

Università di Pisa  
Corso di Laurea in Informatica  
**Corso di Percezione Robotica**



## C. Modulo di Robotica

### **Basi di Navigazione Robotica**

Cecilia Laschi  
cecilia.laschi@sssup.it

# Sommario della Lezione

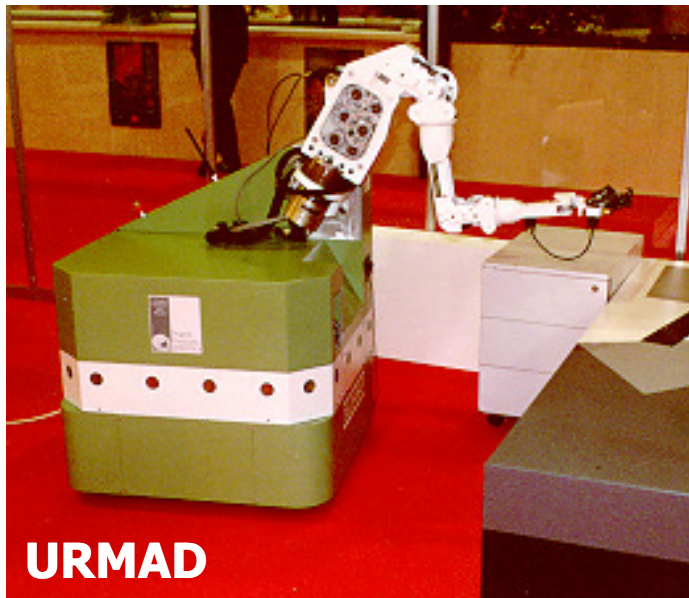


- I problemi fondamentali della Navigazione Robotica
- Mappe e modelli dell'ambiente
  - Mappe Metriche e Mappe Topologiche
- Tecniche di Planning
  - Path Planning e Path Following
- Metodi e Sistemi di localizzazione
  - Odometria e sistemi di localizzazione basati su boe attive e landmark

# Basi Robotiche Mobili



**Movaidd**



**URMAD**



**DustCart, SSSA**



**B21r – Real Word Interface, USA**



**Pioneer I – Real Word Interface, USA**

# Sommario della Lezione



- I problemi fondamentali della Navigazione Robotica
- Mappe e modelli dell'ambiente
  - Mappe Metriche e Mappe Topologiche
- Tecniche di Planning
  - Path Planning e Path Following
- Metodi e Sistemi di localizzazione
  - Odometria e sistemi di localizzazione basati su boe attive e landmark

# 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:

- **Dove sono?**
- **Dove sono gli altri oggetti rispetto a me?**
- **Come posso raggiungere una certa posizione?**

# L'approccio classico al problema

Nell'approccio classico, le risposte alle domande precedenti sono:

- **Localizzazione:** posizione geometrica (coordinate  $X, Y$  e  $\theta$  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

# Sommario della Lezione



- I problemi fondamentali della Navigazione Robotica
- Mappe e modelli dell'ambiente
  - Mappe Metriche e Mappe Topologiche
- Tecniche di Planning
  - Path Planning e Path Following
- Metodi e Sistemi di localizzazione
  - Odometria e sistemi di localizzazione basati su boe attive e landmark



# Mappe e Rappresentazione del mondo



- **Mappe Metriche (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

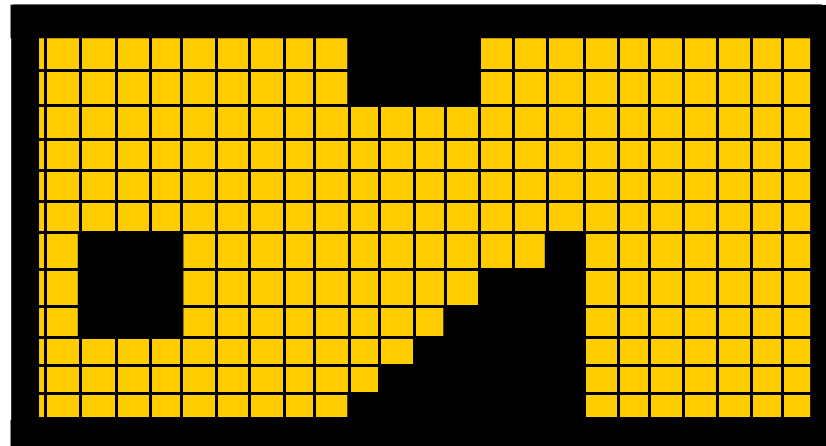


Principali metodi per la rappresentazione dell'ambiente tramite mappe metriche:

- Occupancy grid
- Descrizione geometrica

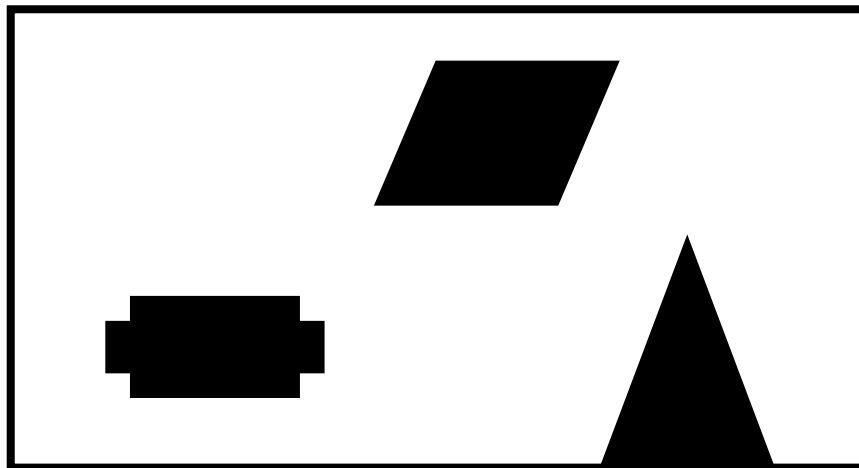
# Occupancy Grid

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



# Descrizione geometrica

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



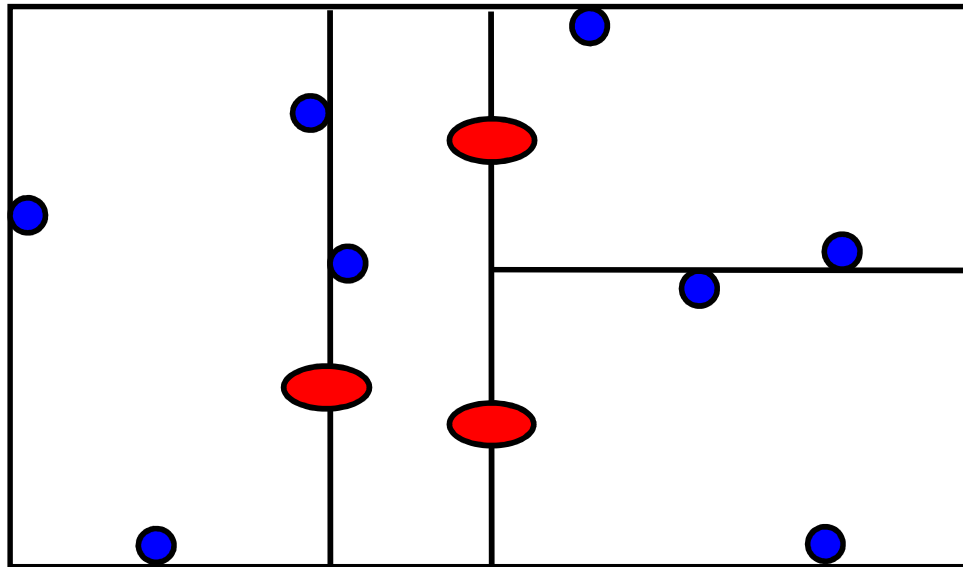
# Mappe Topologiche



- 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



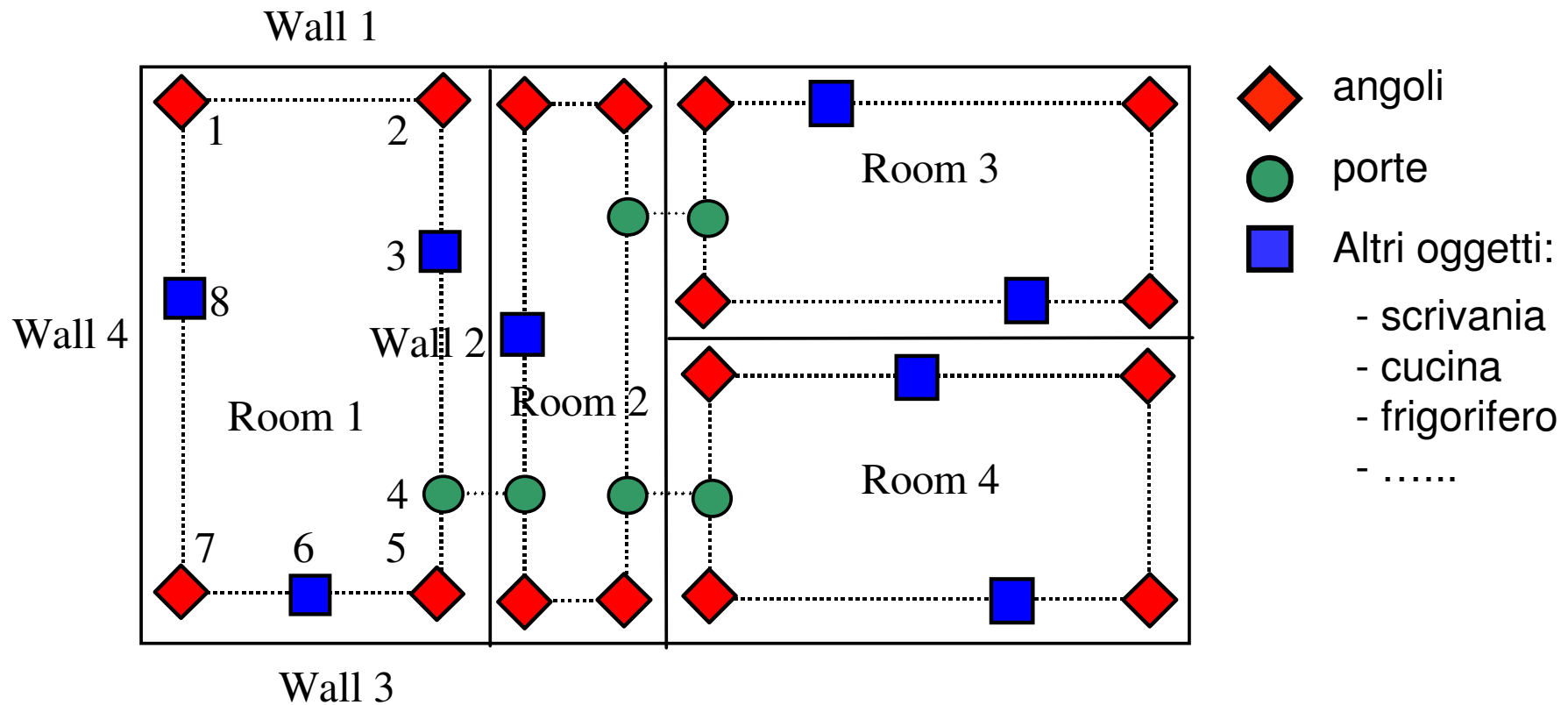
-  **Porta**
-  **Punti di interesse:**
  - Scrivania
  - Letto
  - Frigorifero
  - .....

