

Introduzione alla Robotica

Corrado Santoro

ARSLAB - Autonomous and Robotic Systems Laboratory

Dipartimento di Matematica e Informatica - Università di Catania, Italy

`santoro@dmi.unict.it`

Sistema Robotico

Un **sistema robotico** è un apparato *meccanico/elettronico/informatico* in grado di svolgere un **compito** ben preciso in **autonomia**

Sistema Robotico

Un **sistema robotico** è un apparato *meccanico/elettronico/informatico* in grado di svolgere un **compito** ben preciso in **autonomia**

Elementi di un Sistema Robotico

- Ambiente di lavoro
- Struttura meccanica
- Sensori
- Unità di elaborazione
- Attuatori

Un tipico “sistema robotico”

Un tipico “sistema robotico”



Un tipico “sistema robotico”



E' un Sistema Robotico?

- Ha un obiettivo ben preciso:
lavare i panni!
- Lo fa “da sola”
- Opera in un “ambiente di lavoro”
- Ha una **struttura meccanica**
- Possiede opportuni **sensori**
(temperatura, peso, etc.)
- Possiede una **unità di elaborazione** che comanda il ciclo di lavaggio
- Possiede degli **attuatori** (motore cestello, pompa, resistenza di riscaldamento, etc.)

Un altro tipico “sistema robotico”

Un altro tipico “sistema robotico”



Un altro tipico “sistema robotico”



Il “robot da cucina”

- Ha un obiettivo ben preciso: **cucinare pietanze!**
- Lo fa “da solo”
- Opera in un “ambiente di lavoro”
- Ha una **struttura meccanica**
- Possiede opportuni **sensori** (temperatura, peso, etc.)
- Possiede una **unità di elaborazione** che comanda le operazioni legate alla cottura del cibo
- Possiede degli **attuatori** (motore cestello, resistenza di riscaldamento, etc.)

Un altro robot “domestico”

Un altro robot “domestico”



Un altro robot “domestico”



Il “robot aspirapolvere”

- Ha un obiettivo ben preciso: **togliere la polvere dal pavimento della stanza!**
- Lo fa “da solo”
- Opera in un “ambiente di lavoro” (la stanza)
- Ha una **struttura meccanica**
- Possiede opportuni **sensori** (posizione, ostacoli, scalini, sporcizia, etc.)
- Possiede una **unità di elaborazione** che comanda le operazioni legate alla gestione dei percorsi ed all’attivazione della pompa
- Possiede degli **attuatori** (motori locomozione, aspiratore, etc.)

Robot domestici: dall'utile all'inutile

Robot domestici: dall'utile all'inutile





Aibo: il cane-robot della Sony!

- Ha un obiettivo ben preciso:
comportarsi come un cane!
- Scodinzola... abbaia... mugola...
- Ha una **struttura meccanica**
- Possiede opportuni **sensori** (posizione dei giunti, telecamera, sensori di presenza, sensori di tocco, microfono, etc.)
- Possiede una **unità di elaborazione** che prova a realizzare un comportamento quanto più simile a quello di un cane
- Possiede degli **attuatori** (motori delle zampe, azionamenti per il movimento delle orecchie e della coda, speaker, etc.)



Aibo: il cane-robot!

- Aibo è uno dei primi tentativi commerciali di realizzare un robot con **comportamento simile a quello di un essere vivente**
- Tuttavia ...
- ... le abilità di un cane
- ... o le sue capacità di interazione
- ... **sono molto difficili da emulare artificialmente!**
- A lungo andare Aibo risulta **ripetitivo e noioso**

Robot domestici: dall'inutile allo sperimentale

Robot domestici: dall'inutile allo sperimentale



Robot domestici: dall'inutile allo sperimentale

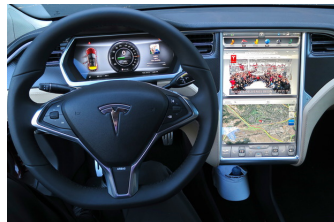


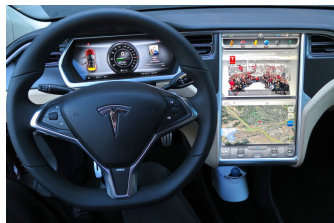
NAO: l'umanoide della Softbank Robotics!

