

Università di Pisa  
Corso di Laurea Magistrale in Informatica



## Corso di Robotica

---

**Architetture di supervisione e  
controllo del comportamento  
di robot**

Cecilia Laschi

[cecilia.laschi@sssup.it](mailto:cecilia.laschi@sssup.it)



# Architetture di supervisione e controllo di robot

---

- Autonomia e Supervisione di un Robot
- I paradigmi per la supervisione dei robot
  - Architetture Gerarchiche
  - Architetture Reattive
    - Architettura Subsumption
  - Architetture Ibride
  - Architetture Distribuite
- La predizione in robotica

*Riferimenti bibliografici:*

R. Murphy, *Introduction to AI Robotics*, MIT Press (Intr. Parte I, Cap.1 e 4)

R. Brooks, *Cambrian Intelligence*, The MIT Press, 2005

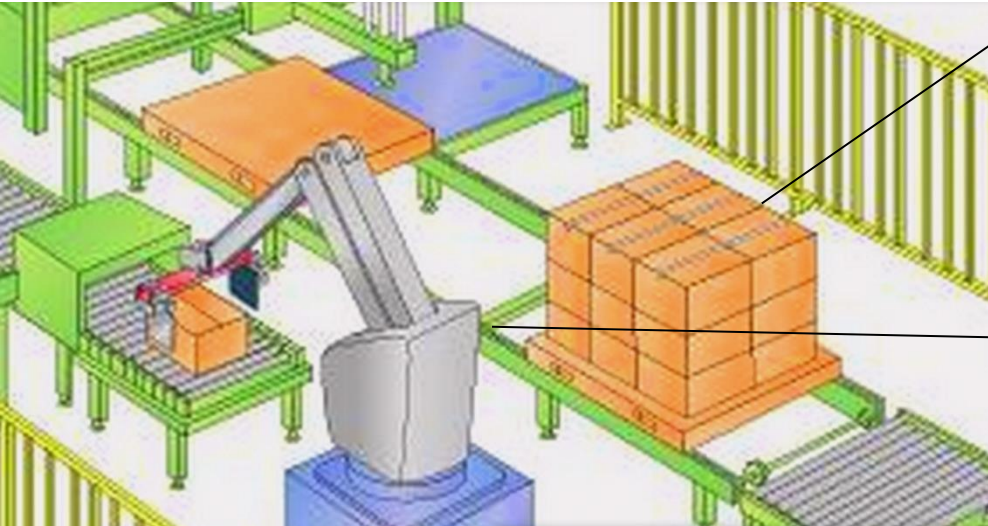


# Definizione di robot

---

- Dal Robot Institute of America:  
“**Manipolatore multifunzionale riprogrammabile** progettato per spostare materiali, parti, utensili o altri dispositivi, per mezzo di movimenti variabili programmati per l’esecuzione di un dato numero di compiti”
- Caratteristiche:
  - Ambiente di lavoro strutturato e noto a priori
  - Ripetitività del compito
  - Accuratezza e velocità

# Tipico scenario manifatturiero per la robotica industriale



Ambiente strutturato sui bisogni del robot

Operatori esperti (formati all'uso)

Procedure ben definite ripetitive

Posizioni predefinite degli oggetti da manipolare



Manipolatori ad elevate prestazioni in termini di accuratezza, ripetibilità, velocità, robustezza

Presenza umana ben delimitata

ISTITUTO  
DI BIORBOTICA



Scuola Superiore  
Sant'Anna

# I robot fuori dalle fabbriche: Robotica di Servizio



**Pulizia, ambiente**



- Ambiente non strutturato
- Condivisione dello spazio di lavoro tra persone e robot
- Capacità percettive
- Comportamento reattivo

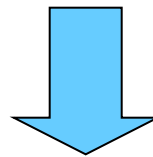
**Ambienti  
pericolosi per  
l'Uomo o  
inaccessibili**



# Definizione di robot autonomo

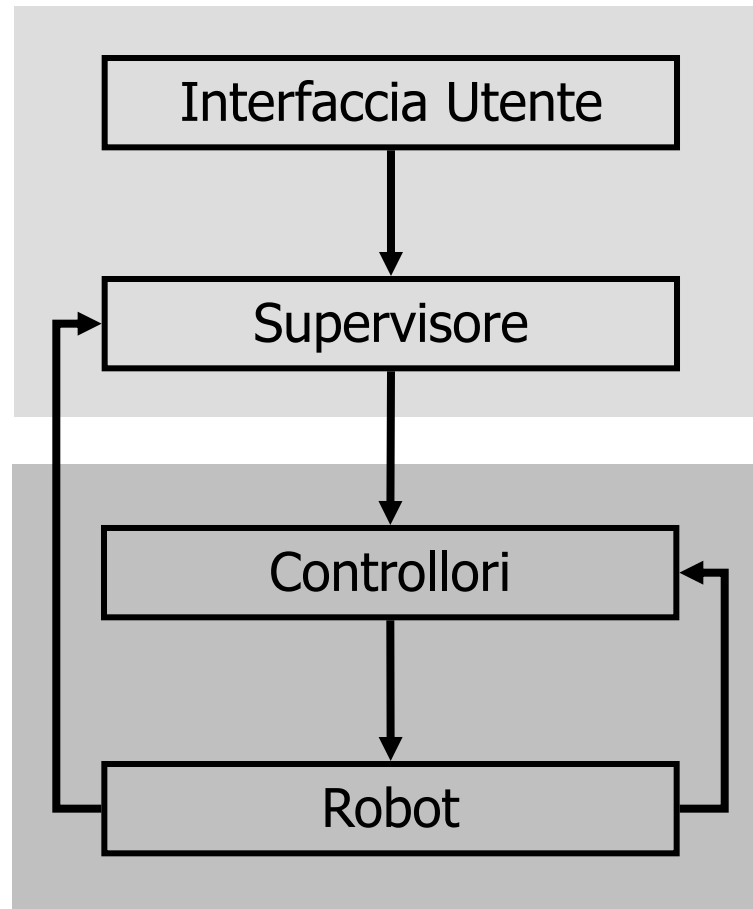
---

- Macchina capace di accettare ed eseguire **autonomamente** comandi o missioni in ambienti **non completamente strutturati senza l'intervento dell'uomo**
- Sfide tecnologiche:  
*"pianificare dinamicamente i comportamenti del robot in un ambiente di lavoro non noto a priori e variabile nel tempo, in funzione della richiesta di esecuzione di un determinato compito"*



**Supervisione o  
pianificazione del comportamento  
di un robot**

# Sistema di supervisione di un robot



Controllo di  
alto livello

Controllo di  
basso livello



# Livelli di autonomia di un robot

---

- Robot autonomo: il robot pianifica ed esegue autonomamente le proprie azioni
- Robot non autonomo: il robot è comandato passo passo da un operatore (teleoperazione)
- Robot semi-autonomo: controllo condiviso tra utente e robot; possono esistere diversi livelli di semi-autonomia

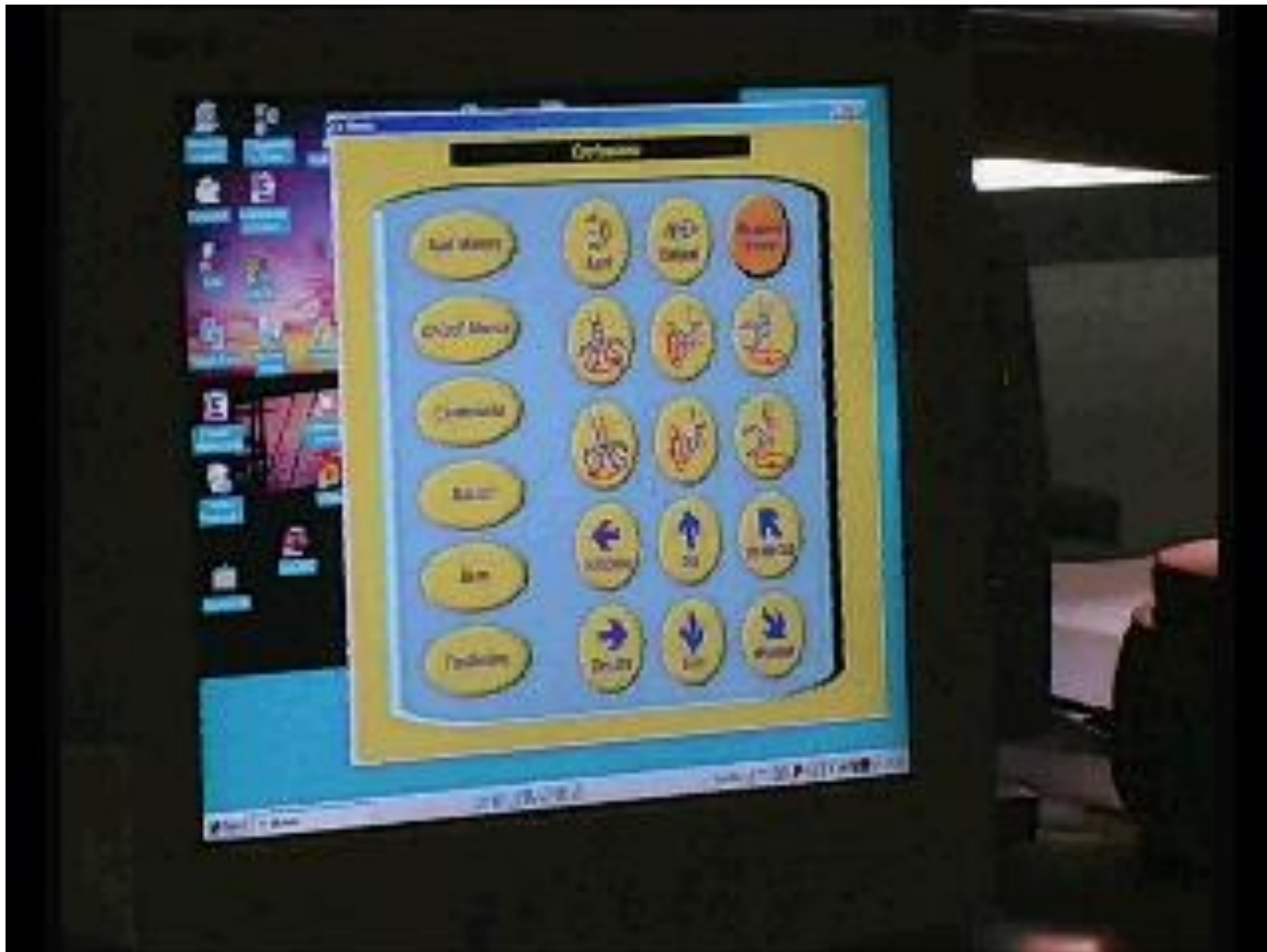


# La teleoperazione di un robot

- La teleoperazione di un robot si ha quando un operatore umano controlla un robot da una postazione remota (robot non autonomo)
- Generalmente l'operatore (locale) controlla il robot (remoto) da una stazione di controllo che fornisce strumenti per la guida (es. joystick) e il monitoraggio (display) del robot

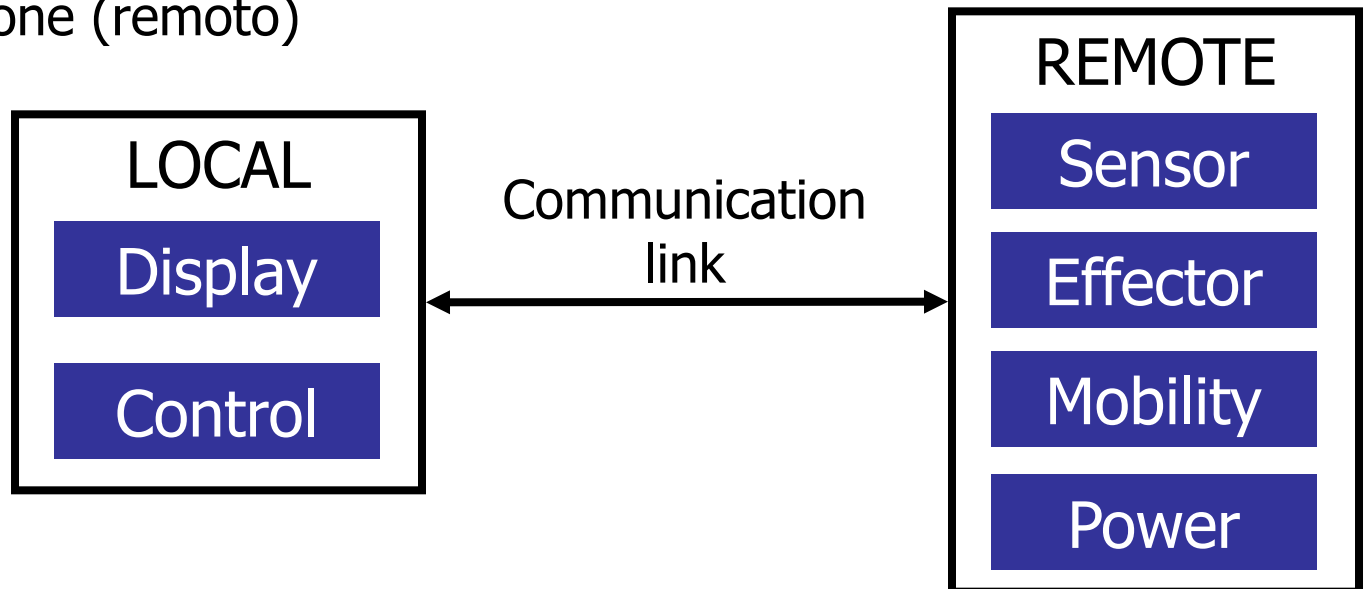


# La teleoperazione di un robot



# La teleoperazione di un robot

- Operatore umano (locale)
  - Display (locale)
  - Interfaccia di controllo e supervisione
- Canale di comunicazione
- Robot (remoto)
  - Sensori/Acquisizione dati sensoriali (remoto)
  - Attuazione (remoto)





# I paradigmi per la supervisione di robot

---

- “Un paradigma è un insieme di assunzioni teoriche e di tecniche risolutive che caratterizza un approccio ad una classe di problemi”
- Nessun paradigma ha validità assoluta: alcuni problemi si affrontano meglio con certi paradigmi piuttosto che con altri
- Scegliere il paradigma giusto per affrontare un problema rende spesso più semplice giungere alla migliore soluzione possibile



# I paradigmi per la supervisione di robot

---

- Tradizionalmente ci sono tre paradigmi principali utilizzati per affrontare il problema della supervisione o pianificazione dei comportamenti dei robot:
  - Paradigma gerarchico
  - Paradigma reattivo
  - Paradigma ibrido (deliberativo/reattivo)



# I paradigmi per la supervisione di robot

---

- I tre paradigmi differiscono per il modo in cui le principali funzioni primitive di un robot sono organizzate
- Le principali funzioni (primitive) di un robot sono:
  - Percezione (SENSE)
  - Pianificazione (PLAN)
  - Azione/Attuazione (ACT)



# I paradigmi per la supervisione di robot

---

- I tre paradigmi possono essere descritti in due modi:
  - descrivendo le relazioni tra le funzioni primitive
  - descrivendo il modo con il quale le informazioni sensoriali sono elaborate e distribuite ai moduli del sistema (locale/globale)



# I paradigmi per la supervisione di robot

---

**PLAN**

**SENSE**

**ACT**





# I paradigmi per la supervisione di robot

<b>ROBOT PRIMITIVES</b>	<b>INPUT</b>	<b>OUTPUT</b>
SENSE	Sensor data	Sensed information
PLAN	Information (sensed or cognitive)	Directives
ACT	Sensed information or directives	Actuator commands



# I paradigmi per la supervisione di robot

---

- Gerarchico: S-P-A

- Reattivo: S-A

- Ibrido: P, S-A

**PLAN**

**SENSE**

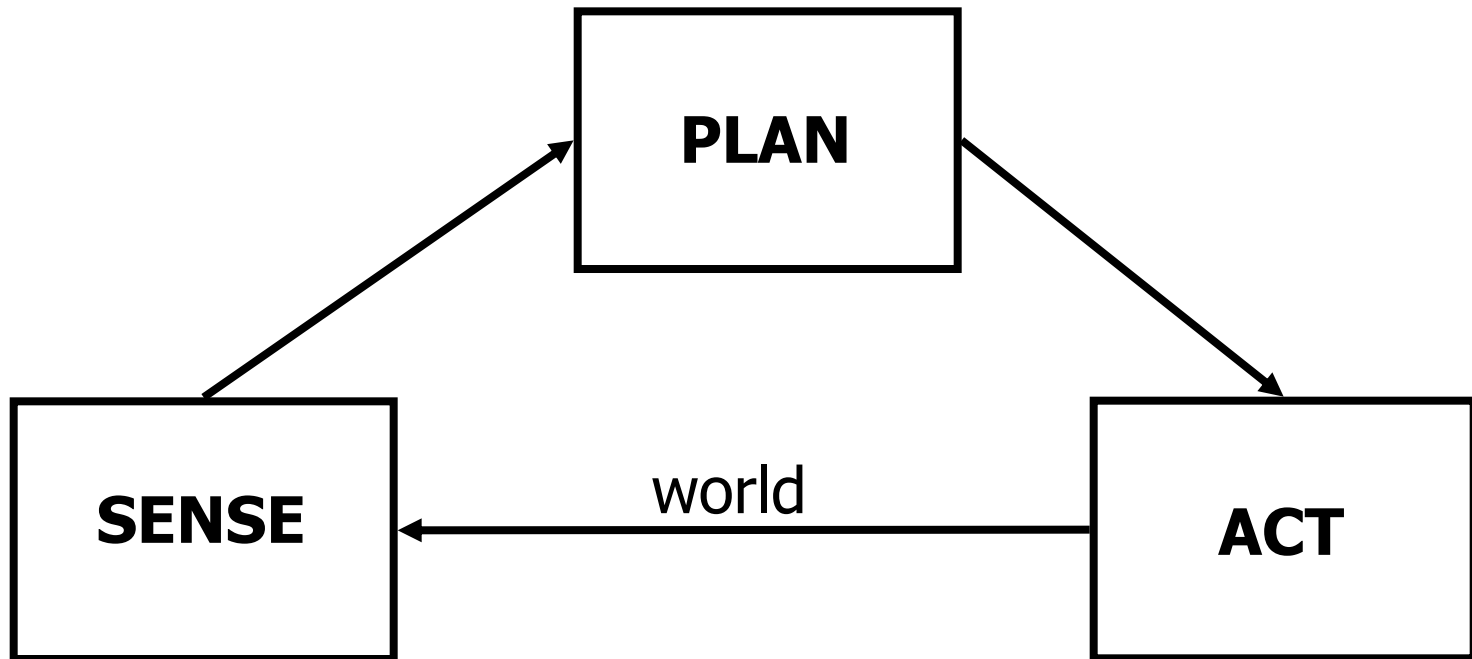
**ACT**



# Il paradigma gerarchico

---

- Gerarchico: S-P-A



# Il paradigma gerarchico

<b>ROBOT PRIMITIVES</b>	<b>INPUT</b>	<b>OUTPUT</b>
SENSE	Sensor data	Sensed information
PLAN	Information (sensed or cognitive)	Directives
ACT	Sensed information or directives	Actuator commands

