

# Appunti Di Robotica Autonoma

Dalle lezioni di Emanuele Menegatti (Unipd)

February 20, 2019

# Contents

<b>1</b>	<b>Per Esame</b>	<b>4</b>
<b>2</b>	<b>Locomozione</b>	<b>5</b>
2.1	Locomozione con Gambe . . . . .	5
2.2	Locomozione con Ruote . . . . .	5
2.3	Gradi di Libertà . . . . .	6
<b>3</b>	<b>Dinamica</b>	<b>6</b>
3.0.1	Procedimento Analitico . . . . .	7
3.0.2	Procedimento Approssimato . . . . .	7
3.1	Marcia Bipede . . . . .	8
<b>4</b>	<b>Cinematica</b>	<b>8</b>
4.1	Cinematica del Robot Mobile . . . . .	8
4.2	Cos'è la Cinematica . . . . .	8
4.3	Il Robot nel Mondo . . . . .	8
4.3.1	Sistema di Riferimento Assoluto . . . . .	8
4.3.2	Sistema di Riferimento Relativo . . . . .	9
4.3.3	Traslazione . . . . .	9
4.3.4	Rotazione . . . . .	9
4.3.5	Angoli di Eulero . . . . .	10
4.3.6	Roll-Pitch-Yaw . . . . .	10
4.3.7	I Quaternioni . . . . .	11
4.3.8	Trasformazioni Omogenee . . . . .	11
4.4	La Cinematica dei Giunti del Robot . . . . .	11
4.5	Problemi Relativi alla Cinematica . . . . .	12
4.5.1	Cinematica Diretta . . . . .	12
4.5.2	Cinematica Inversa . . . . .	12
4.5.3	Cinematica Differenziale . . . . .	12
4.6	Convenzioni di Denavit e Hartenberg . . . . .	13
4.7	Cinematica del Robot Mobile . . . . .	13
4.8	Controllo Cinematico . . . . .	14
4.8.1	Controllo Open Loop . . . . .	14
<b>5</b>	<b>Percezione</b>	<b>14</b>
5.1	Caratterizzazione dei Sensori . . . . .	15
5.1.1	Performance dei Sensori . . . . .	15
5.1.2	Caratteristiche per Ambienti Reali . . . . .	15
<b>6</b>	<b>Architetture Software per Robot</b>	<b>15</b>
6.1	Paradigma SPA . . . . .	15
6.2	Subsumption Architecture . . . . .	16
6.3	Layered Architecture . . . . .	16
6.4	Sistemi Behavior-Based . . . . .	16

<b>7</b>	<b>Localizzazione</b>	<b>17</b>
7.1	Localizzazione Basata su Mappe . . . . .	17
7.2	SLAM . . . . .	19

# 1 Per Esame

Il compito avrà una durata di circa 1 ora e ci saranno una decina di domande a cui rispondere.

1. Quali di queste rappresentazioni non soffrono del fenomeno di gimbal lock, anche definito come l'allineamento di due assi rotanti verso la stessa direzione. Si ricorda che tale blocco causa la perdita di un grado di libertà corrispondente all'asse bloccato:
  - Angoli di Eulero
  - Roll, Pitch, Yaw
  - Quaternioni
2. Spiegare cosa si intende per cinematica di un robot manipolatore, con particolare attenzione alla distinzione tra cinematica diretta e cinematica inversa.
3. Quali paradigmi per la programmazione di architetture software per gestire il comportamento di un robot conosci?
  - ...
  - ...
  - ...
4. Spiegare il significato del termine *robot olonomo*.
5. Spiegare il significato del termine *subsumption architecture*.
6. In robotica cosa significa *SLAM*? Definire lo *SLAM problem*.
7. Quali delle seguenti affermazioni sono vere per un *Fiducial Marker*?
  - Incorpora molte informazioni
  - Vuole essere robusto a diverse illuminazioni e a occlusioni
  - Può essere solo di forma quadrata
8. Dare le definizioni dei seguenti termini:
  - (a) Accuracy
  - (b) Precision
  - (c) Repeatability
  - (d) Resolution
9. Cosa si intende per *localizzazione probabilistica*? Descrivere una tecnica per la localizzazione probabilistica.
10. Descrivere le architetture per robot **reactive** e **deliberative**.
11. Quanti gradi di libertà possiede il robot *synchro drive* e perché.
12. Elencare quattro tipi di sensori per la localizzazione indoor.

## 2 Locomozione

La locomozione è l'atto di muoversi da un posto ad un altro, per farlo si richiede al dispositivo di interagire fisicamente con l'ambiente circostante. La locomozione ha a che vedere con le forze, i meccanismi e gli attuatori che le generano.

