



Corso di Robotica 1

Robot Mobili su Ruote Analisi, Pianificazione, Controllo

Prof. Alessandro De Luca

DIPARTIMENTO DI INFORMATICA
E SISTEMISTICA ANTONIO RUBERTI



SAPIENZA
UNIVERSITÀ DI ROMA



Sommario

- uso dei modelli cinematici di WMR
 - analisi di controllabilità* (sistemi non lineari)
 - odometria
 - trasformazioni del modello
 - scalatura temporale
 - pianificazione di cammini
 - controllo
 - regolazione della configurazione
 - inseguimento di traiettoria

* richiami di geometria differenziale in [appendice](#)



Analisi di controllabilità

$$A(q)\dot{q} = 0 \Leftrightarrow \dot{q} = G(q)v = \sum_{i=1}^m g_i(q)v_i$$

i vincoli differenziali non sono integrabili?

=

*sono accessibili tutti i punti q dello spazio delle configurazioni \mathcal{C} ?
(ossia, il sistema è controllabile?)*

- la risposta viene dalla teoria del **controllo non lineare**
primo strumento: *parentesi (o bracket) di Lie* di campi di vettori $g_1(q)$ e $g_2(q)$

$$[g_1, g_2](q) = \frac{\partial g_2}{\partial q} g_1(q) - \frac{\partial g_1}{\partial q} g_2(q)$$

genera una nuova direzione di moto, oltre a quelle di $g_1(v_2 = 0)$, $g_2(v_1 = 0)$, e delle loro combinazioni lineari $g_1\bar{v}_1 + g_2\bar{v}_2$



Analisi di controllabilità (*continua*)

secondo strumento: *distribuzione di accessibilità* A generata da campi di vettori (mediante operazioni ripetute di *parentesi di Lie*)

ad esempio, per $m=2$

$$\begin{aligned} A &= \{g_1, g_2, [g_1, g_2], [g_1, [g_1, g_2]], \dots\} \\ &= \text{algebra di Lie generata da } g_1, g_2 \end{aligned}$$

Teorema

$$\begin{aligned} \text{rango } \mathcal{A}(q) = n \quad \forall q \in \mathcal{C} \\ \Updownarrow \\ \mathcal{C} \text{ completamente accessibile} \\ \Updownarrow \\ A(q)\dot{q} = 0 \text{ anolonomo} \end{aligned}$$

← con manovre!

Nota 1: è una versione non lineare del test di Kalman per la controllabilità dei sistemi lineari

$$\dot{x} = Ax + Bu \quad \text{rango } [B \ AB \ A^2B \ \dots \ A^{n-1}B] = n$$

Nota 2: i livelli di parentesi di Lie necessari ad ottenere il rango massimo sono indice della difficoltà di manovrare il WMR (cresce il # di manovre elementari per un generico compito)



Controllabilità dell'uniciclo

$$q = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \in \mathcal{C} \quad g_1 = \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix} \quad g_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\dim \mathcal{C} = n = 3$$

$$[g_1, g_2] = \frac{\partial g_2}{\partial q} g_1 - \frac{\partial g_1}{\partial q} g_2 = \begin{bmatrix} \sin \theta \\ -\cos \theta \\ 0 \end{bmatrix} \quad [g_1, g_2]$$

0

è la direzione laterale!

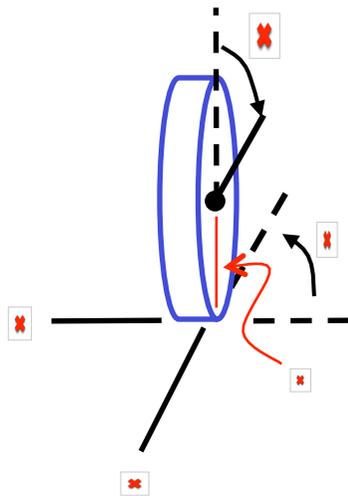
➔ rango $\begin{bmatrix} \cos \theta & 0 & \sin \theta \\ \sin \theta & 0 & -\cos \theta \\ 0 & 1 & 0 \end{bmatrix} = 3 = n!$

N.B. qui è evidente che anche la semplice sequenza "rotazione-traslazione-rotazione" permette di muovere il robot tra due configurazioni qualsiasi...



Controllabilità dell'uniciclo "esteso"

considerando anche l'angolo di rotolamento della ruota ...



$$q = \begin{bmatrix} x \\ y \\ \psi \\ \theta \end{bmatrix} \in \mathcal{C} \quad g_1 = \begin{bmatrix} \cos \theta \\ \sin \theta \\ 1/r \\ 0 \end{bmatrix} \quad g_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\dim \mathcal{C} = n = 4$$

$$[g_1, g_2] = \begin{bmatrix} \sin \theta \\ -\cos \theta \\ 0 \\ 0 \end{bmatrix} \quad [g_2, [g_1, g_2]] = \begin{bmatrix} -\cos \theta \\ -\sin \theta \\ 0 \\ 0 \end{bmatrix}$$

➔ rango $\begin{bmatrix} g_1 & g_2 & [g_1, g_2] & [g_2, [g_1, g_2]] \end{bmatrix} = 4 = n!$

è quindi possibile accedere mediante manovre ad un punto (x,y) del piano, con qualsiasi valori finali dell'orientamento θ e **anche** dell'angolo di rotolamento ψ



Controllabilità per il car-like (RD)

$$q = \begin{bmatrix} x \\ y \\ \theta \\ \phi \end{bmatrix} \in \mathcal{C} \quad g_1 = \begin{bmatrix} \cos \theta \\ \sin \theta \\ \tan \phi / l \\ 0 \end{bmatrix} \quad g_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$\dim \mathcal{C} = n = 4$

direzione di avanzamento

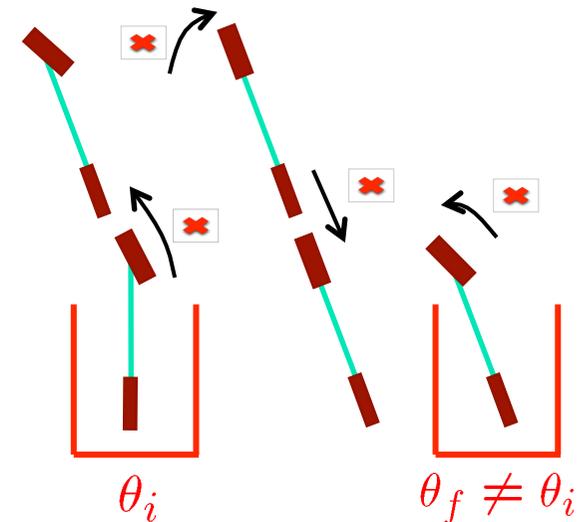
direzione di sterzo

$$[g_1, g_2] = \begin{bmatrix} 0 \\ 0 \\ -1/l \cos^2 \phi \\ 0 \end{bmatrix}$$

"nuova" direzione di rotazione del robot

$$[g_1, [g_1, g_2]] = -\frac{1}{l \cos^2 \phi} \begin{bmatrix} \sin \theta \\ -\cos \theta \\ 0 \\ 0 \end{bmatrix}$$

"nuova" direzione di spostamento laterale del robot



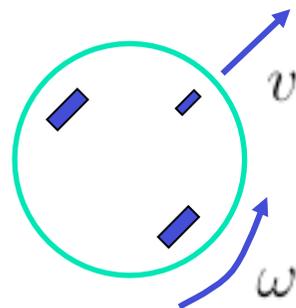
è una sequenza di 8 comandi elementari

$$\text{rango} \begin{bmatrix} g_1 & g_2 & [g_1, g_2] & [g_1, [g_1, g_2]] \end{bmatrix} = 4 = n!$$



Odometria

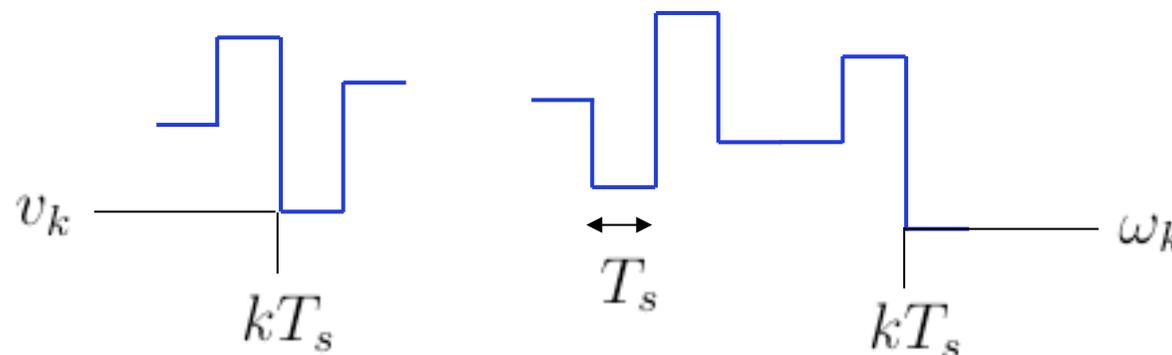
- localizzazione incrementale mediante **odometria**, ossia basata sulle misure degli encoder sulle ruote



$$\begin{aligned}\dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta \\ \dot{\theta} &= \omega\end{aligned}$$

modello unicycle
(SuperMARIO)

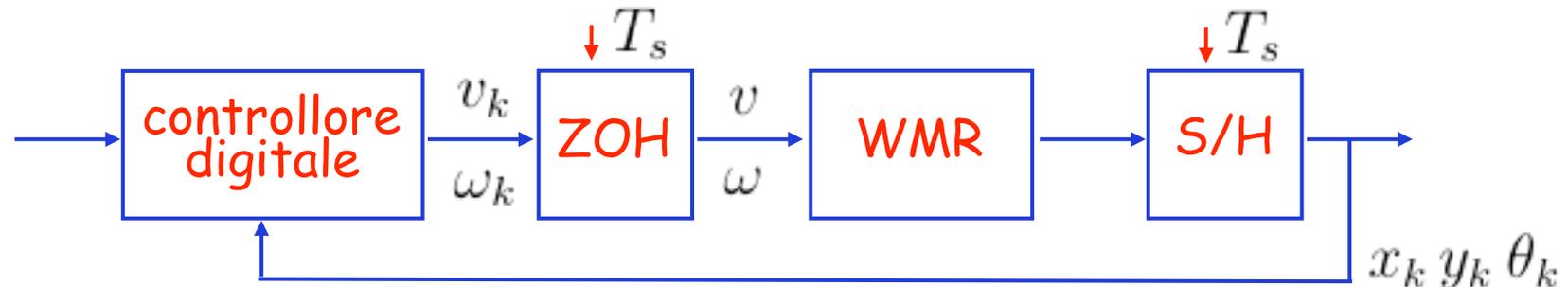
- **ipotesi**: i comandi di velocità (lineare, angolare) sono **costanti** nel periodo di campionamento T_s





