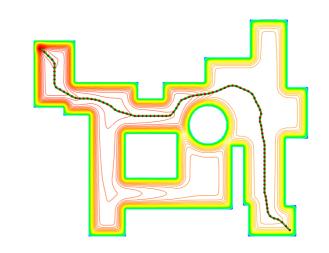


Navigazione Robotica

Giovanni De Luca Laboratorio Progettazione Elettronica www.delucagiovanni.com



Sommario del corso

- Metodi e Sistemi di localizzazione
 - Odometria e sistemi di localizzazione alternativi
- Mappe e modelli dell'ambiente
 - Mappe Metriche e Topologiche
- Tecniche di Planning
 - □ Path Planning e Path Following

Localizzazione

Giovanni De Luca Laboratorio Progettazione Elettronica www.delucagiovanni.com

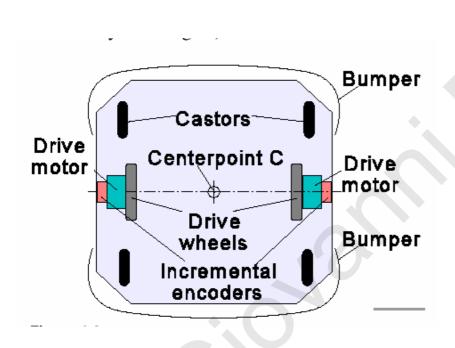
Localizzazione

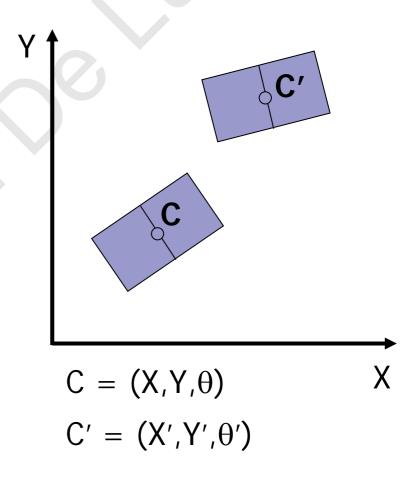
Metodi di localizzazione

- Dead Reckoning Odometria
- Boe Attive
- Sistemi di localizzazione Map-Based
- Landmark Naturali e Artificiali

- Si basa sul calcolo dei giri effettuati dalle ruote del robot (misurata con gli encoder) durante gli spostamenti.
- Fornisce una buona accuratezza su piccoli spostamenti.
- L'errore commesso si accumula nel tempo con la distanza percorsa dal robot (scarsa precisione sulle lunghe distanze).
- L'informazione odometrica viene corretta utilizzando sistemi di localizzazione alternativi.

Esempio di calcolo dell'odometria (per piccoli spostamenti)





Supponiamo che dopo un certo intervallo (I) gli encoder della ruota sinistra e destra abbiano registrato rispettivamente un incremento del numero di impulsi pari a N_L e N_R .

Sia

 $C_m = \pi D/n C_e$

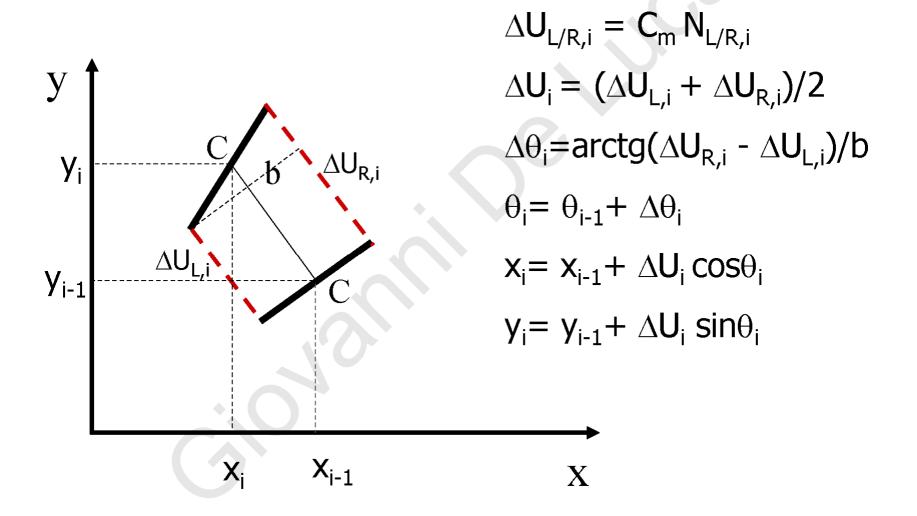
dove

C_m = fattore di conversione che traduce gli impulsi lineari in distanza lineare effettuata delle ruote

D = diametro nominale delle ruote

C_e = risoluzione degli encoder

n = Rapporto di riduzione tra motore (dove è posizionato l'encoder)
 e ruota



Possiamo calcolare la distanza percorsa dalle ruote sinistra e destra, $\Delta U_{L,i}$ e ΔU_{R} , come

$$\Delta U_{L/R,i} = C_m N_{L/R,i}$$

Allora, la distanza percorsa dal centro del robot, ΔU_i risulta essere:

$$\Delta U_{i} = (\Delta U_{L,i} + \Delta U_{R,i})/2$$

Mentre l'angolo di orientazione del robot risulta essere incrementato di

$$\Delta\theta_{i} = (\Delta U_{R,i} + \Delta U_{L,i})/b$$

Dove (**b**) è la distanza tra le due ruote della base (idealmente misurata come la distanza tra i punti di contatto delle le ruote con il pavimento)

La nuova posizione del robot risulta essere:

$$\theta_{i} = \theta_{i-1} + \Delta \theta_{i}$$

$$x_{i} = x_{i-1} + \Delta U_{i} \cos \theta_{i}$$

$$y_{i} = y_{i-1} + \Delta U_{i} \sin \theta_{i}$$

Dove $(x_{i-1}, y_{i-1}, \theta_{i-1})$ era la posizione nello spazio del centro del robot c.

Calcolo spostamento lineare

--- Calcolo spostamento lineare delle ruote Cm -----

Cm = (Pi * diametro_ruote) / (riduzione * impulsi_encoder)

Cm = (3.1415926 * 100) / (43 * 400)

Cm = 314.15926 / 17200

Cm = 0.018265073 per ogni impulso encoder

Inverso 1 / 0,018265073 = 54.74

Giovanni De Luca Navigazione Robotica 11

Calcoli odometrici

```
(Timer 10 mSec o meno)
Differenza(1) = (Val_Encoder(1) - Val_Encoder_Old(1))
Differenza(2) = (Val_Encoder(2) - Val_Encoder_Old(2))
Delta(1) = Differenza(1) / Spostamento_lineare_ruota
Delta(2) = Differenza(2) / Spostamento_lineare_ruota
Delta U = (Delta(1) + Delta(2)) / 2
Delta_Theta = (Delta(2) - Delta(1)) / Interasse_ruote
Theta = Theta + Delta Theta
Xpos = Xpos_old + Delta_U * Cos(Theta)
Ypos = Ypos_old + Delta_U * Sin(Theta)
Angolo = Theta * 180 / PI
Xpos old = Xpos
Ypos_old = Ypos
Val\_Encoder\_Old(1) = Val\_Encoder(1)
Val_Encoder_Old(2) = Val_Encoder(2)
```

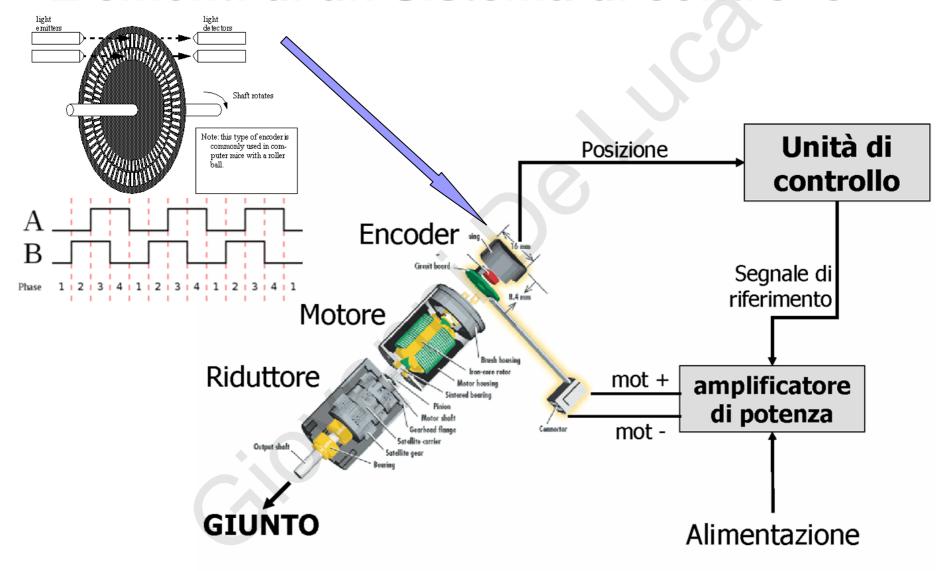
Sistema di controllo

Giovanni De Luca Laboratorio Progettazione Elettronica www.delucagiovanni.com

Sistema di controllo

Un sistema di controllo fornisce un comando in tensione o in corrente agli attuatori (motori) in modo da far assumere ai giunti una configurazione desiderata

Elementi di un Sistema di controllo



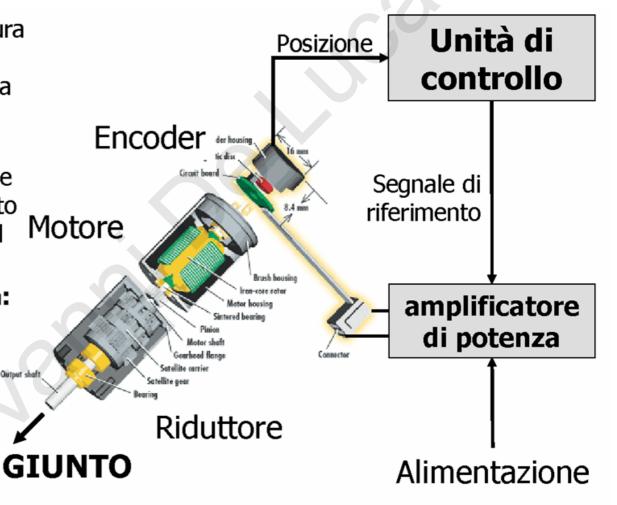
Elementi unità di controllo

Encoder: sensore che misura la rotazione dei giunti in valore relativo o assoluto. La misurazione avviene in "tacche di encoder"