Molti concetti relativi alla locomozione sono ispirati a processi naturali, anche se molto spesso è difficile imitarli. La locomozione per rotolamento non è presente in natura ma è ancora la più efficiente (soprattutto in ambienti ingegnerizzati), per questo molti sistemi utilizzano le ruote e i cingoli.

Come generare il moto?

- Progettazione manuale (per mezzo di editor di moto).
- Progettazione basata sul controllo (ad esempio, per la locomozione bipede, si può usare il modello del pendolo inverso).
- Algoritmi di apprendimento, per trovare i migliori parametri (il moto viene specificato tramite dei parametri i quali vengono determinati con algoritmi di ricerca).
- Movimenti ottenuti da dati umani (apprendimento e imitazione).

### 2.1 Locomozione con Gambe

Il movimento di un camminatore bipede, si può modellare con un poligono che ruota. Il passo di un bipede  $d$  (cioè la distanza tra i suoi piedi nella fase di moto mentre sono entrambi a terra) rappresenta il lato del poligono, mentre la lunghezza delle gambe  $l$ , il raggio. Più il passo si accorcia, più il poligono tende a somigliare ad un cerchio.

Minore è il numero di gambe, più diventa complessa la locomozione: se infatti bastano 3 gambe per la stabilità statica, ne servono 6 per garantire una camminata statica (perché durante il moto alcuni arti devono sollevarsi).

### 2.2 Locomozione con Ruote

Tipi di ruota:

- *standard wheel*, 2 DOF;
- *castor wheel*, 3 DOF;
- *ruota svedese*, 3 DOF;
- *ruota sferica*.

La stabilità è garantita con 3 ruote ma viene solitamente migliorata con 4. Bisogna fare dei compromessi tra *manovrabilità* e *controllabilità*.

### Immobilità

ZMP e proiezione del CoM coincidono

### Moto

ZMP e proiezione del CoM possono trovarsi in punti diversi

## 2.3 Gradi di Libertà

**Gradi di libertà** (*degrees of freedom*) per robot:

- un corpo rigido che trasla e ruota in una dimensione ha un DOF (ad esempio un treno);
- un corpo rigido che trasla e ruota in un piano a due dimensioni ha 3 DOF (2 di traslazione e 1 di rotazione);
- un corpo rigido che trasla e ruota in 3 dimensioni ha 6 DOF (3 di traslazione e 3 di rotazione, come un robot volante).

Un robot si dice *olonomo* se è in grado di muoversi istantaneamente in ogni direzione dello spazio dei suoi gradi di libertà. I robot olonomi esistono, ma richiedono molti motori o una progettazione inusuale.

## 3 Dinamica

La **dinamica** è lo studio delle leggi fisiche necessarie per il moto dei corpi costituenti il robot.  $f$  indica la forza distribuita sul piede caratterizzante il contatto tra piede e suolo.  $\vec{R}$ , forza risultante ZMP, punto di applicazione di  $\vec{R}$  (centro di pressione)

*CoM*, *Center of Mass*: punto nello spazio dove si concentra l'intera massa del corpo  
Sia dato un robot di massa  $M$  con  $N$  link di massa  $m_j$   $1 \leq j \leq N$  - posizione del CoM del corpo  $j$  nel riferimento locale - Posizione di  $j$  nel riferimento assoluto: (posizione e orientazione)

$$c_j = p_j + \vec{R} \cdot c_j$$

- posizione del CoM globale del robot:

$$c = \sum_{j=1}^N m_j c_j / M$$

punto situato nell'intersezione tra la linea di gravità passante per il centro di massa e il suolo

Si tende un elastico attorno ai piedi del robot al livello della superficie di contatto

**Poligono di sostentamento** superficie risultante

ZMP è sempre all'interno del poligono di sostentamento.

La proiezione del CoM può uscire.

$M$  massa totale del robot

$\vec{c} = [x, y, z]^T$  posizione del CoM

$P = [P_x, P_y, P_z]^T$  quantità di moto

$L = [L_x, L_y, L_z]^T$  movimento cinetico

$\dot{c} = \mathcal{P}/M$  La quantità di movimento è il prodotto tra la massa totale del robot e la velocità del suo

$\dot{\mathcal{P}} = f_{all} = Mg + f$  La quantità di movimento dipende dalla risultante delle forze esterne applicate al ro

$\dot{\mathcal{L}} = \tau_{all} = c \times Mg + \tau$  Il momento cinetico dipende dal momento risultante delle forze est