Odometria (continua)

- schema di **controllo digitale**



- si possono **integrare numericamente** le equazioni in modo approssimato (esatto per l'orientamento!) con diversi metodi

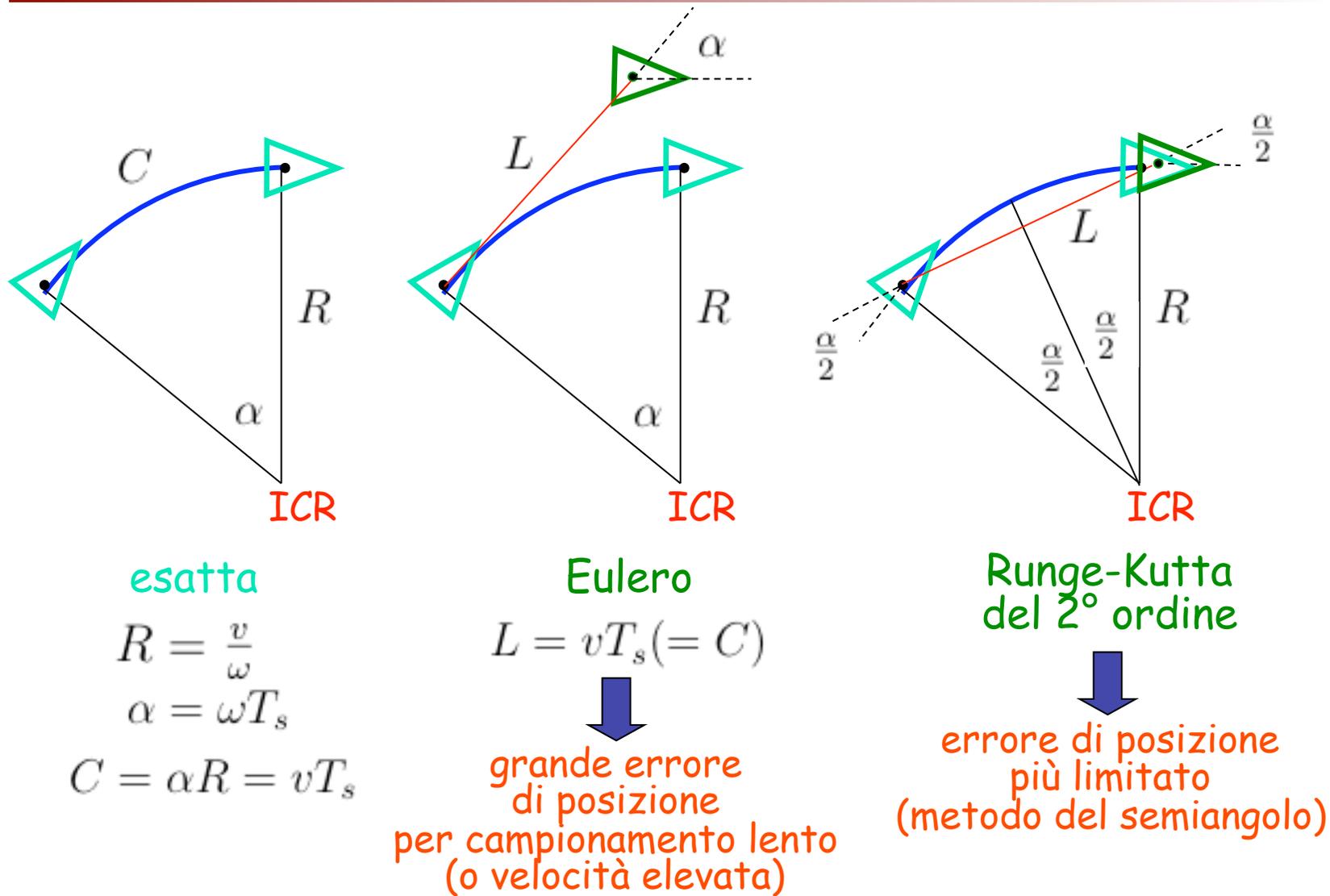
$$\begin{array}{ll} x_{k+1} = x_k + v_k T_s \cos \theta_k & x_{k+1} = x_k + v_k T_s \cos \left(\theta_k + \frac{\omega_k T_s}{2} \right) \\ y_{k+1} = y_k + v_k T_s \sin \theta_k & y_{k+1} = y_k + v_k T_s \sin \left(\theta_k + \frac{\omega_k T_s}{2} \right) \\ \theta_{k+1} = \theta_k + \omega_k T_s & \theta_{k+1} = \theta_k + \omega_k T_s \end{array}$$

Eulero

Runge-Kutta del 2° ordine



Odometria (continua)





Odometria (*continua*)

- alle grandezze di velocità si sostituiscono le **letture degli encoder** di ruota destra $\Delta\psi_R$ e ruota sinistra $\Delta\psi_L$ (incrementi angolari)

$$v_k T_s = r \frac{\Delta\psi_R + \Delta\psi_L}{2}$$
$$\omega_k T_s = r \frac{\Delta\psi_R - \Delta\psi_L}{2d}$$

r = raggio (comune) delle due ruote

- in realtà, in un passo di campionamento a velocità costante, il WMR descrive esattamente un **arco di circonferenza** di raggio $R = \frac{v}{\omega}$
- il calcolo odometrico fornisce una stima approssimata della localizzazione del WMR, che risulta affidabile per passi di **campionamento brevi** e in assenza di **slittamento** delle ruote



Trasformazioni del modello

- si possono effettuare diversi tipi di trasformazioni sul modello cinematico dei WMR, utili per la pianificazione del moto e il progetto di controllori
- una trasformazione comune è quella in **forma a catena** (*chained form*)
- per l'uniciclo, definendo la **trasformazione di coordinate**

$$\begin{array}{l} z_1 = \theta \\ z_2 = x \cos \theta + y \sin \theta \\ z_3 = x \sin \theta - y \cos \theta \end{array} \quad \longleftrightarrow \quad \begin{array}{l} x = z_2 \cos z_1 + z_3 \sin z_1 \\ y = z_2 \sin z_1 - z_3 \cos z_1 \\ \theta = z_1 \end{array}$$

e la **trasformazione di ingressi** (entrambi globalmente invertibili)

$$\begin{array}{l} v_1 = \omega \\ v_2 = v - z_3 \omega = v - (x \sin \theta - y \cos \theta) \omega \end{array} \quad \longleftrightarrow \quad \begin{array}{l} v = v_2 + z_3 v_1 \\ \omega = v_1 \end{array}$$

si ottiene

$$\boxed{\begin{array}{l} \dot{z}_1 = v_1 \\ \dot{z}_2 = v_2 \\ \dot{z}_3 = z_2 v_1 \end{array}} \equiv \text{forma a catena per } n = 3$$

un sistema **polinomiale**, che diviene un sistema **lineare** se il comando v_1 è costante nel tempo



Trasformazioni del modello (*continua*)

- in generale, per

$$\dot{q} = g_1(q)u_1 + g_2(q)u_2 \quad q \in \mathcal{C}, \dim \mathcal{C} = n \geq 2$$

può esistere (almeno) una trasformazione di coordinate e di ingressi

$$z = T(q) \quad v = \beta(q)u$$

tale che il modello cinematico del WMR sia nella **forma a catena**

$$\begin{aligned}\dot{z}_1 &= v_1 \\ \dot{z}_2 &= v_2 \\ \dot{z}_3 &= z_2 v_1 \\ &\vdots \\ \dot{z}_n &= z_{n-1} v_1\end{aligned}$$

- la *chained form* esiste **sempre** (almeno localmente) **per $n \leq 4$** (ad es., uniclo, car-like, veicolo con un rimorchio) e **anche** per sistemi con N rimorchi con aggancio centrato (*zero-hooking*)



Scalatura temporale

- separazione in spazio s e tempo t

$$\dot{q} = \frac{dq}{dt} = \frac{dq}{ds} \frac{ds}{dt} = q' \dot{s}$$

- separazione del modello cinematico dell'uniciclo

$$\begin{aligned}x' &= dx/ds = \cos \theta \tilde{v} \\y' &= dy/ds = \sin \theta \tilde{v} \\ \theta' &= d\theta/ds = \tilde{\omega}\end{aligned}$$

← ingressi "geometrici"

con

$$\begin{aligned}v(t) &= \tilde{v}(s) \dot{s}(t) \\ \omega(t) &= \tilde{\omega}(s) \dot{s}(t)\end{aligned}$$

($s(t)$ = legge oraria in comune)

- dati $\tilde{v}(s), \tilde{\omega}(s)$ per $s \in [0, L]$, il cammino geometrico è univocamente determinato, ma eseguibile con diverse leggi orarie
- vale per tutti i modelli cinematici (del primo ordine) di WMR



Uscite piatte

Flat outputs

- un sistema dinamico nonlineare

$$\dot{x} = f(x) + G(x)u$$

ha la proprietà di “piattezza differenziale”, **se esiste** un insieme di uscite (dette **piatte**) tali che lo stato e l’ingresso siano esprimibili **algebricamente** in funzione di tali uscite e di un certo numero di loro derivate

$$x = x(y, \dot{y}, \ddot{y}, \dots, y^{(r)})$$

$$u = u(y, \dot{y}, \ddot{y}, \dots, y^{(r)})$$

- la traiettoria (nel tempo) o il cammino (nello spazio) dello stato e degli ingressi sono funzioni della sola uscita e delle sue derivate (geometriche o temporali)
- proprietà utile per **pianificare** una riconfigurazione tra due stati iniziale e finale e per trovare i comandi necessari ad essa associati



