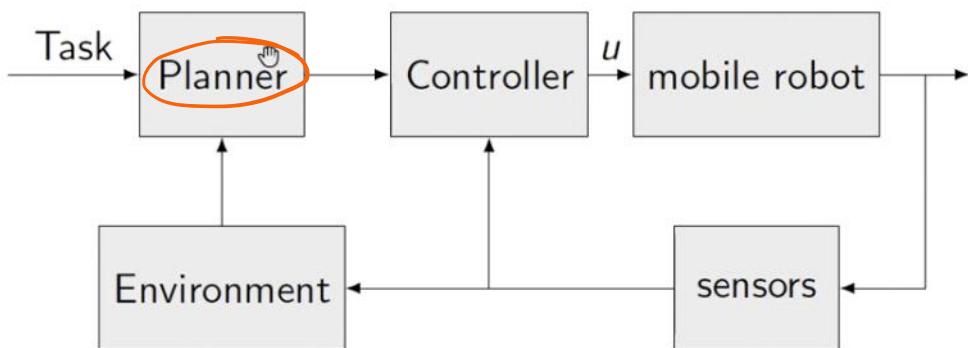


Figure: Mobile robot control scheme

Il path planning non si applica solo ai robot ma anche alle persone.

Considerando i robot più diffusi si possono avere dei clash a causa della sua massa. Le soluzioni sono due:

1) considerare la forma del robot

contro: cambiando robot va modificato l'algoritmo

2) considerare degli ostacoli più grossi per compensare la massa del robot (che è considerato puntiforme)

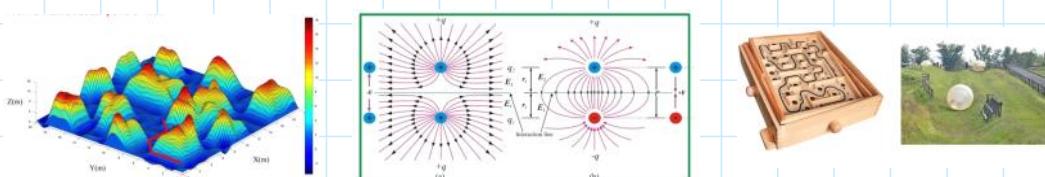
pro: cambiando il robot va solo modificata l'ingresso di dati degli ostacoli

## APPUNTI DI INGEGNERIA

### TECNICA DEI POTENZIALI ARTIFICIALI CONTINUI

È una tecnica che prevede di trasformare lo spazio di movimento in un campo plasmato che presenta

- zone a quote diverse
- zone a potenziale diverso



Il potenziale è una funzione e il campo potenziale è il suo gradiente. Il gradiente rappresenta la crescita del potenziale.

Si fa finta che in uno spazio vi sia un campo potenziale che sia

- repulsivo o bypassante per gli ostacoli
- attrattivo verso i punti

- repulsivo o bypassante per gli ostacoli
  - attrattivo verso il goal
- e ciò permette di matematizzare lo spazio stesso.

La funzione potenziale sarà:

$$J(x) = w_a \cdot \underbrace{f_a(x(t), G)}_{\substack{\text{POTENZIALE ATTRATTIVO} \\ \rightarrow \text{VERSO IL GOAL} \\ \text{Posizione robot}}} + w_o \cdot \sum_{i=1}^{N_{\text{ostacoli}}} \underbrace{f_c(x(t), O_i)}_{\substack{\text{Posizione ostacoli} \\ \text{No}}} + \underbrace{f_r(x(t), O_i)}_{\substack{\text{Posizione robot} \\ \leftarrow \text{POTENZIALI} \\ \text{REPULSIVI DEGLI} \\ \text{OSTACOLI}}}$$

PESI DATI AI POTENZIALI

In alcuni casi è anche utile sapere la direzione e velocità degli ostacoli per evitare di scontrarsi e dunque per scegliere un'altra direzione.

### Potenziale attrattivo:

Per descrivere i potenziali si utilizzano delle funzioni che rispecchiano sia la descrizione matematica dell'ostacolo, sia con antigradiente che permette una ottimizzazione migliore nel percorso verso il goal.