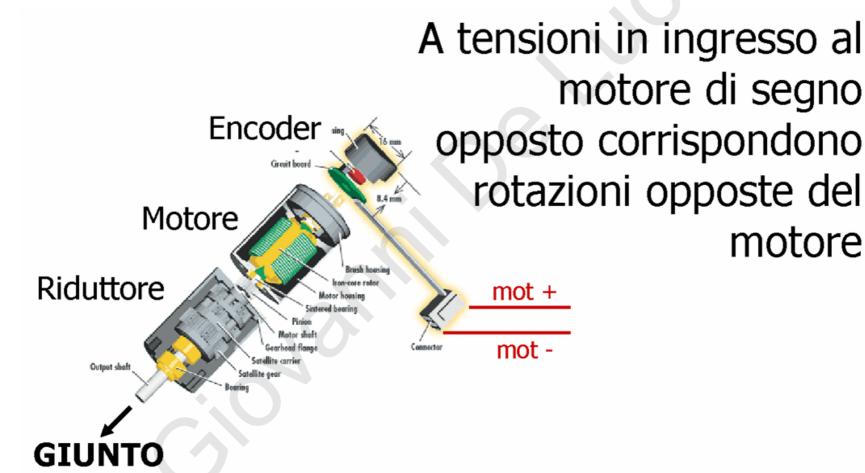
Riduttore: meccanismo che riduce i giri dell'asse montato sul giunto rispetto ai giri del motore (es. riduzione 1:N)

Amplificatore di potenza: amplifica un segnale di riferimento in un segnale di potenza per muovere il motore

Unità di controllo: unità che produce un segnale di riferimento per il motore



Elementi unità di controllo



Relazioni tra le parti

- \bullet : posizione giunto in gradi
- q: posizione giunto in tacche di encoder
- N: rapporto di riduzione del motore
- R: risoluzione dell'encoder (numero di tacche per giro)

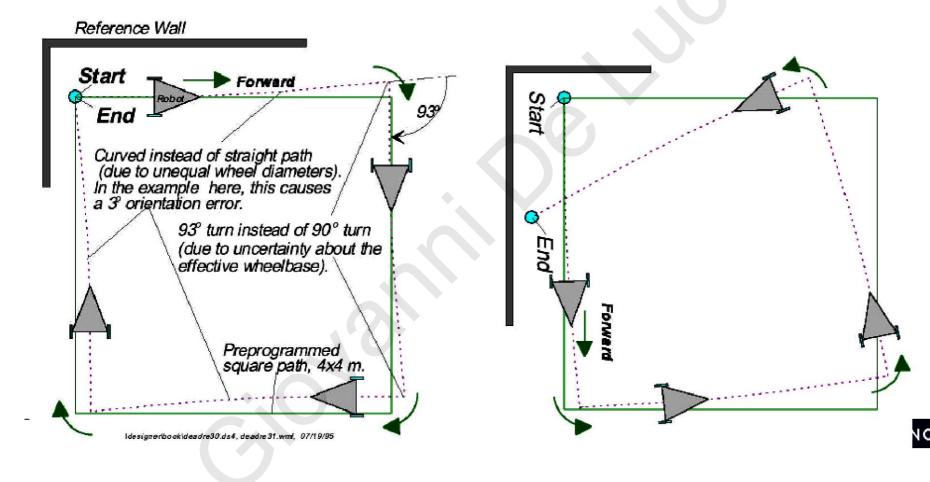
$$\theta = \frac{q}{R \times N} \times 360^{\circ}$$

Odometria – Errori

Gli errori odomoetrici sono di due tipi:

- Errori sistematici, causati da:
 - Diametro differenti delle due ruote.
 - La dimensione reale delle ruote è diversa dalla dimensione nominale.
 - Disallineamento delle ruote.
 - Finita risoluzione degli encoder.
- Errori non sistematici, causati da:
 - Movimenti su pavimenti sconnessi
 - Movimenti su oggetti non previsti
 - Scivolamento delle ruote causato da
 - forti accelerazioni
 - pavimenti scivolosi
 - forze esterne (ostacoli)

Stima dell'errore odometrico



Wake-Up Problem

■ Wake-Up Problem: in questo caso il robot non ha idea della propria posizione iniziale. Compito della localizzazione è aiutare il robot a localizzarsi attraverso l'analisi dell'ambiente circostante. Il problema è più complesso del precedente, e il robot potrebbe avere indicazioni e stime discordanti sulla propria posizione. Le tecniche che risolvono questo problema sono dette tecniche globali.

Kidnapped Robot Problem

• Kidnapped Robot Problem: è il problema più complesso. Il robot è perfettamente localizzato, ma viene improvvisamente spostato in una nuova posizione senza esserne aggiornato. In realtà il Wake-Up Problem è un sottocaso del Kidnapped Robot Problem, in cui il robot è avvertito di essere stato spostato e di doversi rilocalizzare.

La localizzazione robotica affronta i tipi di problemi finora descritti attraverso sistemi di localizzazione relativa e sistemi di localizzazione assoluta.

Sistema di guida per misure assolute

- per ridurre l'incertezza sulla pose del robot
- usa un insieme delle seguenti tecniche
- Guida fissa: seguire dei cammini precedentemente calcolati usando bande magnetiche o ottiche – solo cammini fissi
- Beacons: il robot trova la sua posizione attraverso i segnali lanciati da sensori installati nell'ambiente
- Landmark: cartelli installati nell'ambiente rispetto ai quali il robot riconosce la sua posizione; si parla di landmark naturali se invece il robot riconosce delle caratteristiche stabili dell'ambiente
- Integrate con misure relative:
- Odometria: calcola la propria posizione dai movimenti comandati alle ruote

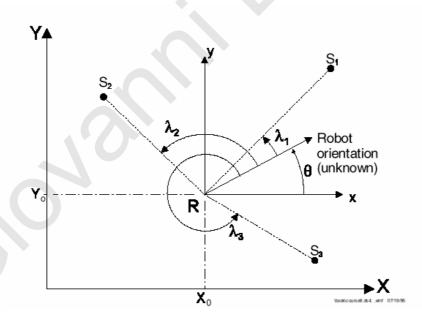
Boe Attive

- I sistemi di localizzazione con boe attive sono costituiti da un insieme di dispositivi ricevitori/trasmettitori (boe) di cui è conosciuta la posizione assoluta nell'ambiente e che sono rilevabili da un sensore trasmettitore/ricevitore posto sul robot.
- L'unità a bordo del robot è un sensore rotante che invia un segnale laser recepibile dalle boe e che codifica l'angolo di rotazione della torretta.
- Le boe ricevono il segnale laser e rispondono con un segnale ad infrarossi, recepibile dalla torretta rotante, che codifica l'angolo di rotazione ricevuto e l' identificativo.

Boe Attive

L'algoritmo di localizzazione è basato sulla procedura di Triangolazione:

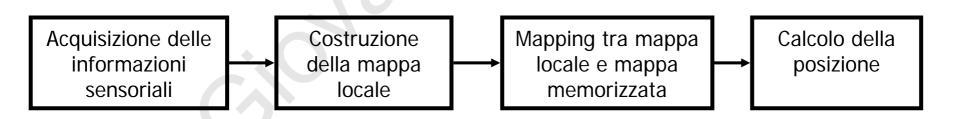
- L' una unità rotante posta a bordo del robot è in grado di misurare gli angoli λ_1 , λ_2 , λ_3
- Conoscendo la posizione delle 3 boe è possibile determinare tramite triangolazione la posizione assoluta del robot (X,Y,θ)



- I sistemi di localizzazione basati sulle mappe, conosciuti anche come map matching, utilizzano uno o più sistemi sensoriali per costruire una mappa locale.
- La mappa locale è poi confrontata con una mappa globale precedentemente memorizzata.
- Se un match viene trovato, il robot calcola la sua posizione e orientamento nello spazio.
- Una mappa può essere un modello CAD o può essere costruita utilizzando i sistemi sensoriali del robot

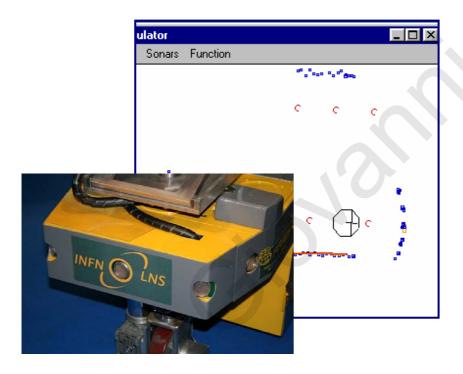
Procedura di localizzazione:

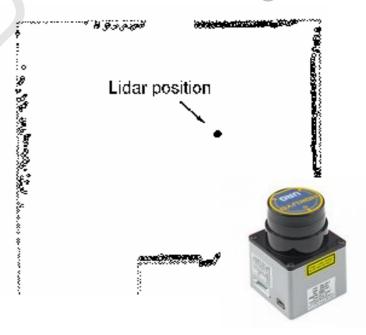
- Per semplificare il problema si assume che la posizione approssimativa corrente del robot (odometria) è conosciuta.
- Passi della procedura di localizzazione:



Sistemi Sensoriali Utilizzati:
 Sensori ad Ultrasuoni

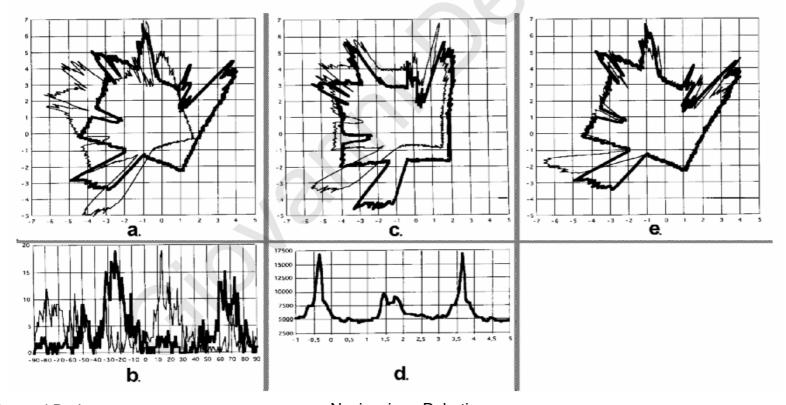
Laser Ranger





Tecniche di mapping utilizzate:

Correlazione



Landmark

- I Landmark sono delle forme caratteristiche che il robot può riconoscere utilizzando i propri sistemi sensoriali.
- I Landmark possono essere forme geometriche (es. rettangoli, linee, cerchi, ..) e possono contenere informazioni aggiuntive (es. Bar-code).
- I Landmark sono scelti in modo tale da essere facilmente riconosciuti dal robot.
- Posizione e caratteristiche dei Landmark devono essere memorizzate nella base di dati del robot.