- Incarna l'idea emergente dei **personal robots**
- E' una **piattaforma sperimentale**
- Possiede 25 motori per il controllo dei movimenti
- ... due telecamere
- ... quattro microfoni
- ... due speaker
- Piattaforma software di sviluppo "open"

Robot Mobili Personali

Robot Mobili Personali





La Tesla e le auto a guida autonoma

- Il **veicolo autonomo** costituisce la prossima sfida nell'ambito dei sistemi di trasporto
- E' un **sistema robotico**
- Ha un **obiettivo**: portarvi sani e salvi verso una destinazione
- Ha dei **sensori**: GPS, velocità, ostacoli, ambiente, etc.
- Ha un'**unità di elaborazione**, che stabilisce come pilotare la guida sulla base del percorso voluto e le informazioni dai sensori
- Ha degli **attuatori**: motore, sterzo, freni, etc.

Robot Mobili Militari

Robot Mobili Militari



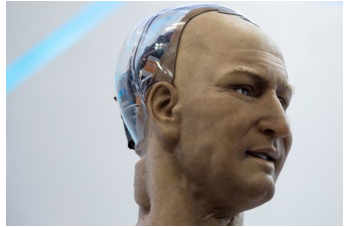


I predator

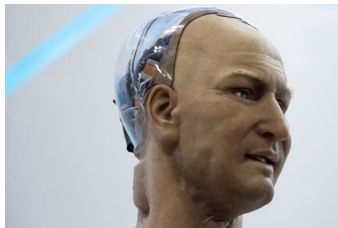
- Sono **velivoli autonomi** (UAV) in dotazione ai paesi NATO
- E' un **sistema robotico**
- Ha un **obiettivo**: effettuare una missione ben precisa
- Ha dei **sensori**: GPS, velocità, ostacoli, ambiente, etc.
- Ha un'**unità di elaborazione**, che stabilisce come effettuare il pilotaggio sulla base del percorso voluto e le informazioni dai sensori
- Ha degli **attuatori**: motore, superfici mobili (alettoni, timone, elevatore)

Dall'Umanoide all'Umano: la mimica facciale

Dall'Umanoide all'Umano: la mimica facciale



Dall'Umanoide all'Umano: la mimica facciale



Zeno, Han e gli altri “umani” della Hanson Robotics

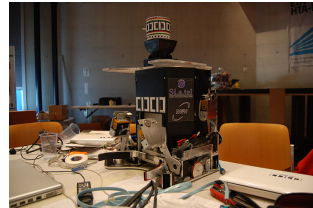
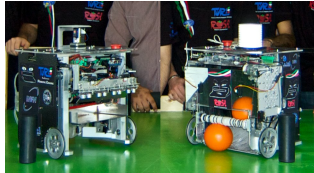
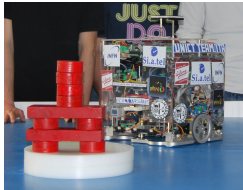
- Nell'imitazione dei comportamenti umani, la nuova sfida è la gestione delle **emozioni**
- **Modelli artificiali** per la rappresentazione delle emozioni con la mimica facciale
- Tecniche per il **riconoscimento** delle emozioni

Dall'Umanoide all'Umano: la deambulazione



I “mostri” della Boston Dynamics

- **Emulazione del movimento** degli arti, sia dei **quadrupedi** che dei **bipedi**
- L'obiettivo è rendere la deambulazione artificiale **altamente affidabile**
- Capacità di muoversi con facilità in **qualsunque ambiente fisico**



Competizioni Robotiche: Eurobot

- Competizione robotica studentesca
- **Obiettivo:** realizzare da zero un robot autonomo in grado di “**giocare**” a un gioco con regole ben precise:
 - **catturare oggetti**
 - **identificarli**
 - **depositarli in zone specifiche**
 - **evitare l'avversario**

Problematiche connesse ai sistemi robotici

Quali problemi deve affrontare il software?