In particolare, la funzione scelta per il potenziale attrattivo deve:

- avere un minimo locale nella posizione di goal
- in nulli così avere un minimo locale

### Potenziale repulsivo:

GAIA BERTOLINO

Nel caso del potenziale repulsivo si usano funzioni asintotiche che permettono di non raggiungere mai l'energia necessaria a toccare l'ostacolo.

Il gradiente pista verso l'ostacolo mentre l'antigradiente contro.

Inoltre, si utilizzano funzioni che più ci si allontana dal picco della funzione (ovvero dall'ostacolo) meno attrazione ci sia. Addirittura, per azzerrarla, si può ricorrere ad una definizione a tratti della funzione che permette anche di assicurare il minimo locale nel goal.

**Attenzione:** i potenziali repulsivi valgono così: deati nell'intorno del robot.

Altrimenti, è il solo potenziale attrattivo a guidare il robot.

Dunque, per arrivare al goal bisogna seguire l'antigradiente della funzione potenziale sopra descritta e la realizzazione diventa un problema di ottimizzazione.

potenziale sopra descritta e la realizzazione diventa un problema di ottimizzazione. Una delle tecniche utilizzate prevede una discesa del gradiente realizzata tramite dei passi definiti che uniti formano il percorso.

Inoltre, vi è il rischio della presenza di minimi locali (dove il gradiente è zero) nel caso in cui ci si trovi su un'asse in direzione di un ostacolo e del goal a causa del compensarsi a vicenda dei parametri attrattivi e repulsivi.



Per evitare che il robot si fermi in un minimo locale senza avanzare più si misura la distanza dal goal e se questa supera una certa soglia e si può:

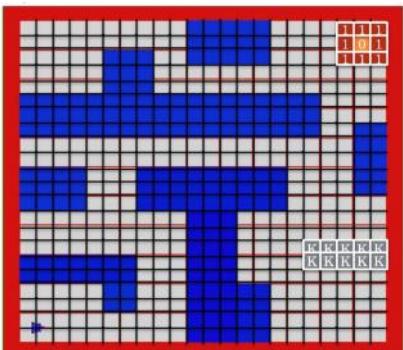
- applicare la tecnica wuro-wuro che prevede che il robot cominci ad ignorare il potenziale e cominci a girarci intorno ad una certa distanza fino a superarlo
- applicare la tecnica dei vortici ovvero utilizzare dei vettori di movimento perpendicolari a quelli di repulsione. Un punto è che ci sono casi, come ad esempio in presenza di più ostacoli, che anche questi potenziali si compensino.
- utilizzare un solo potenziale e usare una somma e dunque switchare fra potenziale attrattivo e repulsivo che singolarmente funzioneranno sicuramente

GAIA BERTOLINO

### TECNICA DEI POTENZIALI ARTIFICIALI DISCRETI

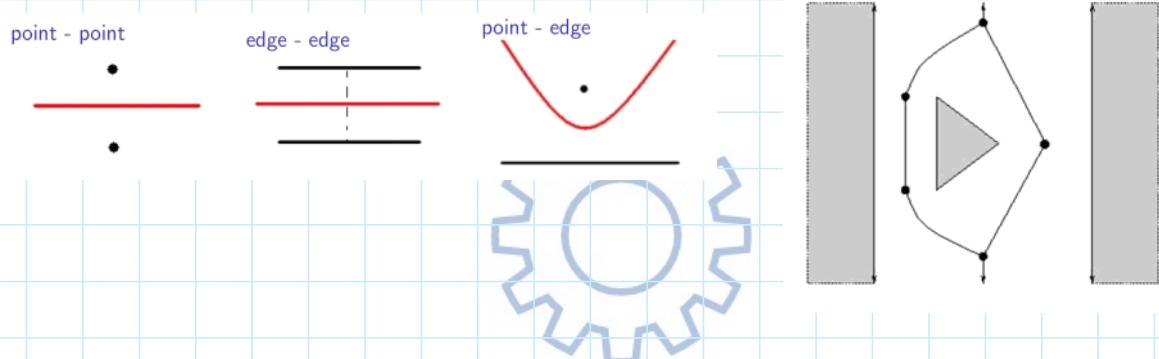
Si divide tutto lo spazio in una griglia e, a partire dalla cella di partenza, si riempiono tutte le celle intorno con  $k_{t+1}$  dove  $k$  è il valore della cella precedente (se tale cella è vuota) o con  $\infty$  (se tale cella contiene anche un solo pezzo di ostacolo).