Landmark

- I landmark possono essere distinti in
 - □ Landmark Naturali: oggetti o caratteristiche già presenti nell'ambiente e che hanno funzionalità proprie (es. luci, corridoi, porte, ecc.).
 - □ Landmark Artificiali: oggetti o marker appositamente sviluppati e piazzati nell'ambiente per consentire la localizzazione del robot.

Landmark Naturali

- Non è necessario strutturare l'ambiente, poiché si utilizzano le caratteristiche dell'ambiente stesso.
- Il problema principale è trovare e mappare le caratteristiche rilevate dal sistema sensoriale di input sulla mappa.
- Generalmente, il sistema sensoriale utilizzato è la visione.
- Landmark naturali facilmente riconoscibili tramite procedure di edge-detection con il sistema di visione sono ad esempio lunghi spigoli verticali (porte e giunzioni di pareti) o le luci dei corridoi.

Landmark Artificiali

- Necessitano strutturazione dell'ambiente.
- Generalmente, il sistema sensoriale utilizzato è la visione.
- Sono più semplici da riconoscere poiché vengono appositamente progettati per avere un ottimo contrasto con l'ambiente e perché dimensioni e forme sono conosciuti a priori.
- Esempi di landmark artificiali utilizzati:
 - rettangolo nero con quattro punti bianchi sugli angoli
 - cerchio metà nero e metà bianco
 - Led attivi
 - Sensori ad infrarossi

Esempi di Basi Mobili e Tecniche di Navigazione

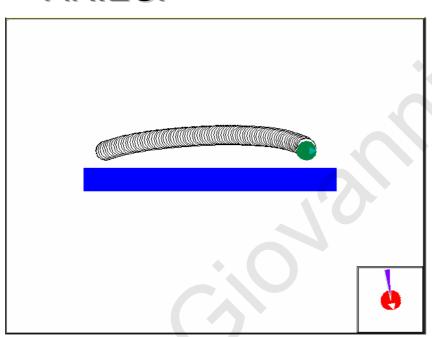


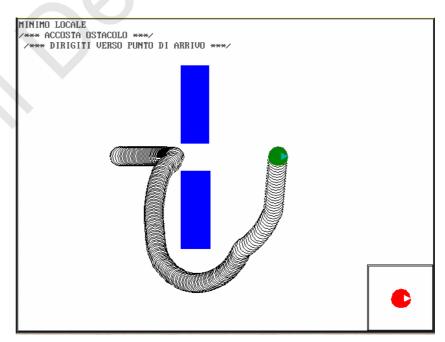
Mobile Robot ARIES:

- Mappa topologica
- Path Planning
- Campi di Potenziale

Esempi di Basi Mobili e Tecniche di Navigazione

La simulazione del Path Following del sistema ARIES:





Esempi di Basi Mobili e Tecniche di Navigazione

Posizione calcolata rispetto Posizione stimata alla parete rilevata dall'odometria int. Y Axis Direzione di Movimento θ_{d} (X_e, Y_e, θ_e) Direzione di Movimento d_{C} d_e Parete Rilevata **Parete** Asse X

 $X_e = X_e$ $Y_e = Y_e + (d_c - d_e)$ $\theta_e = \theta_e + \theta_d$ Navigazione Robotica

Mapping and Planning

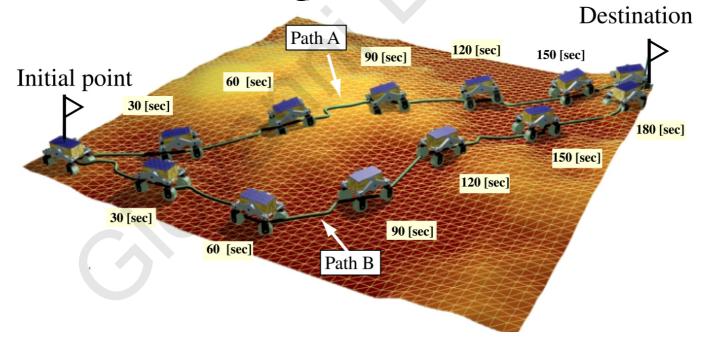
Giovanni De Luca Laboratorio Progettazione Elettronica www.delucagiovanni.com

Introduzione

- Al fine di ampliare il campo di impiego dei robot, è necessario conferire loro un elevato grado di autonomia, che comporta la capacità da parte di questi sistemi di operare in ambienti non strutturati.
- Il robot deve mostrarsi in grado di interagire con il proprio ambiente, sia estraendo una descrizione sintetica e robusta degli oggetti che lo circondano, mediante le informazioni fornitegli dal proprio apparato sensoriale, sia utilizzando in maniera idonea questa descrizione per le proprie attività di pianificazione del moto.
- In questo corso ci occuperemo del processo di costruzione della mappa, comunemente chiamato *map-building*, di un ambiente non strutturato ed inizialmente sconosciuto.
- Tale necessità è abbastanza evidente per i robot autonomi che devono esplorare luoghi di cui mancano completamente informazioni a priori, come la superficie di un pianeta, ma anche per robot mobili che operano in ambienti potenzialmente ostili e pericolosi, come miniere e centrali nucleari, è fondamentale la capacità di saper gestire eventi inaspettati non appena si verificano.

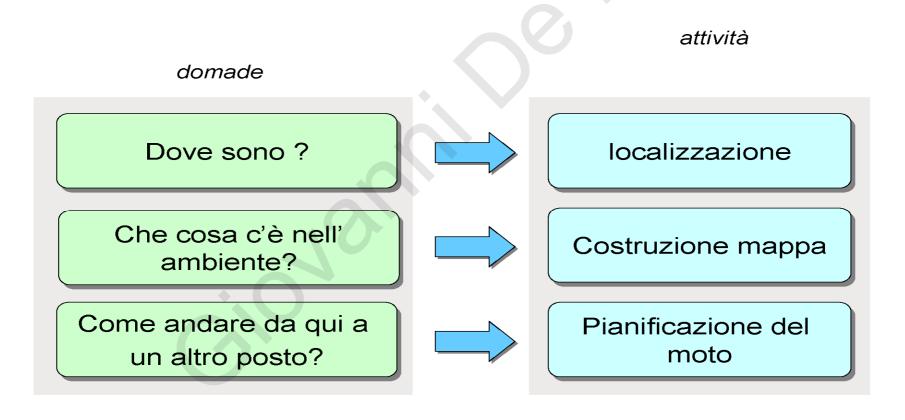
Il problema della Navigazione Robotica

In robotica, il problema della navigazione può essere definito come il problema di raggiungere una posizione finale partendo da una posizione iniziale, specificate in termini geometrici o di stato sensoriale, evitando gli ostacoli.



Il problema della Navigazione Robotica

Le domande classiche che devono essere risolte nel problema della navigazione sono:



L'approccio classico al problema

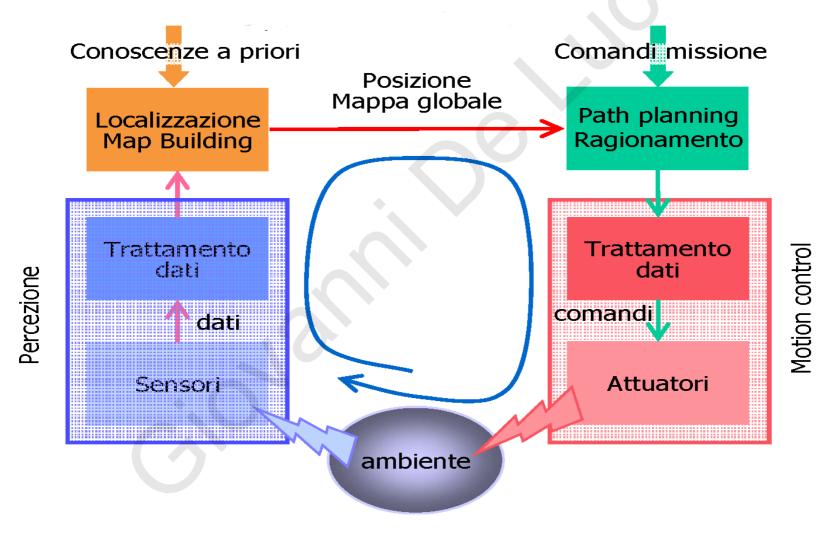
Nell'approccio classico, le risposte alle domande precedenti sono

- Localizzazione: posizione geometrica (coordinate X,Y e θ rispetto ad un sistema di riferimento assoluto) o stato sensoriale nell'ambiente in cui il robot naviga
- Mappe o Modelli: formalizzazione e rappresentazione dell'ambiente
- Planning: pianificazione dei movimenti del robot nell'ambiente