- L'**Ambiente fisico** in cui il robot “vive”
- L'**Impredicibilità** dell'ambiente
- Le **Caratteristiche Dinamiche** dell'ambiente
- Differenti **livelli** e **piani di azione**

Software per sistemi “non-robotici”

- L'ambiente dove opera il software è genericamente **virtuale**, sottoposto al nostro controllo
- Le chiamate a funzione ...
 - 1 ... **“funzionano sempre”**
 - 2 ... restituiscono un'indicazione di **successo/fallimento**
- La seconda categoria restituisce il risultato “immediatamente”, rispetto all'eventuale esecuzione dell'azione (es. invio di un pacchetto in rete)
- La probabilità di fallimenti è (in genere) **molto bassa** e legata a eventi eccezionali
- L'azione è **persistente**

Software per sistemi "robotici"

- L'ambiente dove opera il software è **fisico** ed **evolve secondo sue proprie regole** che non siamo in grado di controllare
- Una chiamata di funzione che agisce sull'ambiente:
 - 1 può **fallire** anche con probabilità elevata
 - 2 l'indicazione di fallimento può **non essere immediata**
 - 3 l'azione (o la sua conseguenza) **non è garantito sia persistente**

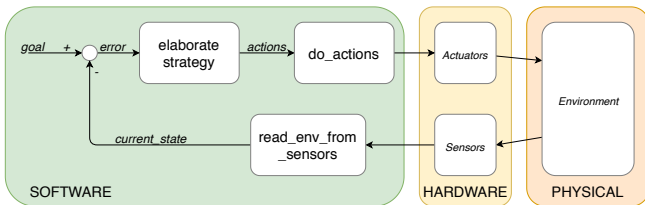
Che significa “controllare”?

- Nei sistemi fisici, effettuare un’azione **non implica** la sua persistenza!
- Esempio: **percorrere un rettilineo in auto richiede che lo sterzo sia “dritto”**
- Impostare lo sterzo e non **toccarlo più, è sufficiente** per assicurare un percorso corretto?
- **No!** Noi **verifichiamo** continuamente la nostra “consegna” (percorso rettilineo) e “aggiustiamo” (controlliamo) lo sterzo sulla base di **errori**

Il Problema del “Controllo”

Il principio dei sistemi di controllo: il **feedback**

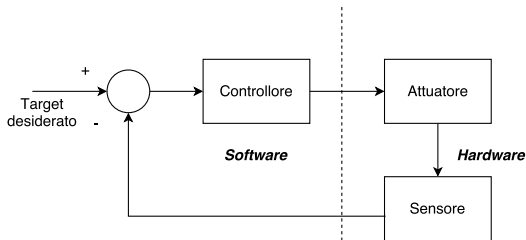
- Data una grandezza fisica da **controllare** (velocità, posizione, etc.) tramite un certo **attuatore**
 - Posto che sia presente un **sensore** in grado di misurare la grandezza effettiva
- 1 **Acquisisci** il dato sulla grandezza effettiva tramite il **sensore**
 - 2 **Confronta** il valore **effettivo** con il valore **desiderato**
 - 3 Sulla base dell'**errore**, **pilota** l'**attuatore** opportunamente



Il Problema del “Controllo”

Algoritmo di Controllo

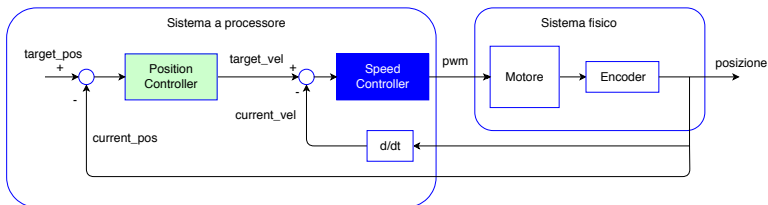
```
while True do  
    On each  $\Delta T$ ;  
     $s \leftarrow \text{read\_sensor\_value}();$   
     $\text{error} \leftarrow \text{target} - s;$   
     $a \leftarrow \text{controller}(\text{error});$   
     $\text{drive\_actuator}(a);$   
end
```



Il Problema del “Controllo”

Azionamenti e “control loops”

- Ogni azionamento è caratterizzato da (almeno) due **loop di controllo**
 - **velocità**
 - **posizione**
- I loop (o blocchi) di controllo sono connessi **in cascata**
- Il software di “basso livello” di un robot contiene una **moltitudine** di **blocchi di controllo**
- In un drone, in navigazione GPS, ne operano ben 12 (almeno!)