Si procede poi a passare dalla cella  $K$  alla  $k_{t+1}$  e ritorno indietro quando uno stato arriverà al goal e non esiste la cella  $k_{t+1}$  per cambiare la sua decisione precedente secondo un approccio di backtracking.



### TECNICA DEI DIAGRAMMI DI VORONOI

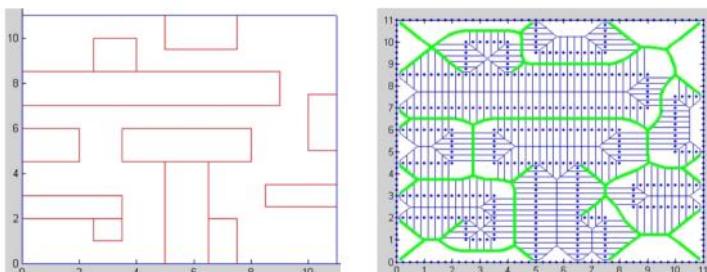
Si partitiona lo spazio in modo che tutti i punti sul contorno di una regione sono i punti a distanza minima fra due regioni confinanti.



esempio:

### APPUNTI DI INGEGNERIA

ICA  
VO



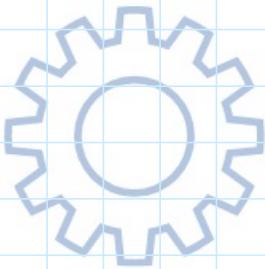
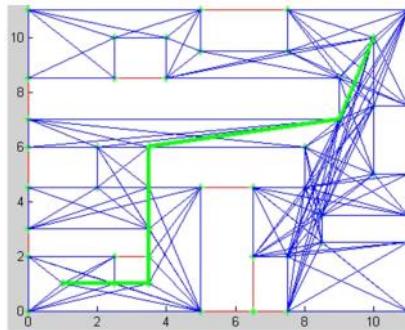
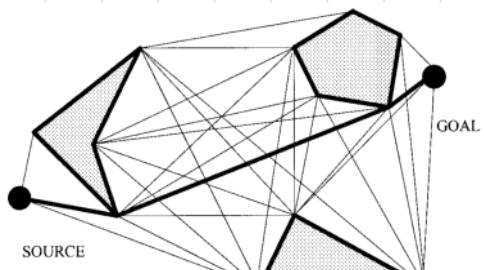
- Si creano i punti
- Si invoca voronoi
- Si studiano le linee per vedere eventuali intersezioni

Una volta calcolato il grafo, possono anche cambiare start e goal ma il grafo sarebbe lo stesso.

### TECNICA DEI GRAFI DI VISIBILITÀ

Si creano un grafo nello spazio che collega i nodi di start e goal e i vertici degli oggetti (solo se il collegamento non interseca un ostacolo). In questo caso si passa però tramite spezzate e sui vertici va allo stesso tempo è più efficiente del percorso di voronoi.

In questo caso si passa però tramite spezzate e sui vertici ma allo stesso tempo è più efficiente del percorso di voronoi.



## APPUNTI DI INGEGNERIA INFORMATICA

GAIA BERTOLINO

## 8. Localizzazione

domenica 5 giugno 2022 18:48

Nel problema di Localizzazione rientra anche la stima dello stato del robot come ad esempio l'orientamento.

È inoltre definito come un problema di **sensor fusion** ovvero fusione e interpretazione congiunta dei dati dei sensori. Questi sono condizionati dal robot e ciò causa un'approssimazione teoricamente limitata nella sua precisione.

### Accelerometro

Esso misura l'accelerazione che agisce sul sensore stesso ed è usato solitamente per valutare l'orientamento.

### Giroscopio

Esso misura la velocità angolare (che è la derivata dell'angolo) e da questo si riesce a valutare l'angolo integrando la velocità angolare

### RSSI

È un parametro utilizzato per misurare il potere di un segnale.