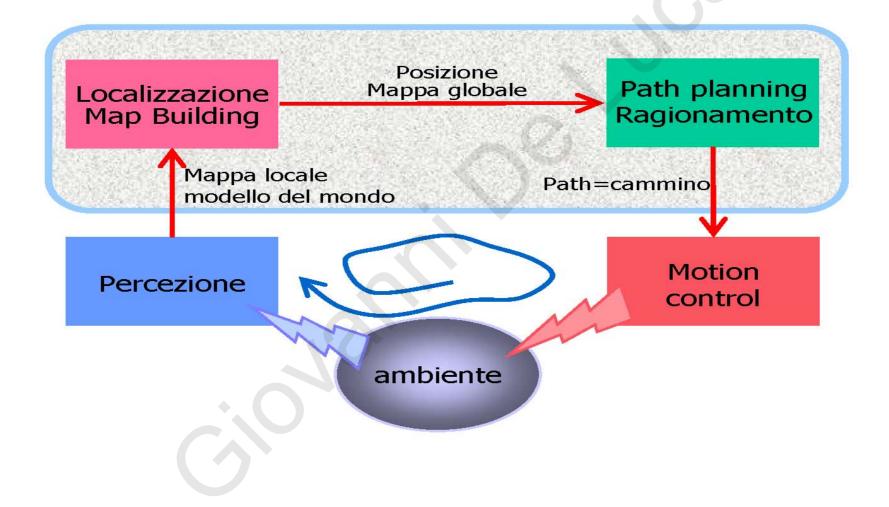
Sistemi Reattivi

- Nei sistemi reattivi il robot interagisce con il mondo tramite sensori ed attuatori.
- La conoscenza non è modellata nè memorizzata, ma è estratta 'on-line' dal mondo stesso tramite i sensori.
- I comportamenti del robot sono definiti come reazione alle informazioni percepite sull'ambiente Plan is not a program

Supervisione e controllo



Supervisione e controllo



Strategie di controllo

- Struttura dell'anello di controllo
 - L'ambiente o "mondo" si modifica dinamicamente
 - Non esiste un modello "compatto" del mondo
 - Vi sono molte sorgenti di incertezza, sia nel mondo sia nel robot
- Due approcci possibili
 - "classico" modello <u>deliberativo</u>
 - Modellazione completa (model-based)
 - Basato su funzioni
 - Decomposizione orizzontale
 - "moderno" modello <u>reattivo</u>
 - Nessun modello o quasi: basato su comportamenti (behavior-based)
 - Decomposizione verticale
 - Approccio bottom-up

Caratteristiche delle strategie di controllo

DELIBERATIVOModel-based

Puramente simbolico

REATTIVO Behavior-based

Riflessivo

Velocità di risposta

Capacità predittive

Dipendenza da modelli accurati del mondo

- Dipendente dalla rappresentazione del mondo
- Risposta lenta
- Intelligenza ad alto livello (cognizione)
- Latenza variabile

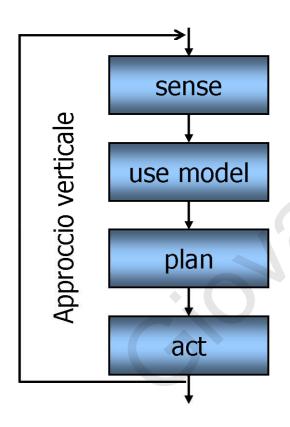
- Indipendente dalla rappresentazione (representation-free)
- Risposta in real-time
- Intelligenza a basso livello (stimolorisposta)
- Calcolo semplice e veloce

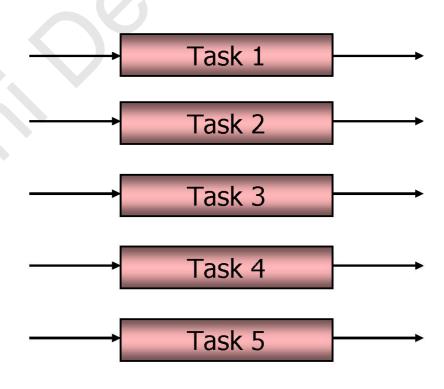
Caratteristiche delle strategie di controllo

Sense – Plan – Act

Questo paradigma può impedire una risposta tempestiva ed efficace del robot Subsumption/Reactive model

http://ai.eecs.umich.edu/cogarch0/subsump/classificazione gerarchizzante

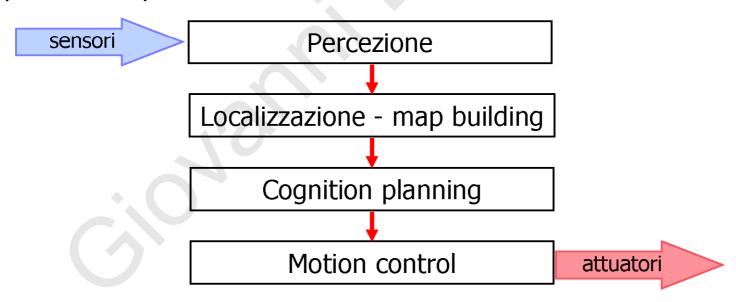




Approccio Model based

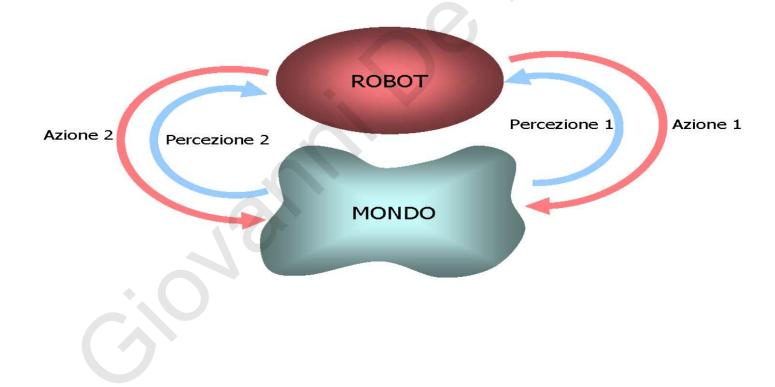
- Modellazione completa del mondo
- Ogni blocco è una funzione computata
- Decomposizione verticale dell'architettura e nested-embodiment delle funzioni:

Un primo esempio



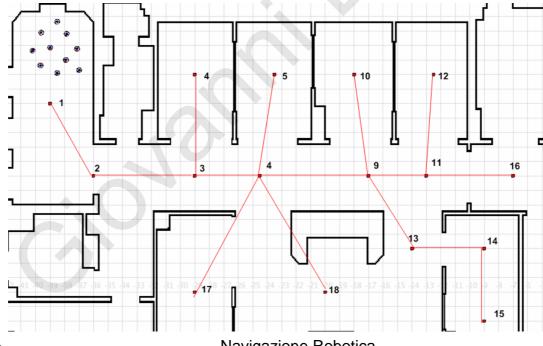
Approccio Behavior based

- Sistema "reattivi"
- Comportamento "riflessivo"
- Percezione-azione



Mappe e Rappresentazione del mondo

- Una mappa è una rappresentazione matematica o geometrica della conoscenza del mondo.
- Un modello è una rappresentazione di alto livello che dà significato alla conoscenza del mondo (ad esempio un grafo).



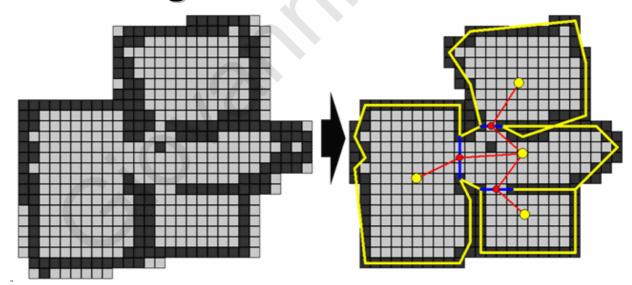
Mappe e Rappresentazione del mondo

- Mappe Geometriche: rappresentano gli oggetti del mondo in base alle loro dimensioni e coordinate rispetto ad un sistema di riferimento assoluto
- Mappe Topologiche: rappresentano gli oggetti del mondo (punti di interesse) in base alle loro caratteristiche e in base alle relazioni tra loro intercorrenti

Mappe Geometriche

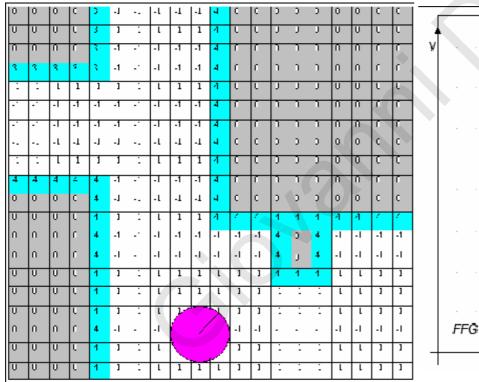
I principali metodi per la rappresentazione dell'ambiente tramite mappe geometriche sono:

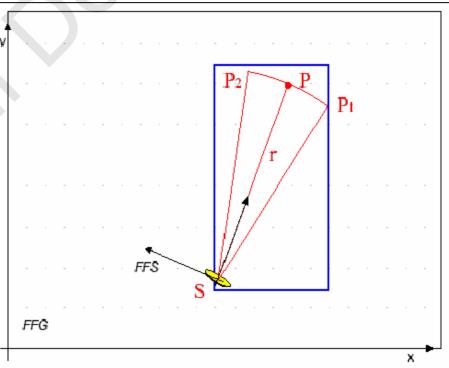
- Occupancy grid
- Descrizione geometrica



Occupancy Grid

- Un ambiente può essere rappresentato tramite una griglia bidimensionale.
- Ad ogni elemento della griglia è associato un valore che indica lo stato della cella (libera/occupata)





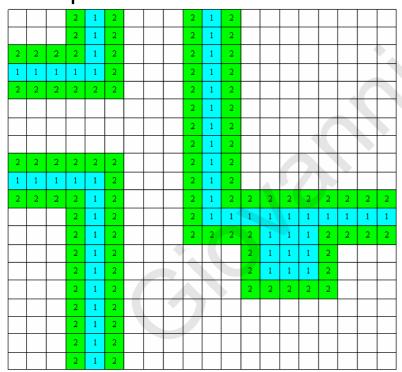
Primo stadio della costruzione di una mappa di traversabilità

Per prima cosa la griglia viene segnata con degli "1" che si trovano in tutte le celle della mappa che corrispondono ai punti occupati dagli ostacoli.