# 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



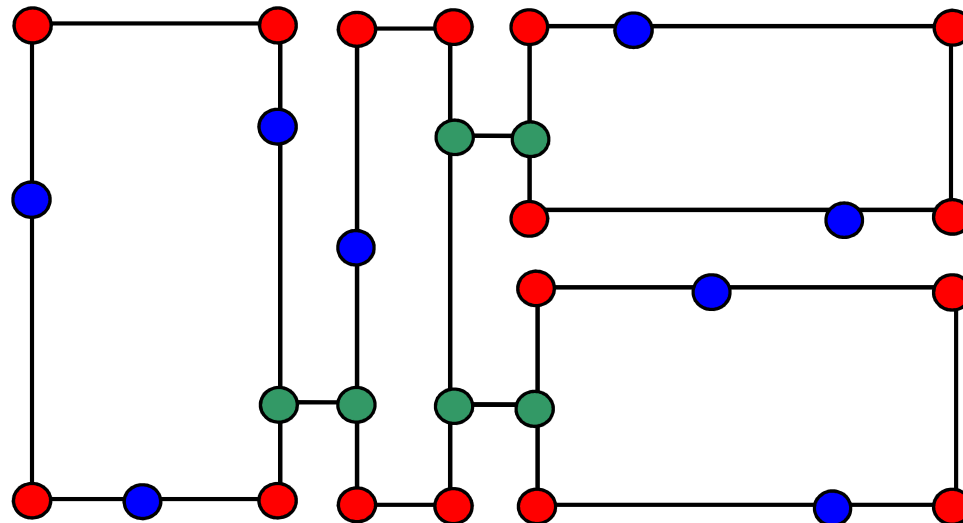


# Rappresentazione di una mappa topologica tramite grafo

$$G = (N, E)$$

$$N = \{\text{punti di interesse}\}$$

$$E = \{(p, q) \mid (p \in N, q \in N, p = q \pm 1) \vee (p \text{ e } q \text{ rappres. la stessa porta per due stanze diverse})\}$$



# Sommario della Lezione



- I problemi fondamentali della Navigazione Robotica
- Mappe e modelli dell'ambiente
  - Mappe Metriche e Mappe Topologiche
- Tecniche di Planning
  - Path Planning e Path Following
- Metodi e Sistemi di localizzazione
  - Odometria e sistemi di localizzazione basati su boe attive e landmark

# Tecniche di Planning e modelli dell'ambiente



L'obiettivo 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  $C_{\text{space}}$  (configurazioni assumibili dal robot nello spazio)
- Il robot è rappresentato in  $C_{\text{space}}$  con un punto
- Gli ostacoli sono rappresentati in  $C_{\text{space}}$
- La regione degli ostacoli è chiamata  $C_{\text{obstacle}}$
- La regione dello spazio libero è chiamata  $C_{\text{free}}$
- Un path è una traiettoria tra due configurazioni  $q_{\text{init}}$  e  $q_{\text{goal}}$  di  $C_{\text{space}}$  contenuta in  $C_{\text{free}}$

# 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
- **Free Way Net**
- **Silhouette**



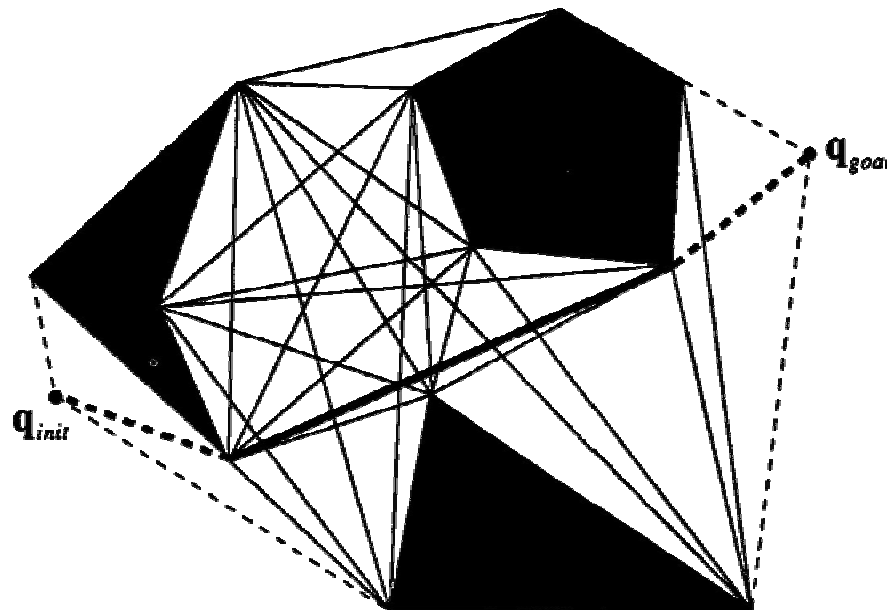
# Visibility Graph



- 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

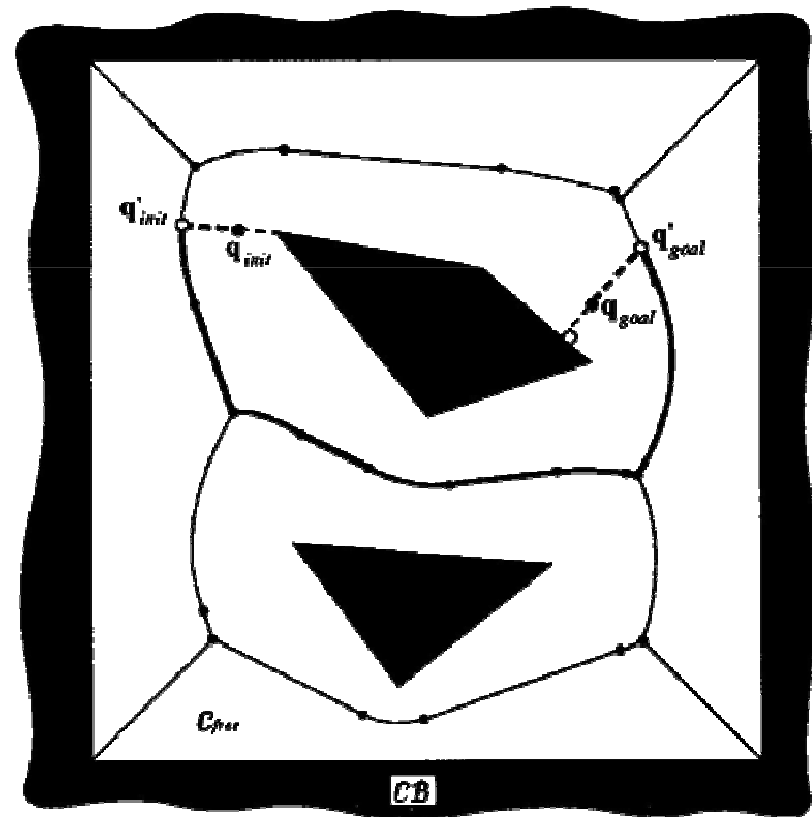


Dati due configurazioni iniziali  $q_{init}$  e  $q_{goal}$ , un percorso è determinato:

