Silvia Rossi

Reactive and Hybrid Architectures

Lezione n. 7

Corso di Laurea:
Informatica

Insegnamento:
Sistemi
multi-agente

Email:
silrossi@unina.it

A.A. 2014-2015

REACTIVE ARCHITECTURES

(W: 5.1)

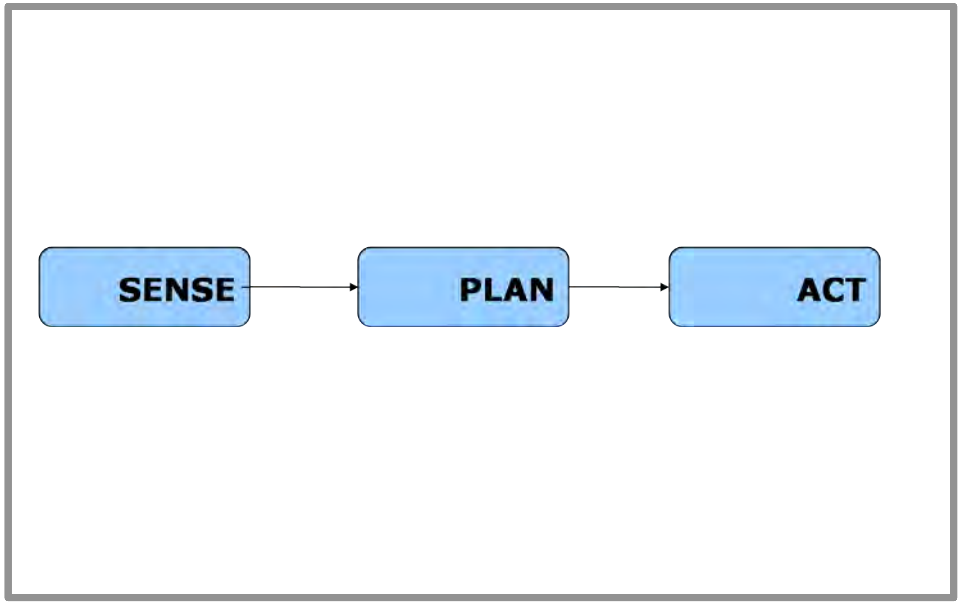
- There are many unsolved (some would say insoluble) problems associated with symbolic AI
- These problems have led some researchers to question the viability of the whole paradigm, and to the development of *reactive* architectures
- Although united by a belief that the assumptions underpinning mainstream AI are in some sense wrong, reactive agent researchers use many different techniques
- In this presentation, we start by reviewing the work of one of the most vocal critics of mainstream AI: Rodney Brooks

Model-based Agent ('70)

Knowledge Representation and
Formal Reasoning

Closed World:
Complete Model of the environment

Functional and horizontal activity decomposition [Shakey 1969]



Sense Plan Act

Stanford AI Laboratory / CMU (Moravec)

Brooks has put forward three theses:

1. Intelligent behavior can be generated **without explicit representations** of the kind that symbolic AI proposes
2. Intelligent behavior can be generated **without explicit abstract reasoning** of the kind that symbolic AI proposes
3. **Intelligence is an emergent property** of certain complex systems

He identifies two key ideas that have informed his research:

1. **Situatedness and embodiment: ‘Real’ intelligence is situated in the world,** not in disembodied systems such as theorem provers or expert systems
2. **Intelligence and emergence**: ‘Intelligent’ behavior arises as a result of an agent’s interaction with its environment. Also, intelligence is ‘in the eye of the beholder’; it is not an innate, isolated property

To illustrate his ideas, Brooks built some based on his *subsumption architecture*

A subsumption architecture is a hierarchy of task-accomplishing *behaviors*

Each behavior is a rather simple rule-like structure

Each behavior 'competes' with others to exercise control over the agent

Lower layers represent more primitive kinds of behavior (such as avoiding obstacles), and have precedence over layers further up the hierarchy

The resulting systems are, in terms of the amount of computation they do, *extremely* simple