Azionamenti e Inerzia

- Se volessimo arrestare la nostra auto in prossimità di un segnale di STOP, azioneremmo il pedale del freno proprio “allo STOP”?
- Se così fosse, l'**inerzia** dell'auto ci porterebbe **oltre** il segnale
- Dobbiamo pensarci prima, ma **quanto prima?**

Controllo dell'Inerzia

- L'**inerzia** introduce una latenza tra l'esecuzione dell'azione ed il suo effetto
- Il controllore, che genera l'azione correttiva, deve dunque considerare questo aspetto

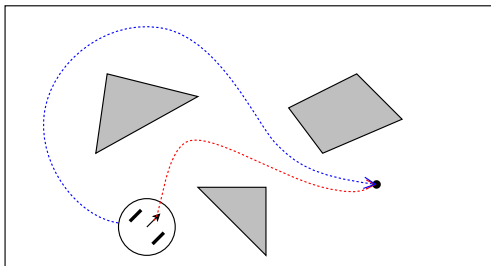
Il “Cervello” di un robot

- Molteplici problematiche software e a diversi “livelli”
- Sistema nervoso **periferico**
 - Controllo specifico dei singoli azionamenti
 - Gestione delle “reazioni”
 - Algoritmi tipici di “regolazione”
- Sistema nervoso **centrale**
 - Gestione del comportamento e pianificazione delle azioni
 - Algoritmi “decisionali”
 - Meccanismi di planning e/o inferenza basati su tecniche di Artificial Intelligence

Il Problema del “Planning”

Dal “Controllo” alla “Pianificazione”

- Dato per certo che gli algoritmi di controllo ci assicurano determinati **azionamenti**
- Il **planning** permette di combinare tali azionamenti (o movimenti) secondo un **obiettivo strategico** ben preciso
- Esempio: **Gestione del moto di un robot**
 - **controllo**: azioni semplici di movimento come *vai dritto*, *ruota*, etc.
 - **planning**: individuazione del percorso da seguire all'interno di un ambiente più o meno complesso



Problema 1

- Siano date diverse azioni $A_1, A_2, \dots, A_n$, determinare (a run-time) il **piano ottimale** $P = \{A_{i_1}, A_{i_2}, \dots, A_{i_n}\}$ per raggiungere un determinato **goal** G
- Esempio: date le azioni di movimento, determinare la sequenza ottimale per effettuare un certo percorso

Problema 2

- Siano dati diversi piani $P_1, P_2, \dots, P_n$, ognuno con differenti azioni ma tutti in grado di raggiungere un determinato **goal** G , determinare (a run-time) il **piano ottimale** P_k
- Esempio: dati diversi percorsi per raggiungere uno stesso punto, determinare quello ottimale

Come può un **Sistema Automatico** effettuare una **scelta consapevole**?

- Ogni **scelta** (sia essa un'azione o un piano), è caratterizzata da:
 - L'**obiettivo finale** (*goal* o *post-condition*)
 - La **fattibilità** (*feasibility* o *pre-condition*)
 - Il **costo/opportunità**
- Esempio, dati n possibili percorsi:
 - **Obiettivo**: punto da raggiungere
 - **Fattibilità**: assenza di ostacoli
 - **Costo**: lunghezza o durata, o una combinazione di essi
- Tali fattori possono essere:
 - Stabiliti staticamente a **design-time**
 - Determinati a **run-time** sulla base dello stato dell'ambiente (catturato dai sensori)

Il “Tipico” Planning Automatico

Algoritmo di Planning

```
foreach  $g \in GOAL\_SET$  do  
   $applicable\_plans \leftarrow \{p \in g.plans \mid is\_feasible(p)\};$   
   $p \leftarrow find\_best(applicable\_plans);$   
   $execute(p);$   
  if  $succeeded(p)$  then  
     $g \leftarrow g - p;$   
    // Remove plan from goal  
    if  $no\_plans(g)$  then  
       $GOAL\_SET \leftarrow GOAL\_SET - g;$   
      // Remove goal from goalset  
    end  
  end  
end
```


Introduzione alla Robotica

Corrado Santoro

ARSLAB - Autonomous and Robotic Systems Laboratory

Dipartimento di Matematica e Informatica - Università di Catania, Italy

`santoro@dmi.unict.it`