In particolare, esso descrive la relazione fra trasmittitore e ricevitore:

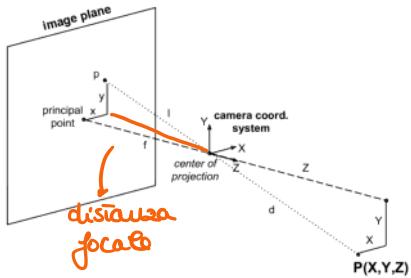
$$\text{RSSI}(d) = \text{RSSI}(d_0) - \frac{10}{u} \cdot \log_{10} \left( \frac{d}{d_0} \right)$$

potenza del segnale alla distanza  $d$   
fattore di propagazione  
distanza di riferimento

da questa deriva

$$d = d_0 \cdot \frac{\text{RSSI}(d_0) - \text{RSSI}(d)}{10 u} \cdot d_0$$

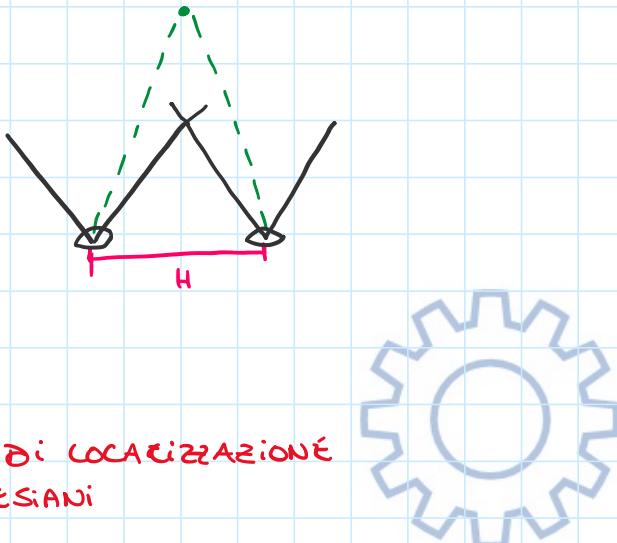
### MODELLO PINHOLE



Modello di rappresentazione dell'acquisizione delle immagini.

Sorge un problema di prospettiva: gli oggetti lontani misurano quanto quelli vicini.

La profondità viene visualizzata tramite la visione con due occhi.  
Tale approccio è detto **stereoscopico**



### TECNICHE DI LOCALIZZAZIONE FILTRI BAYESIANI

Sono degli stimatori che minimizzano gli errori sui valori attesi provocati dal rumore e massimizzare l'affidabilità della stima.

## INFORMATICA

Il concetto alla base è di modellare i dati in maniera sfocata attraverso un modello sfocastico che è in grado di esprimere varianza, media, valore atteso ecc.

### FINCHERPRINTING

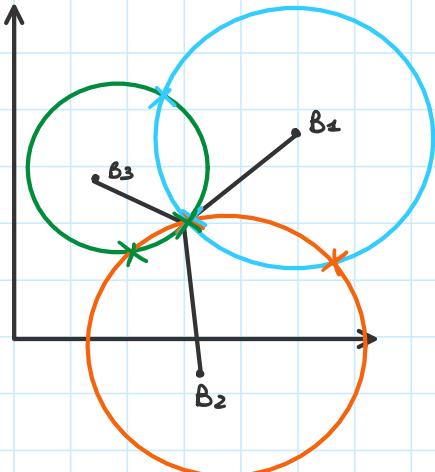
Questa tecnica misura i valori di un segnale nell'ambiente in modo tale da legare ad ogni spazio un valore di riconoscimento.

Dunque, quello che generalmente viene considerato come disturbo, viene ora considerato parte integrante.

### PROSSIMITÀ

Nell'ambiente vengono posti dei sensori e delle trasmettenti e si riesce ad individuare la posizione dell'oggetto in base a qual è il sensore che ricevette il segnale più forte.

## TRILATERAZIONE (O MULTILATERAZIONE)



Le Bi sono delle distanze che misurano le distanze dalla posizione da localizzare.

Con almeno 3 di queste distanze, si ottengono delle circonference la cui intersezione dà il punto.

Dunque la localizzazione si riduce alla risoluzione di un sistema di equazioni.

La Trilaterazione può essere applicata sia in  $\mathbb{R}^2$  che  $\mathbb{R}^3$ , ovvero anche in tre dimensioni.