1. Quando un robot è immobile in piedi, la variazione di quantità del movimento è nulla e le forze di gravità si equilibrano con le forze di reazione del suolo 2. Se le forze di reazione scompaiono, la quantità di movimento aumenta rapidamente verso il basso a cause della forza di gravità caduta libera

1. Dato un robot immobile, questo momento si dovrà equilibrare con quello generato dalla forza peso 2. Nel caso in cui questo equilibrio non avvenga, il momento di reazione del suolo aumenta rapidamente caduta libera

Il calcolo della posizione dello ZMP a seconda dei movimenti effettuati dal robot può essere effettuato secondo due procedimenti:

1. Analitico
2. Approssimato

### 3.0.1 Procedimento Analitico

$$\begin{aligned}\tau_p &= \dot{\mathcal{L}} - c \times Mg + (\dot{\mathcal{P}} - Mg) \times p \\ p_x &= \frac{Mgx + p_z \dot{\mathcal{P}}_x - \dot{\mathcal{L}}_y}{Mg \dot{\mathcal{P}}_z} \\ p_y &= \frac{Mgx + p_z \dot{\mathcal{P}}_y - \dot{\mathcal{L}}_x}{Mg \dot{\mathcal{P}}_z}\end{aligned}$$

### 3.0.2 Procedimento Approssimato

1. Si rappresenta il robot in tutto il suo insieme come un'unica massa puntiforme
2. Si calcola la quantità di movimento ed il momento cinetico all'origine

$$\begin{aligned}\mathcal{P} &= M\dot{c} \\ \mathcal{L} &= c \times M\dot{c}\end{aligned}$$

3. Si calcolano le corrispondenti derivate

$$\begin{bmatrix} \dot{\mathcal{P}}_s \\ \dot{\mathcal{P}} \\ \dot{\mathcal{P}} \end{bmatrix} = \begin{bmatrix} M\ddot{x} \\ M\ddot{y} \\ M\ddot{z} \end{bmatrix}$$

4. Si inseriscono i valori trovati nelle espressioni di  $p_x$  e  $p_y$  del procedimento analitico

$$p_x = x - \frac{(z - p_z)\ddot{x}}{\ddot{z} + g}$$

$$p_y = y - \frac{(z - p_z)\ddot{y}}{\ddot{z} + g}$$

5. Lo ZMP è definito sotto la direzione delle forze di reazione del suolo e di quelle inerziali. I valori di queste ultime si ottengono ponendo  $\ddot{z} = 0$  e  $p_z = 0$

$$p_x = x - \frac{z}{g}\ddot{x}$$

$$p_y = y - \frac{z}{g}\ddot{y}$$

### 3.1 Marcia Bipede

Requisiti: 1. ZMP all'interno del poligono di sostentamento 2. CoM sempre alla stessa altezza Il robot viene modellato come: 1. LIP-3D (Linear inverted pendulum 3D) 2. Carrello sul tavolo  
(da completare, vedi slide)

## 4 Cinematica

### 4.1 Cinematica del Robot Mobile

### 4.2 Cos'è la Cinematica

Un robot può essere modellato come un sistema fisico composto da corpi rigidi, detti anche **collegamenti** (*link*), connessi tra loro da **giunti** (*joint*). Una **posa** (*pose*) rappresenta la *posizione e l'orientazione* di un corpo rigido.

La **cinematica** è lo studio delle relazioni matematiche tra posizioni dei collegamenti e angoli dei giunti. La cinematica dei robot descrive quindi la posa, la velocità, l'accelerazione e le derivate di ordine superiore che corpi che compongono il meccanismo.

Per controllare il movimento del robot dobbiamo definire la posizione e l'orientamento di ciascuno dei suoi segmenti. Per questo scopo abbiamo:

- Il sistema di riferimento assoluto;
- Il sistema di riferimento relativo.

### 4.3 Il Robot nel Mondo

#### 4.3.1 Sistema di Riferimento Assoluto

1. Sia dato un robot nella sua posizione iniziale.
2. Fissare un punto sul terreno sottostante.



3. Tracciare il sistema di riferimento partendo da questo punto.

Otteniamo le coordinate del mondo  $\Sigma_w$  (il mondo in cui si muove il robot). La posizione e l'orientamento sono assoluti se espressi nelle coordinate del mondo.

#### 4.3.2 Sistema di Riferimento Relativo

- Il sistema di riferimento non è fissato al suolo ma ad un segmento (un corpo rigido di cui è composto il robot).
- L'origine si trova nell'asse di rotazione del segmento.
- L'orientazione degli assi (del sistema di riferimento) si riferisce ai movimenti che il segmento può eseguire.