- 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

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



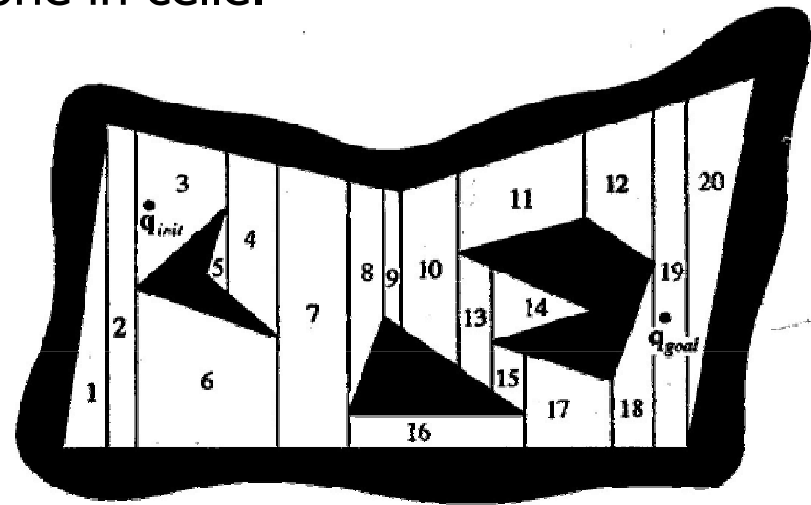
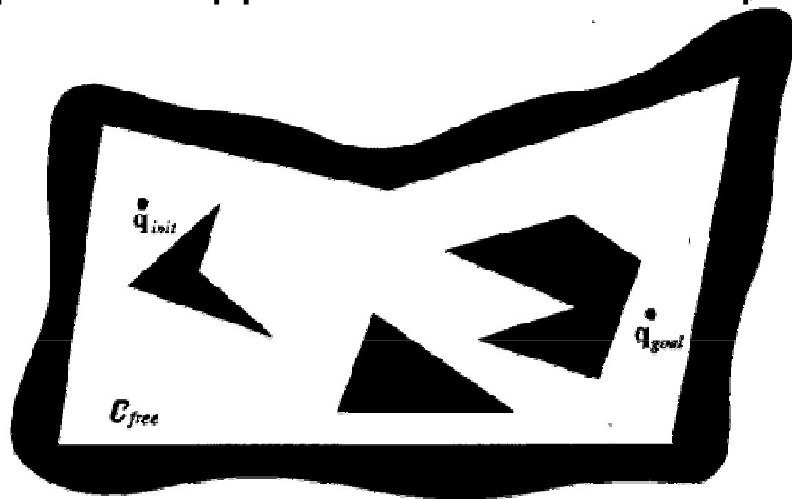
# Cell Decomposition



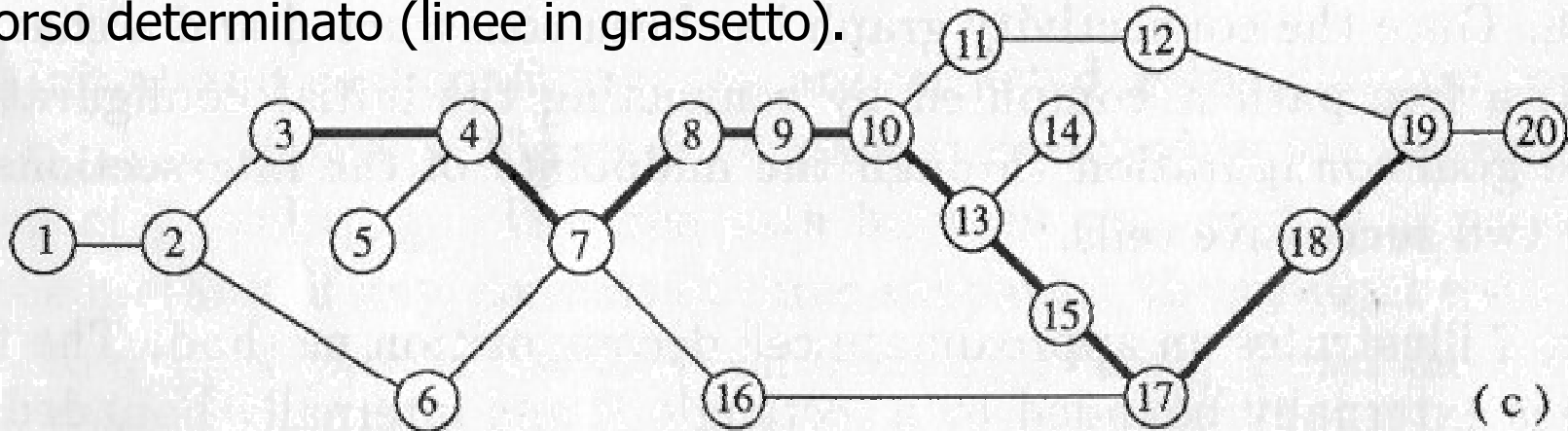
- Consiste nel decomporre lo spazio libero del robot in regioni, dette celle, tali che un percorso tra due celle adiacenti può essere facilmente determinato.
- La mappa è rappresentata tramite un grafo detto **connectivity graph**.
- I nodi del grafo sono costituiti dalle celle estratte dallo spazio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

# Cell Decomposition

Esempio di mappa e della sua decomposizione in celle.



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



(c)

# Cell Decomposition



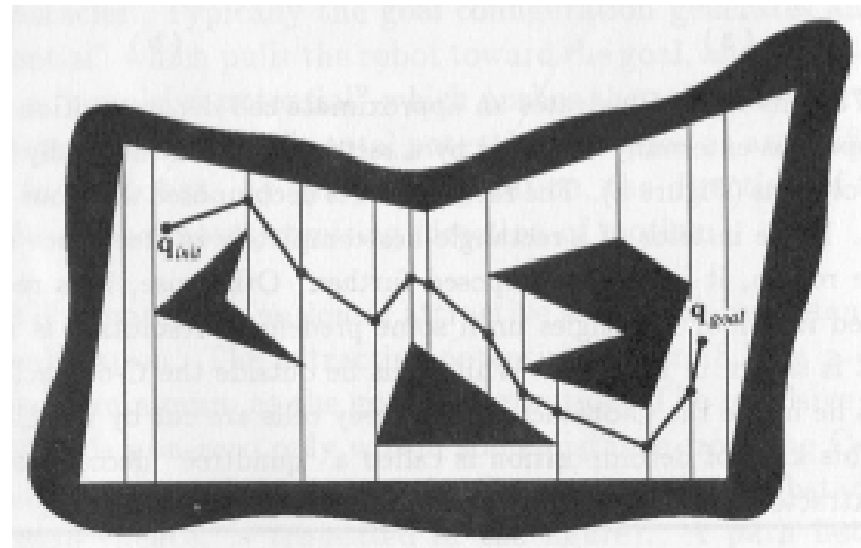
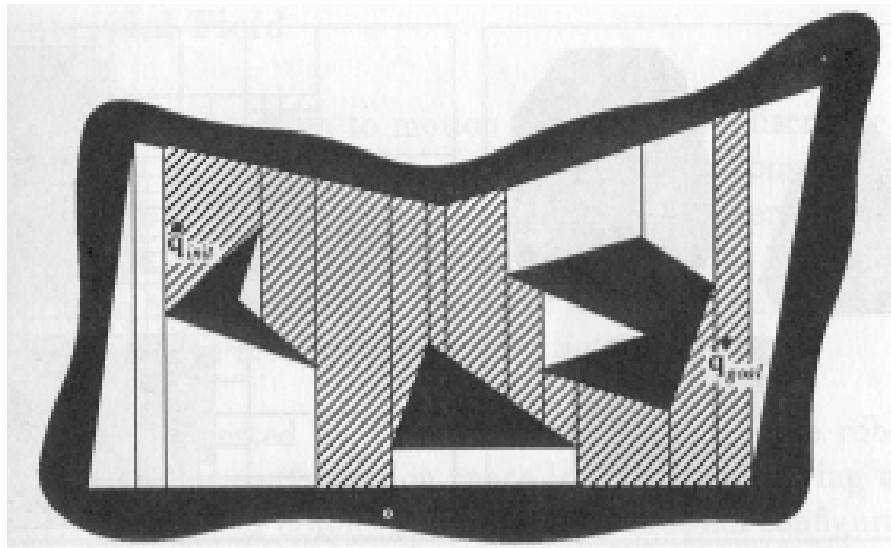
- Una traiettoria per il robot è determinata cercando un cammino sul grafo congiungente i nodi contenenti i punti  $q_{init}$  e  $q_{goal}$
- Il risultato della visita del grafo è una sequenza di celle chiamata **canale**
- Il percorso è determinato congiungendo i punti intermedi dei lati adiacenti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



# Cell Decomposition

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'obiettivo.
- 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.

# Campi di Potenziale

- Funzione Potenziale Differenziabile  $U$  con un minimo locale nel punto  $q_{\text{goal}}$

$$U(q) = U(q)_{\text{att}} + U(q)_{\text{rep}}$$

$U(q)_{\text{att}}$  Funzione Potenziale Attrattiva

$U(q)_{\text{rep}}$  Funzione Potenziale Repulsiva

- Per ogni punto  $q$  dello spazio la direzione di movimento è data dalla funzione  $F$

$$F(q) = -\nabla U(q) = -(F_{\text{att}}(q) + F_{\text{rep}}(q))$$

$$\nabla U(q) = (\delta U / \delta x, \delta U / \delta y)$$

# Campi di Potenziale

## Criteri per la scelta del Potenziale Attrattivo

- Funzione con un minimo locale nel punto  $q_{\text{goal}}$

$$U_{\text{att}}(q) = \frac{1}{2} \xi \rho_{\text{goal}}^2(q) \quad \text{pot. parabolico}$$

dove

$$\rho_{\text{goal}}^2(q) = \|q - q_{\text{goal}}\|^2 \quad \text{distanza euclidea}$$

e quindi

$$F_{\text{att}}(q) = -\xi (q - q_{\text{goal}})$$

# Campi di Potenziale

## Criteri per la scelta del Potenziale Repulsivo

- Creare una barriera protettiva attorno alla regione degli ostacoli che eviti il contatto del robot con essi
- La forza repulsiva non deve incidere sul moto del robot quando esso è lontano dagli ostacoli