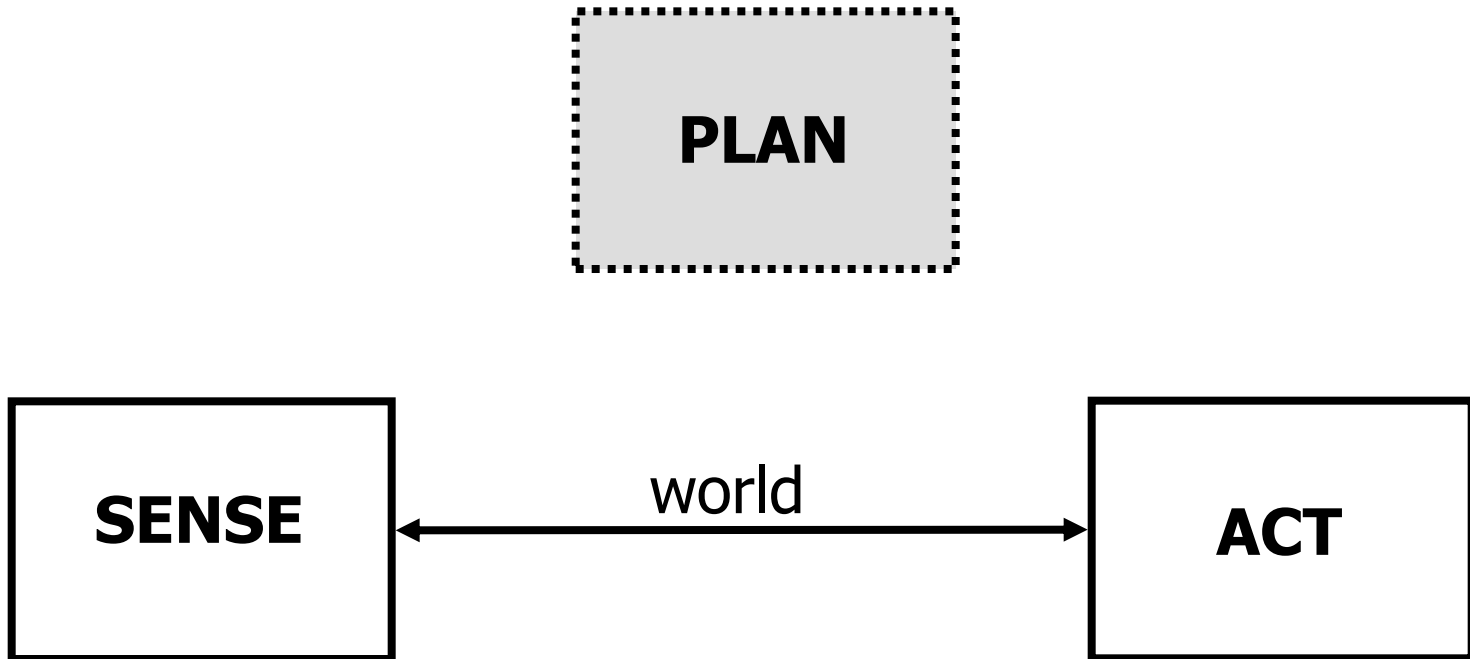
```
graph TD; SENSE[SENSE] --> SENSE_IN[Sensor data]; SENSE_IN --> SENSE_OUT[Sensed information]; PLAN[PLAN] --> PLAN_IN["Information (sensed or cognitive)"]; PLAN_IN --> PLAN_OUT[Directives]; ACT[ACT] --> ACT_IN["Sensed information or directives"]; ACT_IN --> ACT_OUT[Actuator commands]; SENSE_OUT --> PLAN_OUT; PLAN_OUT --> ACT_IN;
```



# Il paradigma reattivo

---

- **Reattivo: S-A**



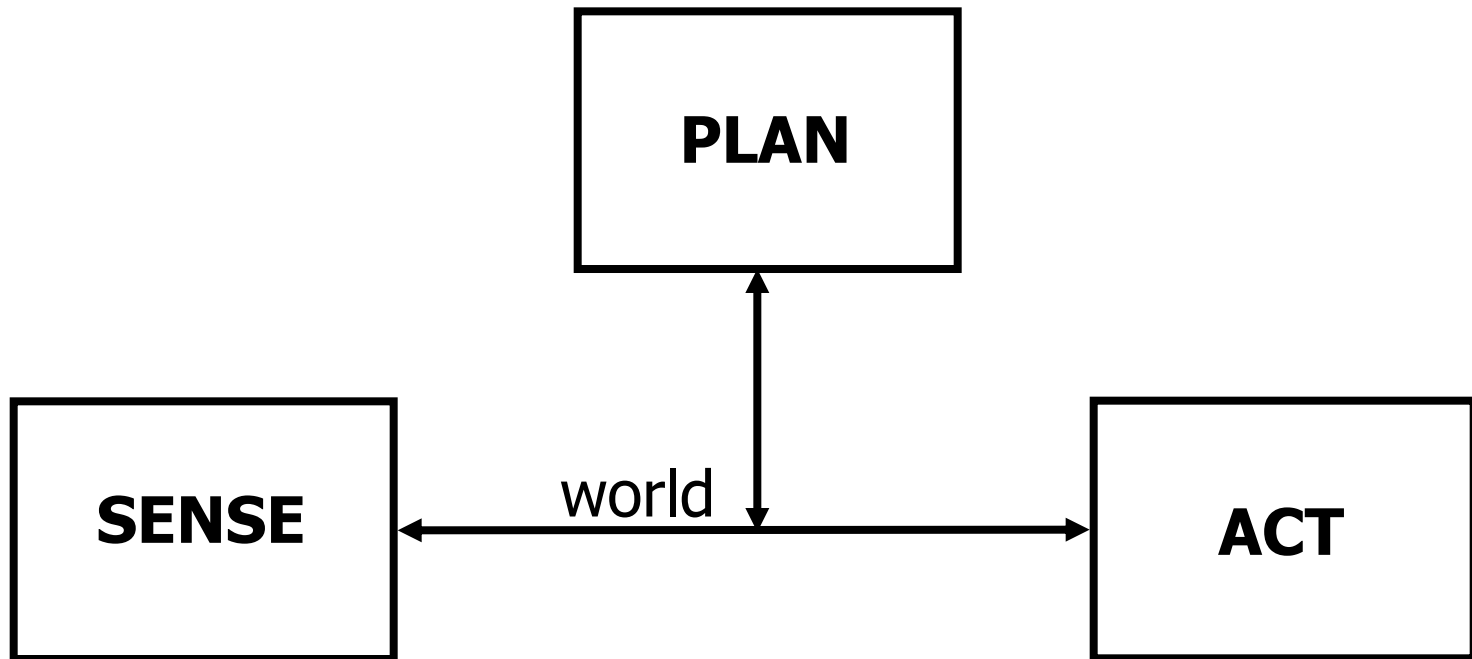
# Il paradigma reattivo

ROBOT PRIMITIVES	INPUT	OUTPUT
SENSE	Sensor data	Sensed information
PLAN	Information (sensed or cognitive)	Directives
ACT	Sensed Information or directives	Actuator commands

```
graph LR; S[Sensor data] --> SI[Sensed information]; SI --> SIDI[Sensed Information or directives]; SIDI --> AC[Actuator commands];
```

# Il paradigma ibrido deliberativo/reattivo

•Ibrido: P, S-A



# Il paradigma ibrido deliberativo/reattivo

ROBOT PRIMITIVES	INPUT	OUTPUT
PLAN	Information (sensed or cognitive) →	Directives
SENSE-ACT	Sensor data →	Actuator commands



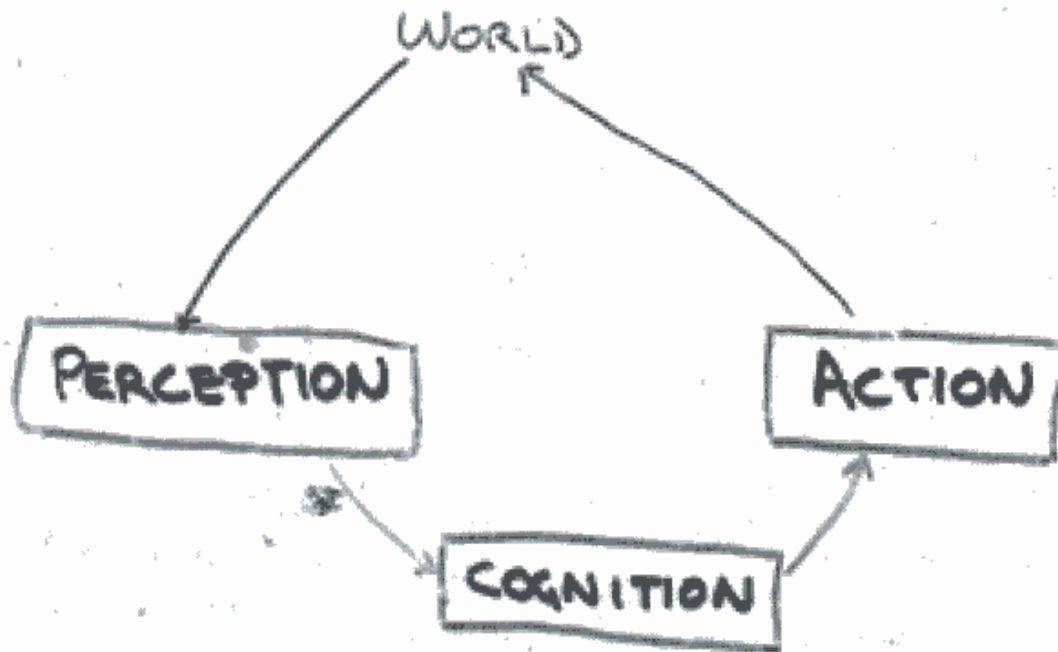


# Architetture di supervisione

---

- Strutture che mettono in relazione in maniera organizzata componenti di base e che definiscono le loro interazioni
- Criteri per la valutazione di architetture:
  - Modularità
  - Applicabilità
  - Portabilità
  - Robustezza

# Architetture Gerarchiche: Modello tradizionale per AI



**Figure 1:** The traditional model where cognition mediates between perceptions and plans of actions.



# Architetture Gerarchiche: Modello tradizionale per AI

---

- La cognizione è utilizzata per interpretare la percezione e per pianificare le azioni degli attuatori
- Il *modulo percezione* genera una descrizione del mondo, utilizzata dal *modulo cognizione* o modulo "pensante" che produce una sequenza di azioni per il *modulo azione*



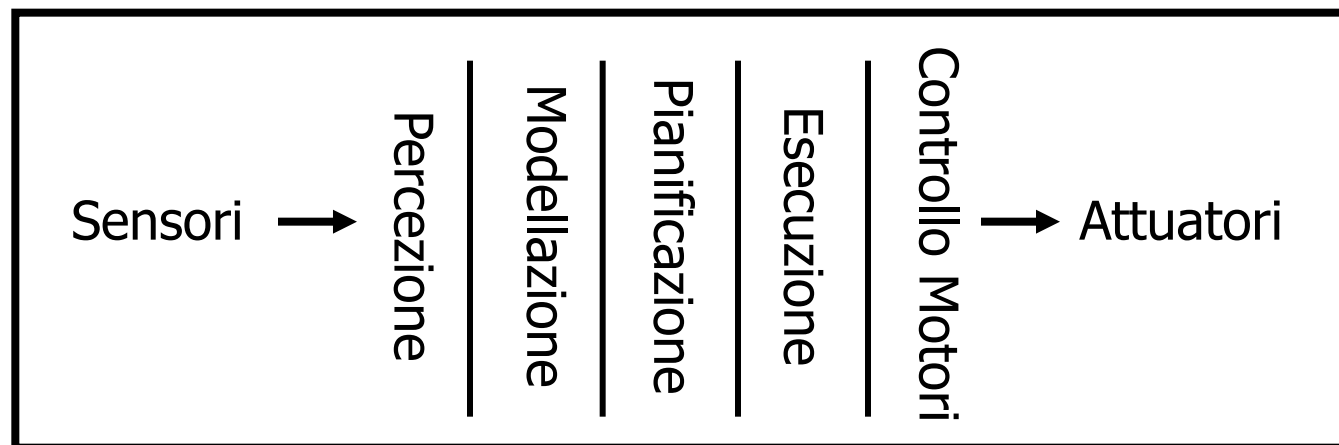
# I modelli Tradizionali: Architetture Gerarchiche

---

- Nelle architetture gerarchiche la percezione viene utilizzata per stabilire e mantenere una corrispondenza tra il **modello interno del mondo** (o mappa) e il mondo esterno.
- Tipicamente il modello del mondo contiene:
  - una rappresentazione a priori dell'ambiente in cui il robot opera
  - l'informazione sensoriale percepita
  - altre informazioni necessarie per l'esecuzione del compito
- La rappresentazione del mondo viene modificata ogni volta che il robot percepisce l'ambiente e il piano delle azioni viene stabilito sulla base di tale rappresentazione

# I modelli Tradizionali: Architetture Gerarchiche

- Suddivisione e distribuzione logica e funzionale dei compiti
- Suddivisione orizzontale e sequenziale della catena di informazioni elaborate dal sistema centralizzato





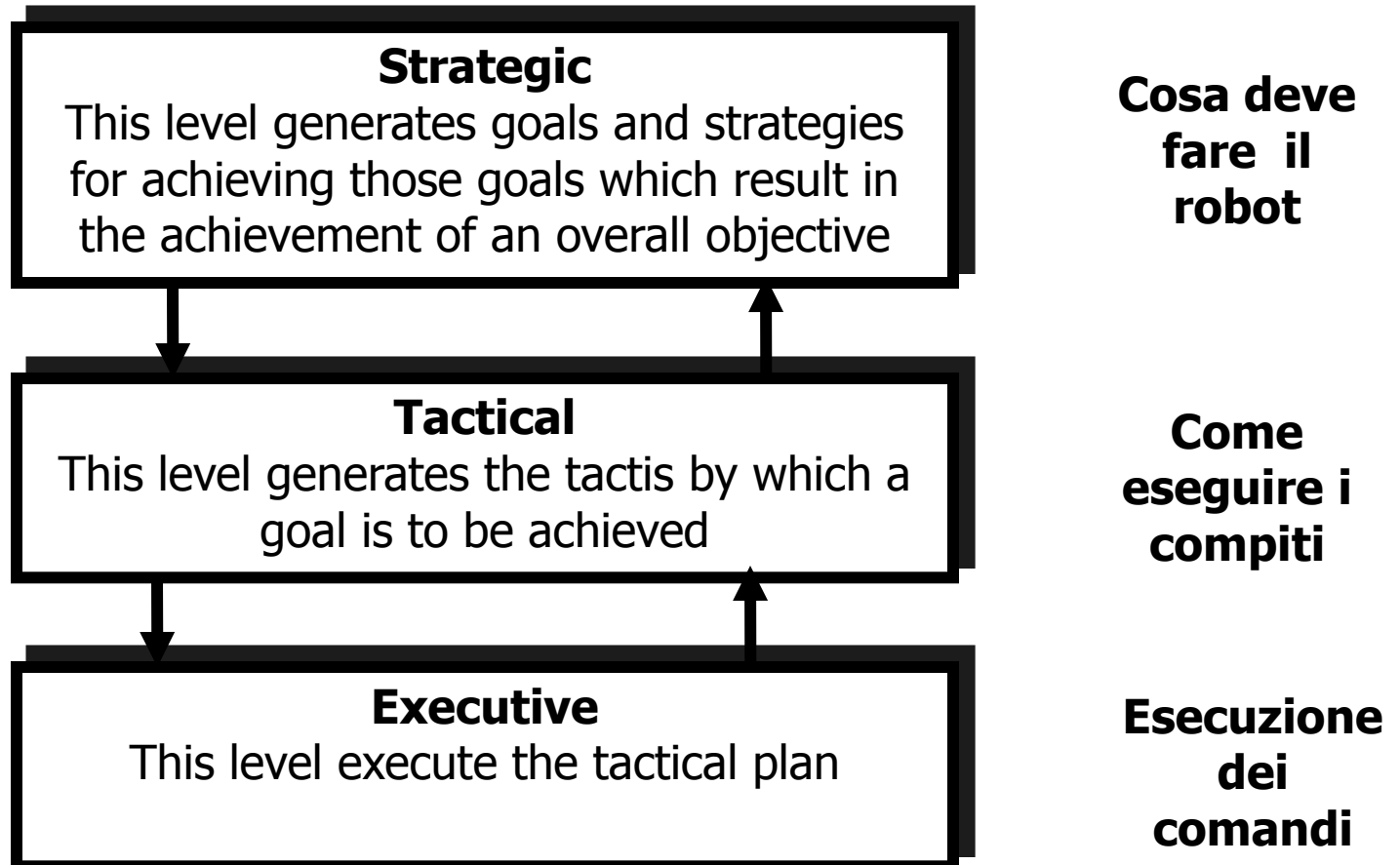
# Architetture Gerarchiche

---

- Generalmente, il modulo Pianificatore è strutturato su tre livelli:
  - Strategico
  - Tattico
  - Esecutivo
- Il livello più alto o strategico genera una strategia in base al compito che il robot deve eseguire
- Il livello intermedio o tattico genera i comandi interpretando istruzioni provenienti dal livello più alto o livello strategico
- Il livello più basso o livello esecutivo riceve macro comandi generati dal livello intermedio e si occupa del controllo in tempo reale degli attuatori