Le coordinate locali sono  $\Sigma_a$ . Si tratta della posa di un corpo espressa relativamente ad un altro corpo (ad esempio, la posizione di una mano relativamente alla posizione della spalla).

#### 4.3.3 Traslazione

La traslazione è uno spostamento nel quale nessun punto del corpo rigido resta nella sua posizione iniziale e tutte le linee rette del corpo rimangono parallele alla loro orientazione iniziale.

#### 4.3.4 Rotazione

La rotazione è uno spostamento nel quale almeno un punto del corpo rigido rimane nella sua posizione iniziale e non tutte le linee nel corpo restano parallele alla loro orientazione iniziale.

Siano dati:

$(\hat{x}_i, \hat{y}_i, \hat{z}_i)$  vettore base del frame  $i$   
 $(\hat{x}_j, \hat{y}_j, \hat{z}_j)$  vettore base del frame  $j$

La matrice di rotazione che si ottiene è:

$$R_i^j = \begin{pmatrix} \hat{x}_i \cdot \hat{x}_j & \hat{y}_i \cdot \hat{x}_j & \hat{z}_i \cdot \hat{x}_j \\ \hat{x}_i \cdot \hat{y}_j & \hat{y}_i \cdot \hat{y}_j & \hat{z}_i \cdot \hat{y}_j \\ \hat{x}_i \cdot \hat{z}_j & \hat{y}_i \cdot \hat{z}_j & \hat{z}_i \cdot \hat{z}_j \end{pmatrix}$$

Assi del sistema di riferimento:

$$e_{ax} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad e_{ay} = \begin{bmatrix} 0 \\ \cos \theta \\ \sin \theta \end{bmatrix} \quad e_{az} = \begin{bmatrix} 0 \\ -\sin \theta \\ \cos \theta \end{bmatrix}$$

Le rotazioni possono essere espresse con matrici  $3 \times 3$ . Le matrici di rotazione si possono combinare con semplici moltiplicazioni matriciali.

$$R_i^k = R_j^k \cdot R_i^j$$

#### 4.3.5 Angoli di Eulero

$(\phi, \theta, \psi)$ : ciascun angolo rappresenta una rotazione su un asse di un frame di coordinate in movimento. La location dell'asse di tutte le successive rotazioni dipende dalla rotazione precedente.

Esempio. Nel sistema  $Z - Y - X$  sono dati:

- $i$  frame in movimento
- $j$  frame fisso
- inizialmente i frame coincidono

Allora

- $\phi$ , implica una rotazione sull'asse  $Z$  del frame  $i$
- $\theta$ , implica una rotazione sull'asse  $Y$  del frame  $i$
- $\psi$ , implica una rotazione sull'asse  $X$  del frame  $i$

#### 4.3.6 Roll-Pitch-Yaw

Traducibile con: rollaggio, picchiata, imbardata. Ogni angolo rappresenta una rotazione attorno ad un asse di un frame di riferimento **fisso**.

Esempio. Nel sistema  $X - Y - Z$  sono dati:

- $i$  frame in movimento
- $j$  frame fisso
- inizialmente i frame coincidono

Allora

- $\phi$ , implica una rotazione roll sull'asse fisso  $X$
- $\theta$ , implica una rotazione pitch sull'asse  $Y$
- $\psi$ , implica una rotazione yaw sull'asse  $Z$

Gli angoli di Eulero e il sistema RPY hanno una singolarità: **il blocco del giunto cardanico** (meglio noto come *gimbal lock*). Consiste nella perdita di un DOF in uno spazio a tre dimensioni che avviene quando gli assi di due dei tre giunti si trovano in una configurazione parallela.

### 4.3.7 I Quaternioni

$$\begin{aligned}\vec{e} &= \epsilon_0 + \epsilon_1 i + \epsilon_2 j + \epsilon_3 k \\ ii &= jj = kk = -1 \\ ij &= k, \quad jk = i, \quad ki = j \\ ji &= -k, \quad kj = -i, \quad ik = -j\end{aligned}$$

Gli  $\epsilon$  sono scalari mentre  $i, j, k$  sono operatori. I quaternioni *non soffrono di singolarità*.

Posizione:

$$\vec{p} = (p_x, p_y, p_z)^T \Rightarrow p = p_x i + p_y j + p_z k$$

Rotazione:

- $\vec{e} \vec{p} \vec{e}$  esegue una rotazione del vettore  $\vec{p}$  sulla direzione  $(\epsilon_1, \epsilon_2, \epsilon_3)^T$ .
- $\epsilon_0$  è l'angolo di rotazione.
- $\epsilon_1, \epsilon_2, \epsilon_3$  sono gli assi di rotazione.