In questa formulazione i valori inferiori indicano valori di minore traversabilità. Per esempio "1" significa che la cella è completamente occupata e quindi non attraversabile.

Invece, valori maggiori di 1 indicano che è possibile attraversare quella

cella o quell'area.



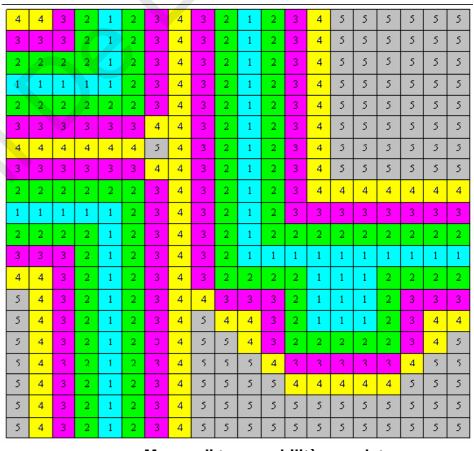
Questo vuol dire che la mappa contiene a priori un'informazione sulla traversabilità di aree non ostacolate. In essa sono indicati infatti dei valori per ciascuna cella corrispondente al punto sulla mappa ed è quindi possibile sapere il suo grado di traversabilità. Per marcare invece aree inesplorate vengono usate dei valori di traversabilità bassa, così che il robot ne stia alla larga.

Dopo aver inizializzato la mappa di traversabilità con tutti "1", sulle *otto* posizioni adiacenti, se libere, vengono messi dei "2" per indicare che la distanza dall'ostacolo più vicina e appunto 2.

Mappa di traversabilità completa

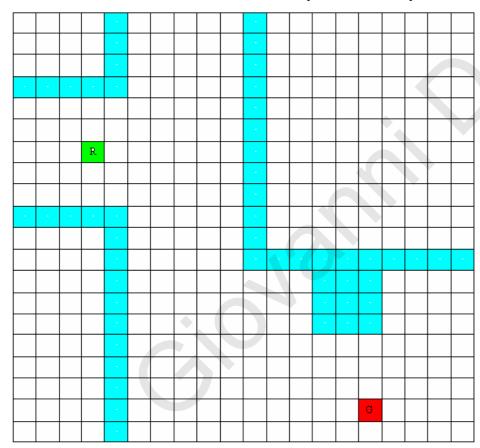
A questo punto per ogni "2" messo sulla mappa, per ognuna della *quattro* posizioni adiacenti, se libere, viene messo un "3". Allo stesso modo per il "3" vengono messi dei "4" e per il "4" vengono messi dei "5" nelle quattro posizioni adiacenti. A questo punto in tutte le altre celle della mappa vengono messi dei "5". Si poteva benissimo continuare per numeri di traversabilità superiori, ma per questa applicazione è

sufficiente fermarsi a questo punto.



Algoritmo di Fronte d'Onda e sua mappa.

Nell'algoritmo standard fronte d'onda la cella rappresentante il punto goal è segnata con "1" e tutte quelle non occupate vicine sono segnate con "2"; tutte quelle non occupate vicine al "2" sono segnate con "3", ecc, fino a che il punto di partenza è raggiunto.



Inizializzazione di una mappa col metodo del Fronte D'onda

Costruzione di una mappa col metodo del Fronte D'onda

					29	28	27	26	25										
					28	27	26	25	24										
					27	26	25	24	23										
					26	25	24	23	22										
30	29	28	27	26	25	24	23	22	21										
29	28	27	26	25	24	23	22	21	20										
28	27	26	25	24	23	22	21	20	19										
27	26	25	24	23	22	21	20	19	18										
26	25	24	23	22	21	20	19	18	17										
					20	19	18	17	16										
					19	18	17	16	15										
					18	17	16	15	14										
					17	16	15	14	13	12	11	10				8	9	10	11
					16 (15	14	13	12	11	10	9				7	8	9	10
					15	14	13	12	11	10	9	8				ń	7	Я	9
					14	13	12	11	10	9	8	7	6	5	4	5	6	7	8
					13	12	11	10	9	8	7	6	5	1	3	1	5	6	7
					12	11	10	9	8	7	6	5	4	3	2	3	4	5	6
					11	10	9	8	7	6	5	4	3	2	1	2	3	4	5
					12	11	10	9	8	7	6	5	4	3	2	3	4	5	6

Percorso del robot sulla mappa

					29	28	27	26	25										
				-	28	27	26	25	24	-									
				-	27	26	25	24	23										
-	-	-	1	-	26	25	24	23	22	1									
30	29	28	27	26	25	24	23	22	21	,									
25	28	27	26	25	24	23	22	21	20	,									
28	27	26	25	24	23	22	21	20	19	1									
25	26	25	24	23	22	21	20	19	18										
26	25	24	23	22	21	20	19	18	17	•									
-	-	-	-	-	20	19	18	17	16										
					19	18	17	16	15										
				-	18	17	16	15	14			-	-	-	-	-	-	-	-
				-	17	16	15	14	13	12	11	10	-	-	-	8	9	10	11
					16	15	14	13	12	11	10	9		-	-	7	8	9	10
				•	15	14	13	12	11	10	9	8	-	-	-	6	7	8	9
					14	13	12	11	10	9	8	7	6	5	4	5	6	7	8
					13	12	11	10	9	8	7	6	5	4	3	4	5	6	7
					12	11	10	9	8	7	6	5	4	3	2	3	4	5	6
				-	11	10	9	8	7	6	5	4	3	2	1	2	3	4	5
				-	12	11	10	9	8	7	б	3	4	3	2	3	4	5	б

Mappa di un campo potenziale

					37	29	27	27	33										
					36	28	26	26	32										
					35	27	25	25	31										
					34	26	24	24	30										
33	37	36	35	34	33	25	23	23	29										
30	29	28	27	26	25	24	22	22	28										
23	27	26	25	24	23	22	21	21	27								-		
28	27	26	25	24	23	22	20	20	26										
33	32	31	30	30	29	21	19	19	25										
					28	20	18	18	24										
					27	19	17	17	23										
					26	18	16	16	22										
					25	17	15	15	21	20	19	18				16	17	18	19
					24	16	14	14	13	12	11	17				15	9	10	11
					23	15	13	12	11	10	10	16				14	8	8	9
					22	14	12	11	10	9	9	15	14	13	12	13	7	7	8
					21	13	11	10	9	8	8	7	6	5	4	5	6	6	7
					20	12	10	9	8	7	6	5	4	3	2	3	4	5	6
					19	11	9	8	7	6	5	4	3	2	1	2	3	4	5
					20	12	10	9	8	7	6	5	4	3	2	3	4	5	6

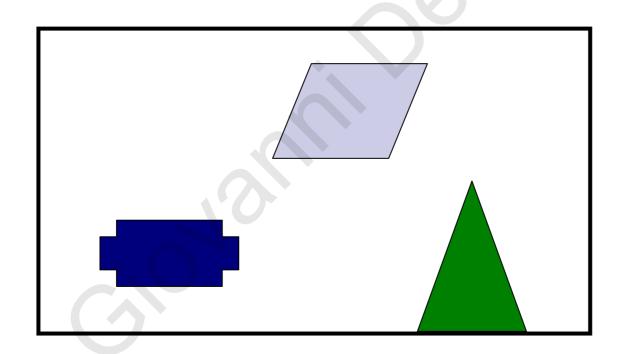
Il campo potenziale finale viene costruito usando una combinazione di trasformazioni fronde d'onda e di griglia di traversabilità. La trasformazione parte dal goal e progredisce fino a che non s'incontra il punto di partenza o fino alla fine se si vuole completare la mappa.

Percorso Iontano dagli ostacoli

																		_	_
					37	29	27	27	33										
					36	28	26	26	32										
					35	27	25	25	31										
					34	26	24	24	30										
38	37	36	35	34	33	25	23	23	29										
30	29	28	27	26	25	24	22	22	28										
28	27	26	25	24	23	22	21	21	27										
28	27	26	25	24	23	22	20	20	26										
33	32	31	30	30	29	21	19	19	25										
					28	20	18	18	24										
					27	19	17	17	23										
					26	18	16	16	22										
					25	17	15	15	21	20	19	18				16	17	18	19
					24	16	14	14	13	12	11	17				15	9	10	11
					23	15	13	12	11	10	10	16				14	8	8	9
					22	14	12	11	10	9	9	15	14	13	12	13	7	7	8
					21	13	11	10	9	8	8	7	6	5	4	5	6	6	7
					20	12	10	9	8	7	6	5	4	3	2	3	4	5	6
					19	11	9	8	7	6	5	4	3	2	1	2	3	4	5
					20	12	10	9	8	7	б	5	4	3	2	3	4	5	б

Descrizione geometrica

Un ambiente è rappresentato tramite la descrizione geometrica, generalmente in termini di poligoni, degli ostacoli e dello spazio libero.