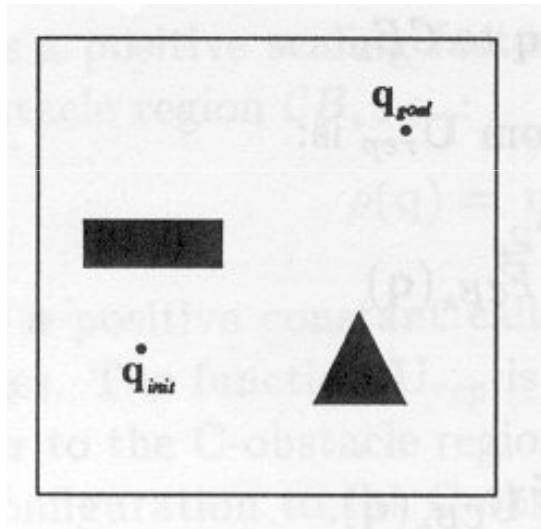
$$U_{\text{rep}}(q) = \begin{cases} \frac{1}{2} \eta (1/\rho(q) - 1/\rho_0)^2 & \text{se } \rho(q) \leq \rho_0 \\ 0 & \text{se } \rho(q) > \rho_0 \end{cases}$$

dove

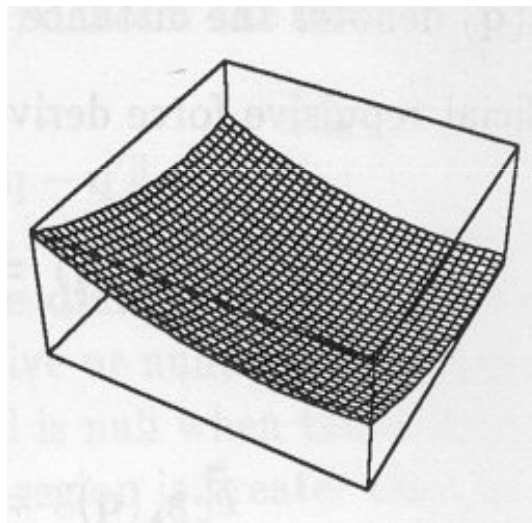
$$\rho(q) = \min \|q - q'\| \quad q' \in C_{\text{obstacle}}$$

# Campi di Potenziale

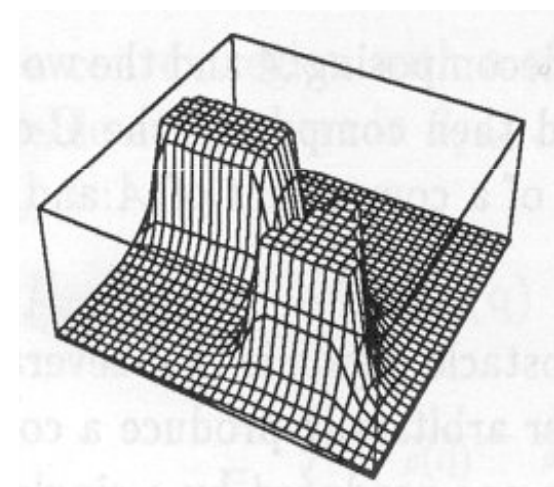
Esempio di potenziale attrattivo e repulsivo



La mappa dell'ambiente

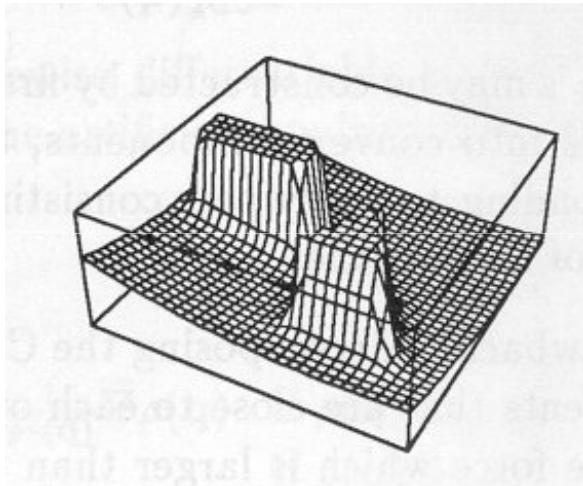


La funzione potenziale attrattivo iperbolica



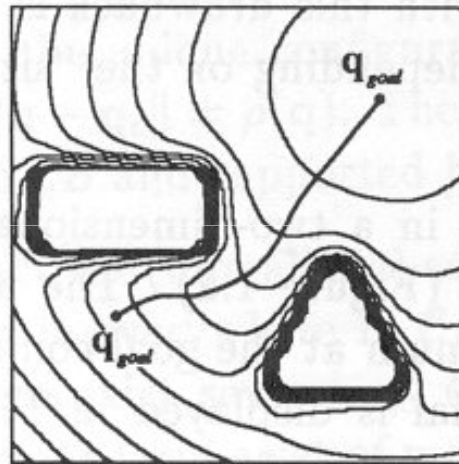
La funzione potenziale repulsivo

# Campi di Potenziale

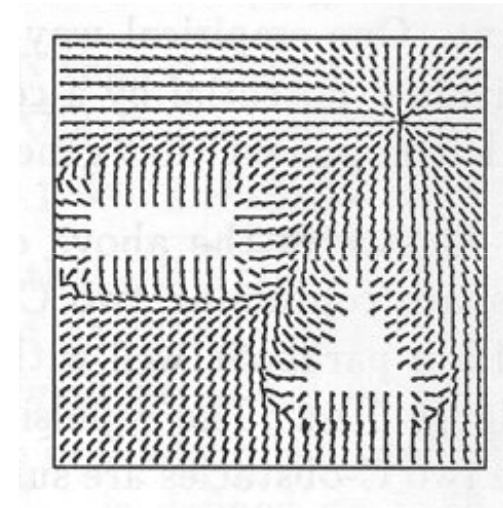


La Funzione  
Potenziale Totale

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



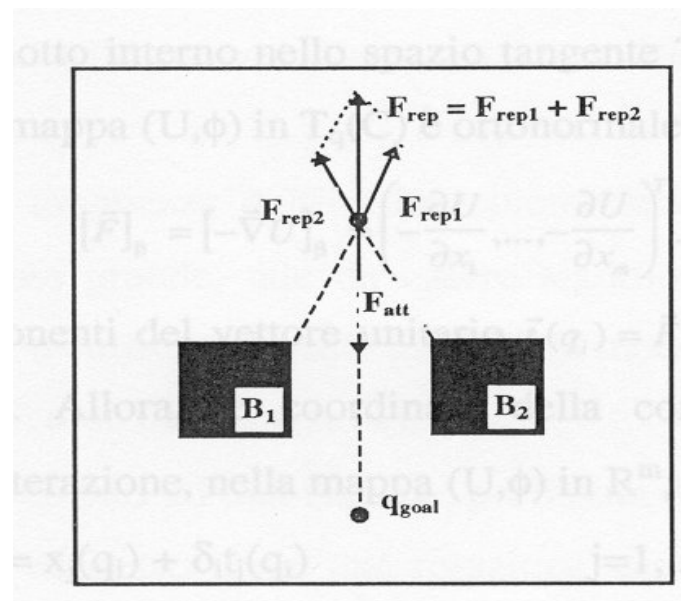
Le Curve della  
Funzione Totale e  
la Traiettoria  
Generata



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  $q_{\text{goal}}$





# Path Planning per Mappe Topologiche

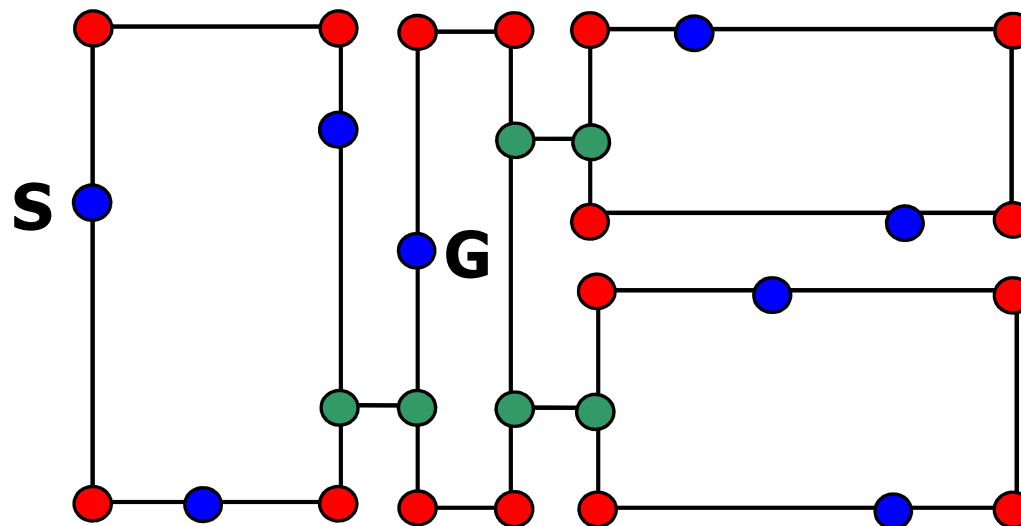


Esempio di cammino per una mappa topologica:

- 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;

# Path Planning per Mappe Topologiche

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



# Path Planning per Mappe Topologiche

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.