# Architetture Gerarchiche



Architettura su tre livelli per il pianificatore del sistema



# Architetture Gerarchiche

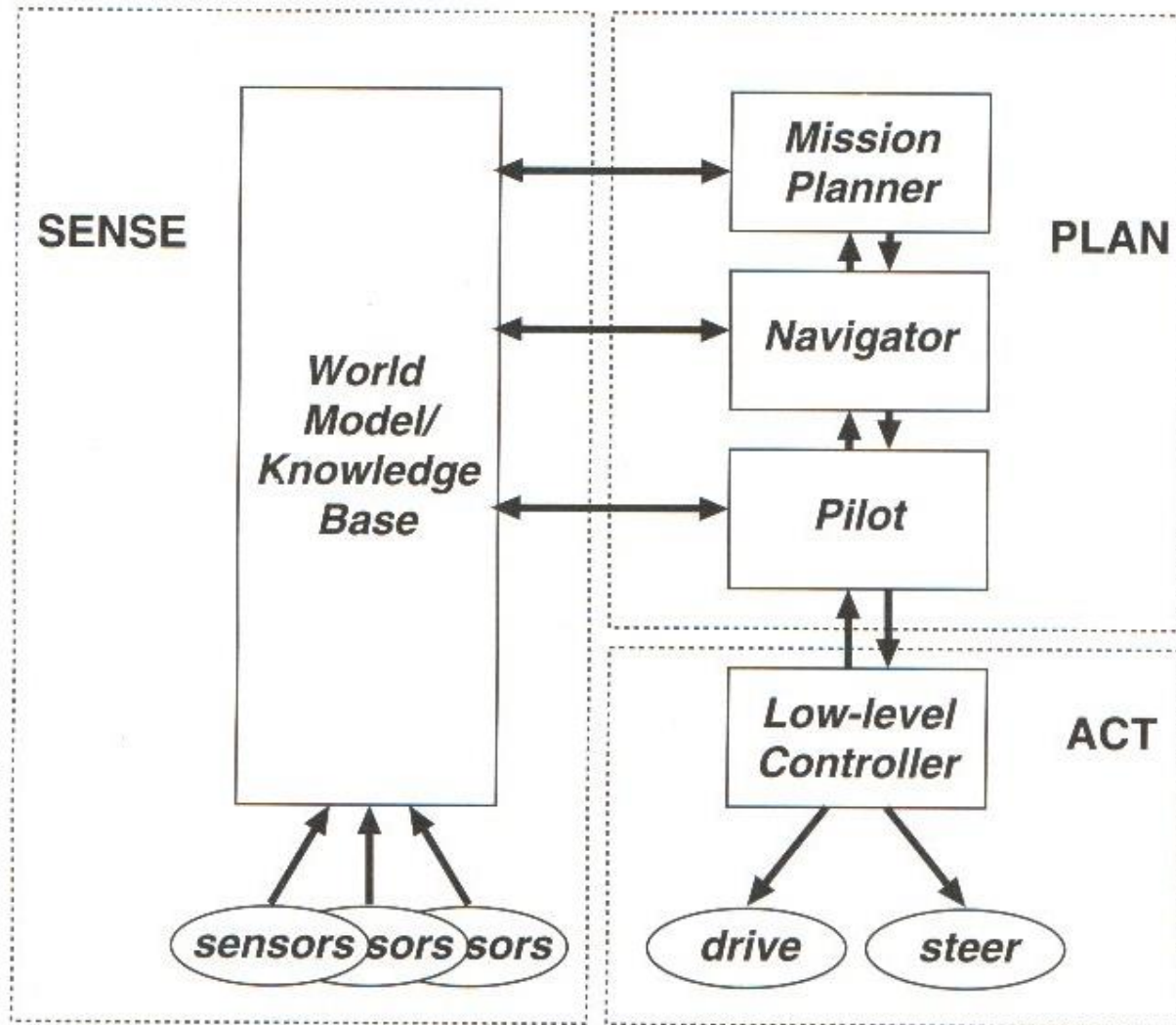
---

Esempio per il task: *"prendi la bottiglia dal frigorifero"*

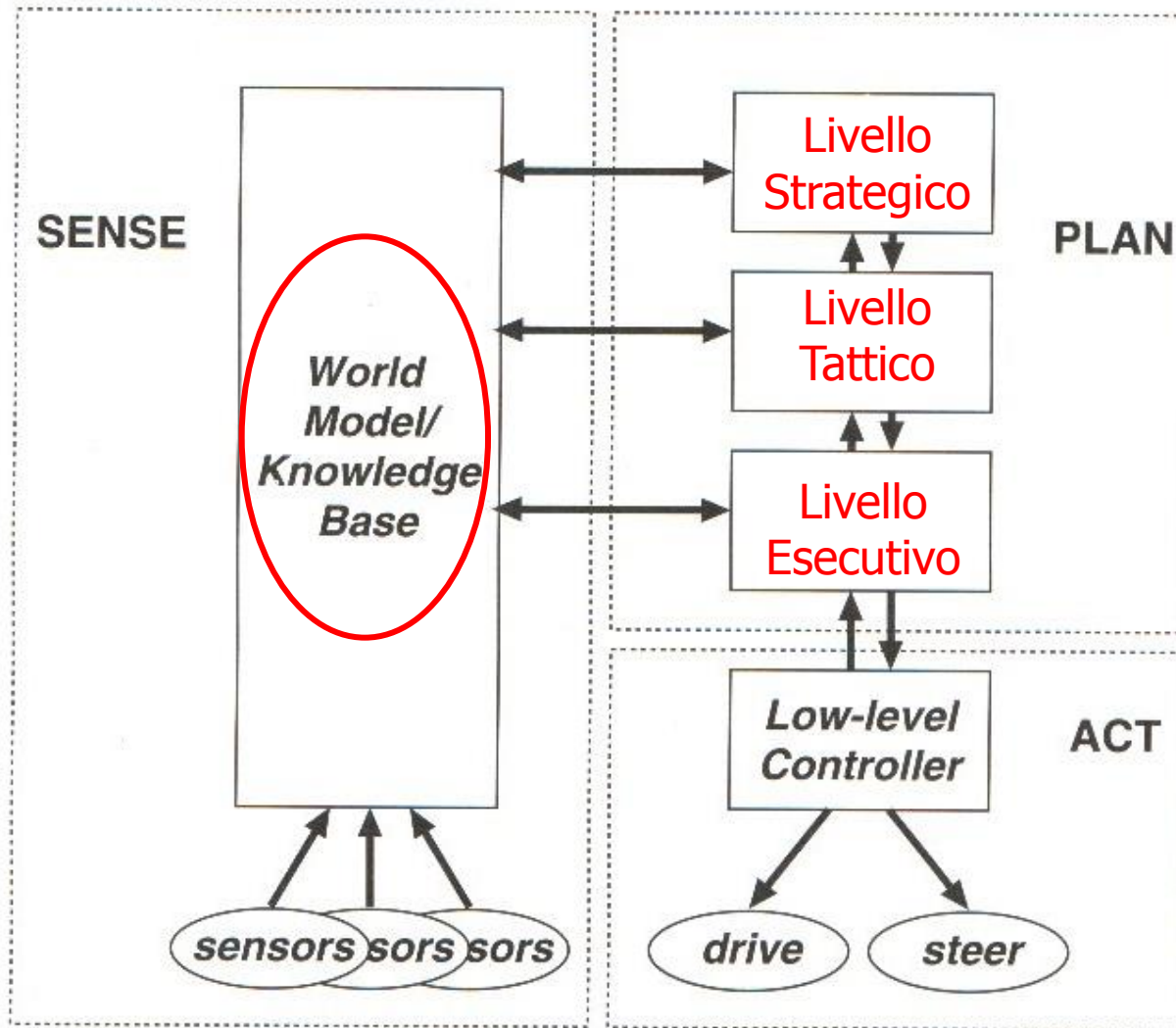
- **Livello strategico:** vai in cucina, vai davanti al frigorifero, apri al frigorifero, prendi la bottiglia...
- **Livello tattico:**
  - Vai in cucina: `muovi_base(X1,Y1); muovi_base(X2,Y2)...`
  - Apri il frigorifero: `muovi_braccio(P1), Apri_Mano()....`
- **Livello esecutivo:**
  - `Muovi_base(X1,Y1); muovi_base(X2,Y2); muovi_braccio(P1)...`



# Nested Hierarchical Controller

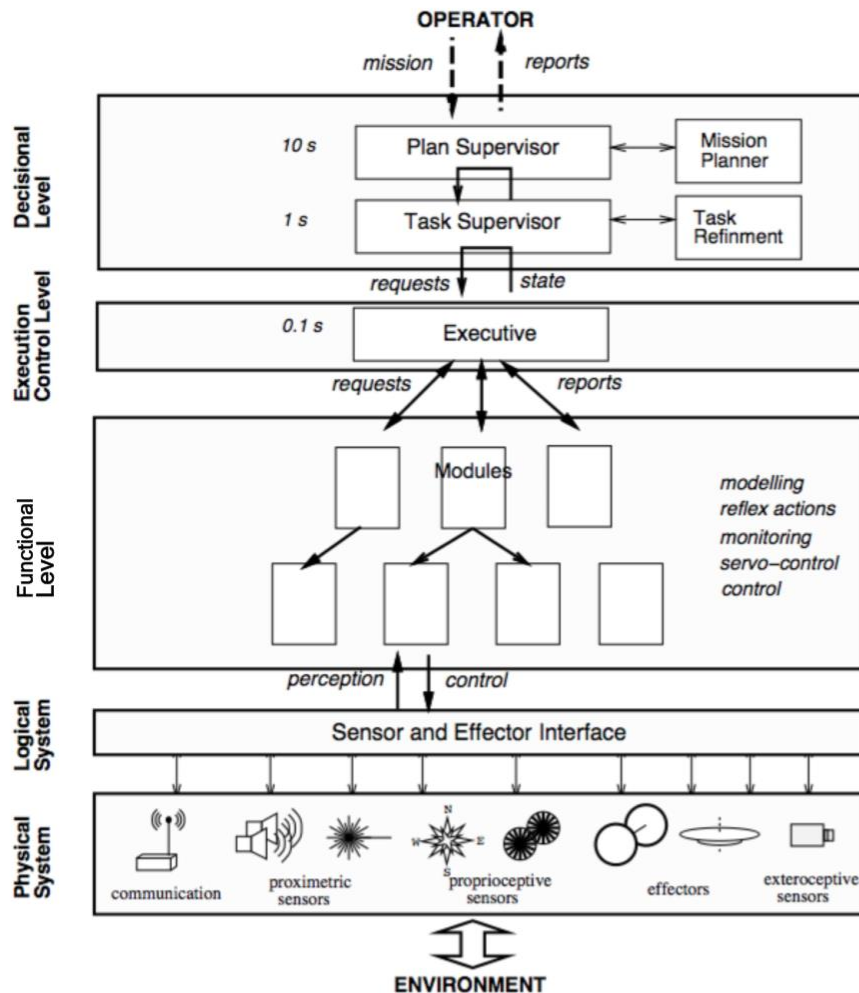


# Nested Hierarchical Controller





# LAAS architecture



- alternative example by LAAS/CNRS in Toulouse
- five levels
  - decision
  - execution (synchronization)
  - functional (modules)
  - logical for interface
  - physical devices

R. Alami *et al.*  
 "An Architecture for Autonomy,"  
*Int. J. of Robotics Research*, 1998



# Architetture Gerarchiche

---

## **Vantaggi**

- Funzionamento predicibile ovvero pianificazione a priori dei comportamenti
- Efficienza e stabilità del sistema



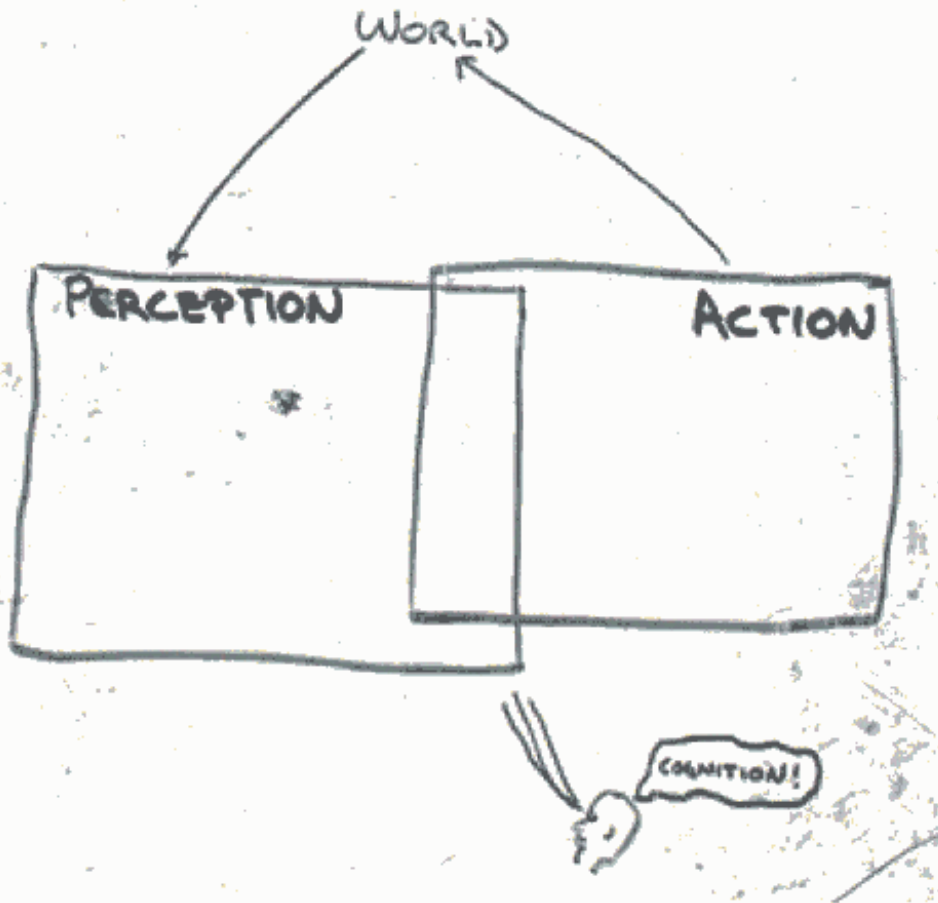
# Architetture Gerarchiche

---

## **Svantaggi**

- Alta complessità computazionale dovuta principalmente alla modellazione dell'ambiente e al ragionamento
- Poca adattabilità alle modifiche in tempo reale dell'ambiente o bassa *reattività*
- Basso parallelismo

# Architetture di controllo Reattive o basate sul comportamento



Non esiste un modulo "Cognizione" vero e proprio e i sistemi di percezione e attuazione collaborano per definire i comportamenti del robot

**Figure 2:** The new model, where the perceptual and action subsystems are all there really is. Cognition is only in the eye of an observer. *Tratto da A. Brooks, "Cambrian Intelligence"*



Alcuni esempi tratti da...

---

# **Vehicles Experiment in Synthetic Psychology**

di Valentino Braitenberg  
The MIT Press

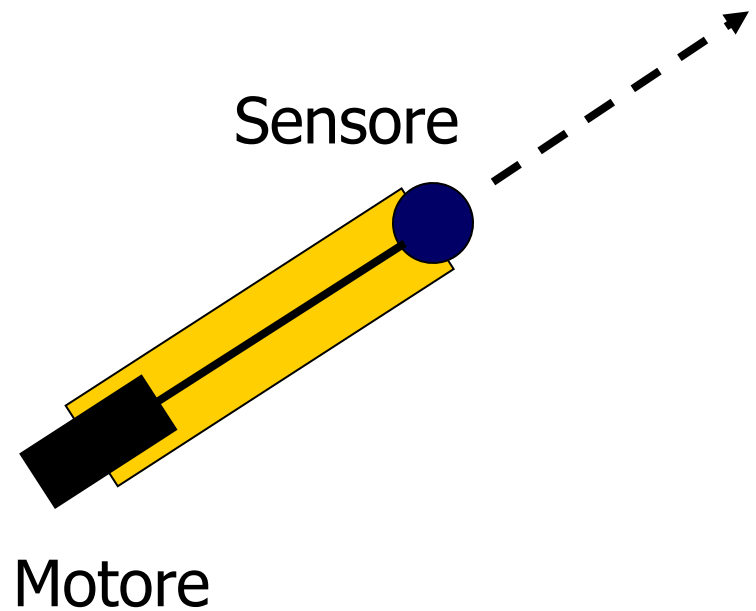
Director of the  
Max Planck Institute For  
Biological Cybernetics





# Esperimento 1

- Il primo veicolo è equipaggiato con un motore ed un sensore di temperatura: il motore è collegato direttamente al sensore
- Il collegamento è tale che la velocità del motore è proporzionale alla temperatura misurata dal sensore
- Il veicolo si muove sempre lungo la stessa direzione, più velocemente nelle zone calde e più lentamente nelle zone fredde





# Esperimento 2: paura e aggressione

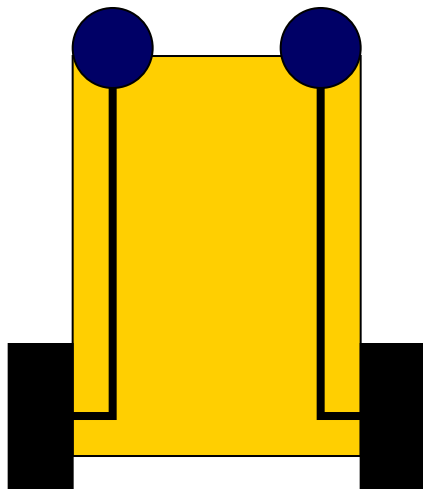
---

- Il secondo veicolo è equipaggiato con due motori e due sensori, due coppie diverse per ogni lato del veicolo
- Ancora una volta i motori sono collegati ai sensori ed il collegamento è tale che la forza esercitata dal motore è proporzionale al valore misurato dal sensore
- Seguendo questo schema è possibile costruire due veicoli che assumono due comportamenti diversi a seconda del tipo di connessione sensori-motori