### 4.3.8 Trasformazioni Omogenee

Siano

- $p_h$ , la posizione della mano rispetto alle coordinate del mondo  $\Sigma_w$
- $r = p_h^a$  la posizione della mano rispetto alla spalla nelle sue coordinate iniziali  $\Sigma_a$
- $r' = R_a r$

Segue la matrice di una trasformata omogenea:

$$T_a = \begin{bmatrix} R_a & p_a \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

É possibile definire un sistema di coordinate locali in un altro sistema di coordinate locali. Ad esempio, si può esprimere la posizione della mano relativamente alla spalla, sfruttando la posizione nota della mano rispetto al gomito. Otteniamo una catena di trasformate.

## 4.4 La Cinematica dei Giunti del Robot

I collegamenti che costituiscono il corpo del robot si considerano perfettamente rigidi e le loro superfici geometricamente perfette. I giunti collegano più corpi rigidi. La **cine-**  
**matica dei giunti** (*joint kinematics*) studia il moto tra due corpi.

- Moto di **rivoluzione** (*revolute*): un link ruota attorno ad un altro, c'è un solo asse di rotazione (1 DOF). Esempio: le cerniere delle porte.
- Moto **prismatico**: un link scivola su un altro (1 DOF). Esempio: pistone idraulico.

- Moto **cilindrico**: un link scivola e ruota sull'altro, è l'unione dei due precedenti moti (2 DOF).
- Moto **elicoidale** (*helical*): ad esempio, una vite.
- Moto **sferico**: si ottiene dal contatto di due superfici sferiche congruenti (una interna e l'altra esterna). Consente di ruotare su tutti gli assi (3 DOF).
- Moto **planare**: si ottiene dal contatto di due superfici piane (3 DOF).

## 4.5 Problemi Relativi alla Cinematica

### 4.5.1 Cinematica Diretta

La **cinematica diretta** (*forward kinematics*) è il calcolo delle configurazioni dei segmenti in funzione dei valori degli angoli dei giunti (associati a tali segmenti).

Angoli dei giunti  $\Rightarrow$  Posizione e orientazione (posa) dei collegamenti.

### 4.5.2 Cinematica Inversa

La **cinematica inversa** (*inverse kinematics*) è il calcolo degli angoli dei giunti partendo dalle posizioni e dalle orientazioni dei segmenti.

Posizione e orientazione (posa) dei collegamenti  $\Rightarrow$  Angoli dei giunti.

Due diversi approcci:

- approccio analitico, che comporta troppi calcoli ed è troppo oneroso (calcolo di tutti gli angoli).
- approccio numerico, cinematica differenziale (Jacobiana per calcolare la correzione).

### 4.5.3 Cinematica Differenziale

La cinematica differenziale è la relazione tra le velocità dei giunti e degli end-effector (velocità sia angolare che lineare). La **Jacobiana** è la struttura matriciale che definisce questa relazione.

Si ricorda la relazione che definisce lo Jacobiano geometrico

$$\vec{v} = \vec{J}(q)\dot{q}$$

I valori di  $q$  per cui la matrice  $\vec{J}$  diminuisce di rango prendono il nome di **singolarità cinematiche**. In presenza di singolarità cinematiche si ha:

- perdita di mobilità (non è possibile imporre leggi del moto arbitrarie);
- possibilità di infinite soluzioni al problema cinematico inverso;
- velocità elevate nello spazio dei giunti (nell'intorno di singolarità).

Le singolarità possono essere:

- **ai confini** dello spazio di lavoro raggiungibile;
- **all'interno** dello spazio di lavoro.

Le ultime sono le più problematiche perché ci si può incorrere con traiettorie pianificate nello spazio operativo.

Disaccoppiamento di singolarità:

- calcolo delle singolarità della struttura portante;
- calcolo delle singolarità del polso.

## 4.6 Convenzioni di Denavit e Hartenberg

???

## 4.7 Cinematica del Robot Mobile

La cinematica del robot mobile è simile alla cinematica del robot manipolatore: lo scopo è quello di descrivere il comportamento meccanico del robot per la progettazione e il controllo. I robot mobili possono muoversi senza limiti nel loro ambiente (non c'è un modo diretto per misurare la posizione del robot).

Per prima cosa, è necessario comprendere come i vincoli delle ruote influiscano nel movimento. Le ipotesi che vengono generalmente utilizzate sono:

- movimento in un piano orizzontale;
- le ruote non sono deformabili;
- moto di puro rotolamento (niente scivolamento o slittamento);
- assi di sterzata ortogonali alla superficie;
- le ruote sono tenute insieme da un supporto rigido e non deformabile (il telaio).