# Path Planning per Mappe Topologiche

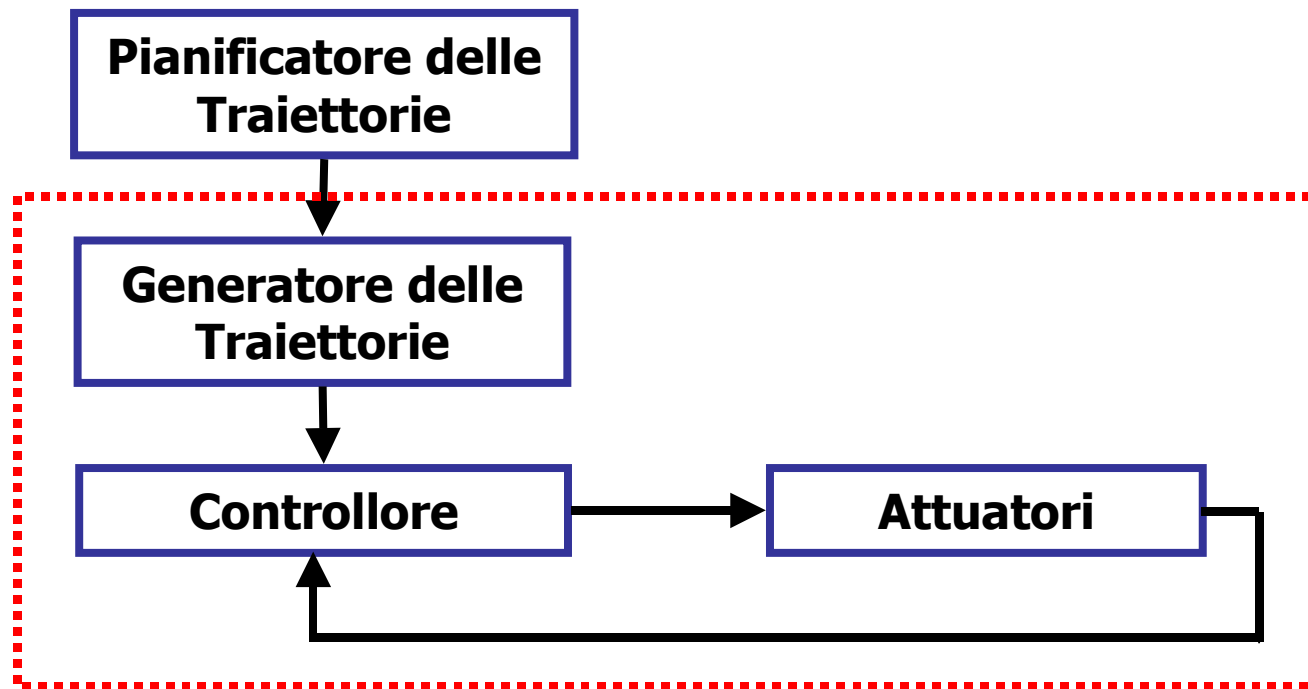
## Regole per la traduzione di un cammino in una sequenza di macro comandi:

- 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 dritto** 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**.



# Path Following

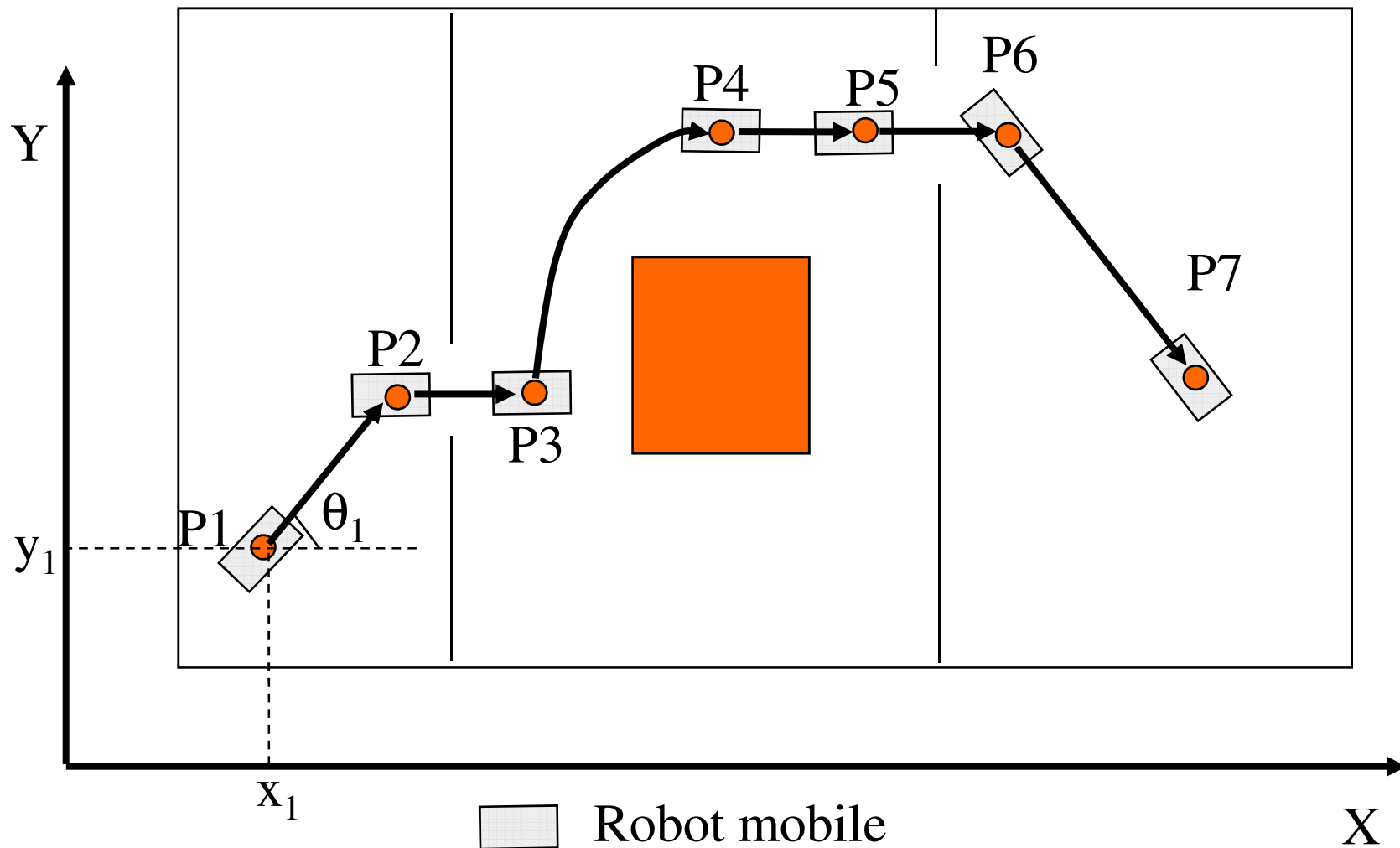
- Si occupa di far eseguire al robot le traiettorie generate dal Path Planner



# Path Following

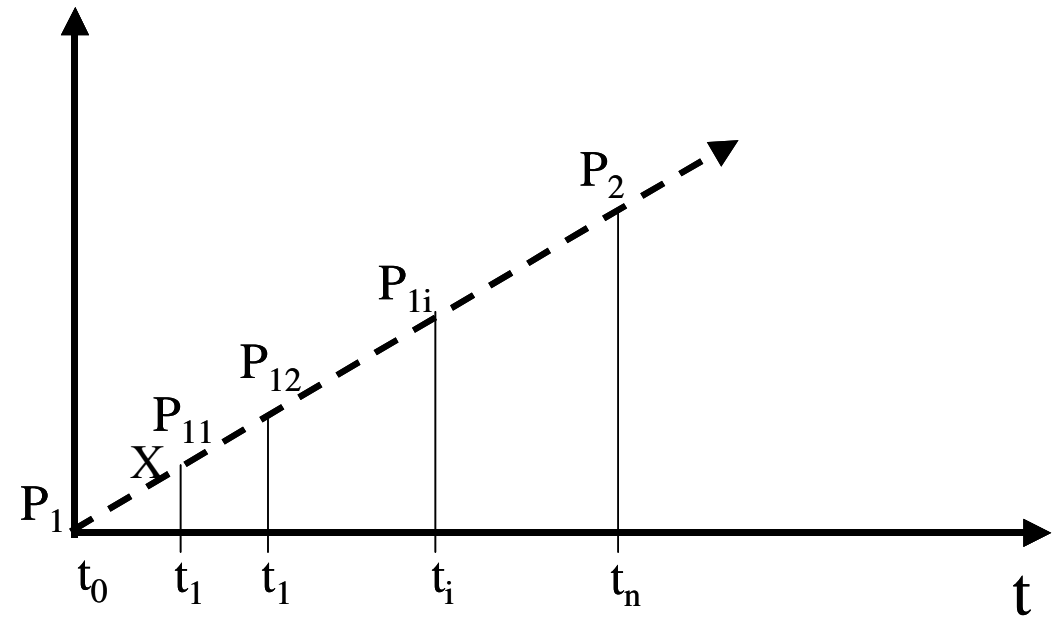
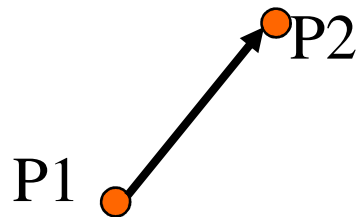
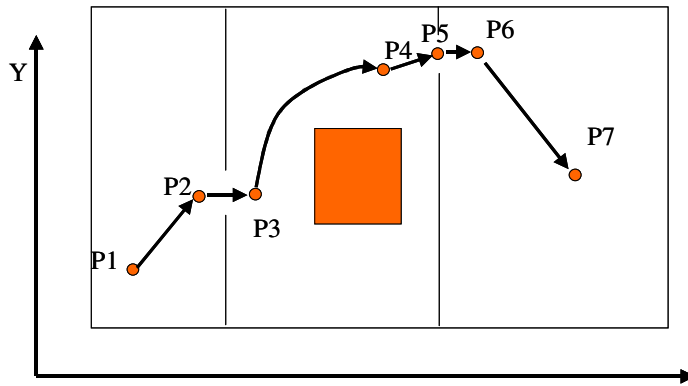
- **Pianificatore delle traiettorie:** determina una sequenza di punti nello spazio che il robot deve visitare (percorso)
  - $(x_1, y_1, \theta_1), \dots, (x_{i-1}, y_{i-1}, \theta_{i-1}), (x_i, y_i, \theta_i), \dots, (x_n, y_n, \theta_n)$
- **Generatore delle traiettorie:** determina la traiettoria 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