Inoltre, la soluzione potrebbe essere un'area e non un punto.

La Trilaterazione può essere ottenuta tramite una soluzione approssimata, iterativa o una Terza modalità



### Trilateration: iterative solution

The problem to be solved can be rewritten as

$$\begin{cases} \|\hat{P} - V_1\| = d_1 \\ \vdots \\ \|\hat{P} - V_N\| = d_N \end{cases} \rightarrow \text{find } \hat{P} \in \mathbb{R}^q \text{ s.t. } \mathcal{F}(\hat{P}) = D$$

where  $q = 2$  or  $q = 3$  in 2D and 3D cases respectively,  $\mathcal{F}(P) : \mathbb{R}^q \rightarrow \mathbb{R}^N$  and  $D = [d_1, \dots, d_N]$ .

Assume to have a **rough initial candidate solution**  $\hat{P}_0$   
then the Taylor series of  $\mathcal{F}(P)$  centered in  $\hat{P}_0$  is

$$\mathcal{F}(P) = \mathcal{F}(\hat{P}_0) + \frac{\partial \mathcal{F}}{\partial P} \Big|_{\hat{P}_0} (P - \hat{P}_0) + \dots$$

EGNERIA  
ICA

IO

Localization 160 / 205

### Trilateration: iterative solution

By truncating the Taylor series now

$$\mathcal{F}(P) \approx \mathcal{F}(\hat{P}_0) + \frac{\partial \mathcal{F}}{\partial P} \Big|_{\hat{P}_0} (P - \hat{P}_0)$$

which becomes, since  $\mathcal{F}(P) = D$  to solve the problem

$$D \approx \mathcal{F}(\hat{P}_0) + \frac{\partial \mathcal{F}}{\partial P} \Big|_{\hat{P}_0} (P - \hat{P}_0)$$

and, by solving w.r.t.  $P$

$$\hat{P} = \hat{P}_0 + J_0^+ (D - \mathcal{F}(\hat{P}_0))$$

where  $J_0 \triangleq \frac{\partial \mathcal{F}}{\partial P} \Big|_{\hat{P}_0}$  and  $J_0^+$  is the pseudoinverse matrix of  $J_0 \in \mathbb{R}^{(N \times q)}$

Localization 161 / 205

## Trilateration: iterative solution

The procedure can be reiterated until a good solution is obtained

Starting from  $\hat{P}_0$ , defining the thresholds  $\bar{\mathcal{E}}$ ,  $\bar{\mathcal{P}}$ , let  $k = 0$  and

1. compute the error  $\mathcal{E}_k = \|\mathcal{F}(\hat{P}_k) - D\|$
2. if  $\mathcal{E}_k \leq \bar{\mathcal{E}}$  break
3. otherwise
  4. update the solution as  $\hat{P}_{k+1} = \hat{P}_k + J_k^+ (D - \mathcal{F}(\hat{P}_k))$  where  $J_k \triangleq \frac{\partial \mathcal{F}}{\partial P} \Big|_{\hat{P}_k}$
  5. if  $\|\hat{P}_{k+1} - \hat{P}_k\| \leq \bar{\mathcal{P}}$  break
  6.  $k = k + 1$  and continue to point 1
6.  $\hat{P} = \hat{P}_k$

To deal with the cases of no solution

a maximum number of iterations should be defined and used at point 2 and 6.

Localization 162/205

## Trilateration: approximate solution

Focus on the 3D case

The single equation to be solved is

$$\|P - V_i\| = d_i \rightarrow \|P - V_i\|^2 = d_i^2$$

which means that

$$(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = d_i^2$$

and then

$$x^2 - 2xx_i + x_i^2 + y^2 - 2yy_i + y_i^2 + z^2 - 2zz_i + z_i^2 = d_i^2 \rightarrow$$

$$(x_i^2 + y_i^2 + z_i^2 - d_i^2) + [-2x_i, -2y_i, -2z_i]^T \begin{bmatrix} x \\ y \\ z \end{bmatrix} + (x^2 + y^2 + z^2) = 0$$