Dato un robot con  $M$  ruote

- ogni ruota impone 0 o più vincoli al moto del robot;
- soltanto le ruote fisse e le ruote standard orientabili (*steerable standard wheels*) impongono vincoli.

La manovrabilità del robot è la combinazione

- della mobilità disponibile sui vincoli di slittamento;
- libertà addizionali date dallo sterzo.

Tre ruote sono sufficienti per garantire stabilità statica ad un robot e avere ruote in più significa doverle sincronizzare.

- Grado di manovrabilità:  $\delta_m$ .
- Grado di sterzata:  $\delta_s$ .
- Manovrabilità del robot:  $\delta_M = \delta_m + \delta_s$ .

Il **grado di libertà** (*degree of freedom*, DOF) è l'abilità del robot nel raggiungere varie pose.

Le velocità del robot raggiungibili in modo indipendente sono dette **grado di libertà differenziabile** (*differentiable degree of freedom*, DDOF) e si indica con  $\delta_m$ . Quindi il DDOF rappresenta l'abilità del robot nell'eseguire diversi percorsi.

$$DDOF \leq \delta_m \leq DOF$$

## 4.8 Controllo Cinematico

L'obiettivo del controllore cinematico è di seguire una traiettoria descritta dalla sua posizione e / o profili di velocità in funzione del tempo. Il controllo del movimento non è così semplice perché i robot mobili sono in genere sistemi non ologoni. La maggior parte dei controller non considera la cinematica del sistema.

### 4.8.1 Controllo Open Loop

Nel **controllo a ciclo aperto** (*open loop control*) la traiettoria, o percorso, è divisa in segmenti di forma ben definita: linee rette e archi di circonferenza.

Il problema in questo caso consiste nel calcolare una traiettoria che sia più liscia (*smooth*) possibile. Precalcolare una traiettoria fattibile in questo modo non è sempre facile, bisogna limitare la velocità e l'accelerazione del robot al fine di mantenerlo in pista. Non si adatta se l'ambiente circostante cambia e le traiettorie risultanti di solito non sono lisce.

## 5 Percezione

(Da slide 4a - Percezione, sensori) I sensori seguono la seguente classificazione.

- Cosa?
  - Sensori **propriocettivi**: misurano valori interni al robot (ad esempio, la velocità del motore, il carico sulle ruote).
  - Sensori **esterocettivi**: misurano informazioni relative all'ambiente esterno, nel quale il robot è immerso (ad esempio, la distanza da un oggetto).
- Come?
  - Sensori **passivi**: energia arriva dall'ambiente (bumper).
  - Sensori **attivi**: emettono energia e misurano la reazione dell'ambiente (sonar).

## 5.1 Caratterizzazione dei Sensori

### 5.1.1 Performance dei Sensori

- **Il Range dinamico.** Rapporto tra il limite inferiore e il limite superiore, di solito espresso in decibel (ad esempio, la potenza misurata tra  $1\text{ mW}$  e  $20\text{ W}$ ).
- **Il Range.** Considerando il limite superiore.
- **La Risoluzione.** Intesa come la differenza minima tra due valori. Per i sensori digitali è di solito la risoluzione del dispositivo A/D (ad esempio  $5\text{V}/255 \rightarrow 8\text{ bit}$ ).
- **La Linearità.** Intesa come la variazione dell'output in funzione dell'input (deve avere le proprietà di una funzione lineare). Questa proprietà è meno importante quando il segnale è trattato con il computer.
- **Banda e frequenza.** La velocità con la quale il sensore fornisce il flusso di letture. Il limite superiore generalmente dipende dal sensore e dal tasso di acquisizione.

### 5.1.2 Caratteristiche per Ambienti Reali

- **La Sensitività.** Il rapporto tra il cambiamento dell'output e il cambiamento dell'input. Generalmente, nel mondo reale il sensore ha un'elevata sensibilità.
- **La Cross-Sensitivity.** La sensibilità ad altri parametri ambientali. Influenza di altri sensori attivi.
- **Errore, Accuratezza.** La differenza tra l'output del sensore e il valore reale.
- **Errori sistematici.** (o deterministici) Causati da fattori che teoricamente possono essere modellati.
- **Errori Casuali.** (o non-deterministici) Sui quali non è possibile fare alcuna predizione.
- **Precisione.** Riproducibilità dei risultati dei sensori.