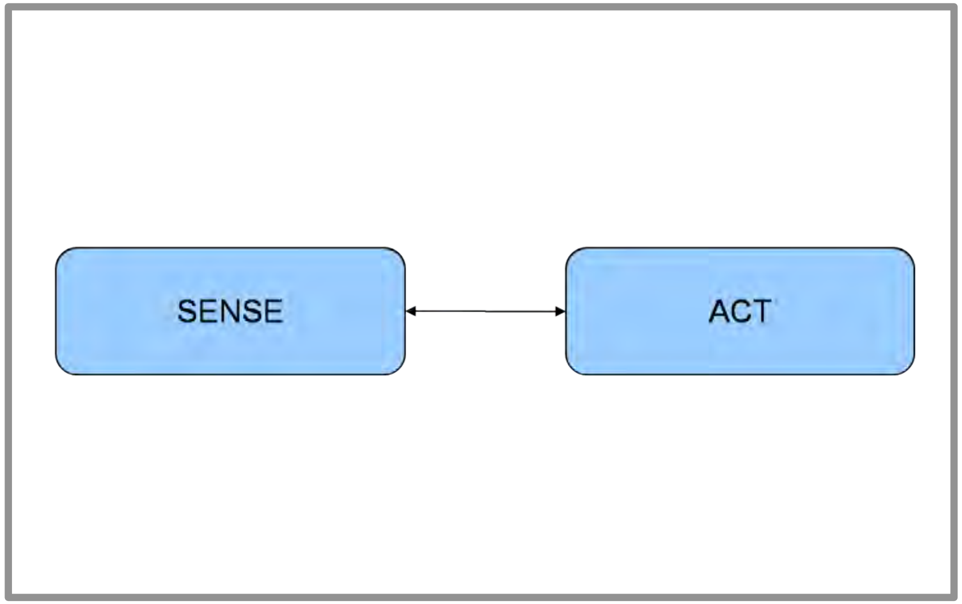
Some of the robots do tasks that would be impressive if they were accomplished by symbolic AI systems

- Situated:
  interacting with the world

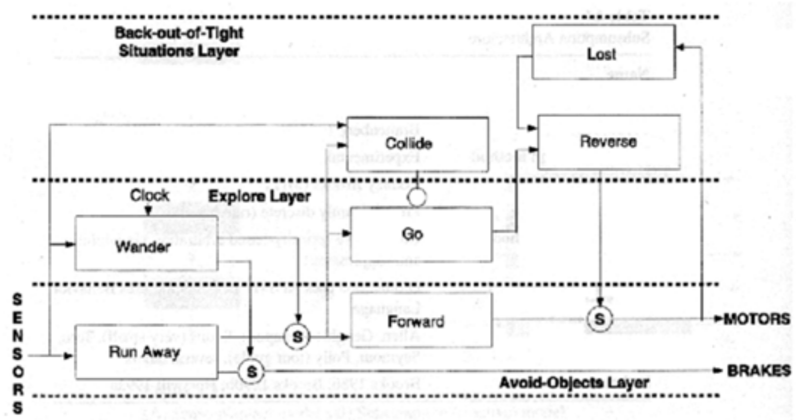
- No Memory, no model:
  memory and model is the
  external env.

- Behavior-based:
  sense and act strictly coupled
  and associated with behaviors