## Trilateration: approximate solution

Starting from

$$(x_i^2 + y_i^2 + z_i^2 - d_i^2) + [-2x_i, -2y_i, -2z_i]^T \begin{bmatrix} x \\ y \\ z \end{bmatrix} + (x^2 + y^2 + z^2) = 0$$

by defining  $\mathcal{K} \triangleq (x^2 + y^2 + z^2)$  it follows that

$$[-2x_i, -2y_i, -2z_i, 1]^T \begin{bmatrix} x \\ y \\ z \\ \mathcal{K} \end{bmatrix} = d_i^2 - (x_i^2 + y_i^2 + z_i^2)$$

which can be written as  $\phi_i^T \theta = b_i$  with

$$\phi_i = [-2x_i, -2y_i, -2z_i, 1], \theta = [x, y, z, \mathcal{K}], b_i = d_i^2 - (x_i^2 + y_i^2 + z_i^2)$$

Localization 164/205

## Trilateration: approximate solution

At this point the linear LMS problem can be solved

$$\begin{bmatrix} -2x_1 & -2y_1 & -2z_1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ -2x_N & -2y_N & -2z_N & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \mathcal{K} \end{bmatrix} = \begin{bmatrix} d_1^2 - (x_1^2 + y_1^2 + z_1^2) \\ \vdots \\ d_N^2 - (x_N^2 + y_N^2 + z_N^2) \end{bmatrix} \rightarrow \Phi \theta = b$$

It is an approximated solution

because, in most of the cases

$$\hat{\mathcal{K}} \neq (\hat{x}^2 + \hat{y}^2 + \hat{z}^2)$$

The more  $\hat{\mathcal{K}} \approx (\hat{x}^2 + \hat{y}^2 + \hat{z}^2)$

the more the approximation should be valid.

$$\text{and then } \hat{\theta} = \Phi^+ b = \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \\ \hat{\mathcal{K}} \end{bmatrix}$$

Localization 165/205

## Trilateration: third solution

Starting from the equations

$$\begin{aligned} E_1. \quad & x^2 - 2xx_1 + x_1^2 + y^2 - 2yy_1 + y_1^2 + z^2 - 2zz_1 + z_1^2 = d_1^2 \\ E_2. \quad & x^2 - 2xx_2 + x_2^2 + y^2 - 2yy_2 + y_2^2 + z^2 - 2zz_2 + z_2^2 = d_2^2 \\ \vdots \\ E_N. \quad & x^2 - 2xx_N + x_N^2 + y^2 - 2yy_N + y_N^2 + z^2 - 2zz_N + z_N^2 = d_N^2 \end{aligned}$$

by computing  $E_1 - E_2$ ,  $E_1 - E_3$ , ...,  $E_{N-1} - E_N$

$$\left\{ \begin{array}{lcl} -2(x_1 - x_2)x & -2(y_1 - y_2)y & -2(z_1 - z_2)z = \tilde{d}_1 - \tilde{d}_2 \\ -2(x_2 - x_3)x & -2(y_2 - y_3)y & -2(z_2 - z_3)z = \tilde{d}_2 - \tilde{d}_3 \\ \vdots \\ -2(x_{N-1} - x_N)x & -2(y_{N-1} - y_N)y & -2(z_{N-1} - z_N)z = \tilde{d}_{N-1} - \tilde{d}_N \end{array} \right.$$

where  $\tilde{d}_i \triangleq d_i - (x_i^2 + y_i^2 + z_i^2)$

Localization 166/205

## Trilateration: third solution

As a consequence

$$\begin{bmatrix} -2(x_1 - x_2) & -2(y_1 - y_2) & -2(z_1 - z_2) \\ -2(x_2 - x_3) & -2(y_2 - y_3) & -2(z_2 - z_3) \\ \vdots \\ -2(x_{N-1} - x_N) & -2(y_{N-1} - y_N) & -2(z_{N-1} - z_N) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \tilde{d}_1 - \tilde{d}_2 \\ \tilde{d}_2 - \tilde{d}_3 \\ \vdots \\ \tilde{d}_{N-1} - \tilde{d}_N \end{bmatrix}$$

which becomes  $\Phi\theta = b$  and then  $\hat{\theta} = \Phi^+ b$ .