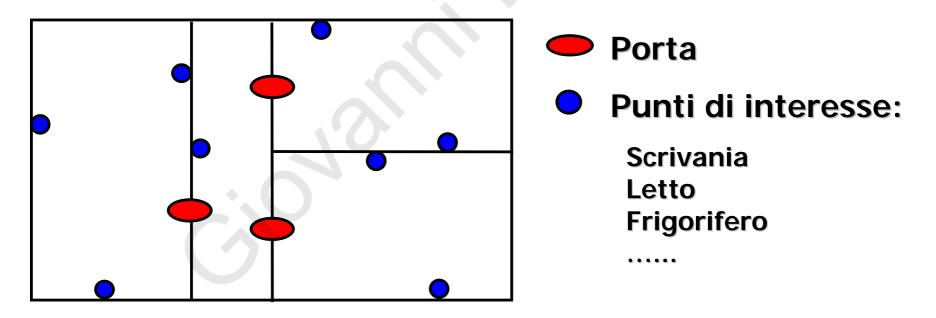


Mappe Topologiche

- Un ambiente è definito in termini di punti di interesse di rilevanza per il robot e delle relazioni intercorrenti tra i punti di interesse.
- Un punto di interesse è un oggetto naturale o artificiale che riveste importanza per la navigazione del robot (es. pareti, porte) o per l'esecuzione di compiti da parte del robot (es tavoli, letti, elettrodomestici).
- Un punto di interesse può essere definito da una posizione nello spazio del robot o da uno stato sensoriale.

Mappe Topologiche

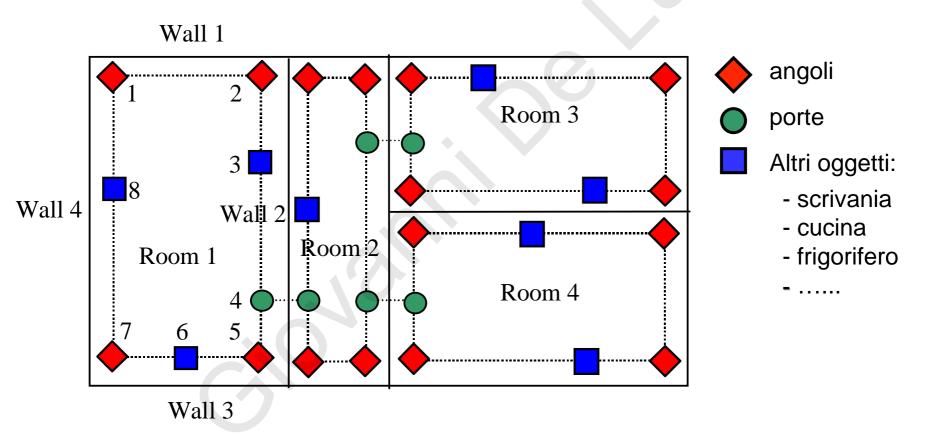
Esempio: la mappa di un ambiente generico con alcuni punti di interesse



Rappresentazione di una mappa topologica tramite grafo

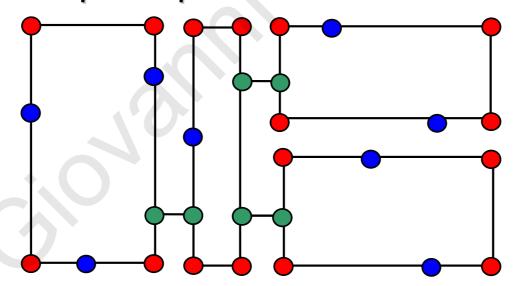
- Assegnare un numero ad ogni stanza
- Per ogni stanza, numerare le pareti in ordine orario
- Per ogni parete, numerare i punti di interesse in ordine orario

Rappresentazione di una mappa topologica tramite grafo



Rappresentazione di una mappa topologica tramite grafo

```
    G = (N,E)
    N = {punti di interesse}
    E = {(p,q) | (p∈N, q∈N, p=q±1) V (p e q rappres.
    la stessa porta per due stanze diverse}
```



Tecniche di Planning e modelli dell'ambiente

L'obbiettivo del **Planning** è determinare una traiettoria che il robot deve eseguire per raggiungere una configurazione finale a partire da una configurazione iniziale (la sua posizione attuale) evitando gli ostacoli.



Tecniche di Planning e modelli dell'ambiente

Il Planning si divide in:

- Path Planning: tecniche per la determinazione delle traiettorie che il robot deve percorrere per raggiungere la configurazione finale evitando gli ostacoli.
- Path Following: tecniche per l'esecuzione delle traiettorie generate dal Path Planning evitando gli ostacoli imprevisti.

Configuration Space

- Lo spazio è chiamato Configuration Space o C_{space} (configurazioni assumibili dal robot nello spazio).
- Il robot è rappresentato in C_{space} con un punto.
- Gli ostacoli sono rappresentati in C_{space}.
- La regione degli ostacoli è chiamata C_{obstacle}.
- La regione dello spazio libero è chiamata C_{free}.
- Un path è una traiettoria tra due configurazioni q_{init} e q_{goal} di C_{space} contenuta in C_{free}.

Path Planning e Modelli dell'ambiente per mappe geometriche

Le principali tecniche di Path Planning basate sulle mappe geometriche sono:

- Roadmap
- Decomposizione in Celle
- Campi di Potenziale

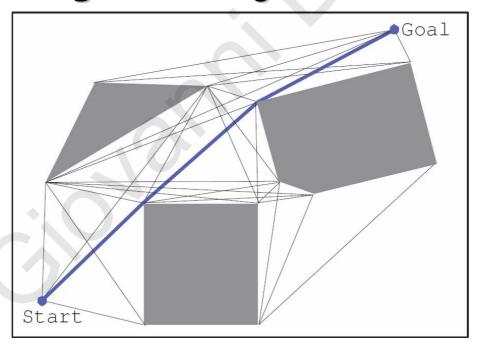
Roadmap

- L'approccio Roadmap consiste nel connettere alcuni punti dello spazio libero di (C) del robot in una rete, chiamata Carta Stradale (Roadmap), di curve unidimensionali giacenti nello spazio libero del robot.
- Il problema base del Path Planning diventa quello di congiungere la configurazione iniziale e finale alla rete e di cercare un percorso in essa.

Roadmap

Le principali tecniche di Path Planning basate sull'approccio Roadmap sono

- Visibility Graph Grafo di Visibilità
- Voronoi Diagram Diagramma di Voronoi

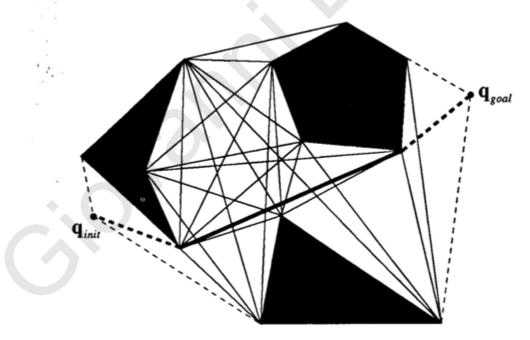


Visibility Graph

- Il grafo di visibilità è un grafo G i cui nodi sono la configurazione iniziale q_{init} e q_{goal} e tutti i vertici dei poligoni che rappresentano gli ostacoli della mappa.
- Gli archi di G sono costituiti da tutti i segmenti che congiungono due nodi di G e che non intersecano i poligoni ostacolo.
- Agli archi può essere associato un peso che corrisponde alla distanza tra i nodi.
- Un cammino da q_{init} e q_{goal} G può essere determinato sul grafo utilizzando un algoritmo dei cammini minimi che minimizza la distanza percorsa.

Visibility Graph

Esempio: il grafo di visibilità e il percorso determinato (linee tratteggiate e in grassetto) per andare da q_{init} a q_{goal}



Voronoi Diagram

- Consiste nel definire tutte le configurazioni libere nello spazio libero del robot equidistanti dalla regione degli ostacoli.
- Se gli ostacoli sono poligoni, il diagramma di Voronoi consiste in un insieme finito di segmenti e curve paraboliche (roadmap).
- Il vantaggio di questa tecnica è che le traiettorie generate tendono a massimizzare la distanza del robot dagli ostacoli

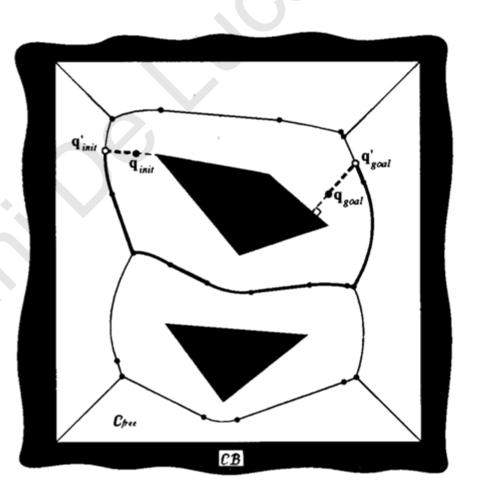
Voronoi Diagram

Dati due configurazioni iniziali q_{init} e q_{goal}, un percorso è determinato:

- collegando le configurazioni q_{init} e q_{goal} alla roadmap nei punti q'_{init} e q'_{goal}.
- determinando un percorso sul diagramma di Voronoi congiungente q'_{init} e q'_{goal}.

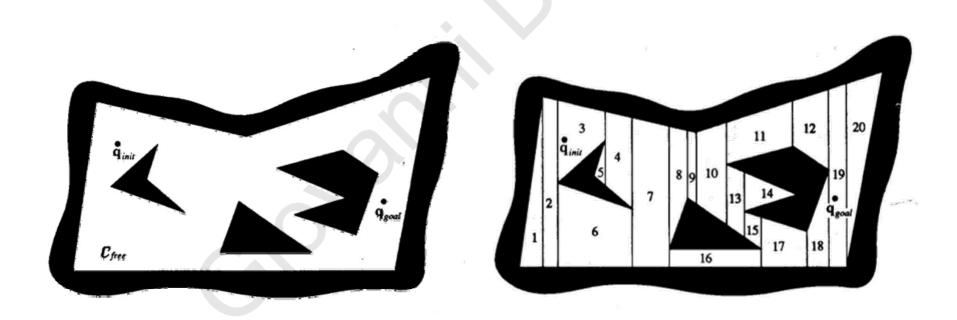
Voronoi Diagram

Voronoi e il percorso determinato (linee tratteggiate e in grassetto) per andare da q_{init} a q_{goal}.