# Esperimento 2: paura e aggressione

**Veicolo 1**

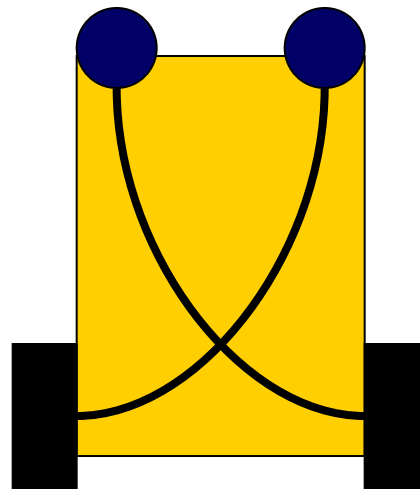
Sensori



Motori

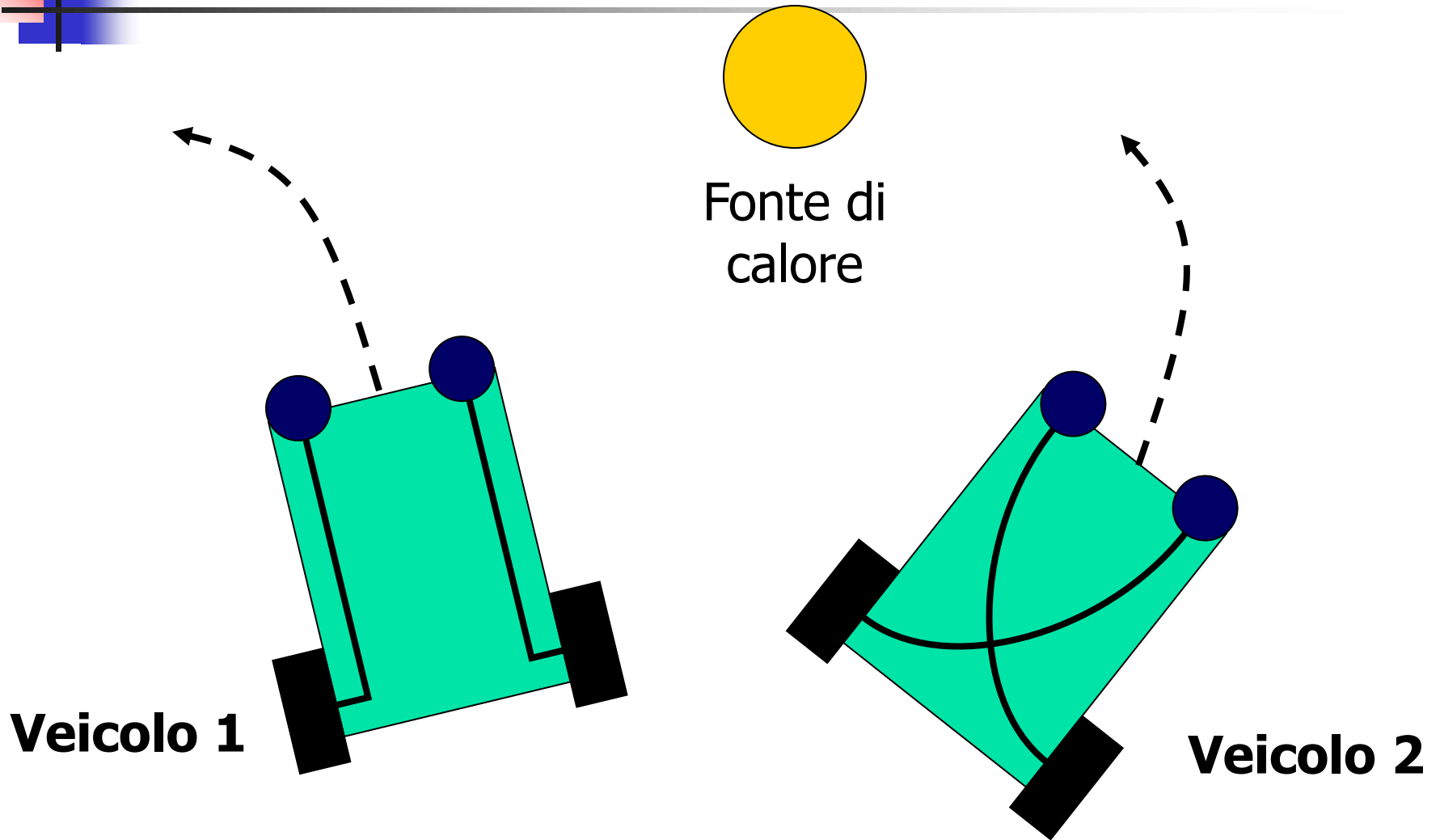
**Veicolo 2**

Sensori



Motori

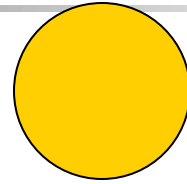
# Esperimento 2: paura e aggressione





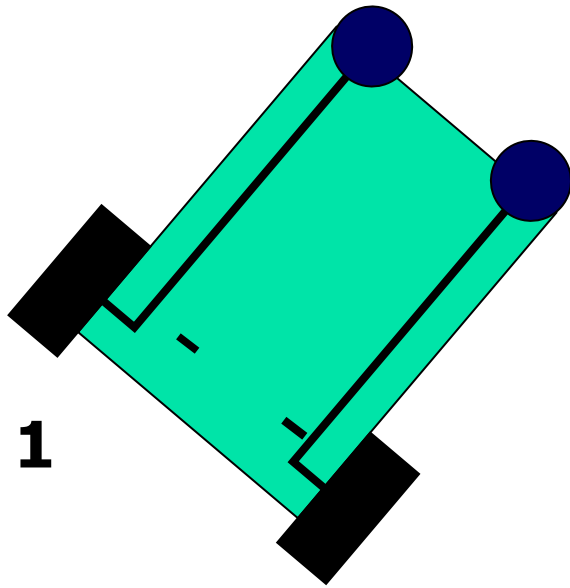
# Esperimento 3: amore

---

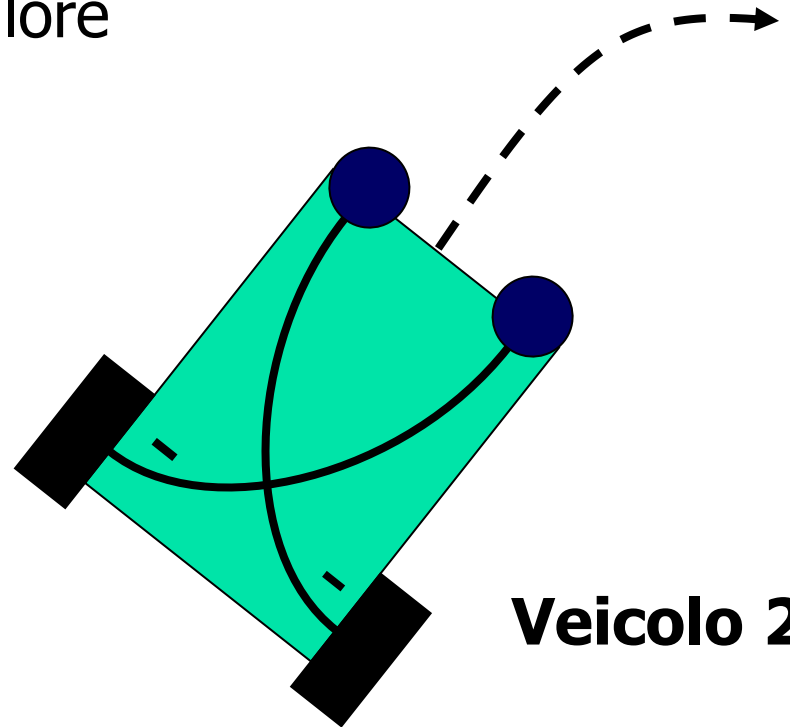


Fonte di calore

**Veicolo 1**



**Veicolo 2**



la velocità del motore è  
inversamente proporzionale al  
valore misurato dal sensore



# Architetture di controllo Reattive o basate sul comportamento

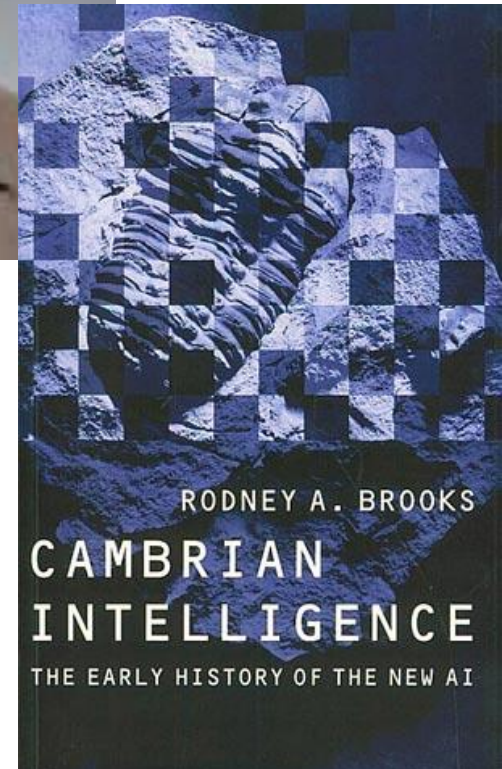
---

- I comportamenti del robot sono reazioni alle informazioni percepite dall'ambiente
- Il modulo di base di tali sistemi è costituito quindi da un **comportamento (behaviour)** che è ottenuto da una relazione diretta tra sensori e attuatori
- Si parla di **behaviour-based robotics** o **sistemi reattivi**, ovvero sistemi capaci di rispondere in tempo reale agli stimoli provenienti dall'ambiente circostante

# Rodney Brooks

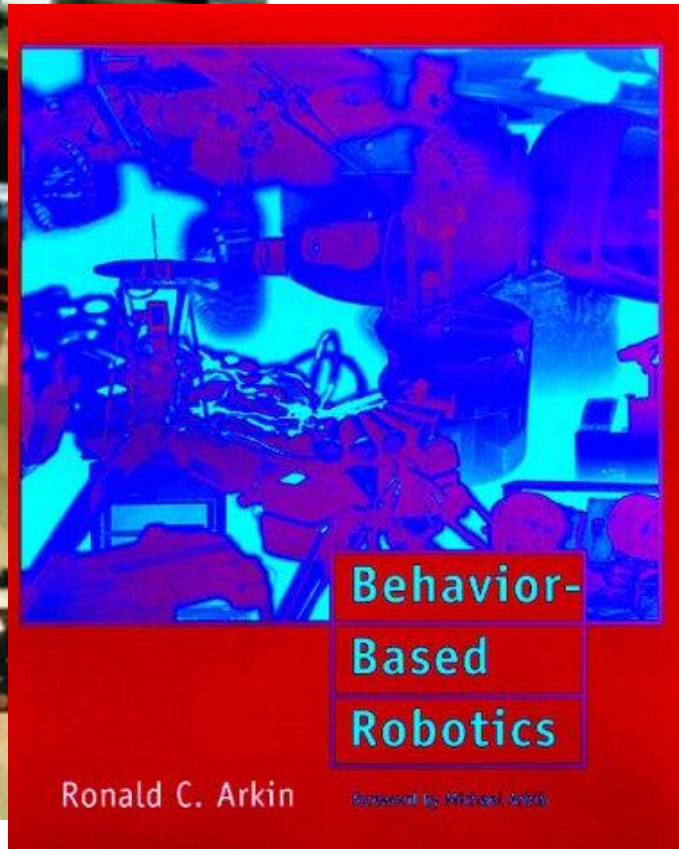
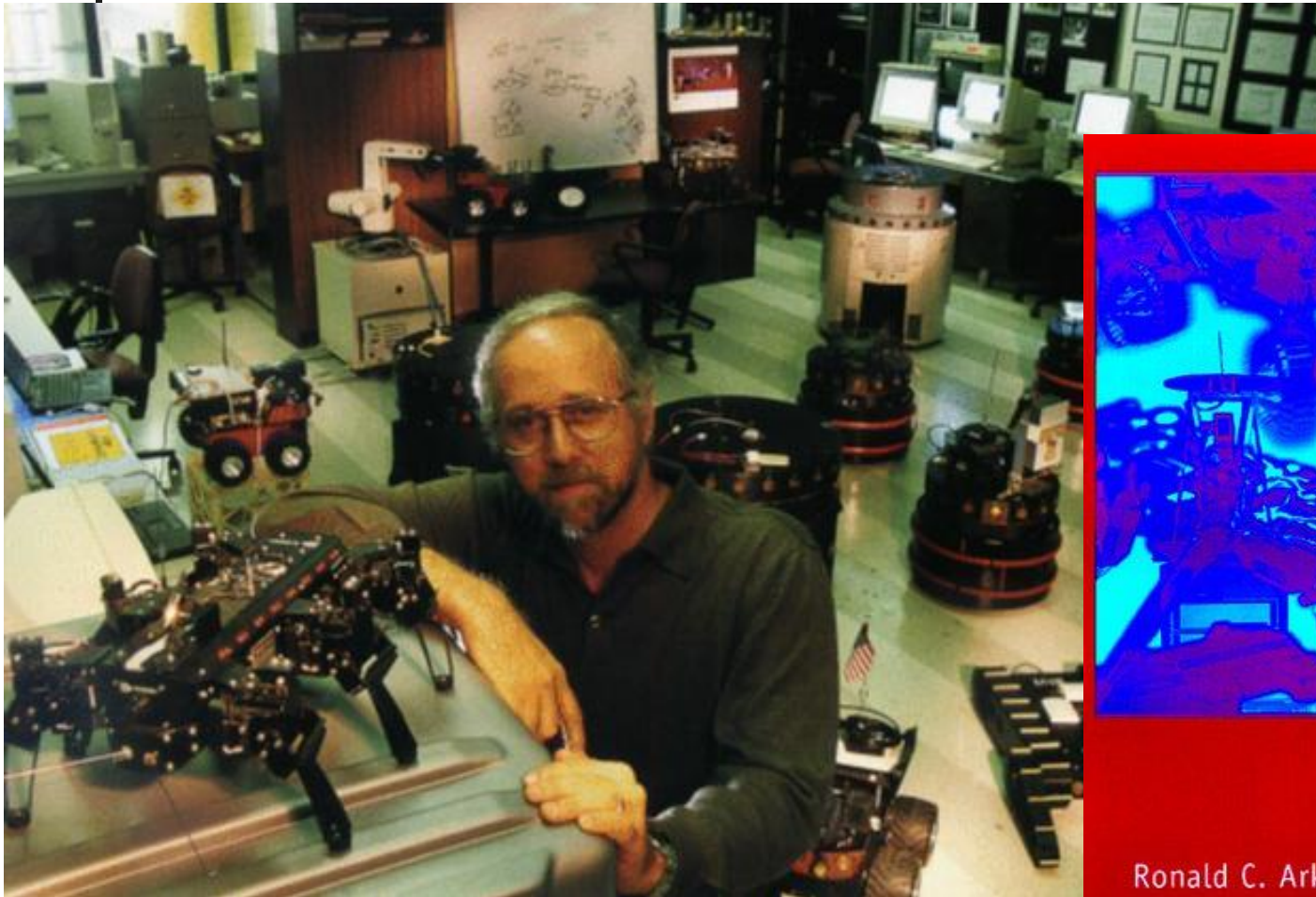
## Computer Science and Artificial Intelligence Laboratory, MIT

*now iRobot*



# Ronald Arkin

## Georgia Institute of Technology





# Dalle architetture gerarchiche a quelle reattive

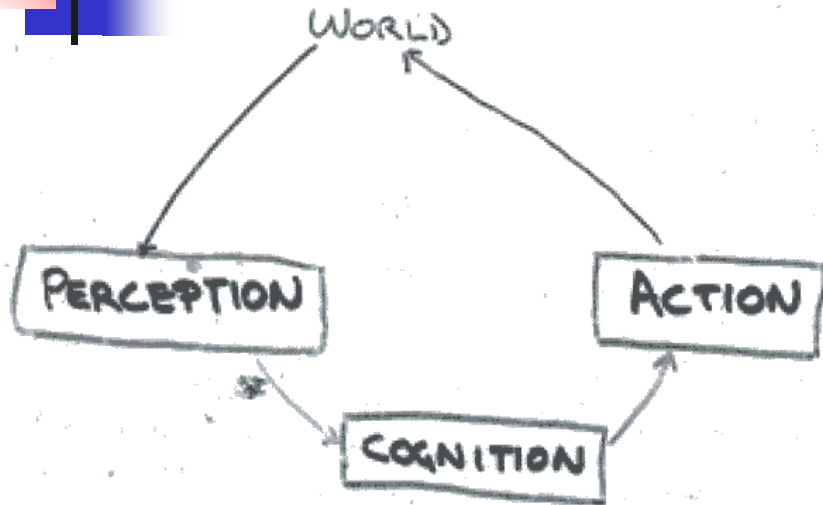


Figure 1: The traditional model where cognition mediates between perceptions and plans of actions.

deliberative, model-based

reactive, behavior-based

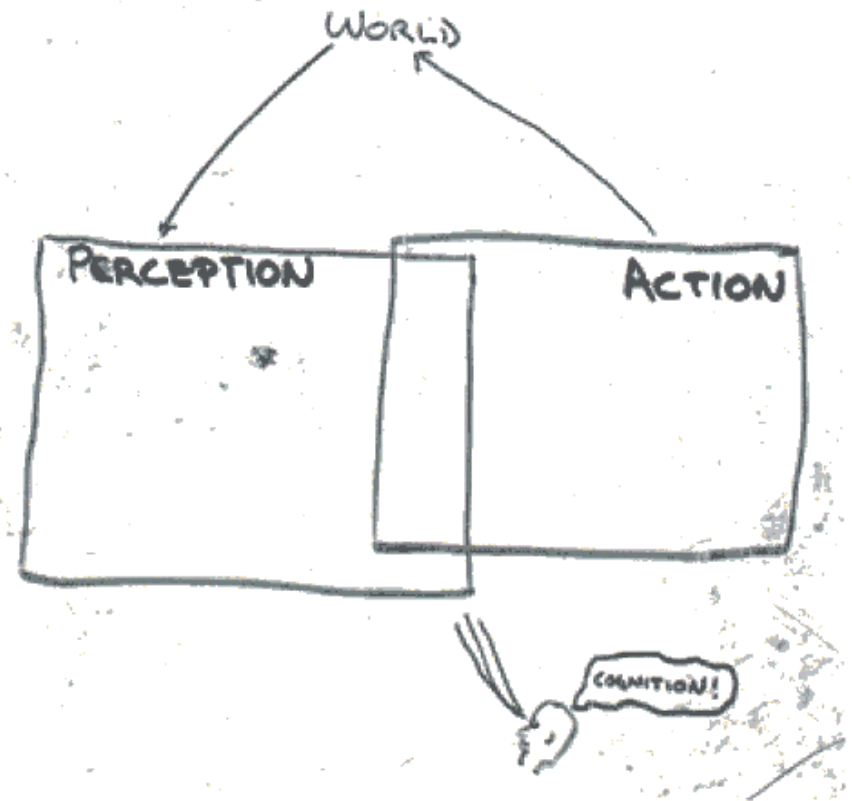


Figure 2: The new model, where the perceptual and action subsystems are all there really is. Cognition is only in the eye of an observer.



# Architetture di controllo Reattive o basate sul comportamento

---

- Il robot interagisce con il mondo attraverso sensori e attuatori
- **Non esiste una rappresentazione del mondo**, (“The world is the best model” di R. A. Brooks, 1986): la conoscenza del mondo non è né modellizzata né memorizzata nel robot ma è estratta in tempo reale dal mondo stesso attraverso i sensori
- Poiché non esiste un modello del mondo, di conseguenza non esiste pianificazione a priori delle azioni del robot

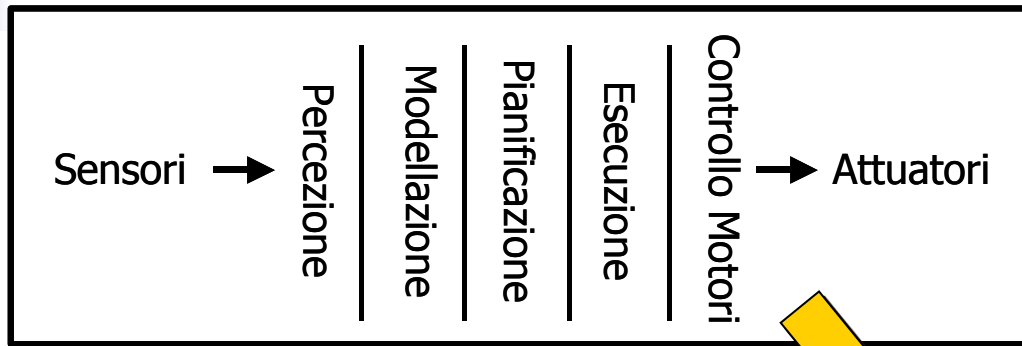


# Architetture di controllo Reattive o basate sul comportamento

---

- Nelle architetture di controllo reattive l'esecuzione di un compito è suddivisa tra moduli ognuno dei quali ha assegnata una specifica competenza e attua un determinato comportamento del robot
- Il comportamento complessivo del robot è determinato dall'insieme dei comportamenti presenti

# Dalle architetture gerarchiche alle architetture reattive

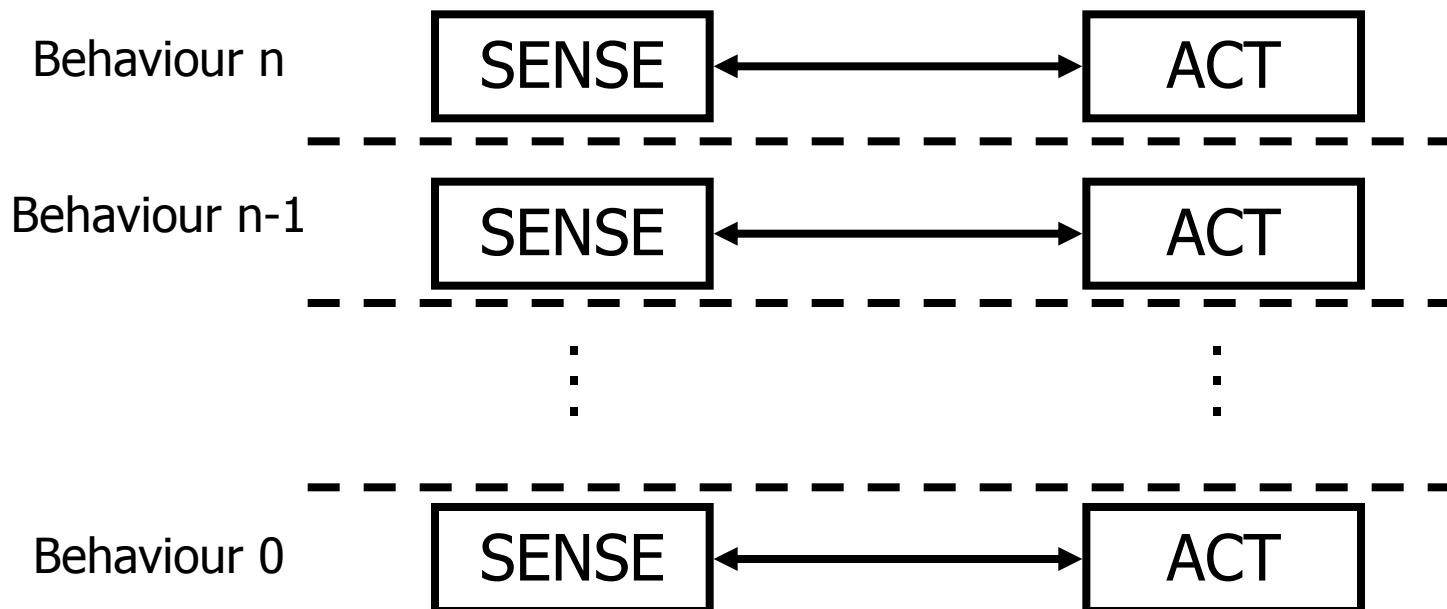


Dalla suddivisione *orizzontale e sequenziale* della catena di informazioni alla suddivisione *verticale e parallela*



# Architetture di controllo Reattive o basate sul comportamento

- Decomposizione verticale che produce flussi multipli di informazione ciascuno relativo ad una particolare funzione assegnata al robot.
- In questo modo, ogni sequenza si occupa di uno specifico aspetto nel funzionamento globale del sistema e può svilupparsi parallelamente ad altri processi





# Architetture di controllo Reattive o basate sul comportamento

---

- **Principio di indipendenza:** i vari moduli devono essere mutuamente indipendenti tra loro. Conseguenza immediata all'applicazione di questo principio è l'impossibilità di mantenere un modello del mondo completo, condivisibile tra tutti i moduli.
- **Principio di località:** ciascun sottocompito richiede, per completarsi, solo una parte limitata di tutta l'informazione sensoriale disponibile. Il robot risponde solo ad eventi del mondo senza mantenere stati persistenti: la memoria di cui ha bisogno è realizzata leggendo direttamente la situazione ambientale che gli indica il modo operativo corrente.