# 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

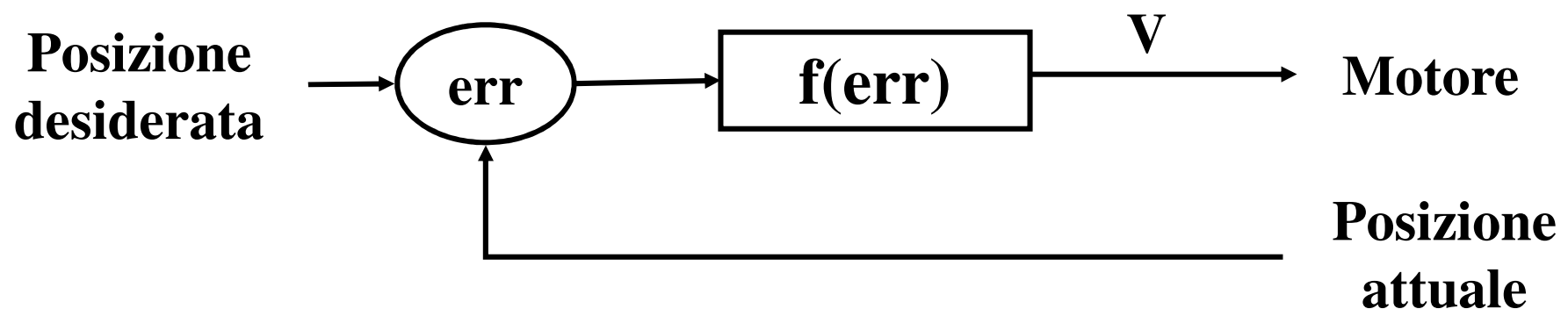
# Controllore



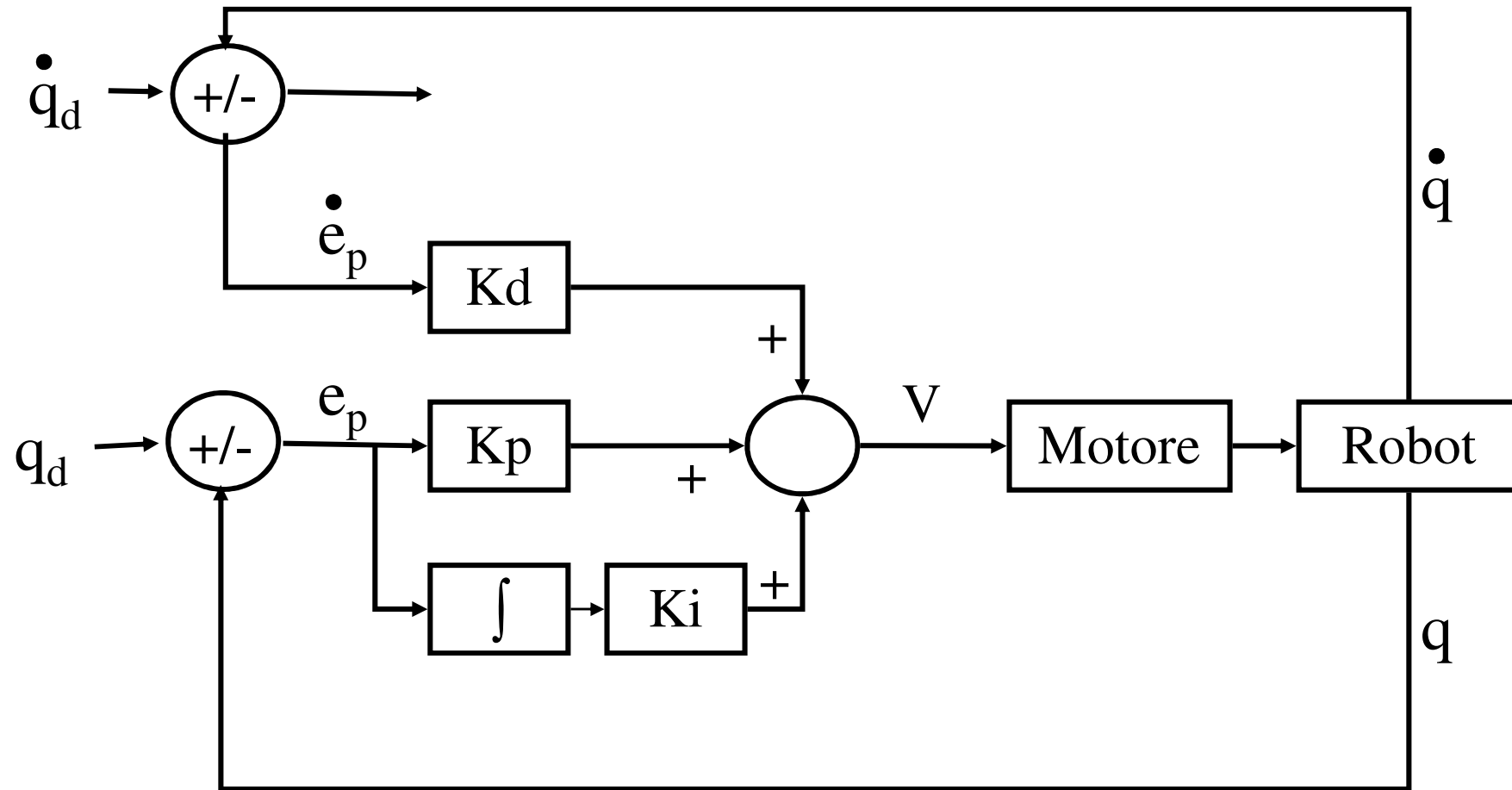
- **Encoder:** sensore che misura la rotazione dei giunti in valore relativo
- Calcolare le tensioni ai motori affinché l'encoder rilevi  $q_{\text{next}} - q_{\text{act}}$  gradi nell'unità di tempo  $dt$

# Controllo in posizione

- La tensione imposta al motore è funzione della differenza tra la posizione effettiva misurata dal sensore e la posizione voluta




# Il controllo Proporzionale, Integrativo e Derivativo (PID)



$$V = K_p e_p + K_d \dot{e}_p + K_i \int e_p$$

# Path Following



- Non sempre è possibile seguire la traiettoria generata dal Path Planner
- Problematiche da affrontare:
  - Basi non omnidirezionali
  - Ostacoli imprevisti

# Path Following



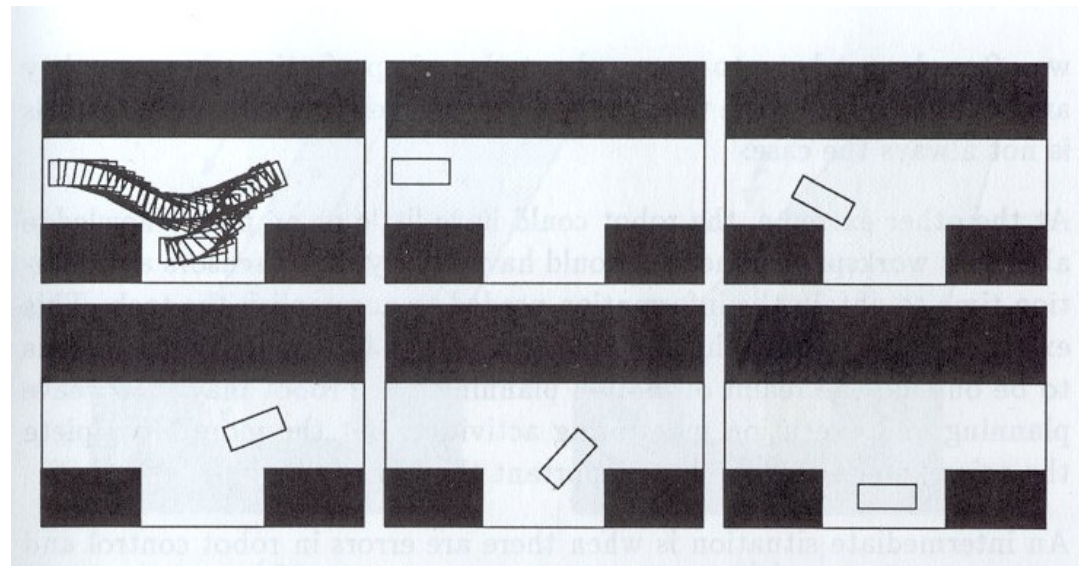
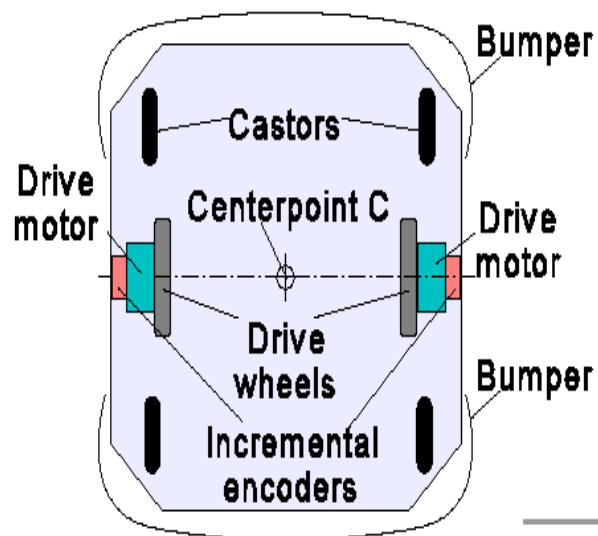
- **Base omnidirezionale:**
  - può muoversi in qualsiasi direzione
  - può seguire la traiettoria generata dal Path Planner