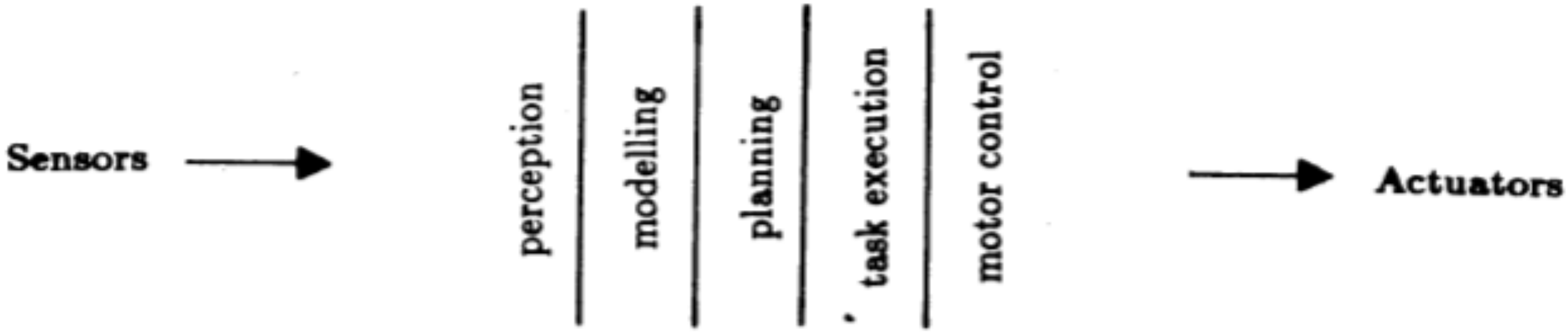
  - Sussumption Architecture
    [Brooks 1986]



Sense-Act Paradigm



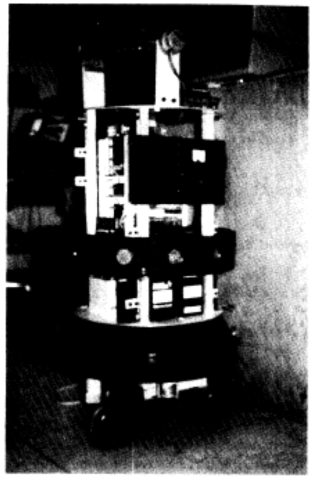
A Traditional Decomposition of a Mobile Robot Control System into Functional Modules