Nel mondo reale, le misurazioni cambiano e sono soggette ad errori, dovuti ad esempio a cambiamenti d'illuminazione, pareti riflettenti, superfici che assorbono luce o suoni.

I sensori vengono modellati su distribuzioni di probabilità (errori casuali). Si hanno poche informazioni sulle cause degli errori, quindi questi possono essere considerati con delle curve Gaussiane.

Esempi di sensori (vedi slide): encoder, inclinometri, bussole, giroscopi.

## 6 Architetture Software per Robot

### 6.1 Paradigma SPA

Il paradigma SPA (*Sense-Plan-Act*) inizia con il robot *Shakey* dell'Università di Stanford negli ultimi anni '60.

1. *Shakey* era dotato di una telecamera e il sistema sensoriale prendeva i dati che questa forniva per tradurli in un *modello interno del mondo*.
2. Il sistema pianificatore prendeva i dati relativi al modello interno del mondo e un *obiettivo* e generava un piano d'azione che avrebbe permesso di raggiungere tale obiettivo.
3. Il piano veniva quindi fornito ad un sistema che si occupava di tradurlo in segnali per gli attuatori.

## 6.2 Subsumption Architecture

Nei primi anni '80 divenne evidente che il paradigma SPA aveva dei problemi. Infatti, il planning richiedeva troppo tempo nel mondo reale e eseguire un piano senza dei sensori poteva risultare pericoloso in un ambiente reale.

La più influente delle architetture che cercavano un'alternativa al paradigma SPA fu la *subsumption architecture* di Brooks. Questa si componeva di diversi *layer* di macchine a stati finiti, ciascuno connesso direttamente con i sensori e gli attuatori. Le macchine a stati finiti erano dette *behavior*. Dato che più *behavior* possono essere attivi contemporaneamente, si rende necessario un meccanismo di *arbitration* che consenta a *behavior* con priorità più alta di sovrascrivere i segnali di quelli con priorità inferiore.

In breve tempo però, anche questo sistema raggiunse i suoi limiti.

## 6.3 Layered Architecture

## 6.4 Sistemi Behavior-Based

(*Pensa, poi agisci.*) Nel *controllo deliberativo*, il robot utilizza tutte le informazioni disponibili fornite dai sensori per decidere come agire in seguito. I sistemi che implementano questo approccio sono solitamente composti da moduli che processano i segnali dei sensori, creano modelli, pianificano e valutano la scelta migliore. La pianificazione però è nota per essere un processo computazionale complesso e il robot deve ripeterlo a ogni ciclo di *sense-plan-act*.

La pianificazione richiede inoltre che il robot possieda una rappresentazione simbolica del mondo al suo interno, con la quale si possano fare previsioni sui futuri stati del sistema. Quando il tempo è sufficiente e il modello è accurato, questo è il miglior approccio ma è sconsigliato nel caso in cui l'ambiente sia dinamico.

(*Non pensare, reagisci.*) Il *controllo reattivo* è una tecnica per connettere strettamente gli input dei sensori con gli output degli attuatori. Questo consente al robot di reagire molto rapidamente in ambienti dinamici e non strutturati. Il controllo deliberativo si ispira al concetto biologico di *stimolo-risposta* e non necessita di complesse rappresentazioni del mondo.

I *sistemi ibridi* cercano dei compromessi ai due paradigmi precedenti prendendo la rapidità di risposta dei sistemi reattivi e l'ottimalità delle decisioni dei sistemi deliberativi. La sfida consiste nell'accordare la componente reattiva (che si occupa delle necessità



immediate del robot, come evitare un ostacolo) con la componente deliberativa (che opera su una scala temporale più lunga) producendo un output coerente.

Ci si riferisce spesso ai sistemi ibridi come *architetture a tre strati* per via della loro struttura:

- modulo reattivo, per l'esecuzione;
- strato intermedio di coordinazione;
- modulo deliberativo, per l'organizzazione e la pianificazione.

Il *controllo behavior-based* utilizza un insieme di moduli distribuiti e interagenti, detti *behavior*, per ottenere un comportamento desiderato a livello di sistema. Ogni *behavior* riceve gli input dai sensori e/o altri *behavior* e fornisce gli output agli attuatori e/o altri *behavior*. Quindi, questo tipo di controllo, è una rete strutturata di comportamenti interagenti, senza una rappresentazione del mondo centralizzata.

## 7 Localizzazione

Per prima cosa è necessario chiedersi se serve veramente localizzare un robot. Ad esempio si può definire un comportamento del robot che gli comunichi di evitare gli ostacoli, seguire sempre una parete (destra o sinistra), determinare se l'area scoperta è quella di arrivo.