# Path Following

- **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



# Path Following – obstacle avoidance

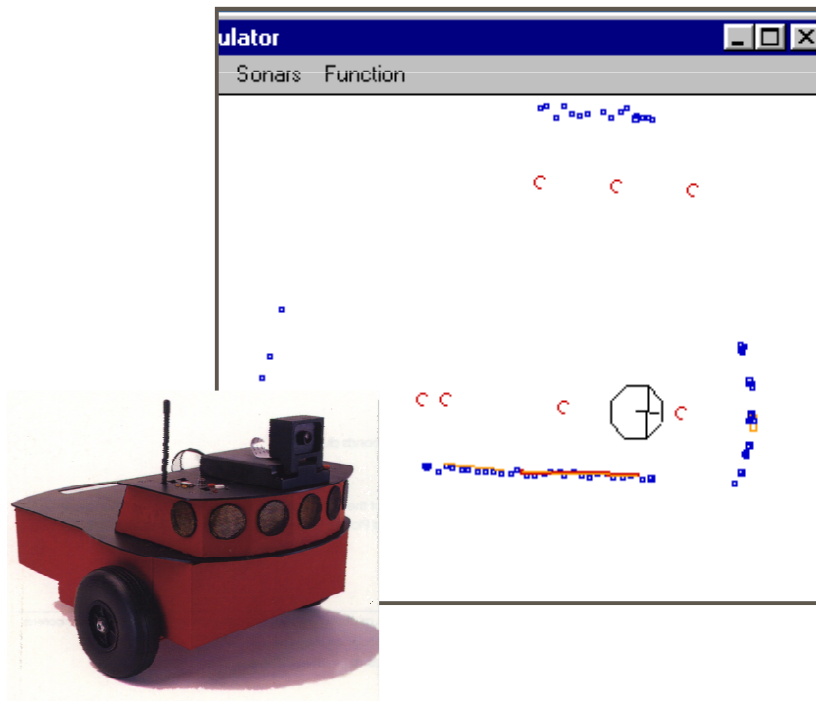


Il problema degli ostacoli imprevisti:

- 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

# Sistemi per la rilevazione di ostacoli

- Sistemi Sensoriali Utilizzati:  
Sensori ad Ultrasuoni



## Laser Ranger



# Sommario della Lezione



- I problemi fondamentali della Navigazione Robotica
- Mappe e modelli dell'ambiente
  - Mappe Metriche e Mappe Topologiche
- Tecniche di Planning
  - Path Planning e Path Following
- **Metodi e Sistemi di localizzazione**
  - **Odometria e sistemi di localizzazione basati su boe attive e landmark**

# Localizzazione



## Metodi di localizzazione

- Dead Reckoning - Odometria
- Boe Attive
- Landmark Naturali e Artificiali

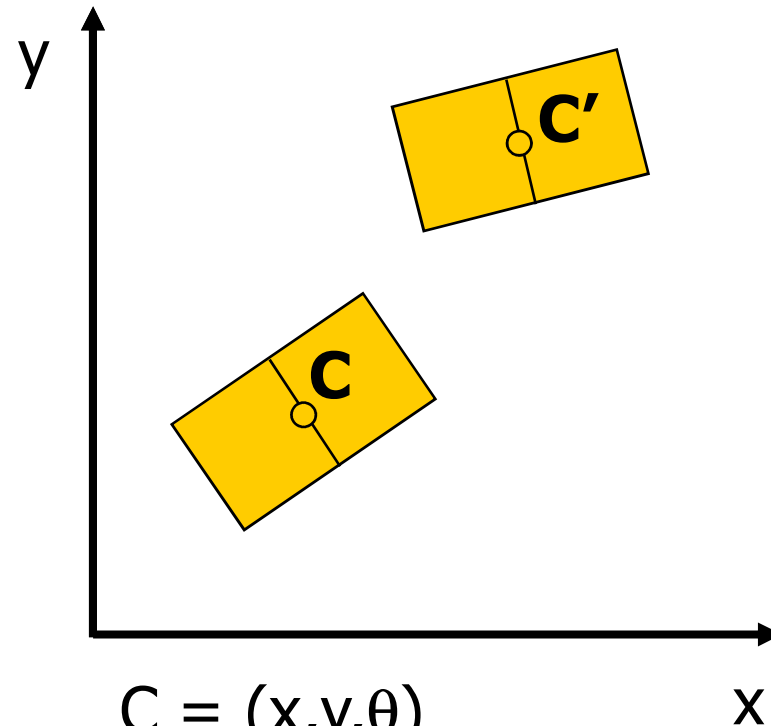
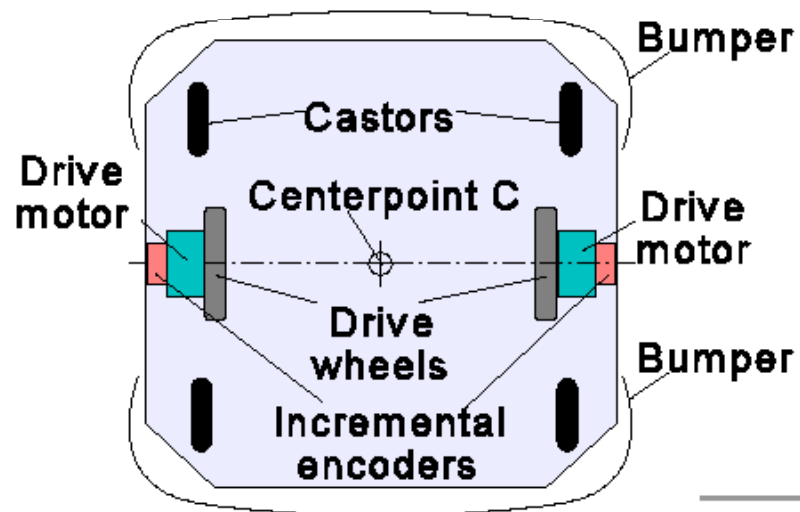
# Odometria - Dead Reckoning



- 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.

# Odometria - Dead Reckoning

Esempio di calcolo dell'odometria (per piccoli spostamenti)



$$C = (x, y, \theta)$$

$$C' = (x', y', \theta')$$

# Odometria - Dead Reckoning

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

$$\mathbf{C}_m = \pi \mathbf{D}/n \mathbf{C}_e$$

dove

$\mathbf{C}_m$  = fattore di conversione che traduce gli impulsi lineari in distanza lineare effettuata delle ruote

$\mathbf{D}$  = diametro nominale delle ruote

$\mathbf{C}_e$  = risoluzione degli encoder

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



# Odometria - Dead Reckoning

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

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

Allora, la distanza percorsa dal centro del robot,  $\Delta U_i$  risulta essere:

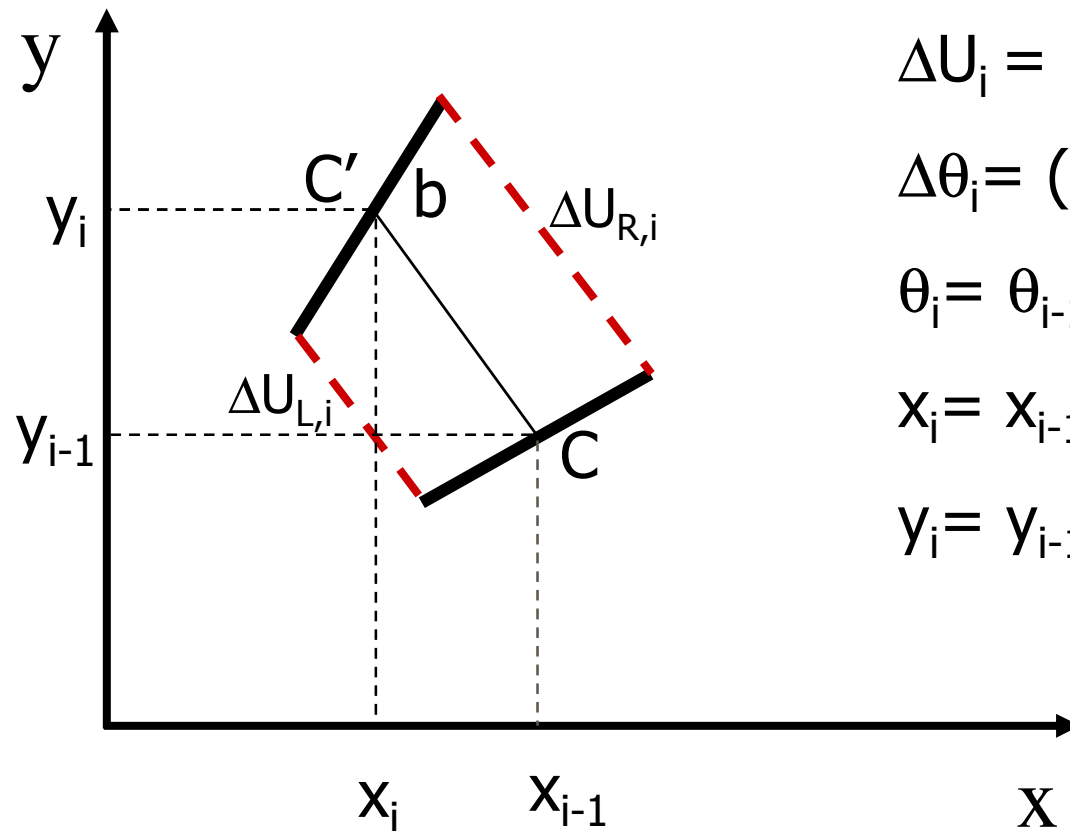
$$\Delta U_i = (\Delta U_{L,i} + \Delta U_{R,i})/2$$

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 ruote con il pavimento)

# Odometria - Dead Reckoning



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

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

$$\Delta \theta_i = (\Delta U_{R,i} - \Delta U_{L,i})/b$$

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

$$x_i = x_{i-1} + \Delta U_i \cos \theta_i$$

$$y_i = y_{i-1} + \Delta U_i \sin \theta_i$$

# Odometria - Dead Reckoning



Gli errori odometrici sono di due tipi:

- **Errori sistematici**, causati da:
  - diametri differenti delle due ruote
  - la dimensione reale delle ruote è diversa dalla dimensione nominale
  - disallineamento delle ruote
  - risoluzione finita 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)