Uscite piatte

Esempio elementare: l'uniciclo

- l'uniciclo è piatto differenzialmente rispetto alle coordinate del suo centro (uscite flat)
- ad esempio, in termini **geometrici**, date $x(s)$ e $y(s)$, per $s \in [0, 1]$

$$\theta(s) = \text{Atan2}(y'(s), x'(s)) + k\pi \quad k = 0, 1$$

cammino eseguito a marcia **avanti** o **indietro**

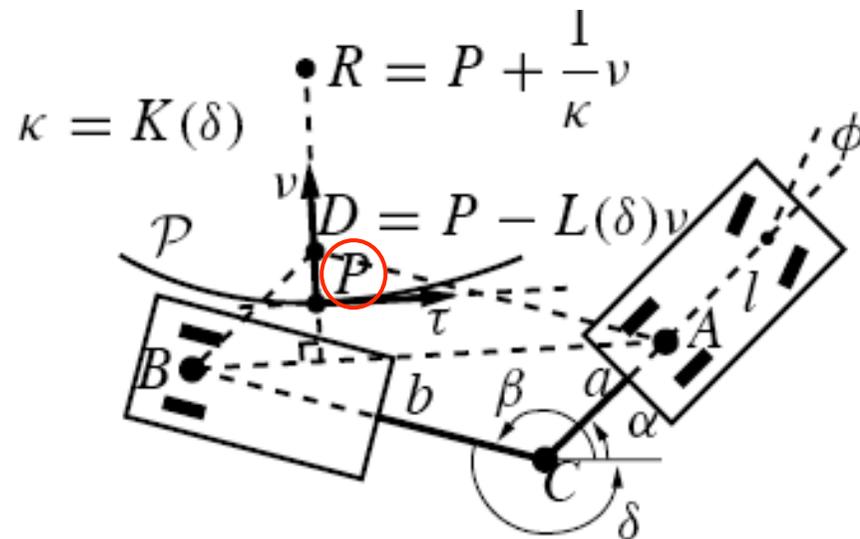
$$\tilde{v}(s) = \pm \sqrt{(x'(s))^2 + (y'(s))^2}$$
$$\tilde{\omega}(s) = \frac{y''(s)x'(s) - x''(s)y'(s)}{(x'(s))^2 + (y'(s))^2}$$



Uscite piatte

Esempi più complessi

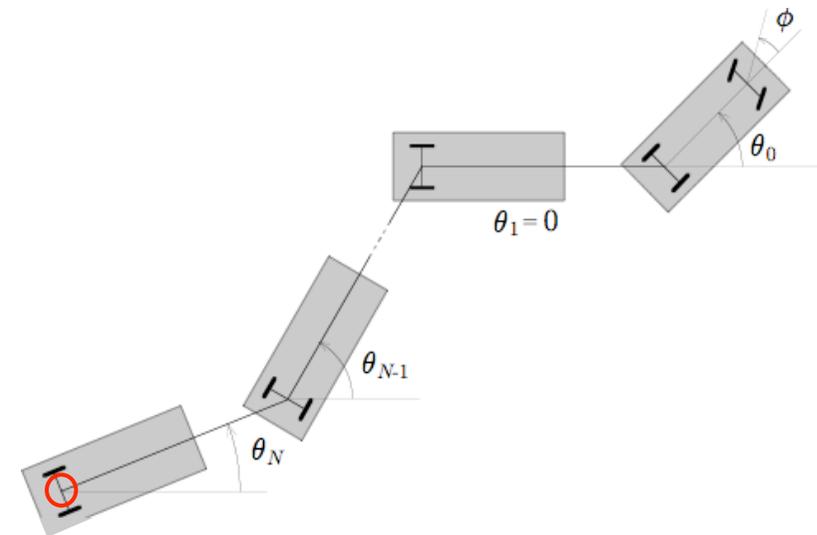
- car + 1 rimorchio con aggancio generale



l'uscita flat è il **punto P** (variabile con la geometria, in funzione di δ)

- tutti i sistemi driftless a due ingressi (trasformabili) **in forma chained** (uscite flat: z_1 e z_3)

- car + N rimorchi con aggancio centrato (**zero-hooking**)

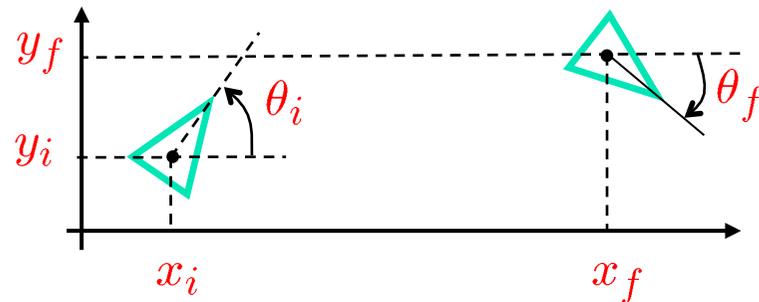


l'uscita flat è il **centro asse dell'ultimo rimorchio**



Pianificazione di cammini

trovare un **cammino** punto-punto (tra configurazioni) per l'uniciclo



compito
di moto

è un problema di interpolazione; si utilizzano polinomi **cubici** nelle **uscite flat**

$$\begin{aligned}x(s) &= -(s-1)^3 x^i + s^3 x^f + \alpha_x s^2 (s-1) + \beta_x s (s-1)^2 \\y(s) &= -(s-1)^3 y^i + s^3 y^f + \alpha_y s^2 (s-1) + \beta_y s (s-1)^2 \\s &\in [0, 1]\end{aligned}$$

che soddisfano automaticamente le condizioni sulle posizioni

$$\begin{aligned}x(0) &= x^i & x(1) &= x^f \\y(0) &= y^i & y(1) &= y^f\end{aligned}$$

e hanno parametri sufficienti ad imporre orientamento iniziale e finale



Pianificazione di cammini (*continua*)

il problema si risolve imponendo le altre **quattro** condizioni

$$\begin{aligned}x'(0) &= K^i \cos \theta^i & x'(1) &= K^f \cos \theta^f \\y'(0) &= K^i \sin \theta^i & y'(1) &= K^f \sin \theta^f\end{aligned}$$

con i due parametri liberi $K^i \neq 0$, $K^f \neq 0$ (concordi e **positivi!**)

posto ad esempio $K^i = K^f = K$ ← è importante "ottimizzare" questo valore...

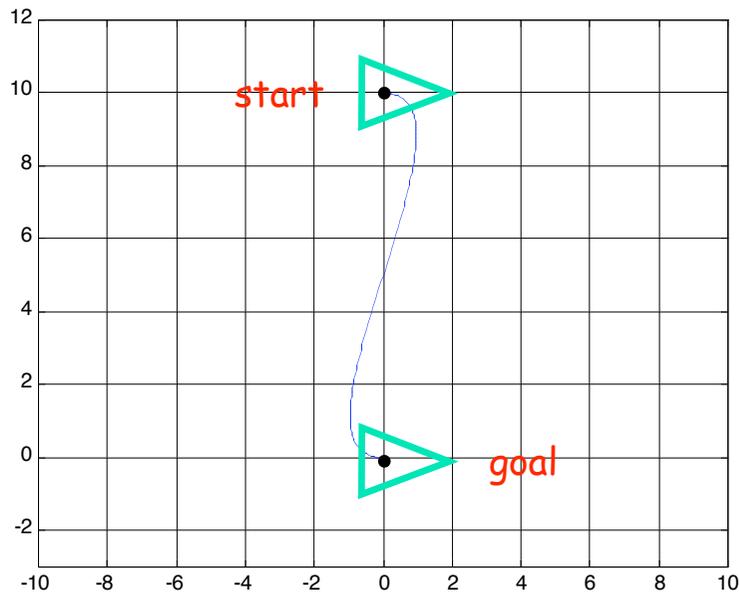
$$\Rightarrow \begin{bmatrix} \alpha_x \\ \alpha_y \end{bmatrix} = \begin{bmatrix} K \cos \theta^f - 3x^f \\ K \sin \theta^f - 3y^f \end{bmatrix}, \quad \begin{bmatrix} \beta_x \\ \beta_y \end{bmatrix} = \begin{bmatrix} K \cos \theta^i + 3x^i \\ K \sin \theta^i + 3y^i \end{bmatrix}$$