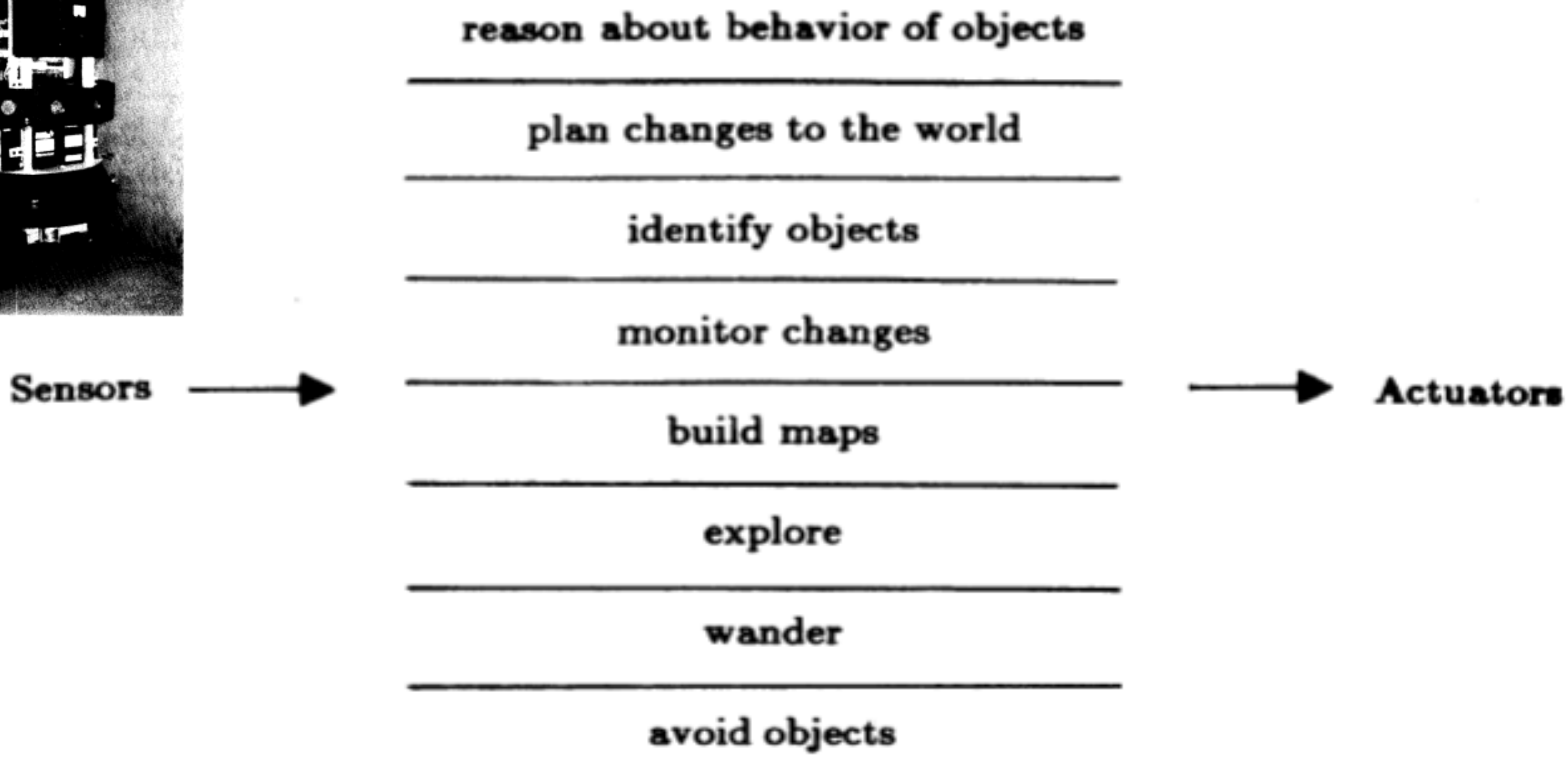


Stanford AI Laboratory / CMU (Moravec)