# Boe Attive

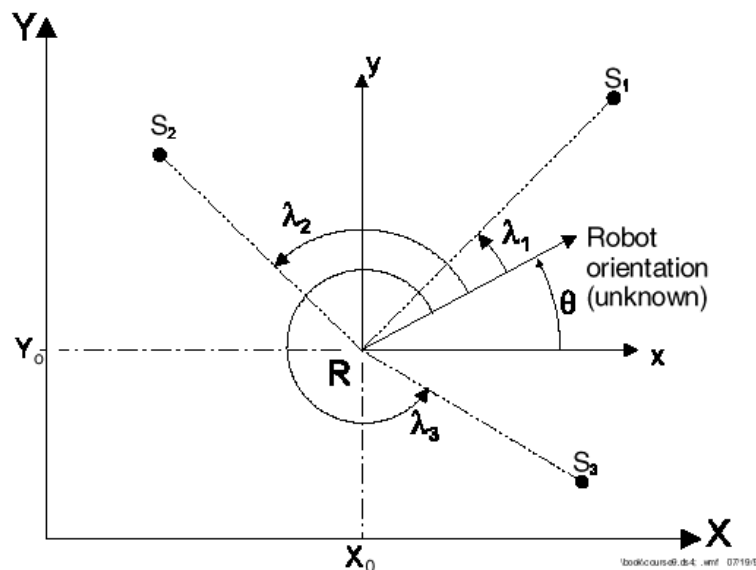


- I sistemi di localizzazione con boe attive sono costituiti da un insieme di dispositivi ricevitori/trasmittitori (boe o beacon) di cui è conosciuta la posizione assoluta nell'ambiente e che sono rilevabili da un sensore trasmettitore/ricevitore posto sul robot

# Boe Attive

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

- L'una unità posta a bordo del robot è in grado di misurare gli angoli  $\lambda_1, \lambda_2, \lambda_3$  a cui si rilevano 3 boe
- Conoscendo la posizione delle 3 boe è possibile determinare tramite triangolazione la posizione assoluta del robot  $(X, Y, \theta)$



# Sistemi di Localizzazione basati sulle mappe

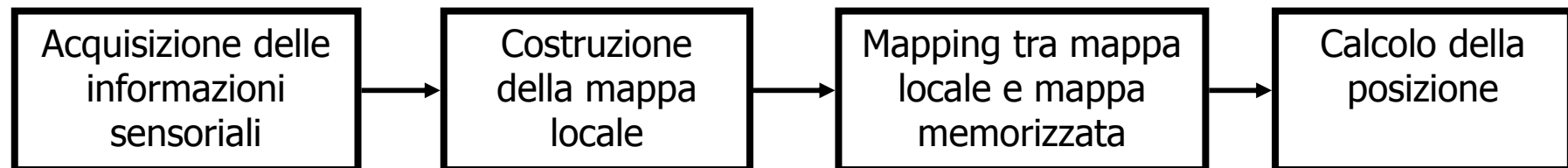


- I sistemi di localizzazione basati sulle mappe, conosciuti anche come *map matching*, utilizzano uno o più sistemi sensoriali per costruire una mappa locale.
- La mappa locale è poi confrontata con una mappa globale precedentemente memorizzata.
- Se un matching viene trovato, il robot calcola la sua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

# Sistemi di Localizzazione basati sulle mappe

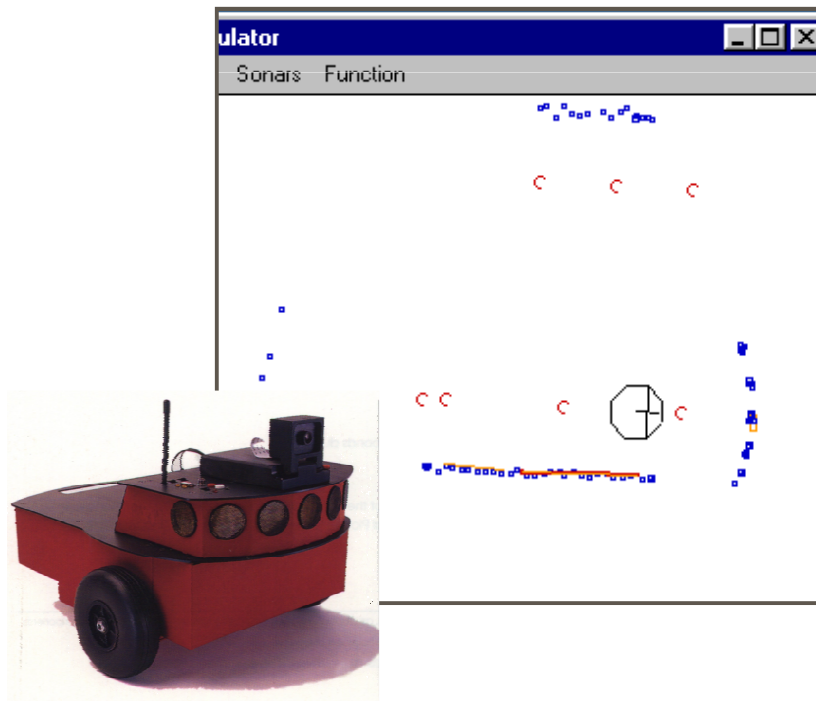
Procedura di localizzazione:

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



# Sistemi di Localizzazione basati sulle mappe

- Sistemi Sensoriali Utilizzati:  
Sensori ad Ultrasuoni



## Laser Ranger

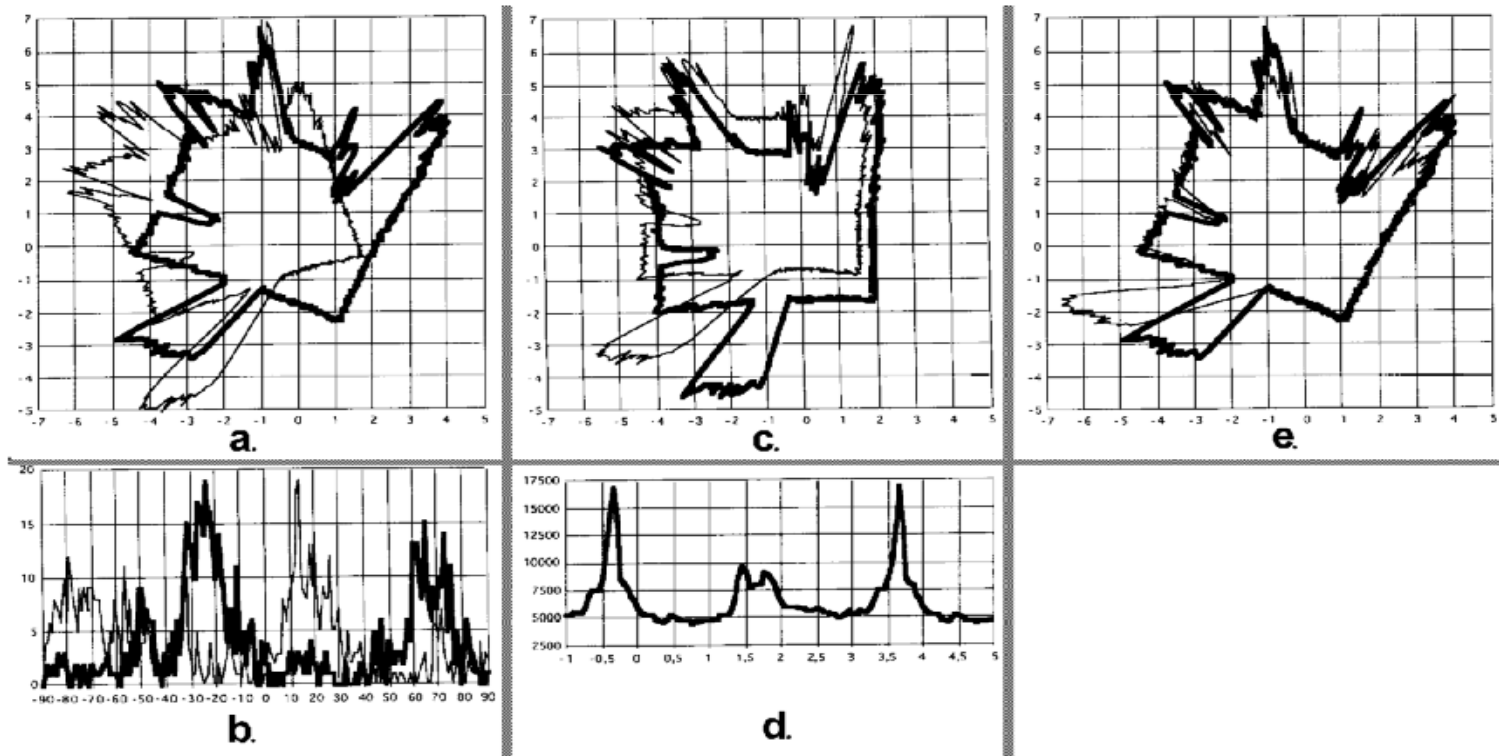




# Sistemi di Localizzazione basati sulle mappe

Tecniche di mapping utilizzate:

- Correlazione



# Landmark



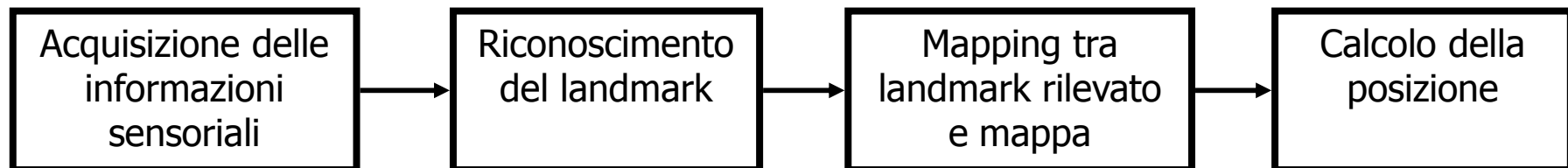
- I Landmark sono delle forme caratteristiche che il robot può riconoscere utilizzando i propri sistemi sensoriali.
- I Landmark possono essere forme geometriche (es. rettangoli, linee, cerchi, ..) e possono contenere informazioni aggiuntive (es. Bar-code).
- I Landmark sono scelti in modo tale da essere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

# Landmark



Procedura di localizzazione:

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



# 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 bianche sugli angoli
  - cerchio metà nero e metà bianco
  - LED attivi
  - infrarossi