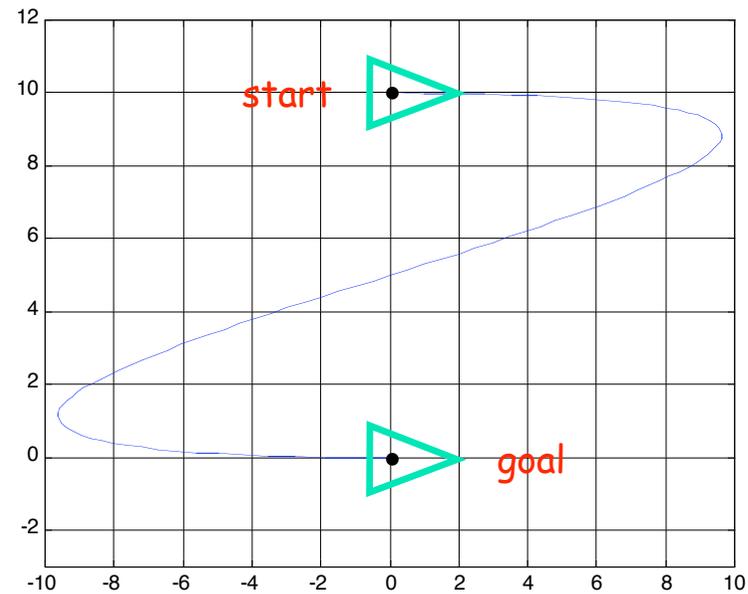
Risultati numerici

$$q = (x, y, \theta) \quad q^i = (0, 10, 0) \rightarrow q^f = (0, 0, 0)$$

$K = 10$



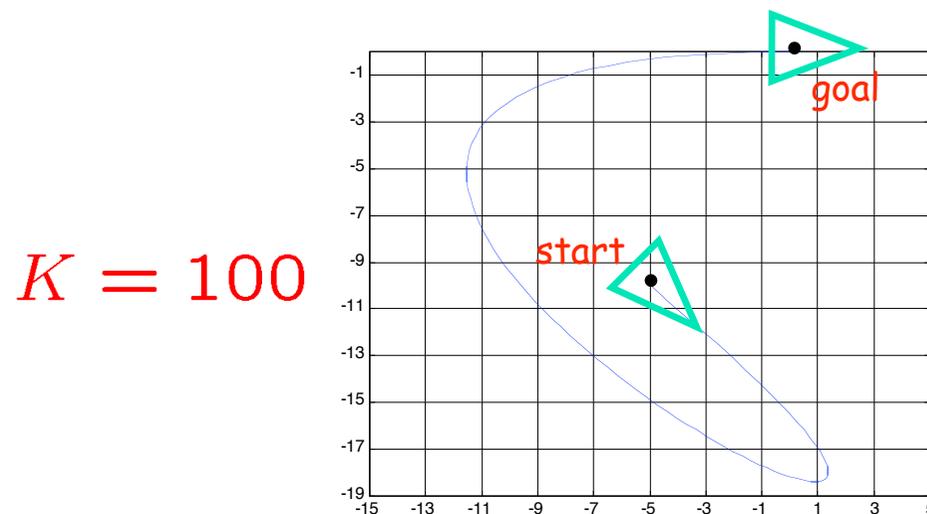
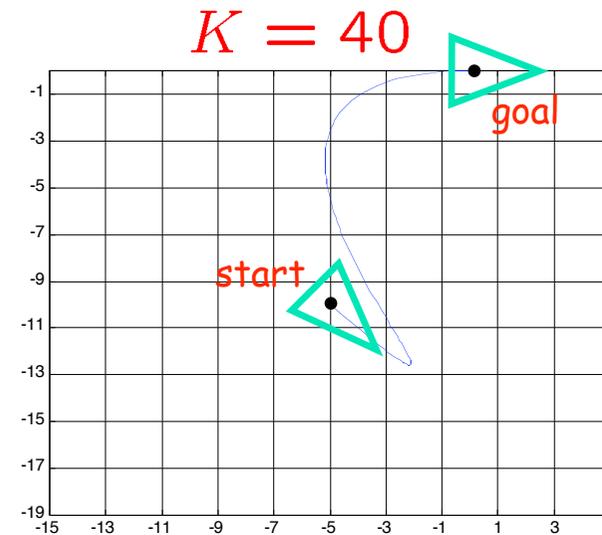
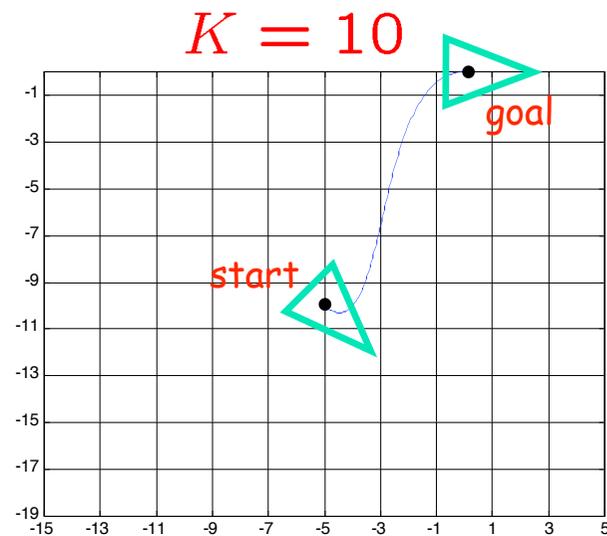
$K = 100$



parcheggio "parallelo"



Risultati numerici (continua)



$$q^i = (-5, -10, -\pi/4)$$

↓

$$q^f = (0, 0, 0)$$



Pianificazione di cammini

uso della forma chained

in modo alternativo, si può lavorare sulla **forma a catena**

$$\begin{aligned} (x, y, \theta)^i &\longrightarrow (z_1, z_2, z_3)^i \\ (x, y, \theta)^f &\longrightarrow (z_1, z_2, z_3)^f \end{aligned} \quad \text{tenendo presente che: } z_2 = z'_3/z'_1$$

e, come prima, procedere con polinomi cubici (nelle relative **uscite flat**)

è però possibile ridurre al minimo il numero di parametri da ricavare usando una combinazione **lineare/cubica**

$$\begin{aligned} z_1(s) &= z_{1,f}s - (s-1)z_{1,i} \\ z_3(s) &= s^3 z_{3,f} - (s-1)^3 z_{3,i} + \alpha_3 s^2 (s-1) + \beta_3 s (s-1)^2 \end{aligned} \quad s \in [0, 1]$$

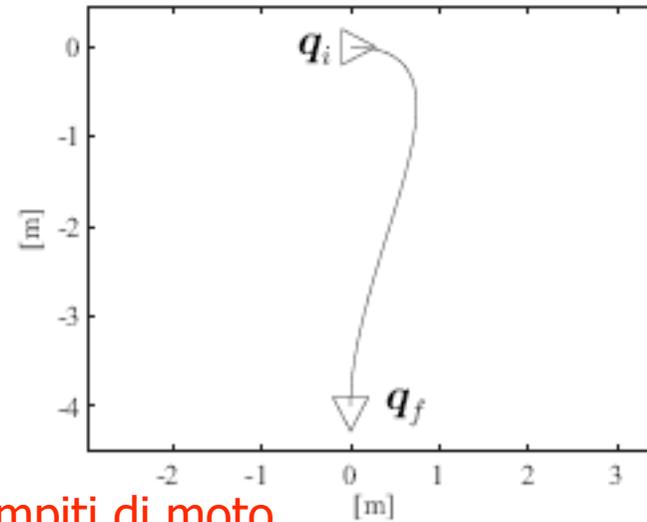
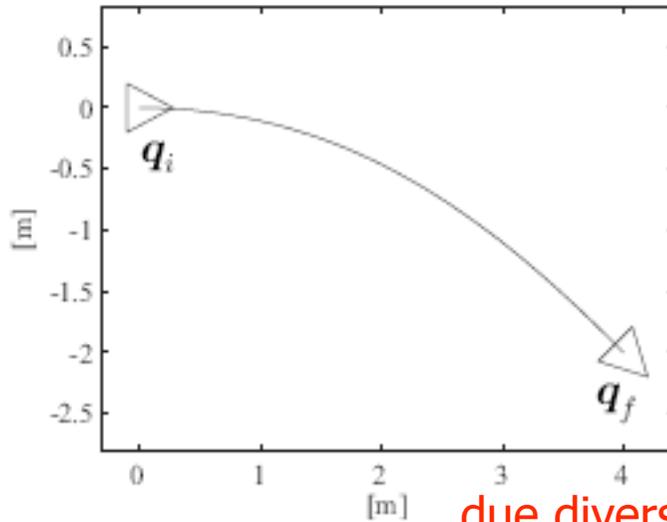
e imponendo le condizioni al contorno

$$\frac{z'_3(0)}{z'_1(0)} = z_{2i} \quad \frac{z'_3(1)}{z'_1(1)} = z_{2f} \quad \longrightarrow \quad \begin{aligned} \alpha_3 &= z_{2,f}(z_{1,f} - z_{1,i}) - 3z_{3,f} \\ \beta_3 &= z_{2,i}(z_{1,f} - z_{1,i}) + 3z_{3,i} \end{aligned}$$

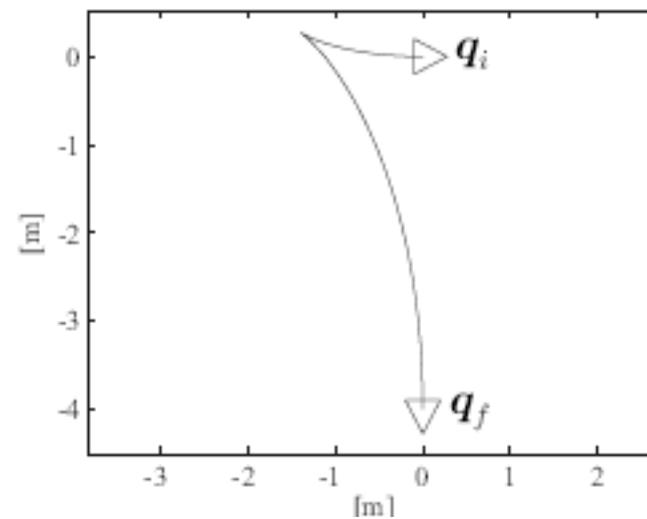
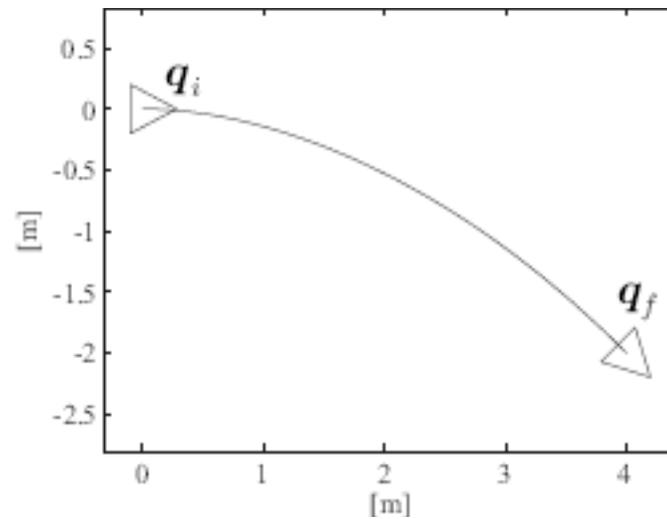
valide **solo** se $z'_1(s) = z_{1,f} - z_{1,i} = \theta_f - \theta_i \neq 0$ (altrimenti serve un "via point" con orientamento diverso)