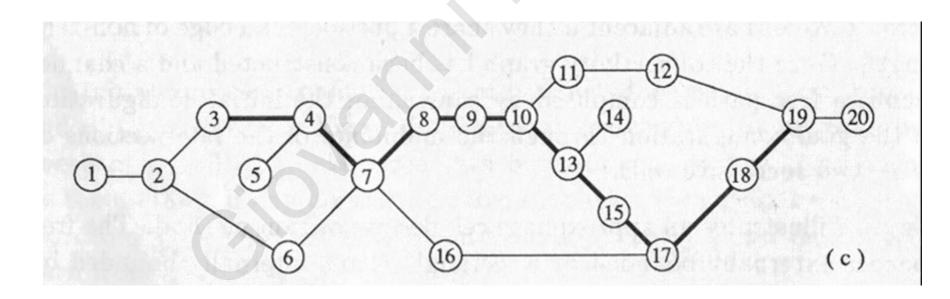


- Consiste nel decomporre lo spazio libero del robot in regioni, dette celle, tali che un percorso tra due celle adiacenti può essere facilmente determinato.
- La mappa è rappresentata tramite un grafo detto connectivity graph.
- I nodi del grafo sono costituiti dalle celle estratte dallo spazio libero del robot.
- Due nodi del grafo sono connessi se e solo se le due celle che rappresentano sono adiacenti.

Esempio di mappa e della sua decomposizione in celle.



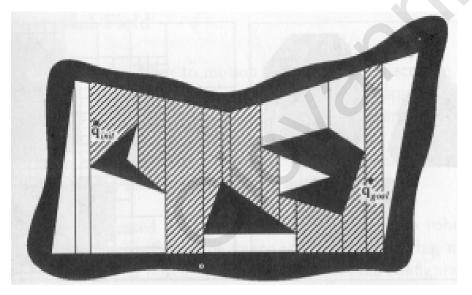
Il grafo associato alla mappa (connectivity graph) e il percorso determinato (linee in grassetto).

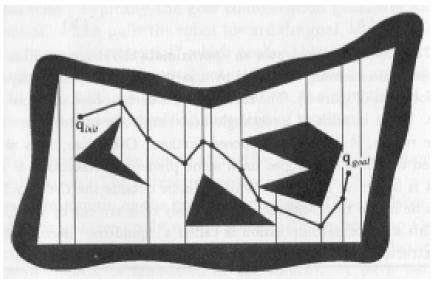


- Una traiettoria per il robot è determinata cercando un cammino sul grafo congiungente i nodi contenenti i punti q_{init} e q_{qoal}.
- Il risultato della visita del grafo è una sequenza di celle chiamata canale.
- Il percorso è determinato congiungendo i punti intermedi dei lati adiacenti alle celle del canale.

Il risultato della ricerca del grafo:

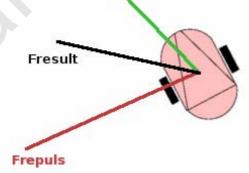
- Il canale (celle grigie).
- Il percorso determinato (linea in grassetto).





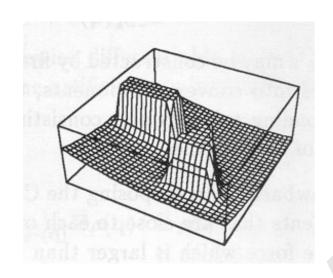
Campi di Potenziale

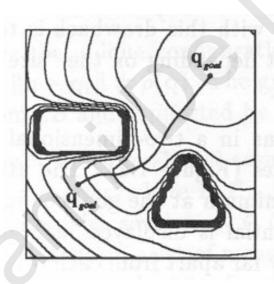
- Il robot è rappresentato da un punto nello spazio che si muove sotto l'influenza di un potenziale artificiale prodotto dalla configurazione finale e dagli ostacoli.
- La configurazione finale genera un potenziale attrattivo che spinge il robot verso l'obbiettivo.
- Gli ostacoli generano un potenziale repulsivo che spingono il robot lontano da essi.
- La somma dei potenziali attrattivo e repulsivo si traduce in una forza che muove il robot verso la configurazione finale evitando gli ostacoli.

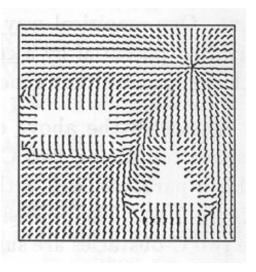


Fresult = a*Fatrac+b*Frepuls

Campi di Potenziale







La Funzione Potenziale Totale

$$U = U_{att} + U_{rep}$$

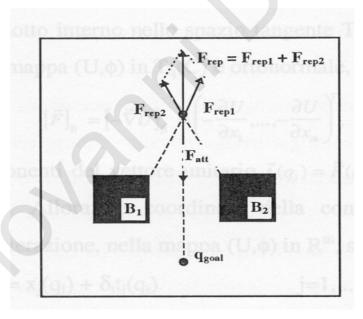
Le Curve della Funzione Totale e la Traiettoria Generata

Navigazione Robotica

La Matrice degli Orientamenti dei Vettori Gradienti Negati

Campi di Potenziale

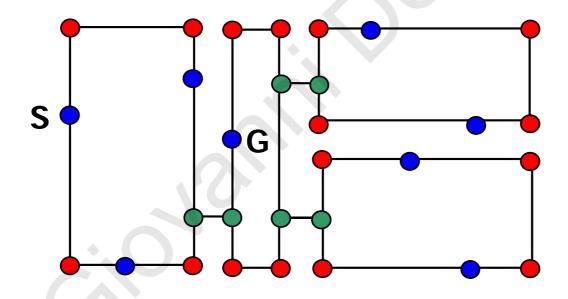
Il problema dei minimi locali della funzione risultante: Si possono generare quando la somma della forza repulsiva annulla la forza attrattiva in punti diversi da q_{qoal}



Esempio di cammino per una mappa topologica:

- segui la parete sulla destra;
- gira a destra;
- segui la parete sulla destra;
- fermati davanti alla porta;
- entra nella porta e gira a sinistra;
- segui la parete sulla sinistra;
- fermati quando hai raggiunto la scrivania;

Il problema del Path Planning si riduce a determinare un percorso sul grafo a partire dal nodo **Start** al nodo **Goal**



Esempio: è possibile applicare l'algoritmo dei cammini minimi (SPT) per trovare il percorso sul grafo che minimizza il costo totale (distanza, sicurezza, ...) dei pesi associati agli archi.

La lista dei nodi generati dall'algoritmo rappresenta la sequenza di punti di interesse che il robot deve visitare per raggiungere l'obiettivo.

Il cammino generato viene tradotto in comandi che il robot deve eseguire.

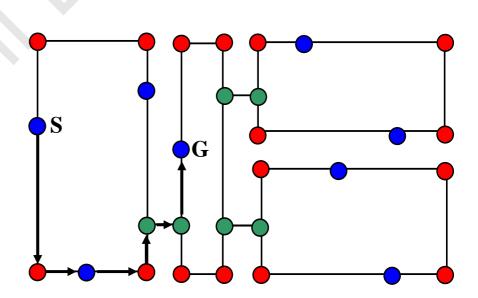
Regole per la traduzione di un cammino in una sequenza di macro comandi:

- Per il nodo Start, il macro comando generato è Segui la parete sulla destra o Segui la parete sulla sinistra, a seconda dell'ordine del nodo adiacente che deve essere raggiunto.
- Per i nodi intermedi di tipo Angolo, il macro comando generato è Cambia parete sulla destra o Cambia parete sulla sinistra.
- Per i nodi intermedi di tipo Porta il comando generato è Vai a diritto se il robot non deve entrare nella stanza adiacente o Entra nella porta e gira a sinistra (destra) se il robot deve entrare nella stanza adiacente e deve seguire la parete sulla sinistra (destra).
- Per gli altri nodi intermedi diversi dal nodo goal il comando è Segui la parete.
- Quando il robot raggiunge il nodo Goal il macro comando generato è Stop.

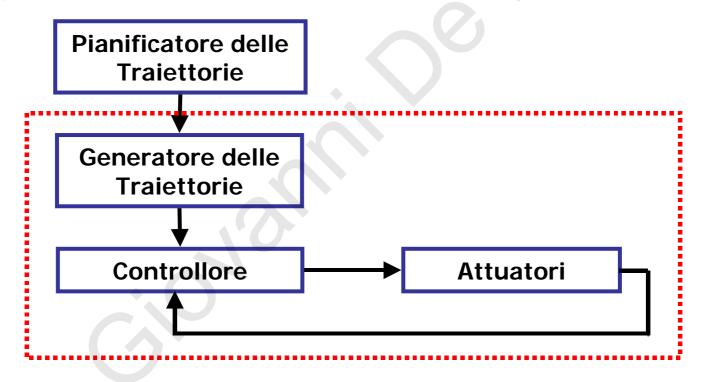
Cammino generato per raggiungere il punto finale G partendo dal punto iniziale S:

- 1) segui la parete sulla destra;
- 2) cambia parete sulla destra;
- 3) segui la parete sulla destra;
- 4) cambia parete sulla destra;
- 5) entra nella porta e gira a sinistra;
- 6) segui la parete sulla sinistra;
- 7) ferma quando il punto di

interesse è raggiunto.