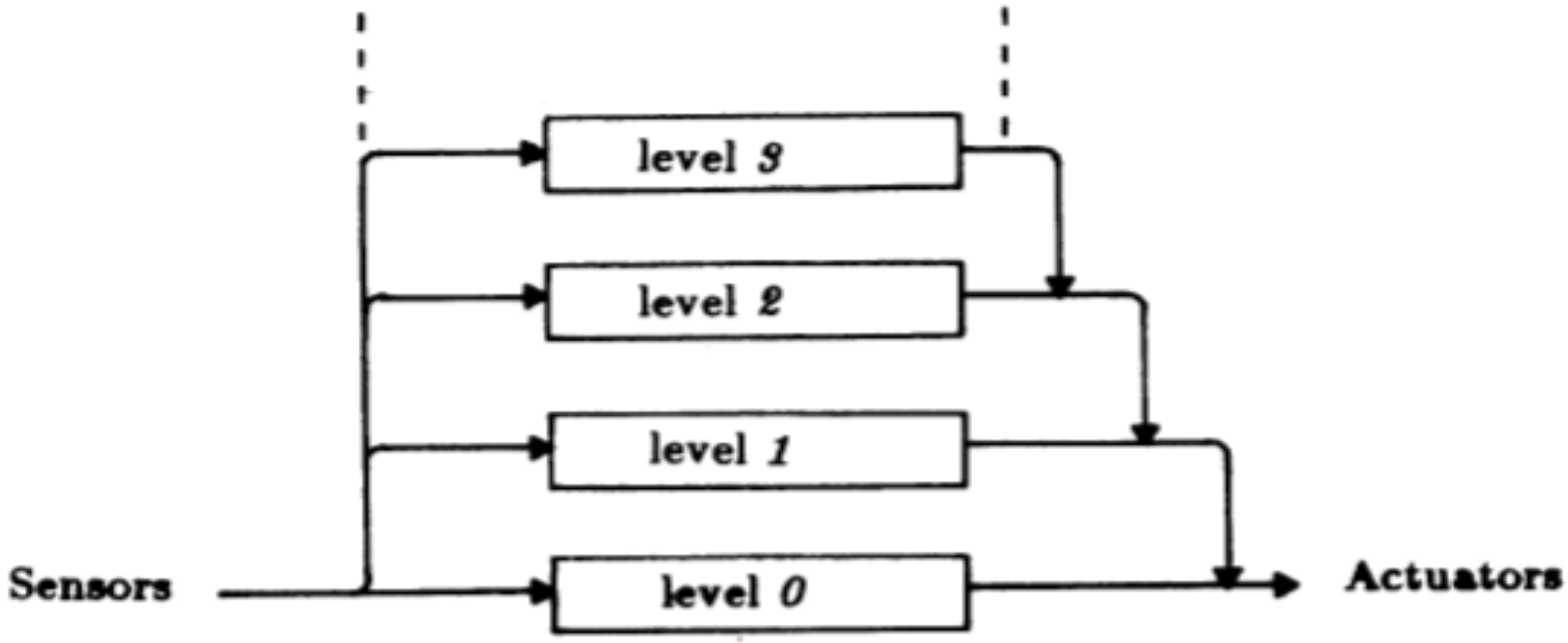
Reactive: Sense-Act

A Decomposition of a Mobile Robot Control System Based on Task Achieving Behaviors





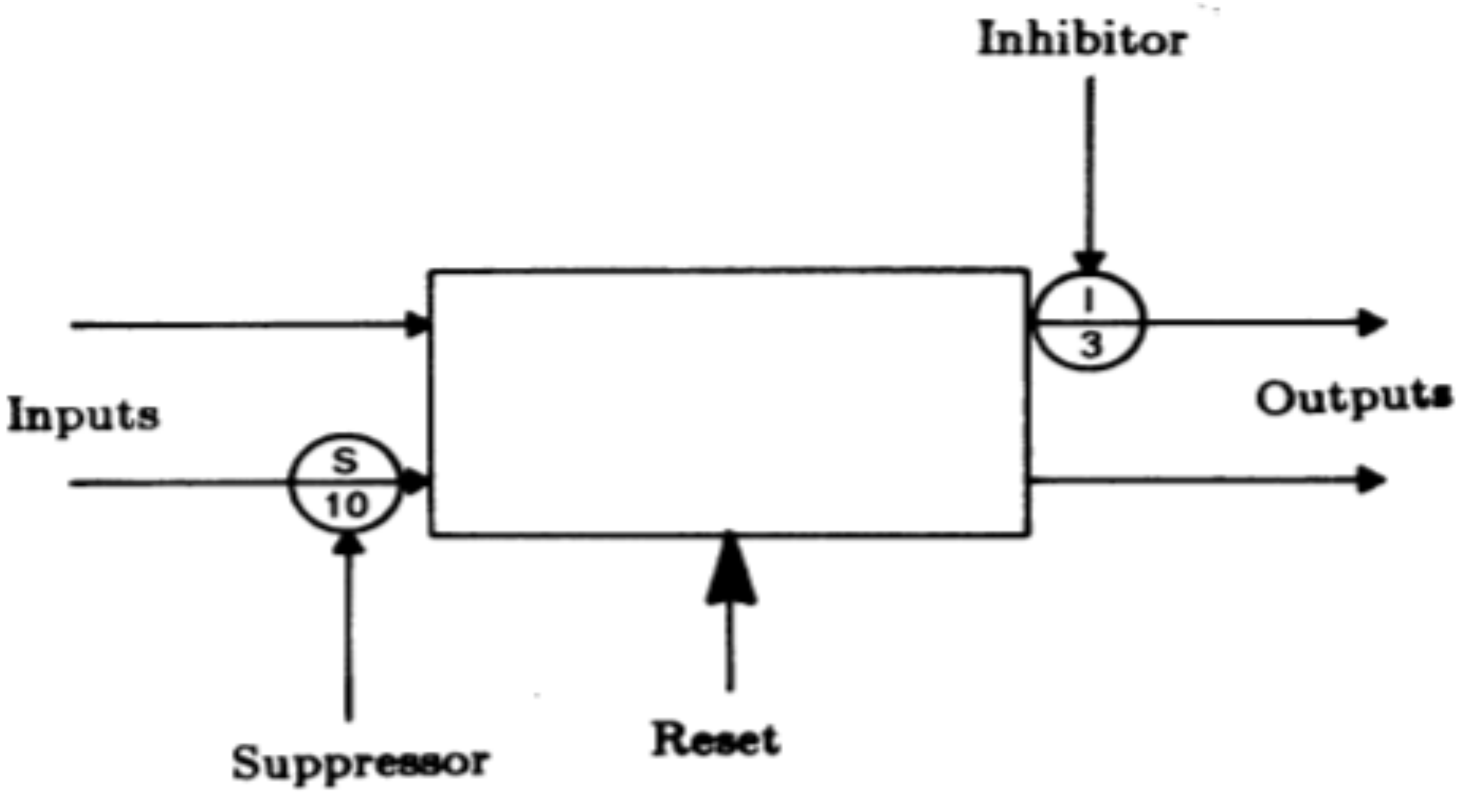
Layered Control in the Subsumption Architecture