Risultati numerici di confronto



due diversi compiti di moto



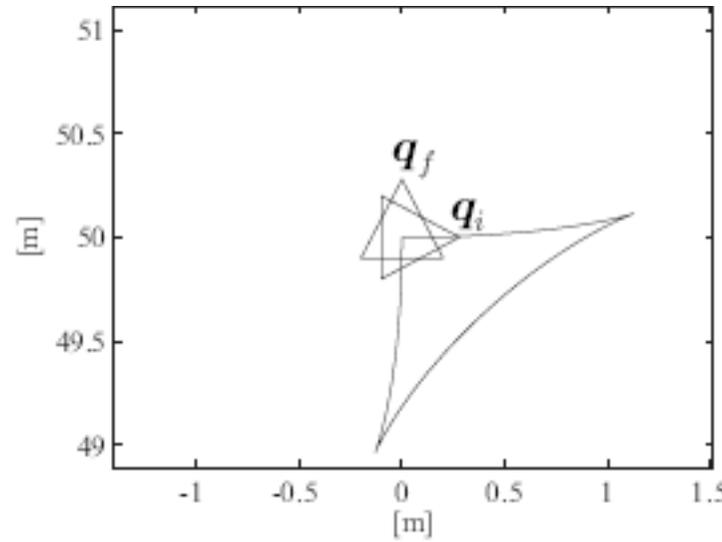
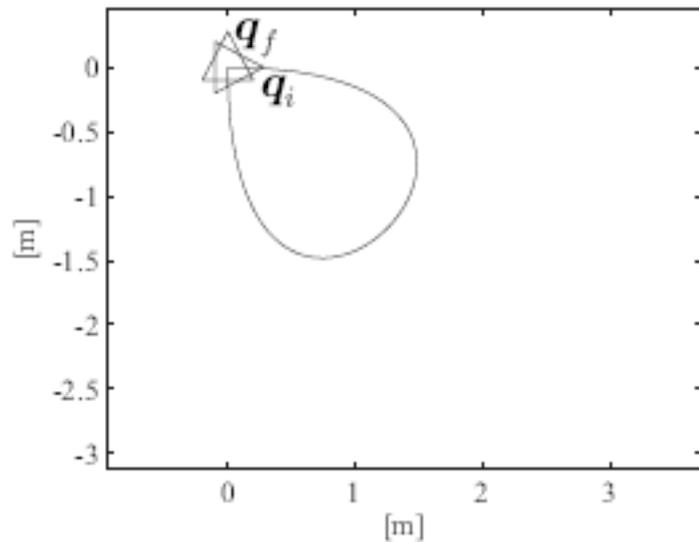
con polinomi
cartesiani
cubici
($K = 5$)



utilizzando
la forma a
catena

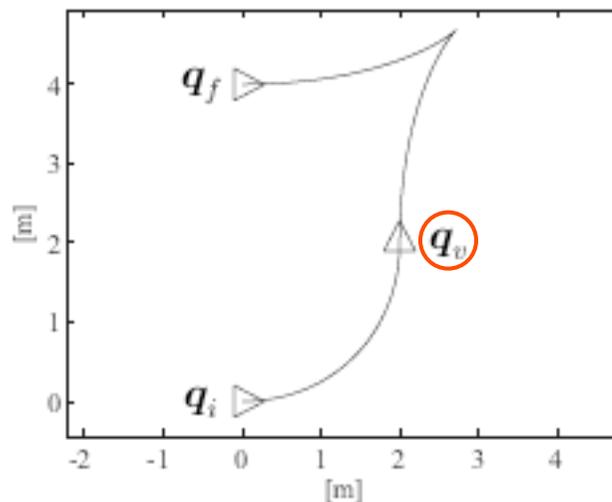


Risultati numerici di confronto (*continua*)



puro
riorientamento
di 90°

polinomi cartesiani cubici ($K = 10$) ↔ con la forma a catena



parallel parking
uso di un "via point"
nella pianificazione
con la forma a catena



Controllo del moto

- schemi di controllo per robot mobili tipo unicycle
 - regolazione della configurazione (*è il problema più difficile qui!*)
 - senza perdita di generalità $(x_d, y_d, \theta_d) = (0, 0, 0)$, l'origine
 1. basata su una trasformazione in coordinate polari
 2. basata sulla linearizzazione esatta del modello cinematico con feedback dinamico (DFL)
 - inseguimento di traiettoria (*più interessante in pratica...*)
 - ancora con DFL (modifica del metodo 2. di regolazione)
 - basato sulla linearizzazione/disaccoppiamento esatto di un legame ingresso-uscita con feedback statico (I-O SFL)
 - tutti schemi di controllo con feedback non lineare dallo stato

$$\begin{array}{l} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = \omega \end{array}$$

modello cinematico
del WMR



Regolazione in coordinate polari

- cambiamento di coordinate

$$\rho = \sqrt{x^2 + y^2}$$

$$\gamma = \text{ATAN2}(y, x) - \theta + \pi$$

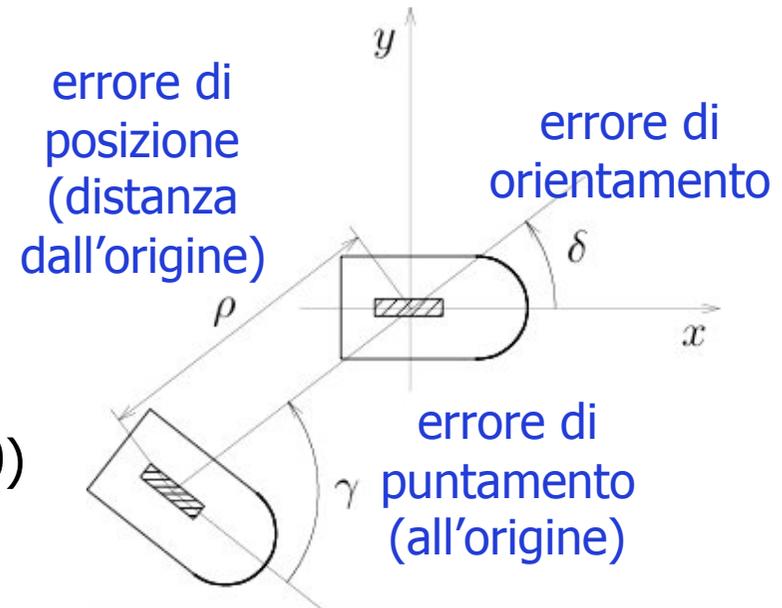
$$\delta = \gamma + \theta$$

- legge di controllo (con $k_1, k_2, k_3 > 0$)

$$v = k_1 \rho \cos \gamma$$

$$\omega = k_2 \gamma + k_1 \frac{\sin \gamma \cos \gamma}{\gamma} (\gamma + k_3 \delta)$$

- si dimostra la convergenza **asintotica** dell'errore a zero mediante un'analisi con funzione di Lyapunov





Regolazione con DFL

1. si introduce uno **stato** ξ nel **controllore** e si comanda il robot con la legge di controllo

$$\begin{aligned}\dot{\xi} &= u_1 \cos \theta + u_2 \sin \theta \\ v &= \xi \\ \omega &= \frac{u_2 \cos \theta - u_1 \sin \theta}{\xi}\end{aligned}$$

2. cambiamento di coordinate

$$\begin{aligned}z_1 &= x \\ z_2 &= y \\ z_3 &= \dot{x} = \xi \cos \theta \\ z_4 &= \dot{y} = \xi \sin \theta\end{aligned}$$

3. il sistema risultante è **lineare**: **doppi integratori** disaccoppiati

$$\begin{aligned}\ddot{z}_1 &= u_1 \\ \ddot{z}_2 &= u_2\end{aligned}$$

4. **regolazione** mediante PD

$$\begin{aligned}u_1 &= -k_{p1}x - k_{d1}\dot{x} \\ u_2 &= -k_{p2}y - k_{d2}\dot{y}\end{aligned}$$

$$k_{pi} > 0, k_{di} > 0 \quad (i = 1, 2)$$

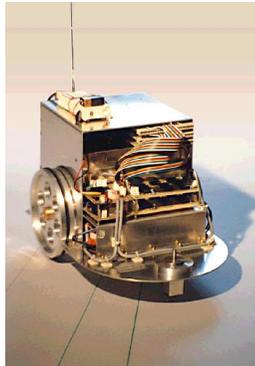
- si ha convergenza **esponenziale** dell'errore a zero **se** si sceglie

$$k_{d1}^2 - 4k_{p1} = k_{d2}^2 - 4k_{p2} > 0$$

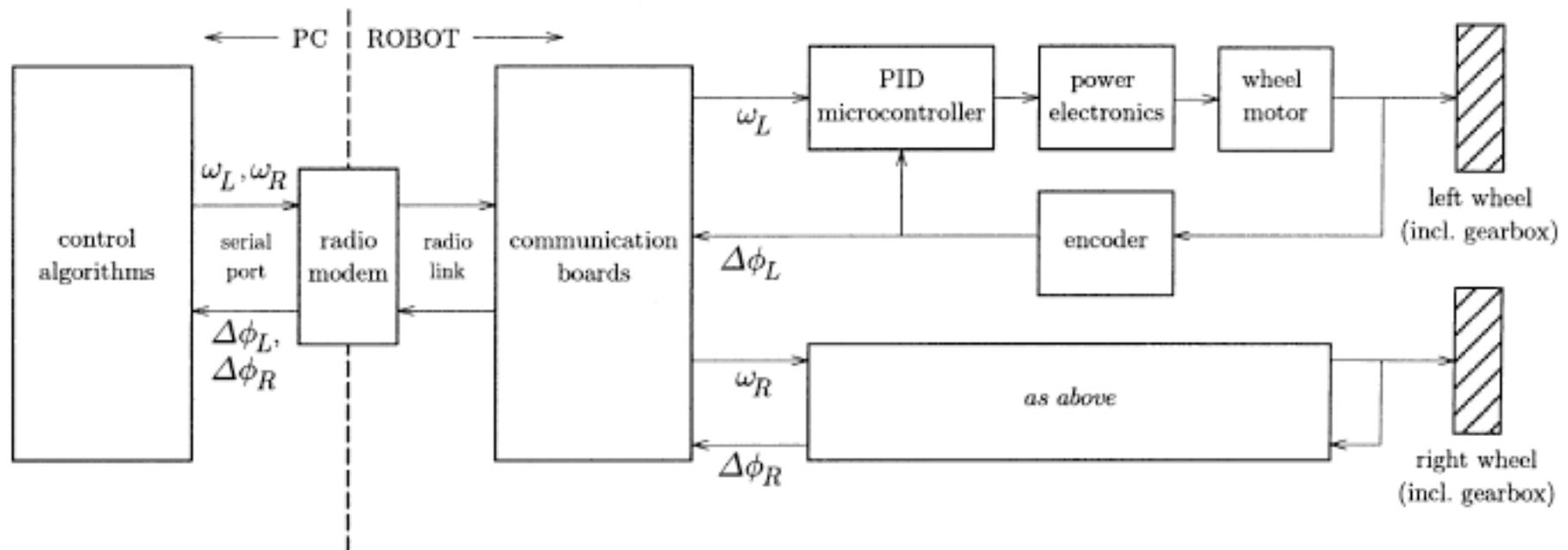
$$k_{d2} - k_{d1} > 2\sqrt{k_{d2}^2 - 4k_{p2}}$$