# Architetture di controllo Reattive o basate sul comportamento

---

## **Vantaggi**

- Non esiste un modello del mondo
- Alta adattabilità alle modifiche dell'ambiente (risposte in tempo reale)
- Bassa complessità di ogni livello e basso costo computazionale complessivo del sistema
- E' possibile avere parallelismo nel controllo
- L'estensione dei comportamenti è relativamente semplice



# Architetture di controllo Reattive o basate sul comportamento

---

## **Svantaggi**

- Difficoltà nel prevedere a priori il comportamento globale del robot
- Gestione della concorrenza tra moduli
- Aumentando i comportamenti si aumenta anche la complessità della gestione della concorrenza tra moduli con conseguente difficoltà nella risoluzione di conflitti





# Un esempio di architettura reattiva: architettura subsumption

---

Sistema di controllo per la navigazione  
senza collisioni di un robot mobile  
dotato di sensori ad ultrasuoni

R. A. Brooks, "A Robust Layered Control System  
for a Mobile Robot", in *Cambrian Intelligence*,  
The MIT Press

R.A. Brooks, "A Robust Layered Control System  
for a Mobile Robot", *IEEE Journal of Robotics  
and Automation*, Vol. Ra-2, No. 1, March 1986



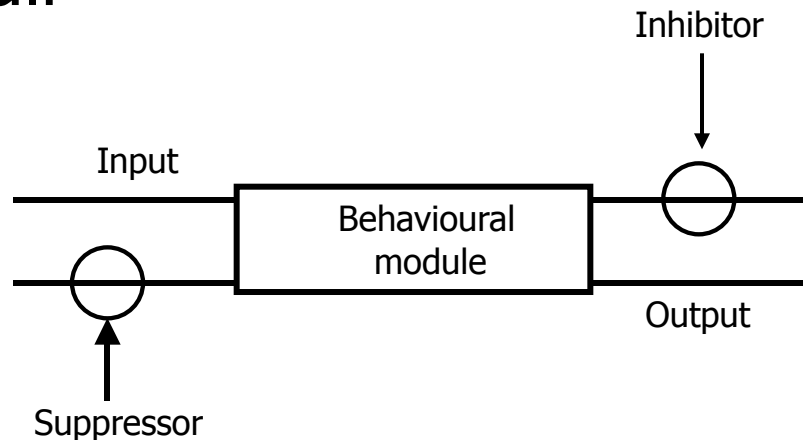
# Subsumption Architecture (R. Brooks)

---

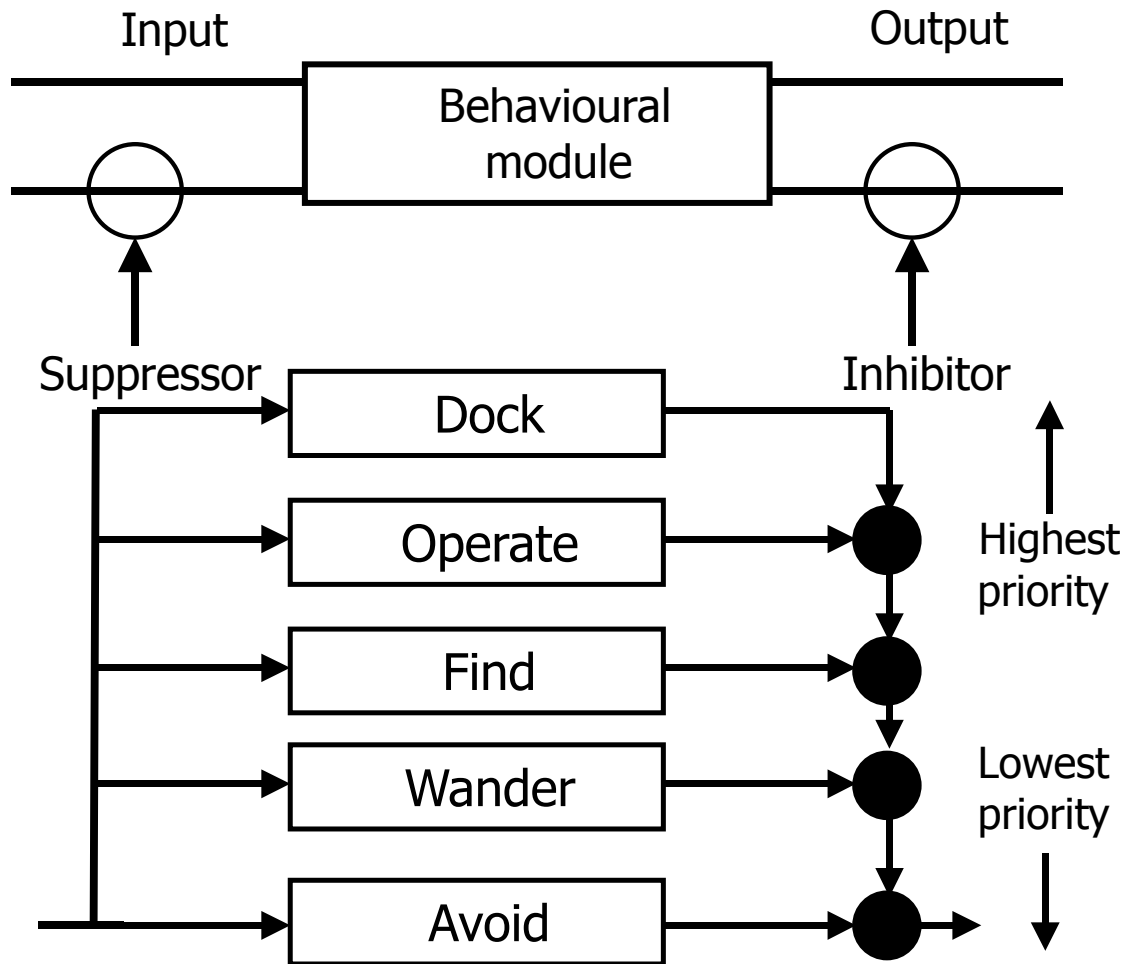
- Nelle architetture subsumption l'esecuzione di un compito è suddivisa tra moduli ognuno dei quali ha assegnata una specifica competenza e attua un determinato comportamento del robot
- I comportamenti sono organizzati in una architettura a strati ed hanno priorità diversa a seconda del loro livello
- Comportamenti a livello più alto (high priority) possono inibire le operazioni o sopprimere l'input dei comportamenti a livello più basso (low priority)

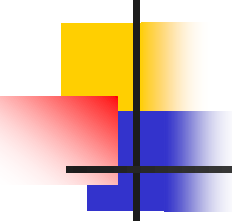
# Architettura Subsumption: soppressione e inibizione

- Ogni modulo ha delle linee di output e di input.
- Linee di output di un modulo possono essere collegate a linee di input o di output di altri moduli



# Subsumption Architecture (R. Brooks)





# Architettura Subsumption: i livelli di competenza dell'architettura

---

---

Reason about behavior of objects

---

Plan changes to the world

---

Identify objects

---

Monitor changes

---

Build maps

---

Explore

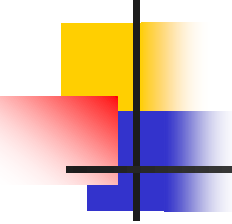
---

Wander

---

Avoid objects

---



# Architettura Subsumption: i livelli di competenza dell'architettura

---

- Livello 0: evitare gli ostacoli
- Livello 1: navigazione casuale nell'ambiente
- Livello 2: esplorazione del mondo e 'identificazione' dei punti di interesse
- Livello 3: costruzione della mappa dell'ambiente (relazioni tra punti)
- Livello 4: rilevazione dei cambiamenti dell'ambiente statico
- Livello 5: ragionamento sul mondo in termini di esecuzione di task in relazione al rilevamento di determinati oggetti
- Livello 6: formulazione ed esecuzione di comportamenti che determinano cambiamenti dello stato del mondo
- Livello 7: ragionamento sui comportamenti degli oggetti e modifica dei piani in accordo

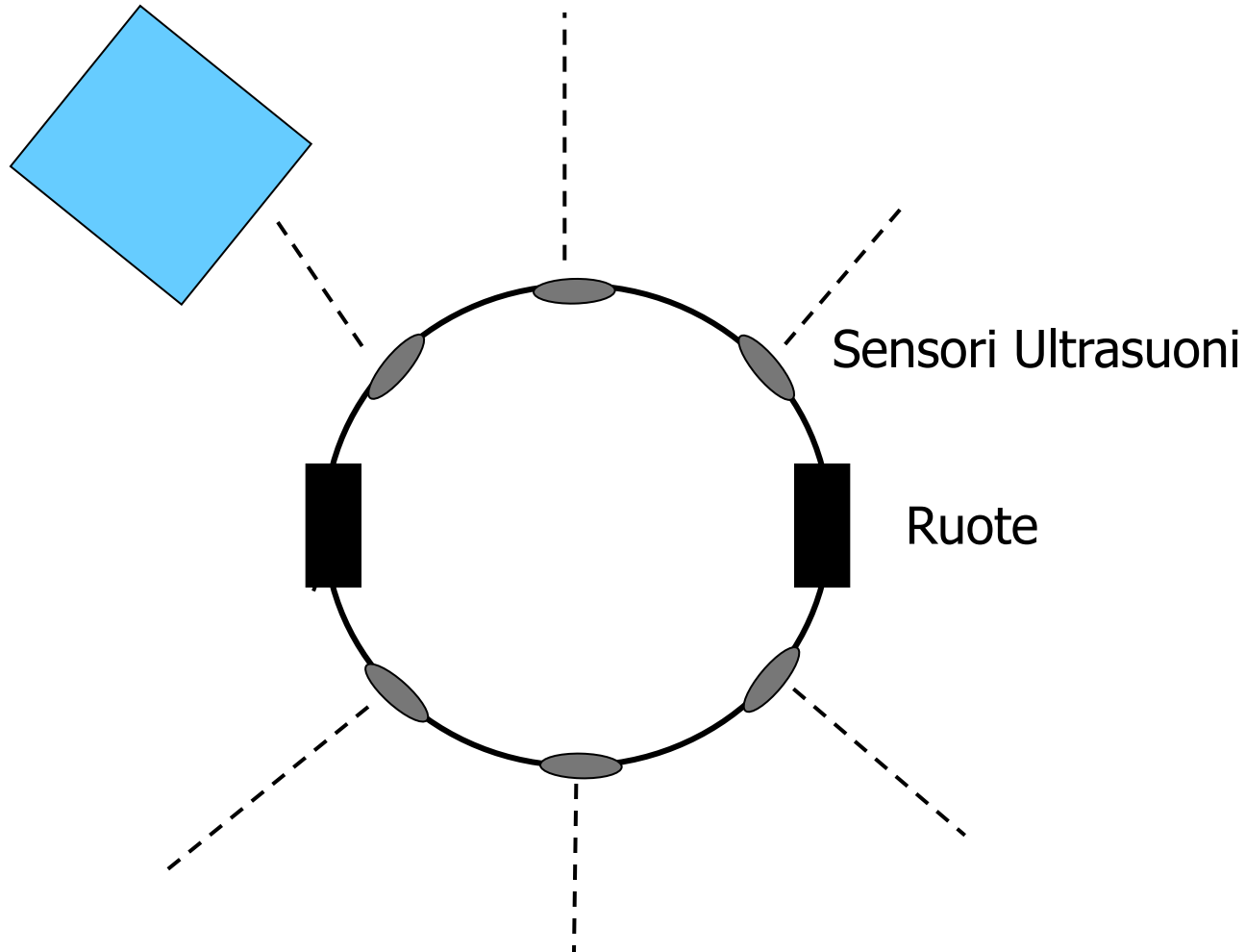


# Architettura Subsumption: implementazione del sistema

---

- Ogni strato dell'architettura è implementato con una macchina a stati finiti in grado di inviare messaggi alle altre e di gestire proprie strutture di dati
- Non esiste memoria globale condivisa
- Ogni macchina agisce in maniera asincrona rispetto agli altri, monitorando i propri input e inviando messaggi di output
- Input ai moduli possono essere soppressi e output inibiti dagli output di altri moduli (meccanismo tramite il quale strati superiori assumono il controllo degli stati inferiori)

# Architettura Subsumption: implementazione del sistema





# Sensori US: Misurazione della distanza in base al tempo di volo

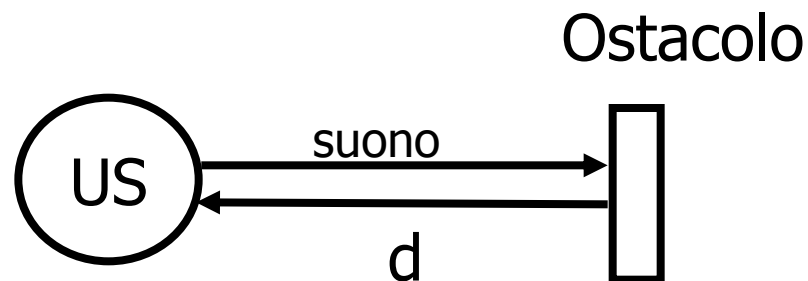
La misura della distanza di un oggetto è data dalla misura del tempo che un dato segnale (suono) impiega a raggiungere l'oggetto e a tornare indietro