In alternativa si definisce la *map based navigation*.

- *Global localization*: al robot non è fornita la sua posizione iniziale, che deve essere stimata da zero.
- *Posizion tracking*: il robot conosce la sua posizione iniziale e deve soltanto considerare gli errori di odometria quando si muove.

La localizzazione viene fatta per mezzo di

- sensori esterni, punti di riferimento (artificiali), beacon (ad esempio punti altamente riflettenti con un laser che li identifica oppure beacon colorati e fotocamera omnidirezionale);
- odometria;
- localizzazione basata su mappa, senza sensori esterni o punti di riferimento artificiali, solamente con i sensori di bordo.

### 7.1 Localizzazione Basata su Mappe

Si consideri un robot mobile in un ambiente conosciuto, non appena inizia a muoversi da una posizione nota, può tenere traccia dei suoi spostamenti con l'odometria. Ingredienti:

- teoria delle probabilità, propagazione degli errori;
- rappresentazione della credenza, discreta continua;

- modello di movimento, modello odometrico;
- sensori, modelli per la misurazione.

Conoscere la posizione esatta (ad esempio con un GPS) spesso non è sufficiente.

Il rumore nel segnale fornito dai sensori è spesso influenzato dall'ambiente o da altre misurazioni (ad esempio, altri sensori a ultrasuoni). Il rumore riduce drasticamente l'utilità delle informazioni che si ottengono. La soluzione sta nell'effettuare misurazioni multiple e usare metodi che permettano di unire queste misurazioni (metodi temporali o multi sensore).

Odometria e dead reckoning:

- l'aggiornamento della posizione è basato su sensori propriocettivi;
- odometria: solo sensori ruota;
- dead reckoning: anche i sensori di direzione

Il movimento del robot, rilevato con encoder ruota e/o i sensori di direzione sono integrati nella posizione. L'utilizzo di ulteriori sensori di direzione (ad esempio il giroscopio) potrebbe aiutare a ridurre gli errori cumulativi, ma i problemi principali rimangono.

- Gli errori deterministici possono essere eliminati con una corretta calibrazione del sistema.
- Gli errori non deterministici devono essere descritti da modelli di errore e porteranno sempre a una stima di posizione incerta.

Le principali fonti di errore sono:

- Risoluzione limitata durante l'integrazione (incrementi di tempo, risoluzione della misurazione);
- Disallineamento delle ruote (deterministico);
- Diametro della ruota ineguale (deterministico);
- Variazione nel punto di contatto della ruota;
- Contatto del pavimento ineguale (scivoloso, non planare).

Classificazione degli errori:

- **errore di portata** (*range error*): relativo alla lunghezza del percorso (distanza) che il robot ha compiuto;
- **errore di rotazione** (*turn error*): simile al precedente, ma per le rotazioni;
- **errore di deriva** (*drift error*): la differenza nell'errore delle ruote causa un errore nell'orientamento angolare del robot.

Caso continuo:

- Precisione legata dai dati del sensore
- Stima di posa ipotetica in genere
- Perso quando divergenti (per singola ipotesi)
- Rappresentazione compatta e in genere ragionevole in potenza di elaborazione.

Caso discreto:

- Precisione vincolata dalla risoluzione della discretizzazione
- Stima di posa ipotetica in genere
- Mai perso (quando le divergenze convergono in un'altra cella)
- È necessaria memoria importante e potenza di elaborazione.

## 7.2 SLAM

Il problema del *Simultaneous Location And Mapping* (SLAM) è definito come segue. Un robot mobile vaga per un ambiente sconosciuto, partendo da una posizione con coordinate note. Il suo moto è incerto, rendendo gradualmente più difficile determinare le sue coordinate globali. Mentre si muove, il robot può percepirne l'ambiente. Il problema SLAM consiste nel costruire una mappa dell'ambiente mentre si determina contemporaneamente la posizione del robot rispetto a questa mappa.

I principali paradigmi dello SLAM sono:

- *Extended Kalman Filters* (EKF): storicamente è stato il primo approccio e anche il più influente. Si proponeva di utilizzare un vettore di singoli stati e un insieme di caratteristiche dell'ambiente con associata una matrice di covarianza, che rappresenta l'incertezza nelle stime.
- *Graph-Based Optimization Techniques*: questo paradigma considera il robot e i landmark come nodi in un grafo, mentre ogni coppia di luoghi è tenuta insieme da un arco rappresentante le informazioni dell'odometria. Esistono altri archi che connettono i luoghi con i landmark.
- *Particle Methods*: si basa sui particle filter che sono diventati popolari solo ddi recente.