DFL = **D**ynamic **F**eedback **L**inearization

Architettura di controllo



- SuperMARIO
 - con odometria (encoder sulle due ruote)
 - con localizzazione esterna (da telecamera fissa)



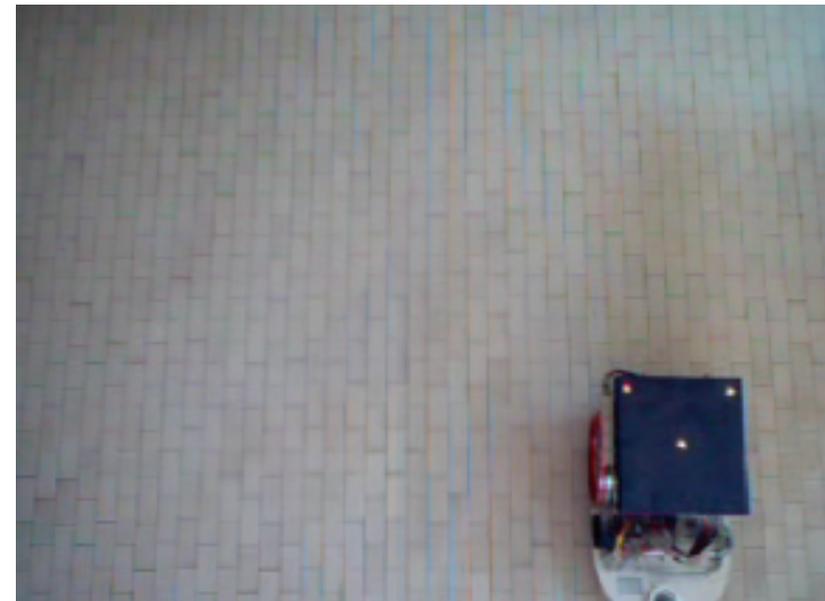


“Parcheggio” di SuperMARIO

- manovre effettuate solo mediante **controllo in feedback** dell'errore rispetto alla configurazione desiderata finale (problema di **regolazione, senza pianificazione!**), elaborando l'immagine visiva (tre led visti da telecamera fissa al soffitto) per localizzare il robot



in coordinate polari



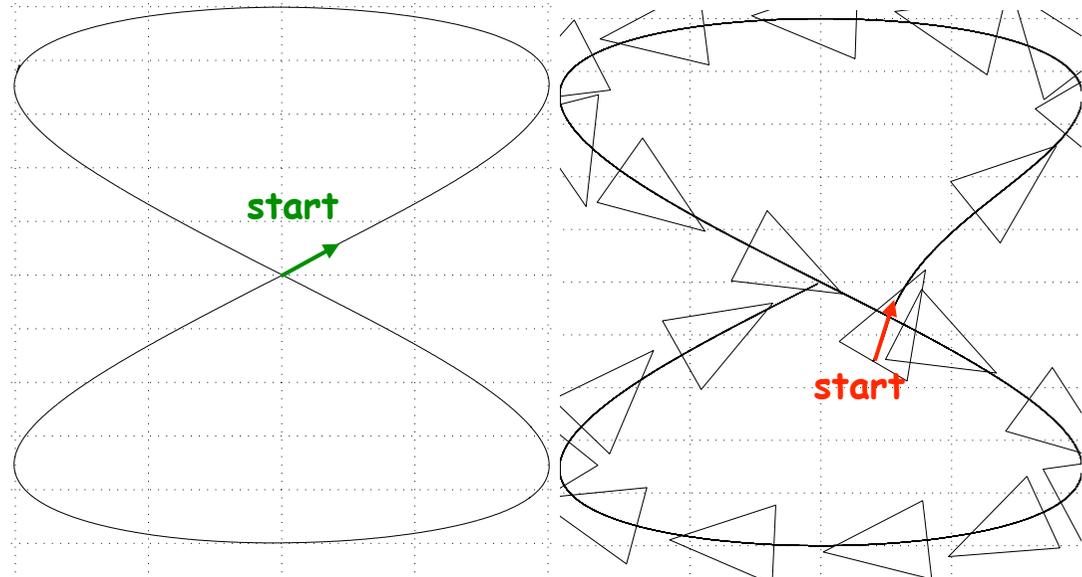
con linearizzazione dinamica

Inseguimento di traiettoria con DFL



- basandosi sulla linearizzazione dinamica, basta stabilizzare al passo 4. l'errore di traiettoria con un PD + feedforward di accelerazione

$$\begin{aligned} u_1 &= \ddot{x}_d + k_{p1}(x_d - x) + k_{d1}(\dot{x}_d - \dot{x}) \\ u_2 &= \ddot{y}_d + k_{p2}(y_d - y) + k_{d2}(\dot{y}_d - \dot{y}) \end{aligned} \quad k_{pi} > 0, k_{di} > 0 \quad (i = 1, 2)$$



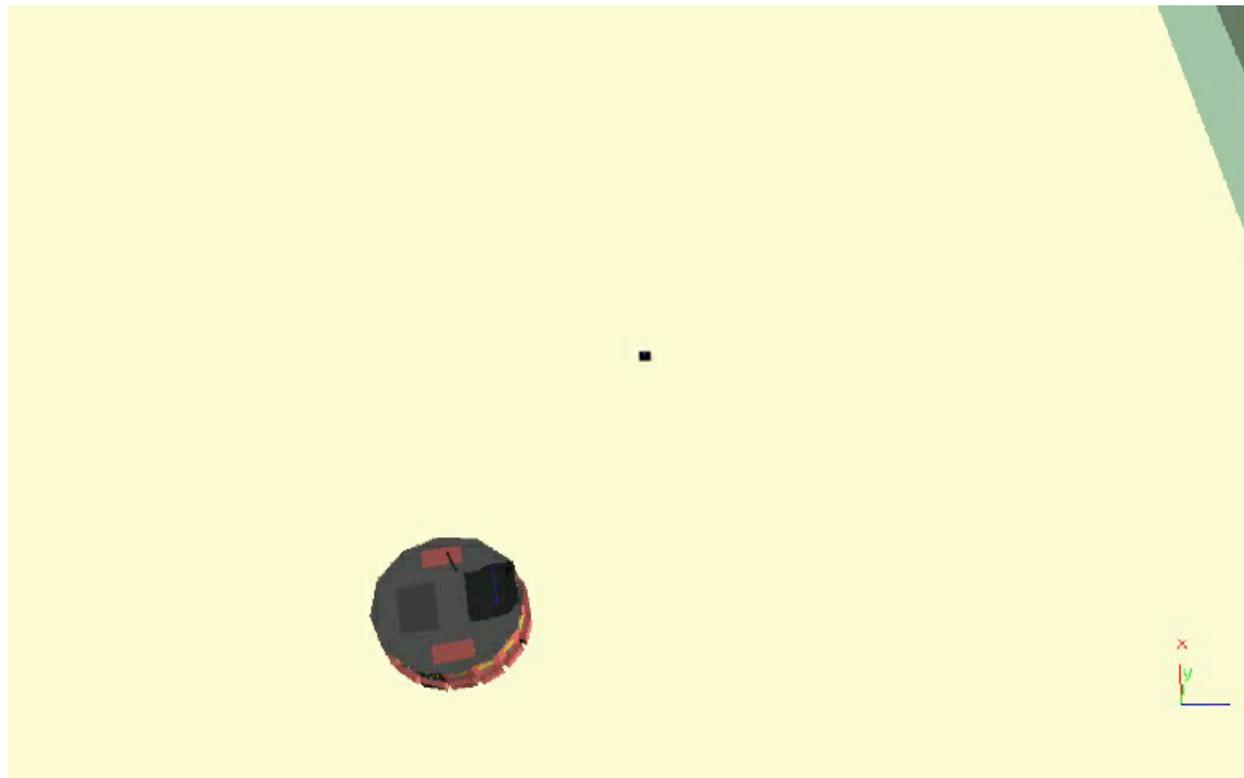
esecuzione ad errore iniziale **nullo** di un cammino "ad otto"

con errore iniziale

Inseguimento di traiettoria con DFL



- simulazione del Magellan Pro con **Webot** (campionamento a **32 msec**)
- a partire da un **errore iniziale** di traiettoria





Commenti sul metodo DFL

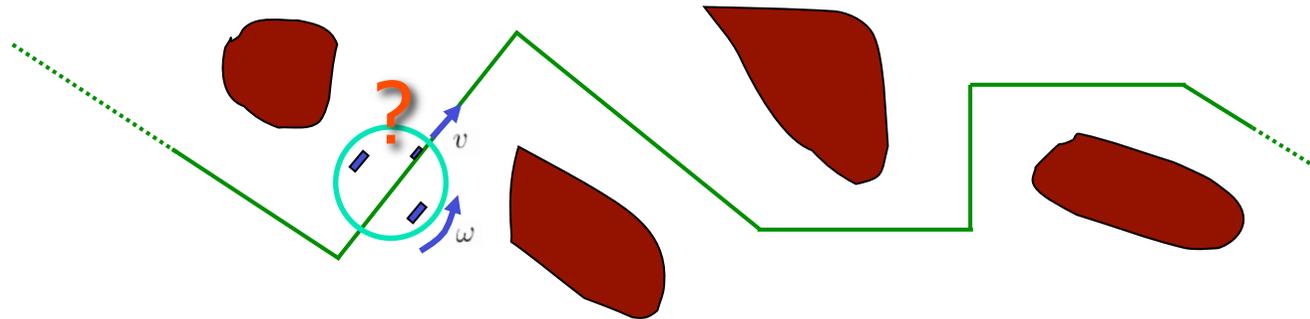
- lo schema di controllo precedente è **piuttosto complesso**
 - inizializzazione critica dello stato del controllore dinamico
 - problemi all'avvio e all'arresto (quando la velocità lineare svanisce)
- richiede inoltre **traiettorie sufficientemente "smooth"**
 - il pianificatore deve fornire cammini a curvatura almeno continua
 - in corrispondenza ad eventuali discontinuità sulla curvatura o addirittura sulla tangente, occorre **fermare** il robot (modificando la legge oraria)
- esistono altre leggi di controllo per l'**inseguimento di traiettorie** basate su **feedforward** nominale + regolazione via **feedback** (lineare o nonlineare) dall'errore di traiettoria
 - richiedono altre restrizioni operative: **ammissibilità** della traiettoria nominale (quindi cammini a tangente continua) e/o **persistenza** della traiettoria con **errori piccoli** (validità locale del feedback lineare)

Motivazione

per il controllo I-O SFL



è possibile per l'uniciclo percorrere in modo esatto a velocità **costante** un cammino avente **tangente discontinua**?