$$d = (v \times t)/2$$

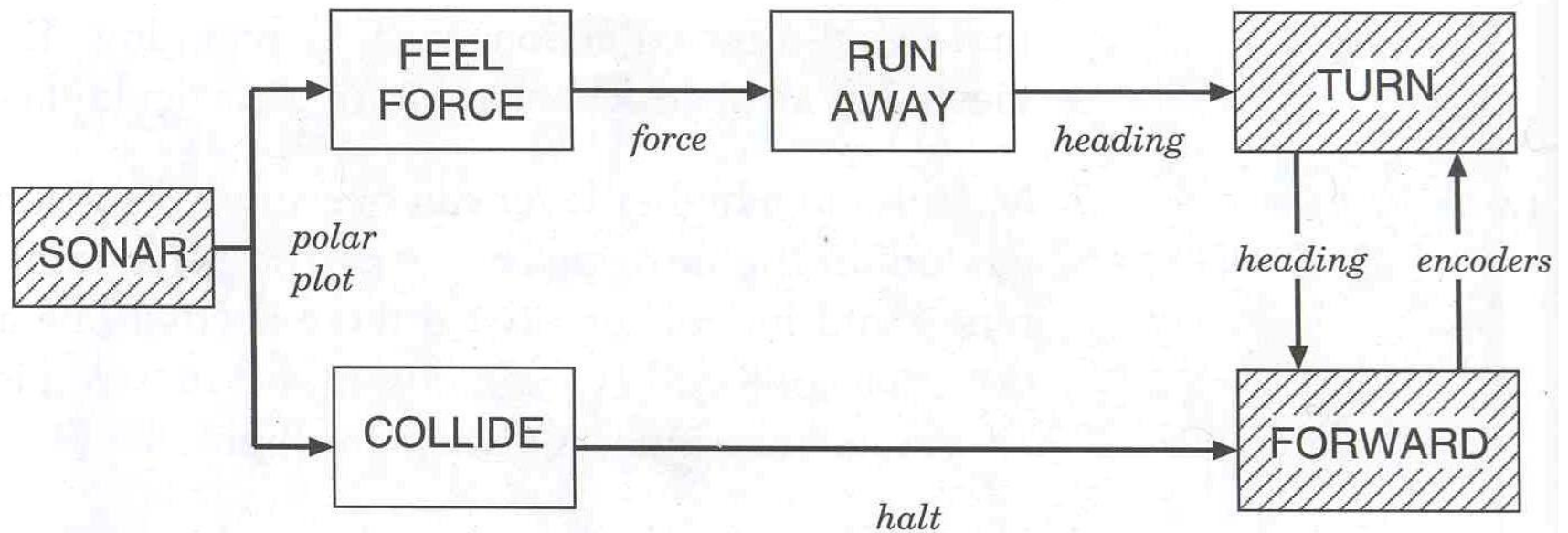
d = distanza dell'oggetto

v = velocità del segnale

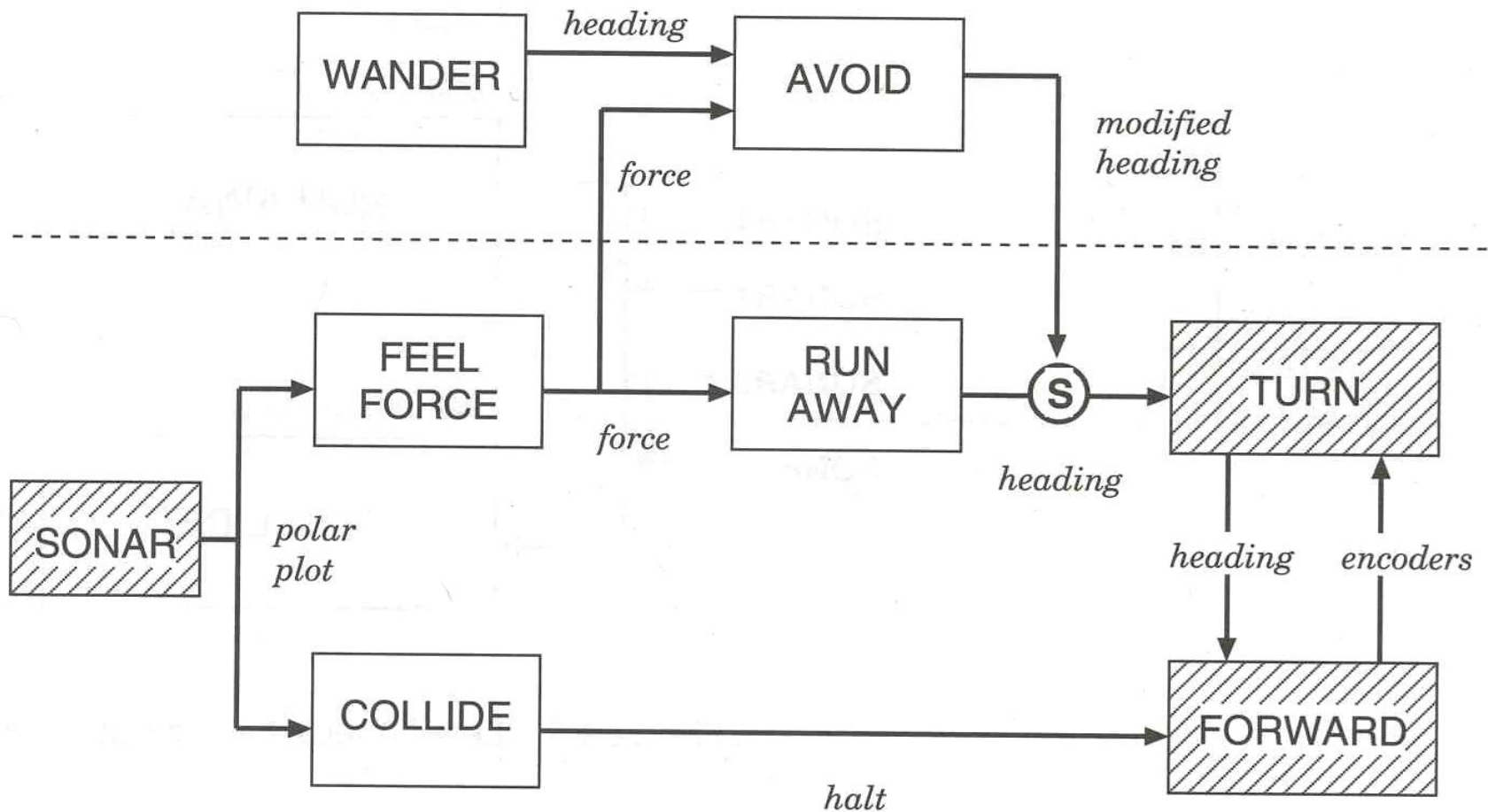
t = tempo impiegato dal segnale per raggiungere l'oggetto e tornare indietro



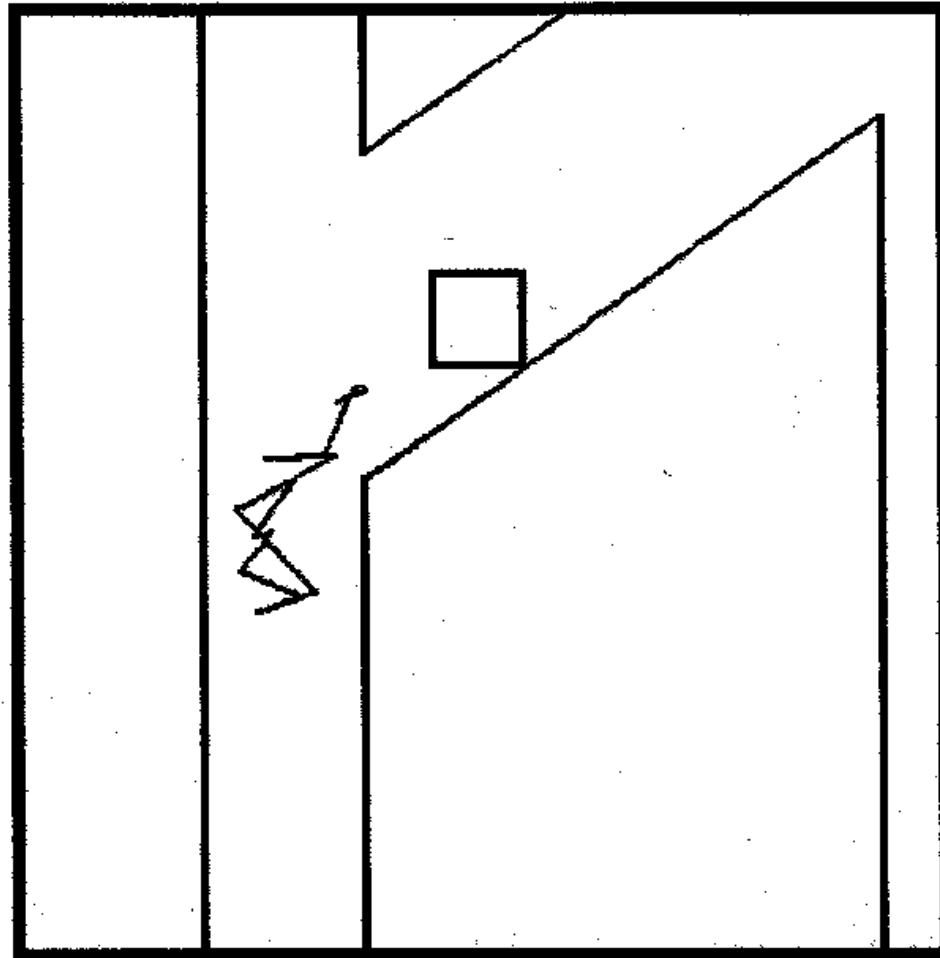
# Il livello 0 - Avoid



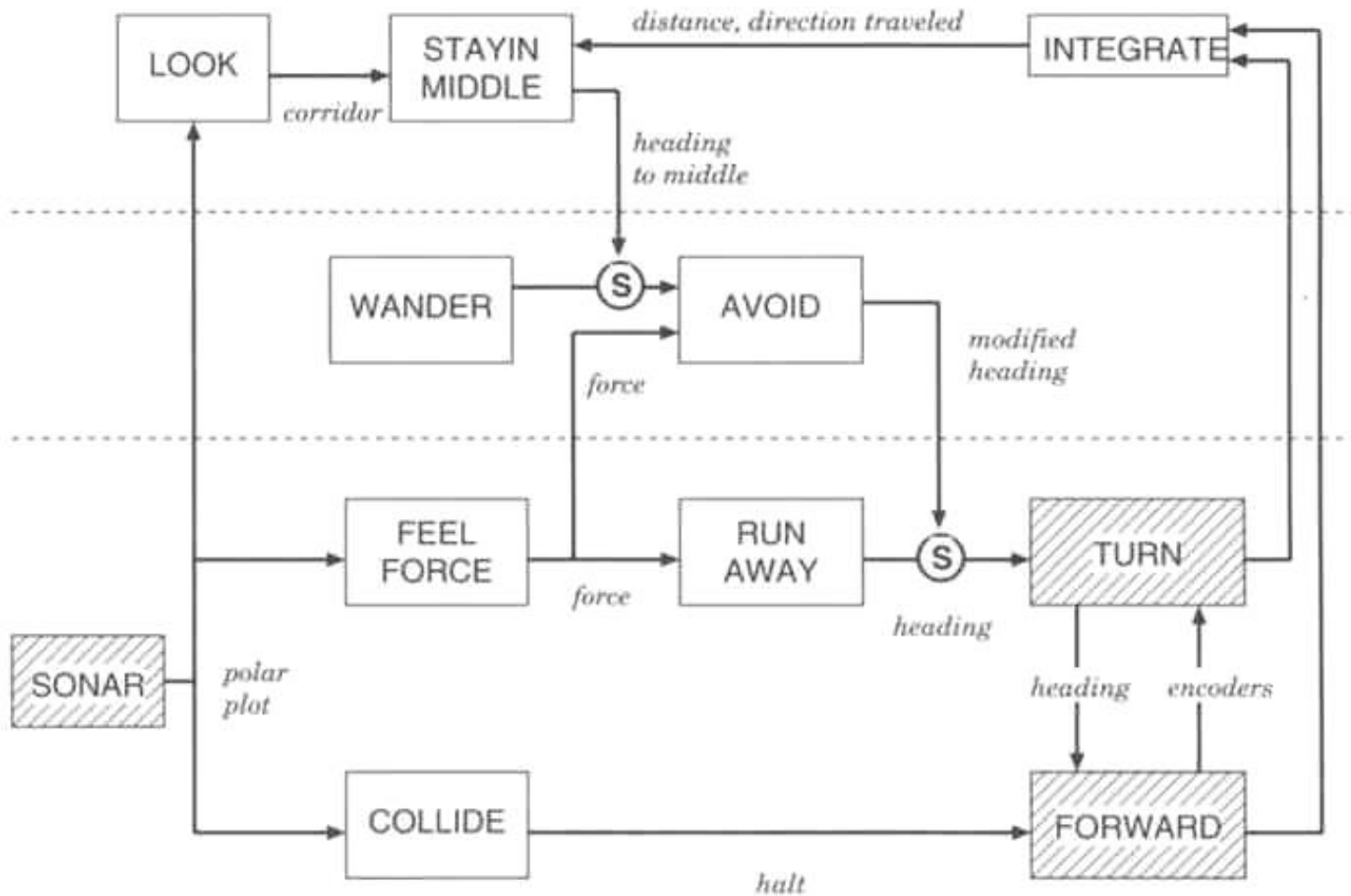
# Il livello 1 - Wander



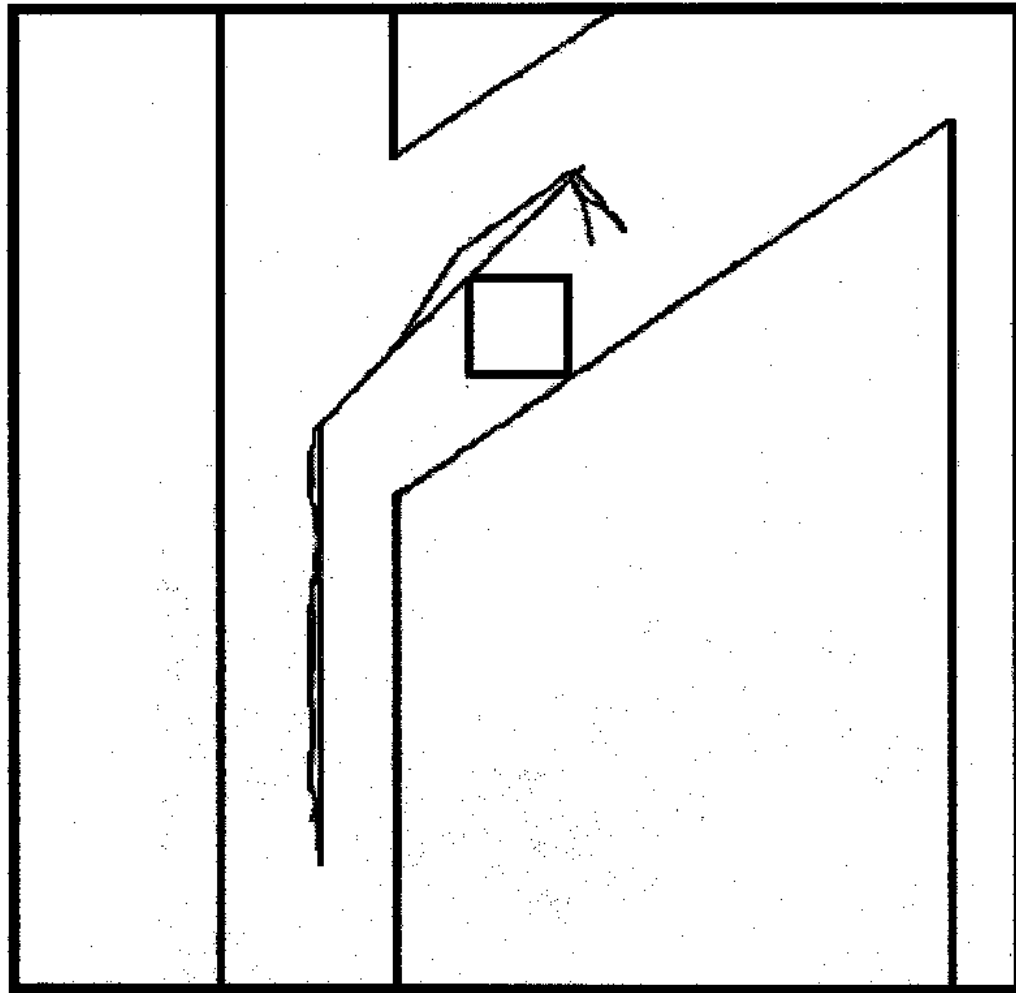
# Risultati delle simulazioni del livello 0 e 1



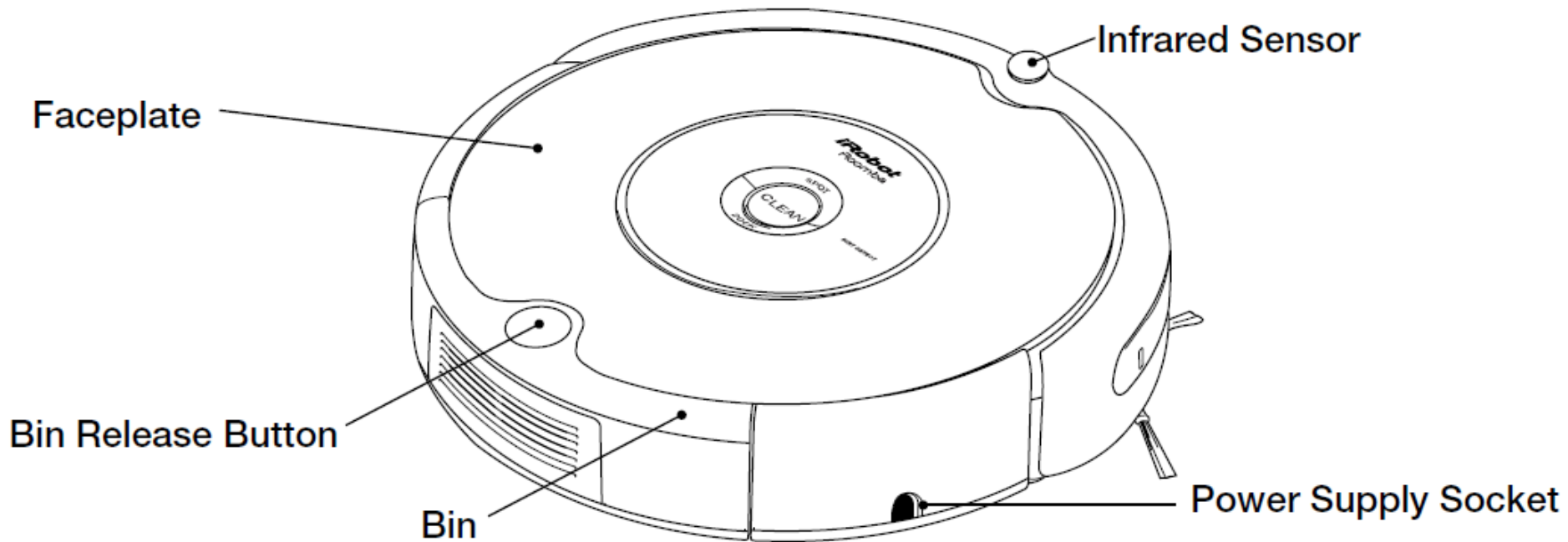
# Il livello 2 - Explore



# Risultati delle simulazioni del livello 0,1 e 2



# iRobot Roomba



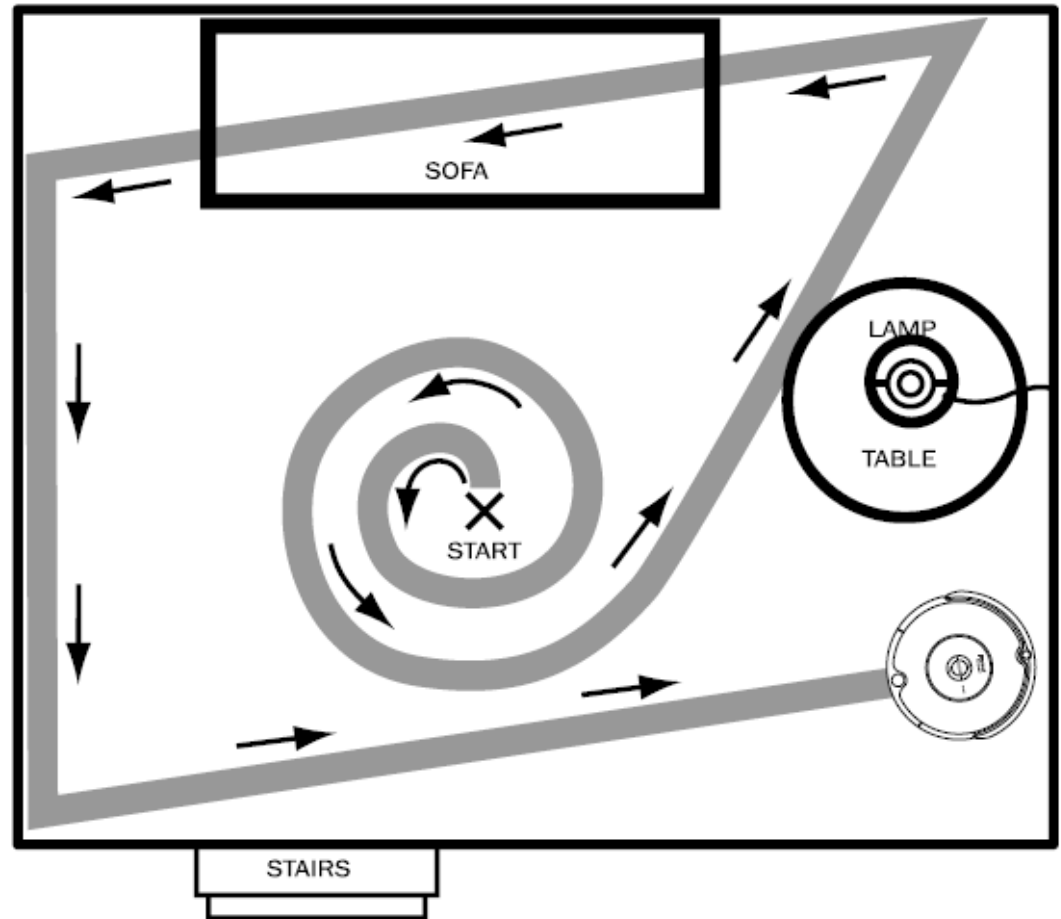
# iRobot Roomba – comportamenti

**Spiraling:** Roomba uses a spiral motion to clean a concentrated area.

**Wall Following:** Roomba uses this technique to clean the full perimeter of the room and navigate around furniture and obstacles.