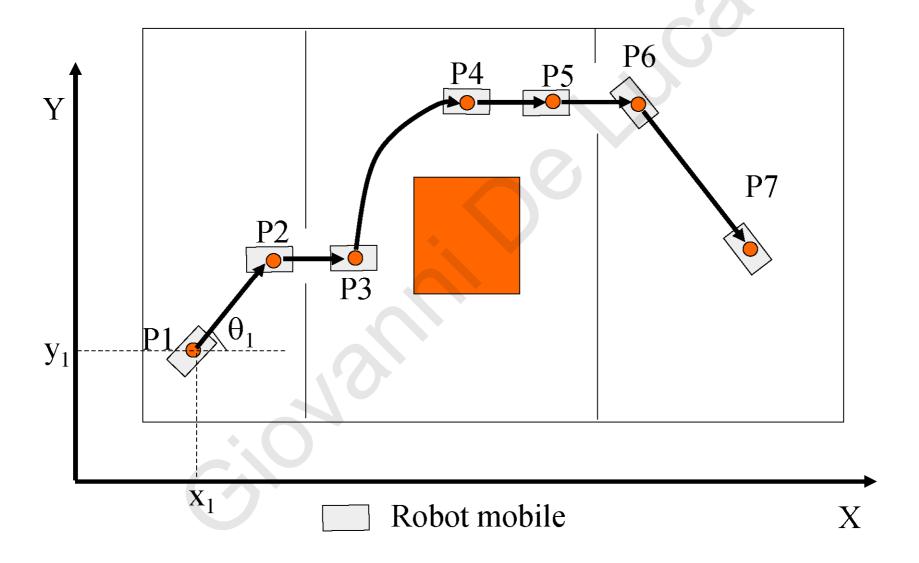


Si occupa di far eseguire al robot le traiettorie generate dal Path Planner.

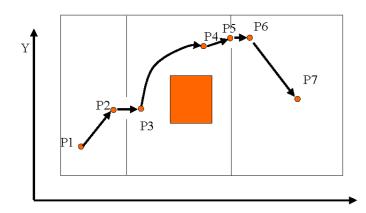


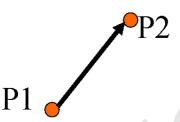
- Pianificatore delle traiettorie: determina una sequenza di punti nello spazio che il robot deve visitare (percorso)
 - $(x_1,y_1\theta_1), ... (x_{i-1},y_{i-1}\theta_{i-1}), (x_i,y_i\theta_i), ... (x_n,y_n\theta_n)$
- Generatore delle traiettorie: determina la traiettoria e i tempi che il robot deve seguire tra ogni coppia di punti (non necessariamente la traiettoria è lineare)
- Controllore: fa eseguire al robot la traiettoria determinata dal generatore delle traiettorie

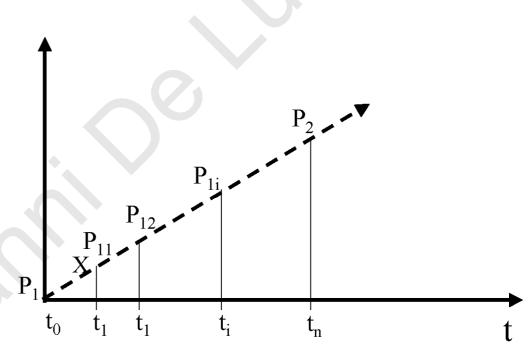
Pianificatore delle traiettorie



Generatore delle traiettorie

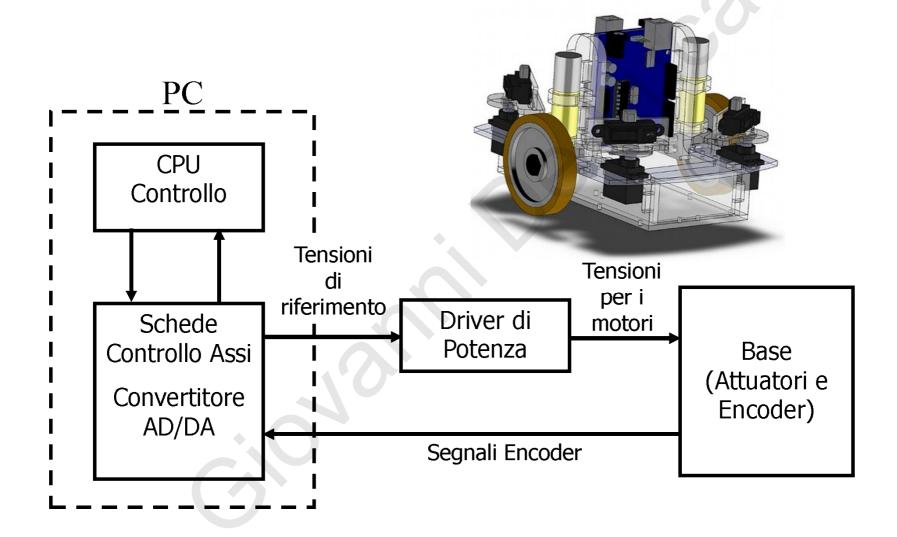






Traiettoria generata dal generatore delle traiettorie

Architettura HW di una base mobile



Controllore

Controllo degli attuatori:

- Controllo in velocità: consiste nell'impostare una velocità e una accelerazione ai motori delle ruote.
- Controllo in posizione: consiste nell'impostare una posizione da raggiungere.
 Il controllore del robot esegue i calcoli delle velocità e
 - delle accelerazioni da impostare ai motori per raggiungere la posizione voluta (cinematica inversa).

Controllore

- Controllo in posizione: Il controllore del robot esegue i calcoli delle velocità e delle accelerazioni da impostare ai motori per raggiungere la posizione voluta nel tempo desiderato
- Determinare le correnti (Volt) da applicare ai motori per far assumere ai motori velocità e accelerazioni calcolate dalla cinematica inversa

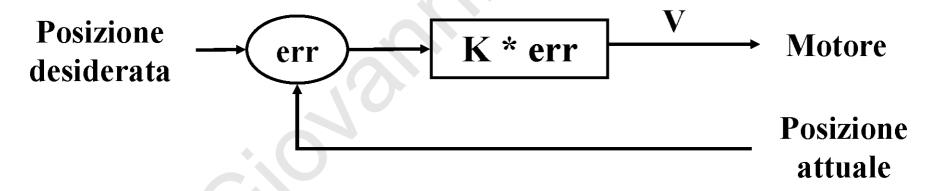
Controllore

 Encoder: sensore che misura la rotazione dei giunti in valore relativo

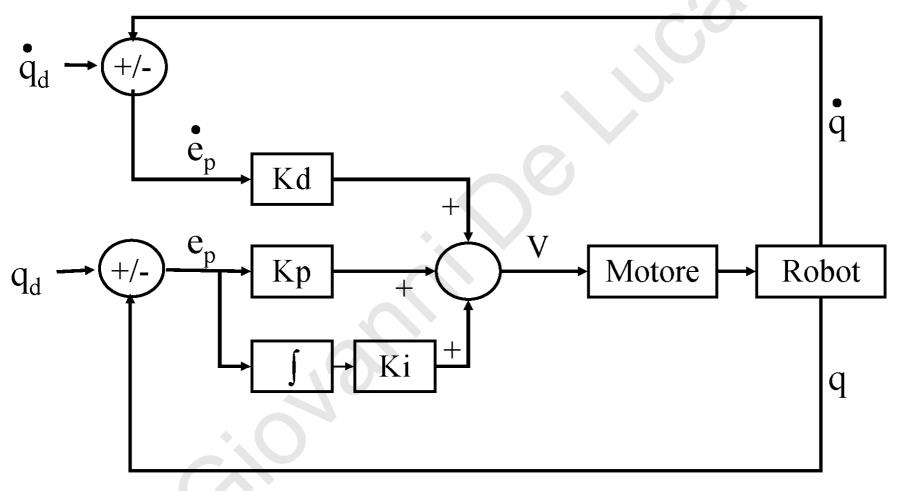
 Calcolare le tensioni ai motori affinchè l'encoder rilevi q_{next} – q_{act} gradi nell'unità di tempo dt

Controllo in posizione

 La tensione imposta al motore è proporzionale alla differenza tra la posizione effettiva misurata dal sensore e la posizione voluta



Il controllo PID



$$V = K_p e_p + K_d e_p + K_i \int E_p$$

Non sempre è possibile seguire la traiettoria generata dal Path Planner

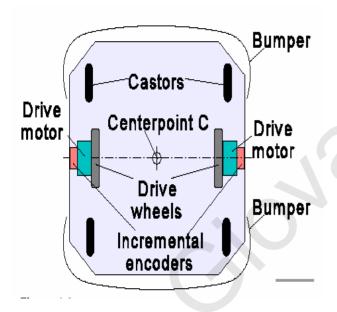
- Problematiche da affrontare:
 - Basi non omnidirezionali
 - Controllo degli attuatori
 - □ Ostacoli imprevisti

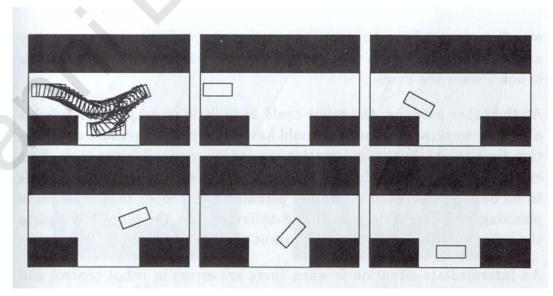
Base omnidirezionale:

- può muoversi in qualsiasi direzione.
- può seguire la traiettoria generata dal Path Planner.

Base non omnidirezionale:

- non può muoversi in qualsiasi direzione a causa della sua struttura (es. car-like robot)
- non sempre può seguire la traiettoria generata dal Path Planner.



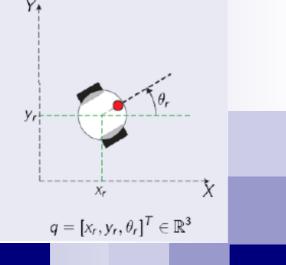


Controllo degli attuatori:

- Controllo in velocità: consiste nell'impostare una velocità e una accelerazione ai motori delle ruote.
- Controllo in posizione: consiste nell'impostare una posizione da raggiungere.
 Il controllore del robot esegue i calcoli delle velocità e delle accelerazioni da impostare ai motori per raggiungere la posizione voluta (Cinematica inversa).

Il problema degli ostacoli imprevisti:

- Gli ostacoli imprevisti sono rilevati dal robot tramite sensori ad ultrasuoni o laser.
- Il controllore del robot deve modificare la traiettoria da seguire per evitare gli ostacoli
- Tecniche di obstacle avoidance:
 - □ basate su occupancy grid
 - □ basate su campi di potenziale



Fine

Giovanni De Luca Laboratorio Progettazione Elettronica www.delucagiovanni.com