- la risposta dipende dal punto preso come riferimento del robot (**uscita del sistema**), ossia al quale vogliamo assegnare il moto desiderato
 - ad esempio, il punto (x,y) del centro asse non può mai avere una velocità laterale rispetto all'orientamento del veicolo (quindi la risposta è **negativa**)
 - **idem** per qualsiasi altro punto posto lungo l'asse delle ruote...

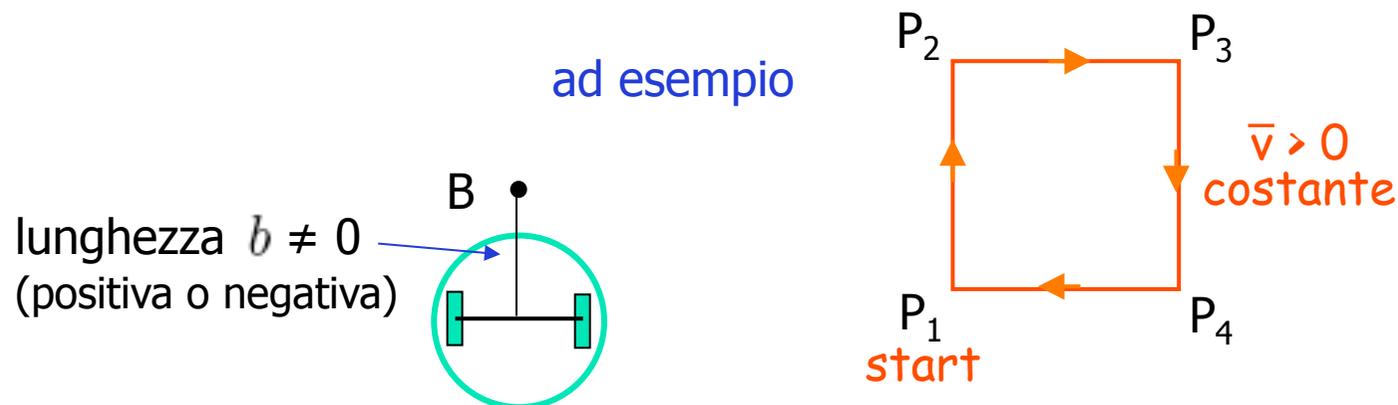


Controllo mediante I-O SFL

- se però si prende come uscita del sistema un punto B fuori dall'asse delle ruote, di coordinate

$$x_B = x + b \cos \theta \quad y_B = y + b \sin \theta$$

è possibile controllare il moto del robot in modo da eseguire con tale punto anche cammini con tangente discontinua avendo velocità lineare sempre diversa da zero (ad es., costante)



I-O SFL = Input-Output Static Feedback Linearization



Linearizzazione ingresso-uscita

1. con la trasformazione di coordinate $(x, y, \theta) \rightarrow (x_B, y_B, \theta)$ si ha

$$\dot{x}_B = v \cos \theta - \omega b \sin \theta$$

$$\dot{y}_B = v \sin \theta + \omega b \cos \theta$$

$$\dot{\theta} = \omega$$

2. nelle prime due equazioni, la dipendenza dagli ingressi è invertibile

$$\det \begin{bmatrix} \cos \theta & -b \sin \theta \\ \sin \theta & b \cos \theta \end{bmatrix} = b \neq 0$$

3. definita la legge di controllo **statica**, in funzione di due nuovi ingressi

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos \theta & -b \sin \theta \\ \sin \theta & b \cos \theta \end{bmatrix}^{-1} \begin{bmatrix} v_{dx} \\ v_{dy} \end{bmatrix} = \begin{bmatrix} v_{dx} \cos \theta + v_{dy} \sin \theta \\ \frac{1}{b} (v_{dy} \cos \theta - v_{dx} \sin \theta) \end{bmatrix}$$

si ha il sistema **lineare** e **disaccoppiato** (sui canali **ingresso-uscita**)

$$\leftarrow \dot{x}_B = v_{dx} \quad \leftarrow \text{direzione } x$$

$$\leftarrow \dot{y}_B = v_{dy} \quad \leftarrow \text{direzione } y$$

$$\dot{\theta} = \frac{1}{b} (v_{dy} \cos \theta - v_{dx} \sin \theta)$$



Linearizzazione ingresso-uscita (*continua*)

4. per condizioni iniziali "agganciate" alla traiettoria, la seguente scelta degli ingressi ausiliari esegue il "quadrato a velocità costante"

nell'esempio

$$\begin{aligned} P_1 \rightarrow P_2 & : & v_{dx} &= 0 & v_{dy} &= \bar{v} \\ P_2 \rightarrow P_3 & : & v_{dx} &= \bar{v} & v_{dy} &= 0 \\ P_3 \rightarrow P_4 & : & v_{dx} &= 0 & v_{dy} &= -\bar{v} \\ P_4 \rightarrow P_1 & : & v_{dx} &= -\bar{v} & v_{dy} &= 0 \end{aligned}$$

5. nel caso di errore iniziale (o ad un qualsiasi istante), si aggiunge un termine proporzionale all'errore (in posizione) di traiettoria

$$e = \begin{bmatrix} e_x \\ e_y \end{bmatrix} = \begin{bmatrix} x_{Bd}(t) - x_B \\ y_{Bd}(t) - y_B \end{bmatrix} \quad \begin{bmatrix} v_{dx} \\ v_{dy} \end{bmatrix} \xrightarrow{k_x, k_y > 0} \begin{bmatrix} v_{dx} + k_x e_x \\ v_{dy} + k_y e_y \end{bmatrix}$$

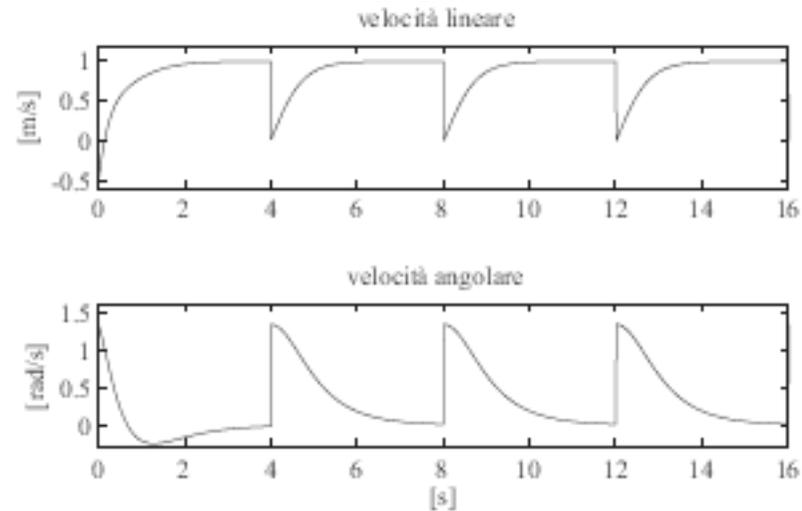
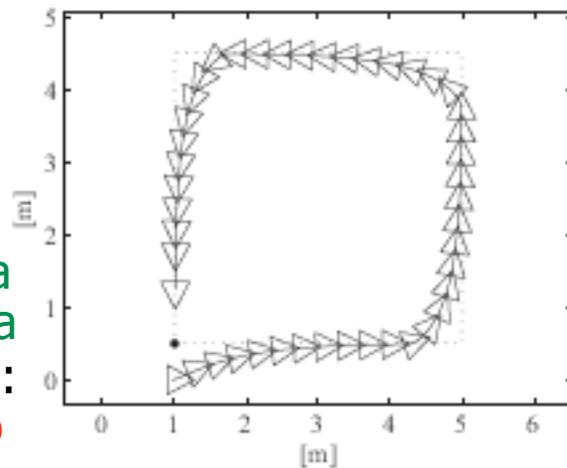
l'errore converge a zero **esponenzialmente**, in modo **indipendente** per ogni componente cartesiana

$$\dot{e}_x = -k_x e_x \quad \dot{e}_y = -k_y e_y$$

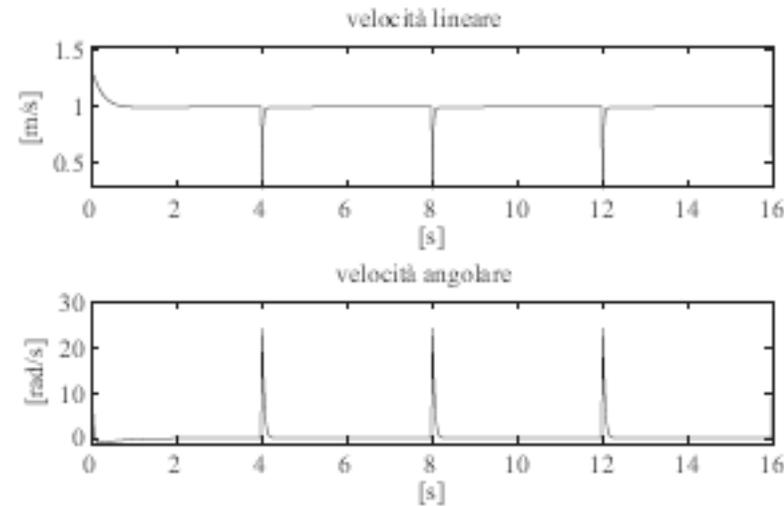
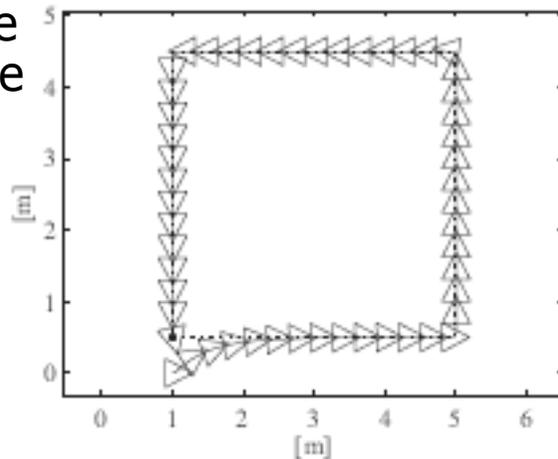