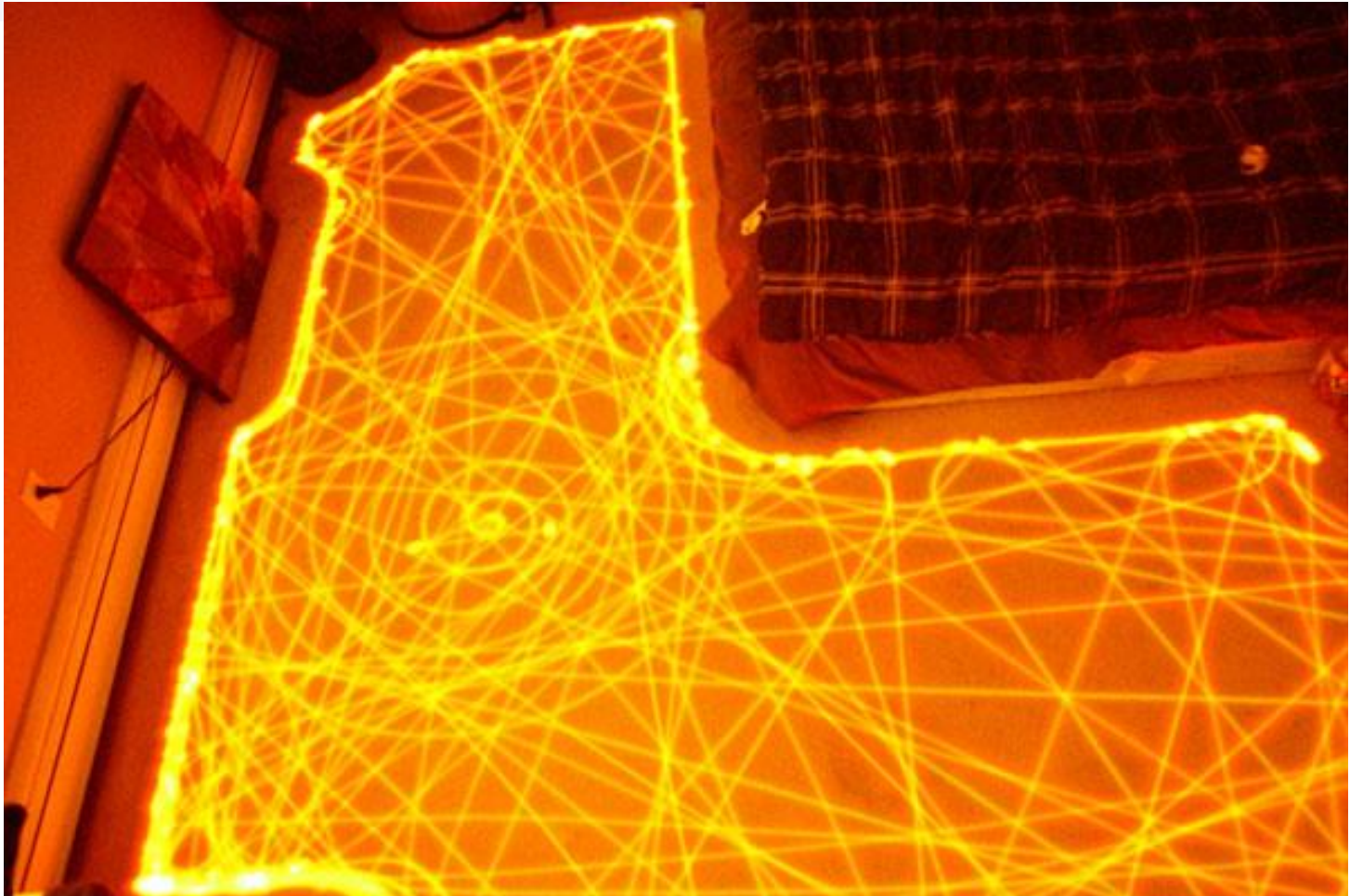
**Room Crossing:** Roomba crisscrosses the room to ensure full cleaning coverage.

**Dirt Detection** (selected models): When Roomba senses dirt, the blue Dirt Detect™ light is lit and Roomba cleans more intensely in that area.





# iRomba – esempio di funzionamento





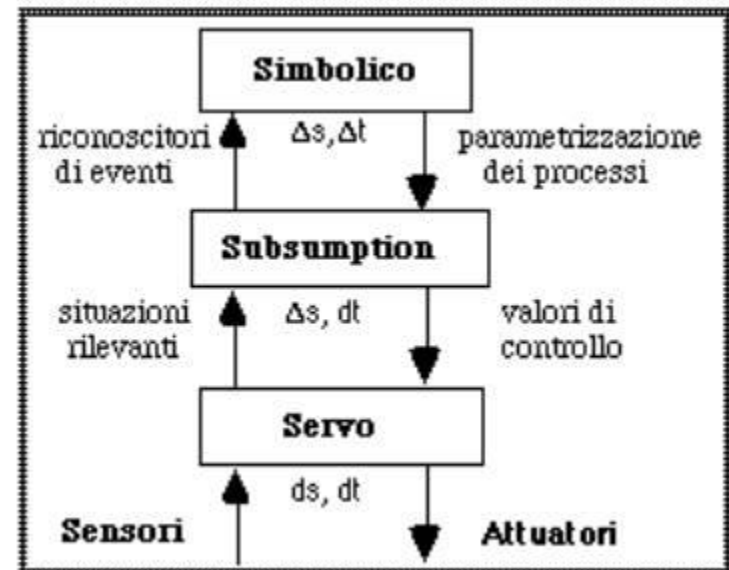
# Architetture Ibride

---

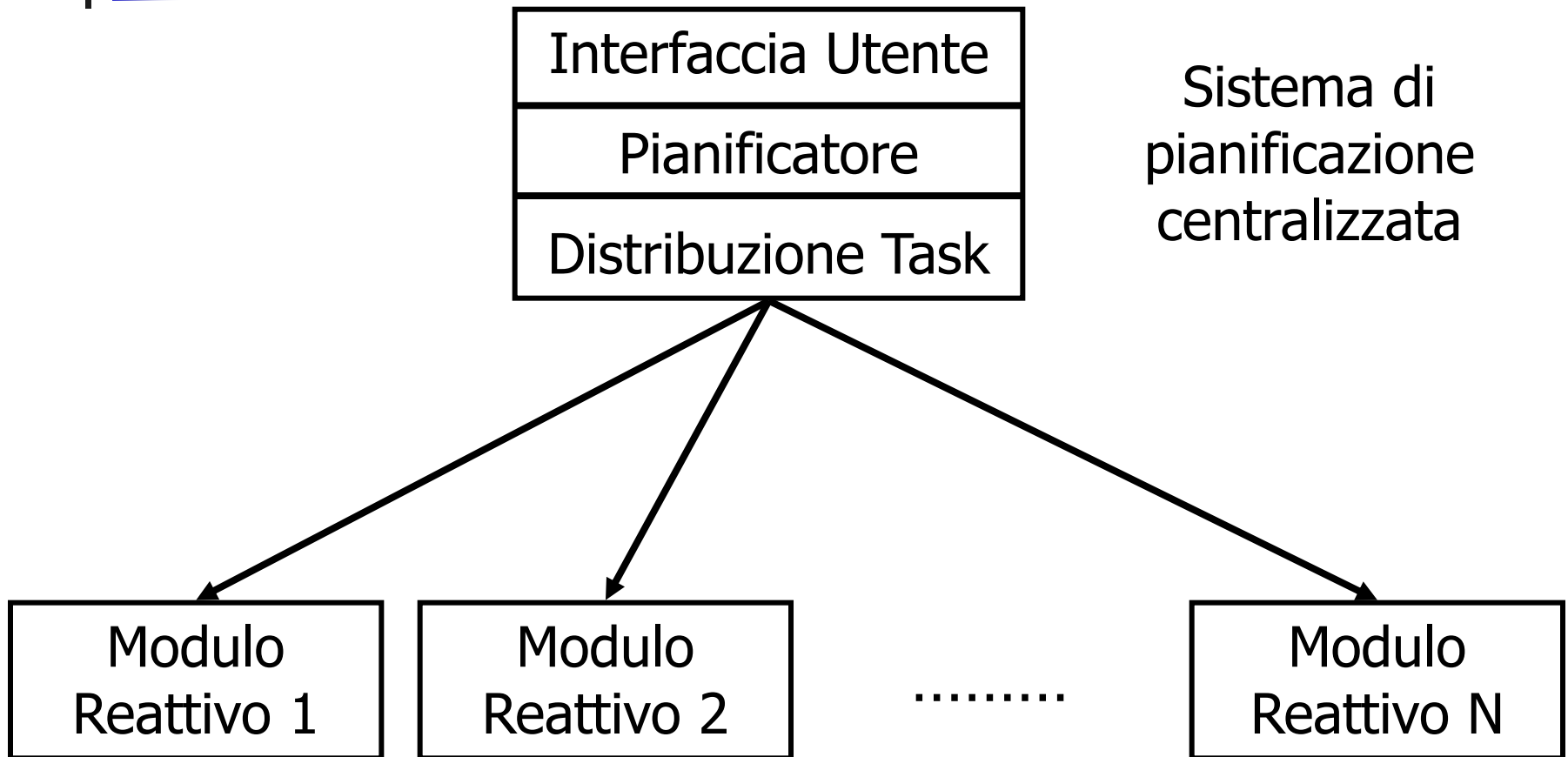
- Un approccio puramente reattivo dota il robot della capacità di eseguire compiti semplici in maniera efficiente, come ad esempio evitare gli ostacoli e adattarsi alle variazioni del mondo, ma non garantisce l'esecuzione di task più complessi, che includono modellizzazione dell'ambiente e comportamenti più complessi
- L'integrazione dei metodi reattivi con i metodi gerarchici combina l'efficienza della pianificazione con la flessibilità dei sistemi di controllo reattivo

# Architetture Ibride

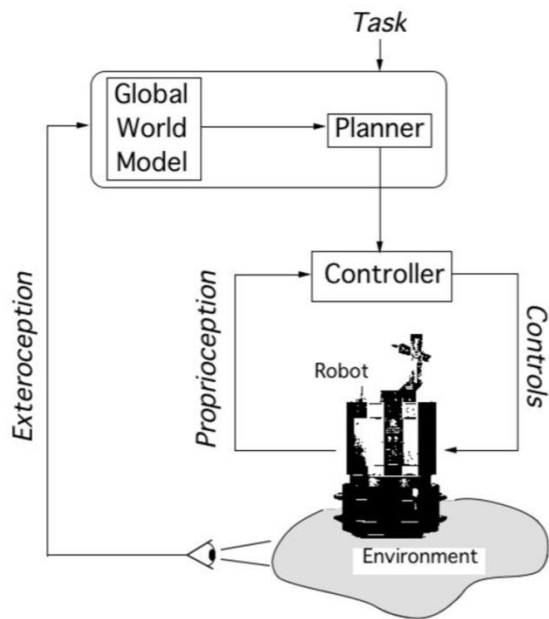
- Tipicamente una architettura ibrida comprende un modulo pianificatore strategico e un modulo pianificatore tattico per la gestione dei comportamenti di un robot.
- Il pianificatore strategico pianifica a lungo termine le azioni del robot, individuando la sequenza di sotto-obiettivi da realizzare per raggiungere il goal e passando i risultati per l'esecuzione al pianificatore tattico.
- Il pianificatore tattico inizializza e monitora i comportamenti prendendosi cura degli aspetti temporali per la loro coordinazione.



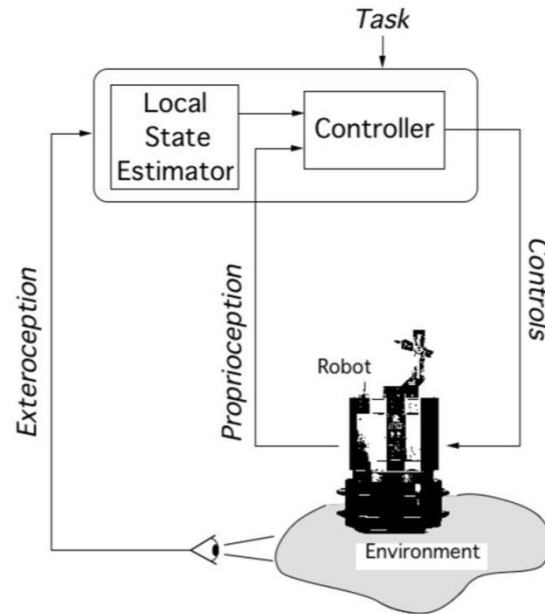
# Schema generale di una architettura ibrida



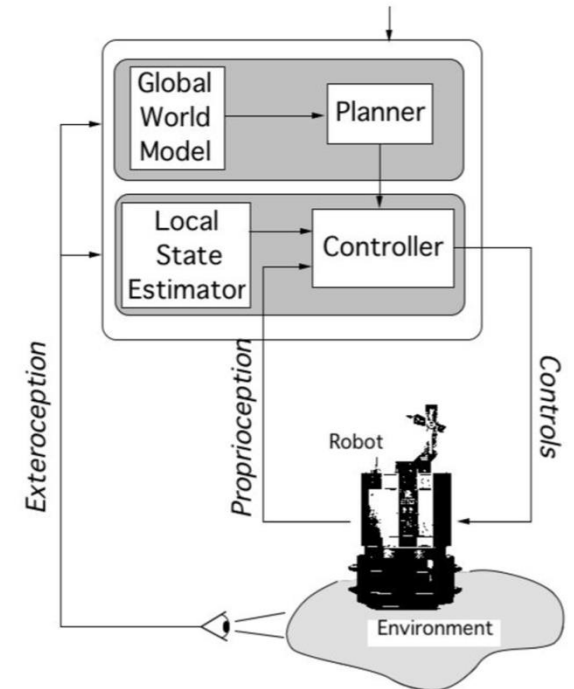
# Architetture gerarchiche, reattive e ibride



Gerarchica



Reattiva

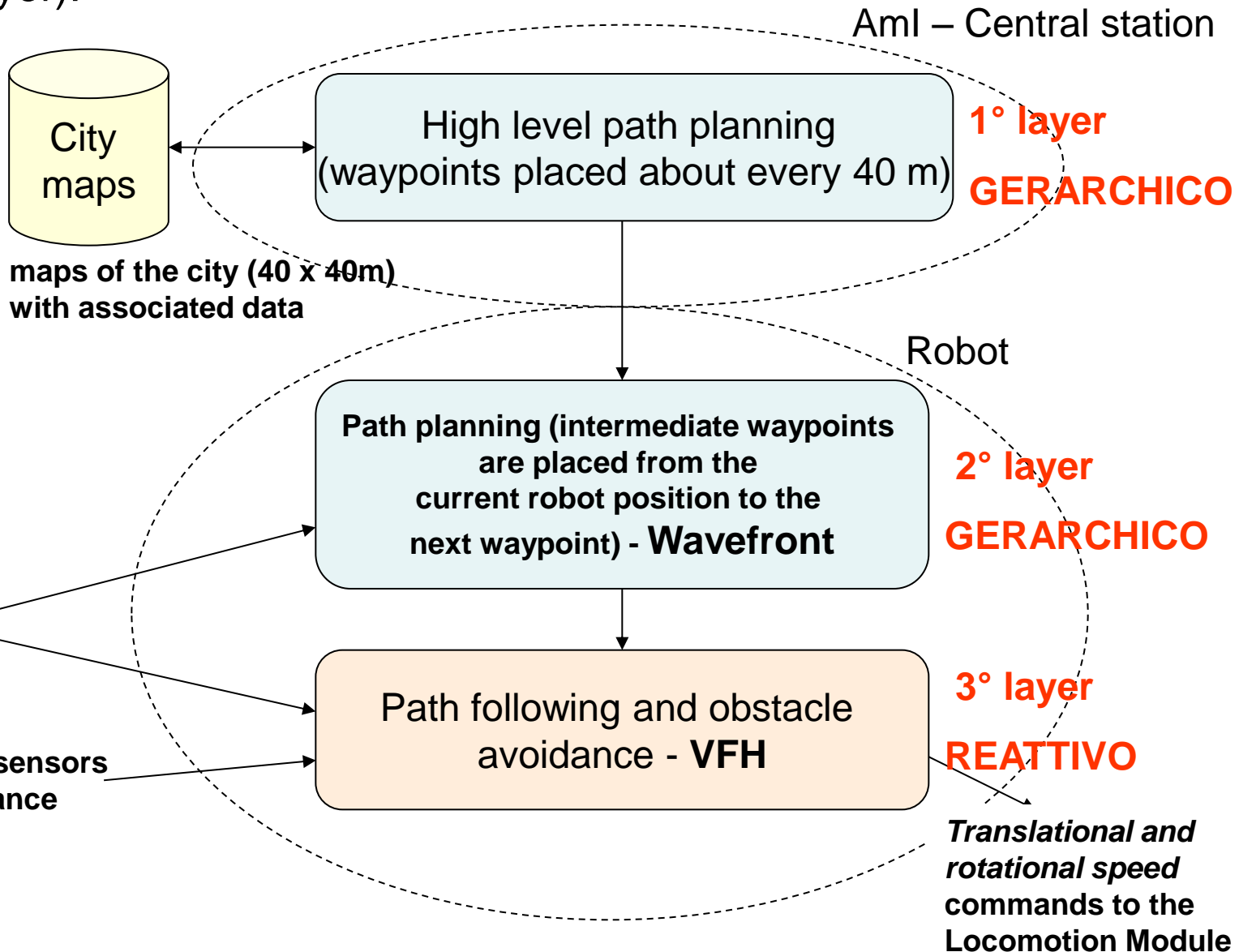


Ibrida



# Architettura ibrida di DustCart

3 livelli (layer):