A set of  $N(N - 1)/2$  equations is used in this case

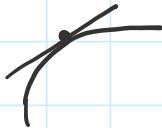
Localization 167/205



## 9. Controllo

martedì 7 giugno 2022 01:27

Il controllo è un compito non lineare e la linearizzazione stessa è un concetto legato all'interno di linearizzazione stessa; infatti, al di fuori di questa si perde la sua affidabilità.



I problemi fondamentali sono

- **posture regulation** ovvero far cambiare posura al robot => PARCHEGGIO
- **trajectory Tracking** ovvero fare in modo che il robot si muova lungo la traiettoria definita => movIMENTO

### trajectory tracking

Affinché un obiettivo sia ammesso, devono esistere  $w$  (velocità angolare) e  $v$  (velocità lineare). Tali da rispettare il modello matematico del robot il quale è

$$\begin{cases} \dot{x}(t) = v(t) \cdot \cos(\theta(t)) \\ \dot{y}(t) = v(t) \cdot \sin(\theta(t)) \\ \dot{\theta}(t) = w(t) \end{cases}$$

#### Obiettivi

ovvero si deve avere che  $\theta(t) = \tan^{-1}\left(\frac{\dot{y}(t)}{\dot{x}(t)}\right)$

Ne consegue che l'errore sarà definito come

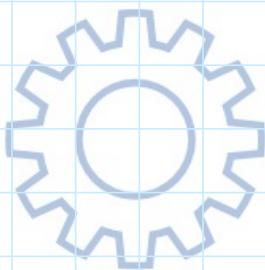
$$e = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^* - x \\ y^* - y \\ \theta^* - \theta \end{bmatrix}$$

Posizione del robot

Sistema solido con il robot fotografato

Sistema guidato con  
il robot teleoperato

Cioè che si cerca di fare è portare  $x \rightarrow x^*$  ovvero  $x - x^* \rightarrow 0$  (equivalente a minimizzare l'errore).



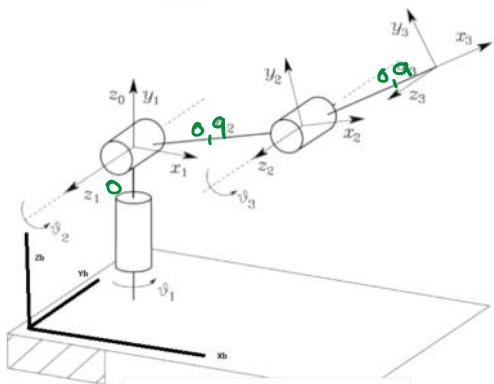
# APPUNTI DI INGEGNERIA INFORMATICA

GAIA BERTOLINO

## Esercizio braccio

mercoledì 15 giugno 2022 11:31

eu



$\Theta = 0$  gradi

$$R_0^b = \begin{bmatrix} 1 & 0 & 0 & 0,5 \\ 0 & 1 & 0 & 0,5 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_1^0 = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \xrightarrow{q_1} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

INFORMATICA

$$R_2^1 = \begin{bmatrix} c_2 & -s_2 & 0 & 0,9 \cdot c_2 \\ s_2 & c_2 & 0 & 0,9 \cdot s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \xrightarrow{q_2} \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0,9 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_3^2 = \begin{bmatrix} c_3 & -s_3 & 0 & 0,9 \cdot c_3 \\ s_3 & c_3 & 0 & 0,9 \cdot s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \xrightarrow{q_3} \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0,9 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_3^0 = R_0^b \cdot R_1^0 \cdot R_2^1 \cdot R_3^2$$

Ogni matrice sarà

$$\begin{bmatrix} c_i & -s_i & 0 & l_i \cdot c_i \\ s_i & c_i & 0 & l_i \cdot s_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\cos \theta_1 (\cos \theta_2 \cdot \cos \theta_3 - \sin \theta_2 \cdot \sin \theta_3)$$

$$0,8 = -0,78 ($$

Punto iniziale

Punto di arrivo  $[0,8 \quad 0,8 \quad 0,5]$

Matrice di posizione

$$\begin{bmatrix} 0,998 & -0,02 & 0,01 & 2,2997 \\ 0,0100 & -0,0002 & -1 & 0,5180 \\ 0,0200 & -0,998 & 0 & 1,0270 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

APPUNTI DI INGEGNERIA  
INFORMATICA  
GAIA BERTOLINO