Risultati di simulazione

traiettoria desiderata (punto B): quadrato percorso a $v = 1 \text{ m/s}$ (con errore di posizione iniziale)



$b = 0.75$



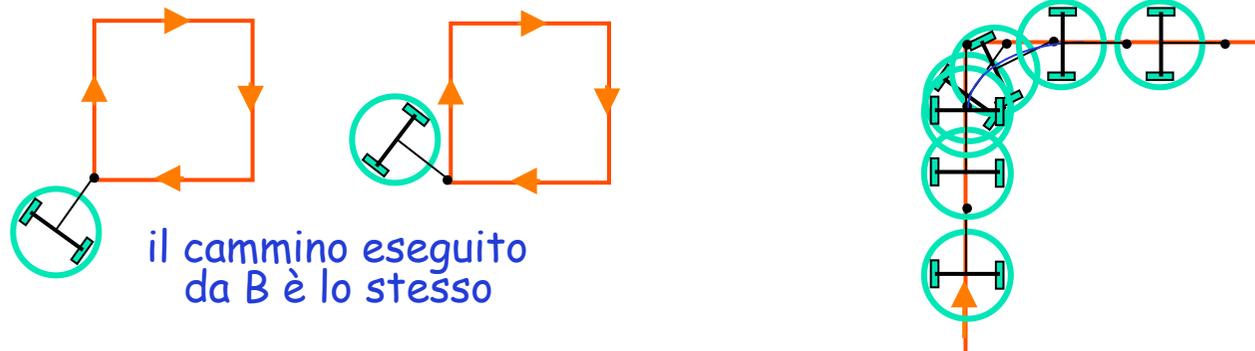
$b = 0.2$

... picchi eccessivi sulla ω



Commenti sul metodo I-O SFL

- l'esecuzione esatta della traiettoria per l'**off-set point B** non dipende dall'orientamento iniziale del robot
 - il punto (x,y) "arrontonda" i punti angolosi del cammini



- questo schema risolve **in generale** il problema del controllo di traiettoria del robot unicycle
 - occorre però un **canale libero** intorno al robot per tenere conto dell'area "spazzata" dal veicolo nei cambi di direzione
 - se si sceglie $b < 0$, il moto preferito è "a marcia indietro"



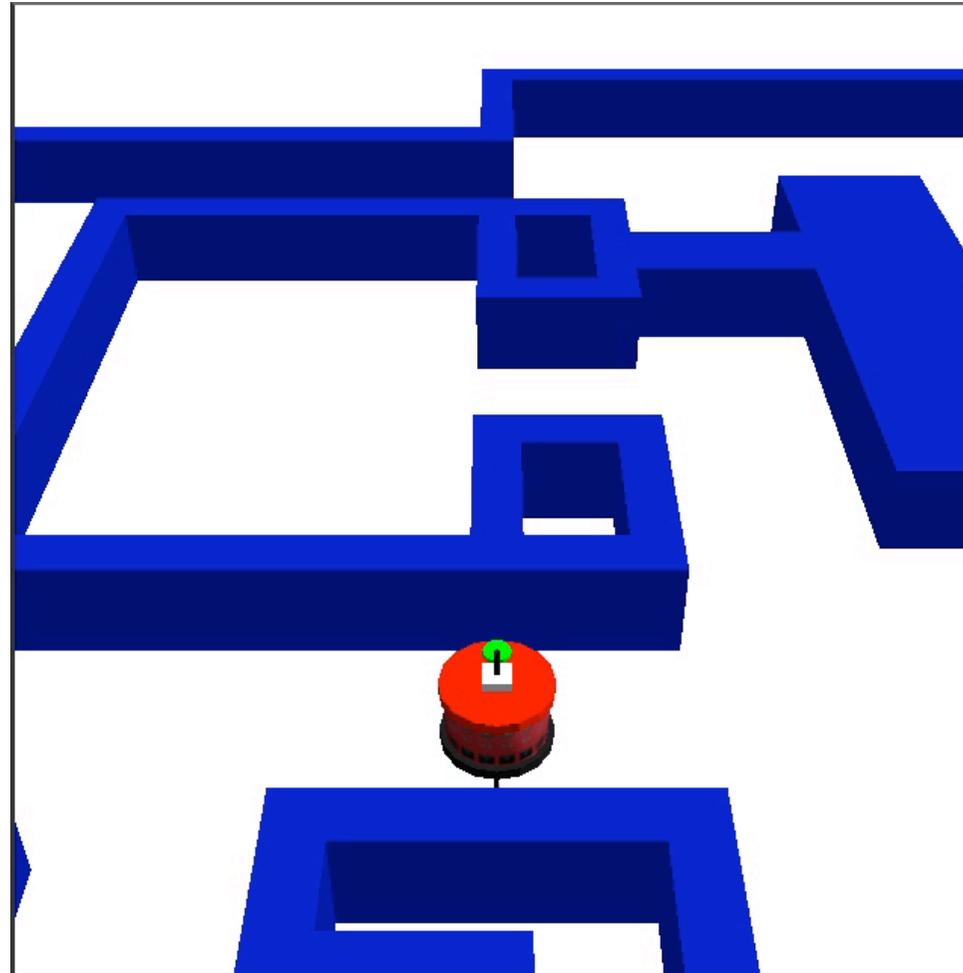
Altri argomenti interessanti ...

- pianificazione in presenza di vincoli
 - ingressi di velocità limitati
 - angolo di sterzo limitato (car-like)
 - cammini a lunghezza minima e traiettorie a tempo minimo
 - presenza di ostacoli
- navigazione
 - localizzazione del robot mobile nell'ambiente (noto)
 - costruzione di mappe ambientali con sensori esteroceettivi
 - SLAM = Simultaneous Localization And Mapping
 - esplorazione di ambienti ignoti
- locomozione su gambe



Esplorazione randomizzata

Robot
Magellan
con laser SICK
a bordo





Locomozione su gambe

Accompanying video submitted to ICRA'07

Development of a multimode navigation system for an assistive robotics project

A. Cherubini G. Oriolo F. Macri

F. Aloise F. Cincotti D. Mattia

Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Eudossiana 18, 00184 Roma, Italy
{cherubini, oriolo} @dis.uniroma1.it

Fondazione Santa Lucia IRCCS
Via Ardeatina 306, 00179 Roma, Italy
{f.cincotti, d.mattia} @hsantalucia.it

- progetto ASPICE (Telethon)
- robot SONY Aibo
- comandi mediante
BCI = Brain
Computer Interface

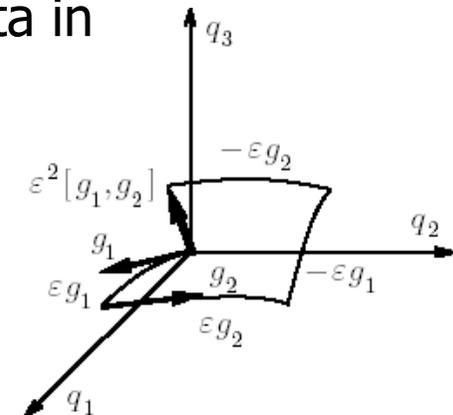


Appendice

Geometria differenziale - 1

- un **campo di vettori** (differenziabile) $f : \mathbb{R}^n \mapsto T_q \mathbb{R}^n$ è un'applicazione da ogni punto di \mathbb{R}^n allo spazio tangente $T_q \mathbb{R}^n$
- per un'equazione differenziale $\dot{q} = f(q)$, il **flusso** $\phi(q)_t^f$ del campo di vettori f è l'applicazione che associa a ciascuna q la "soluzione" che da essa evolve: $\frac{d}{dt} \phi_t^f(q) = f(\phi_t^f(q))$
- il flusso ha la proprietà di **gruppo** $\phi_t^f \circ \phi_s^f = \phi_{t+s}^f$
- nei sistemi dinamici **lineari** $f(q) = Aq$, il flusso è $\phi(q)_t^f = e^{At}$
- partendo da q_0 , un flusso **infinitesimale** di durata ϵ lungo g_1 , poi lungo g_2 , poi lungo $-g_1$ e infine lungo $-g_2$, porta in

$$\begin{aligned} q(4\epsilon) &= \phi_\epsilon^{-g_2} \circ \phi_\epsilon^{-g_1} \circ \phi_\epsilon^{g_2} \circ \phi_\epsilon^{g_1}(q_0) \\ &= q_0 + \underbrace{\epsilon^2 \left(\frac{\partial g_2}{\partial q} g_1(q_0) - \frac{\partial g_1}{\partial q} g_2(q_0) \right)}_{\text{parentesi di Lie}} + O(\epsilon^3) \end{aligned}$$





Appendice

Geometria differenziale - 2

- **proprietà** delle parentesi di Lie
 - 1) $[\alpha f + \beta f', g] = \alpha[f, g] + \beta[f', g], \quad \forall \alpha, \beta \in \mathbb{R}$
 - 2) $[f, g] = -[g, f]$
 - 3) $[f, [g, h]] + [h, [f, g]] + [g, [h, f]] = 0$
- una **distribuzione** Δ associata ad un insieme di campi di vettori differenziabili $\{g_1, \dots, g_m\}$ assegna a ciascun punto q un sottospazio del suo spazio tangente
$$\begin{aligned} \Delta &= \text{span}\{g_1, \dots, g_m\} \\ &\Updownarrow \\ \Delta_q &= \text{span}\{g_1(q), \dots, g_m(q)\} \subset T_q \mathbb{R}^n \end{aligned}$$
- una distribuzione è **regolare** se $\dim \Delta_q = \text{cost}, \forall q$
- l'insieme dei vettori differenziabili su \mathbb{R}^n con l'operazione definita dalla parentesi di Lie è un'**algebra di Lie**
- una distribuzione Δ è **involutiva** se è chiusa sotto l'operazione di parentesi di Lie
$$[g_i, g_j] \in \Delta \quad \forall g_i, g_j \in \Delta$$