# DustCart





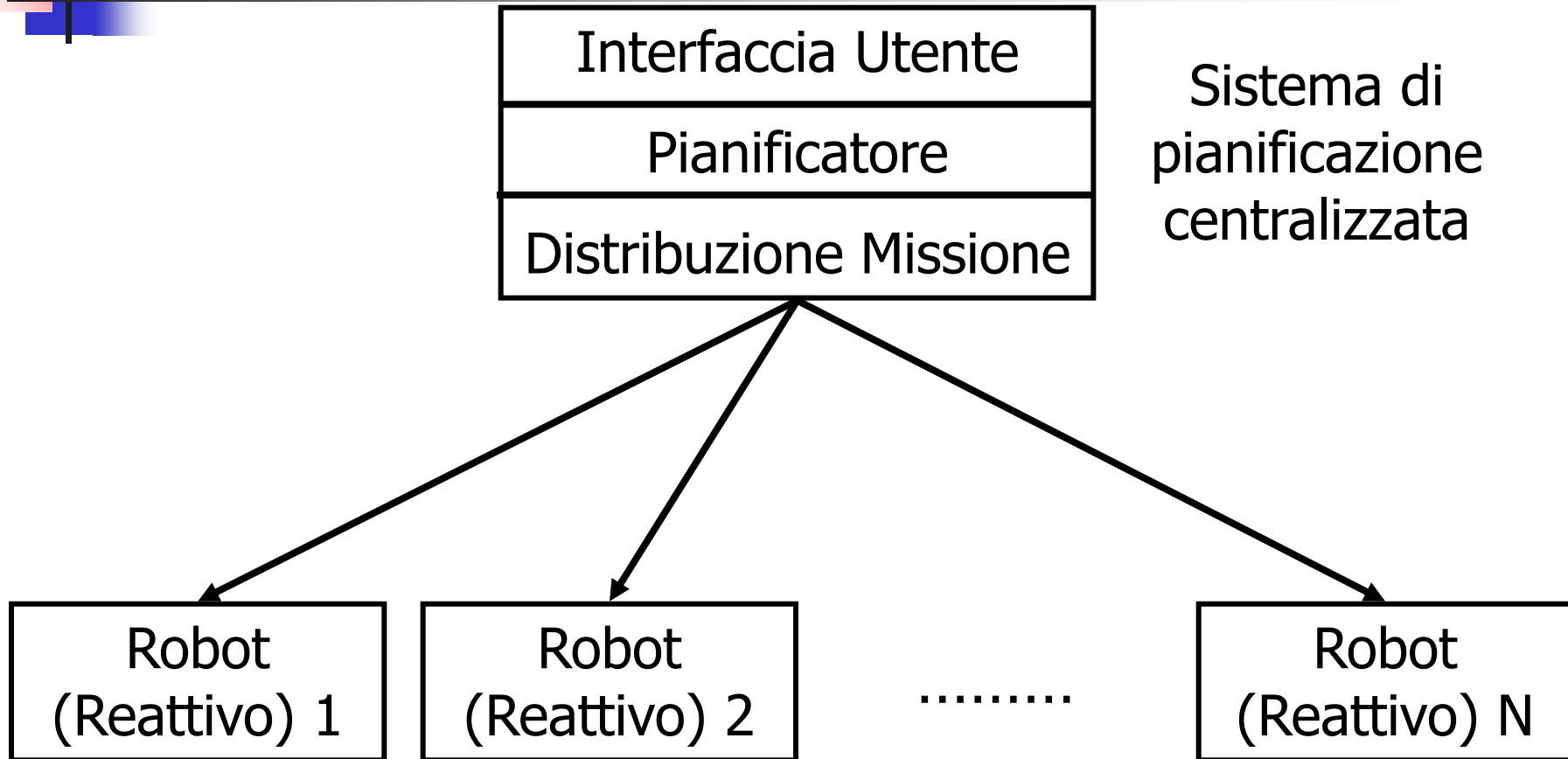
# Architetture decentralizzate o distribuite

---

- Un approccio alternativo all'uso di un unico sistema robotico per l'esecuzione di compiti è l'uso di un gruppo di sistemi robotici, ciascuno di complessità inferiore, che cooperano per l'esecuzione dello stesso compito.
- L'intelligenza è distribuita tra i vari sistemi, ognuno dei quali è autonomo



# Architetture distribuite (eventualmente ibride) per robot cooperanti



# Traditional classification of robotic architectures

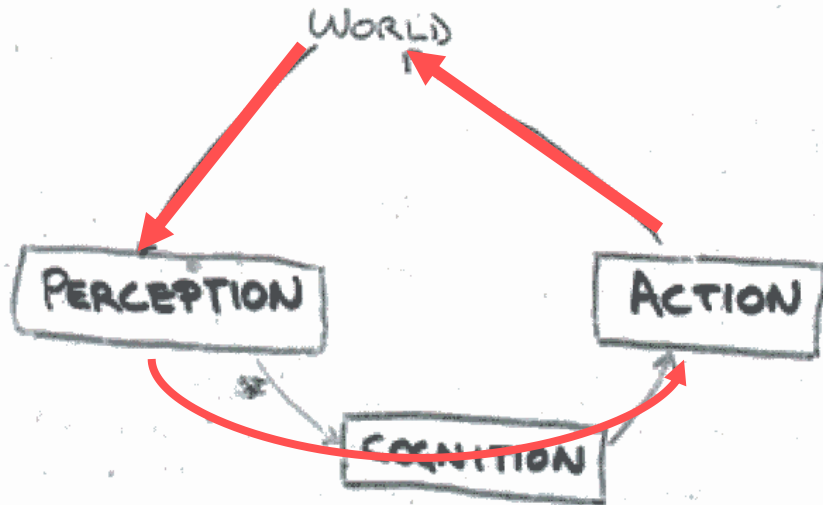


Figure 1: The traditional model where cognition mediates between perceptions and plans of actions.

deliberative, model-based

reactive, behavior-based

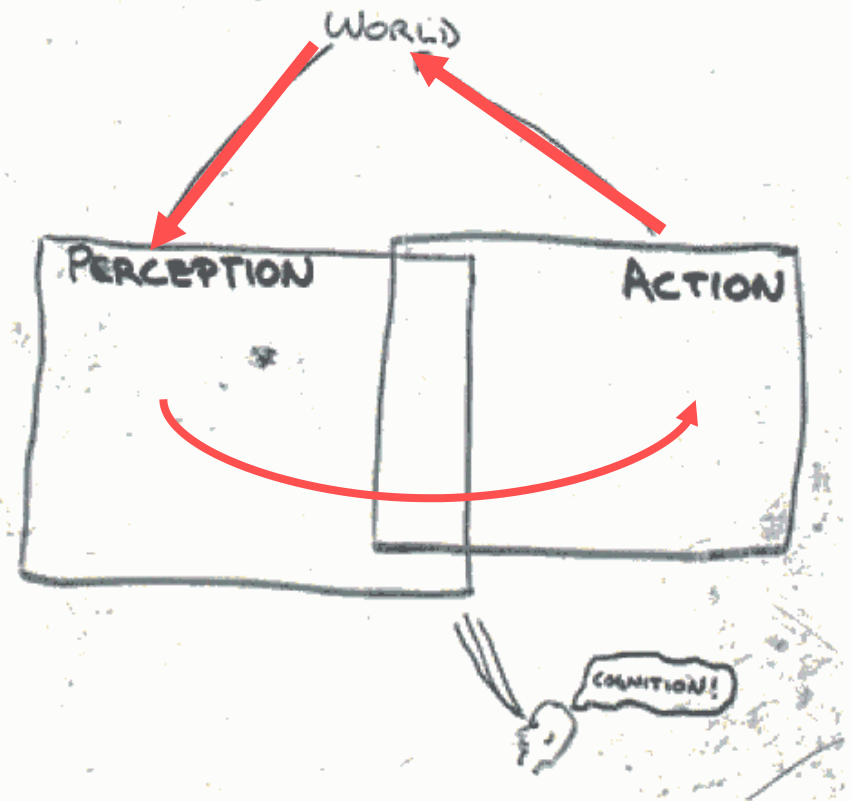
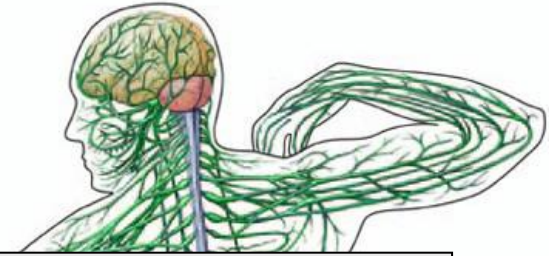


Figure 2: The new model, where the perceptual and action subsystems are all there really is. Cognition is only in the eye of an observer.

# Delays in the human nervous system



"In motor control **delays** arise in **sensory transduction**, **central processing**, and in the **motor output**. [...] These **delays** combine to give an **unavoidable feedback delay** within the **negative feedback control loop**, and can lie between about **30ms** for a spinal reflex up to **200-300 ms** for a visually guided response."

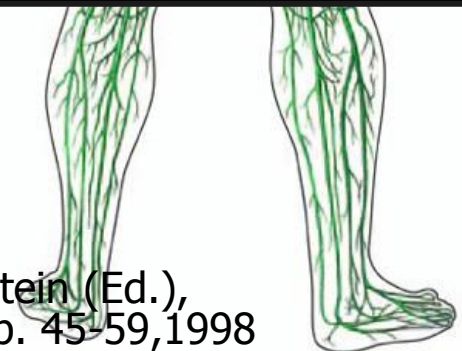
R.C. Miall, D.J. Weir, D.M. Wolpert, J.F. Stein, "Is the cerebellum a Smith predictor?", *Journal of Motor Behavior*, vol. 25, no. 3, pp. 203-216, 1993

"Fast and coordinated arm movements **cannot be executed under pure feedback control** because biological feedback loops are both too slow and have small gains"

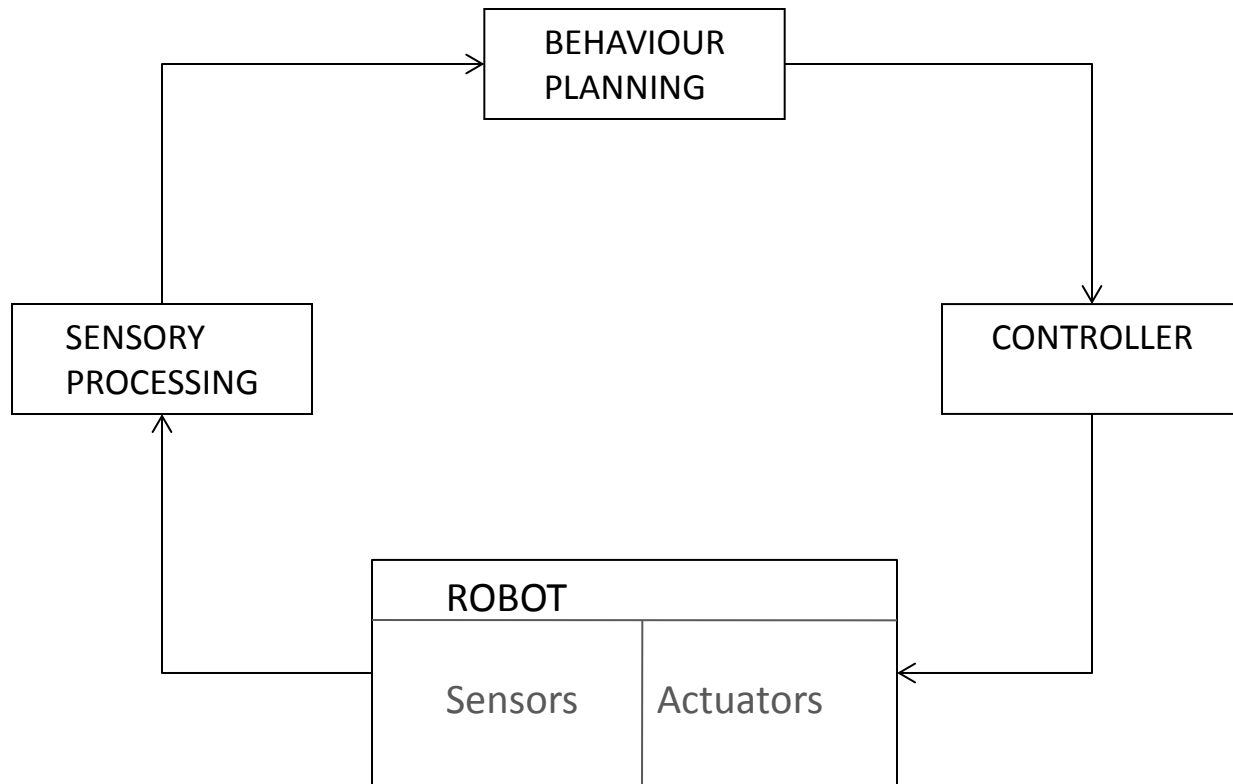
M. Kawato, Internal models for motor control and trajectory planning. *Current Opinion in Neurobiology*, 9, 718-727(1999). Elsevier Science Ltd.

## A lesson from neuroscience: **anticipation**

- A. Berthoz, *Le sens du mouvement*. Odile Jacob, Paris, 1997
- R.S. Johansson, "Sensory input and control of grip", in M. Glickstein (Ed.), *Sensory Guidance of Movements*. John Wiley, Chichester, UK, pp. 45-59, 1998

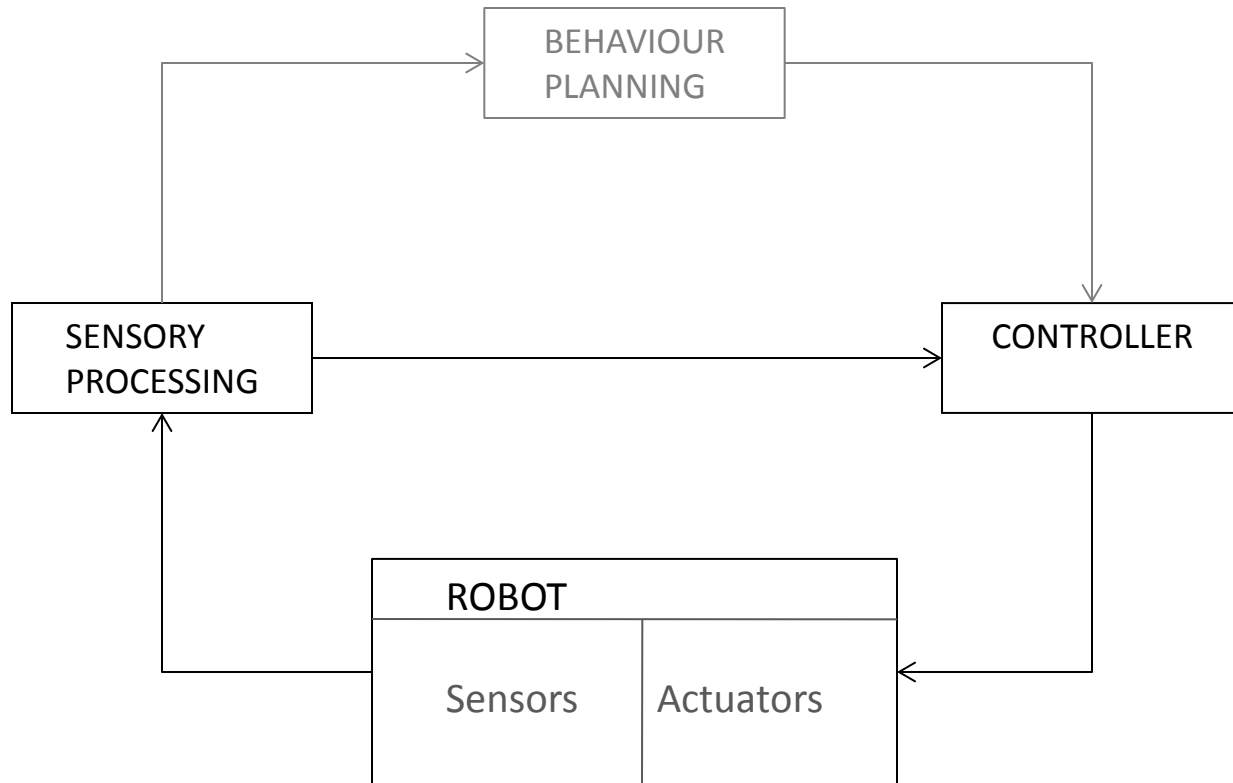


# Basic scheme for robot behaviour control



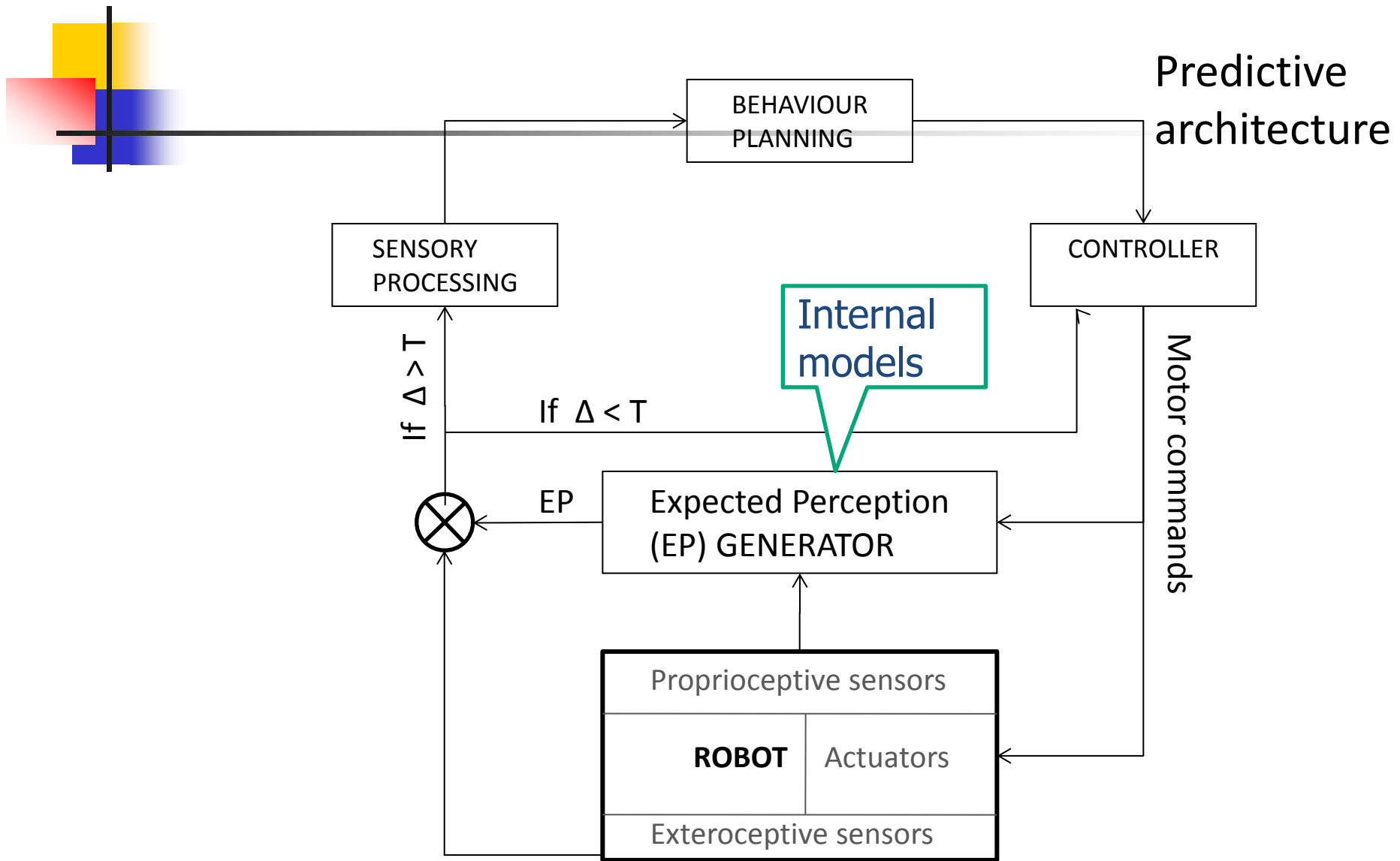
Hierarchical architectures

# Basic scheme for robot behaviour control



Reactive architectures

# Basic scheme for robot behaviour control



# Predictive smooth pursuit eye movement



The retinal slip (target velocity onto the retina) reaches zero after the algorithm convergence.

When the target is unexpectedly stopped, the system continues to follow the target for a short period.

# Punching a moving target - robot experiments



The prediction is iterated ahead 0.5 seconds  
As the predicted target is inside the arm workspace, the robot executes a movement to punch the ball in the ***predicted position***