Example of a Module – Avoid

```
(defmodule avoid
  :inputs (force heading)
  :outputs (command)
  :instance-vars (resultforce)
  :states
    ((nil (event-dispatch (and force heading) plan))
     (plan (setf resultforce (select-direction force heading))
           go)
     (go (conditional-dispatch (significant-force-p resultforce 1.0)
                               start
                               nil))
     (start (output command (follow-force resultforce))
            nil)))
```

Schematic of a Module



Level 0

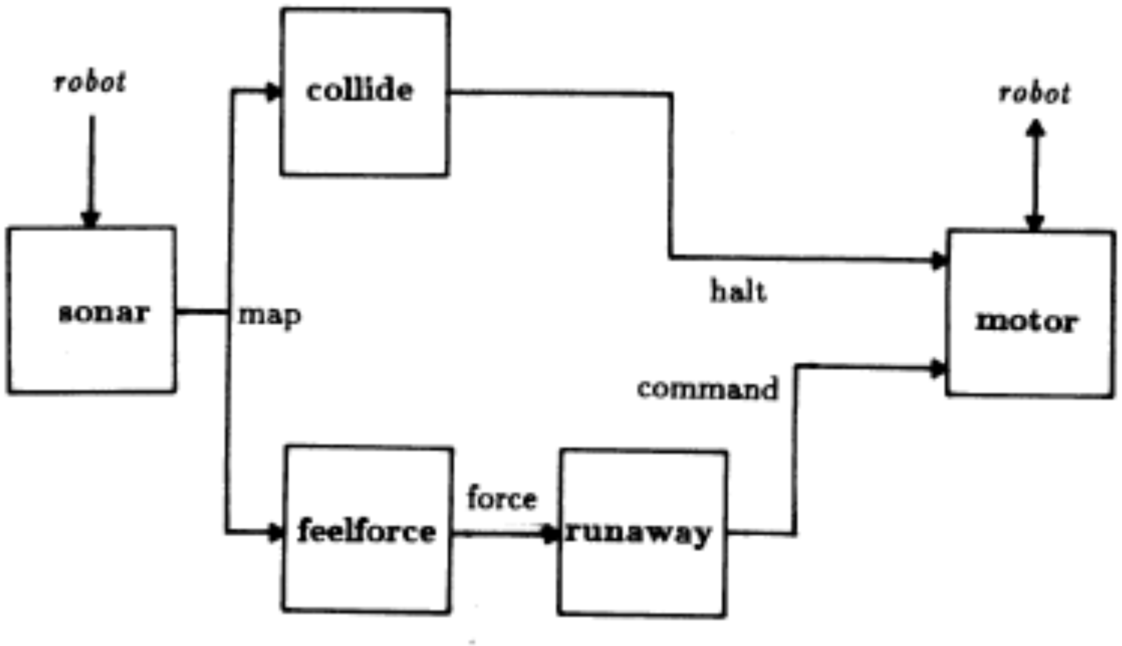


Figure 5. The level 0 control system.

Level 0, 1

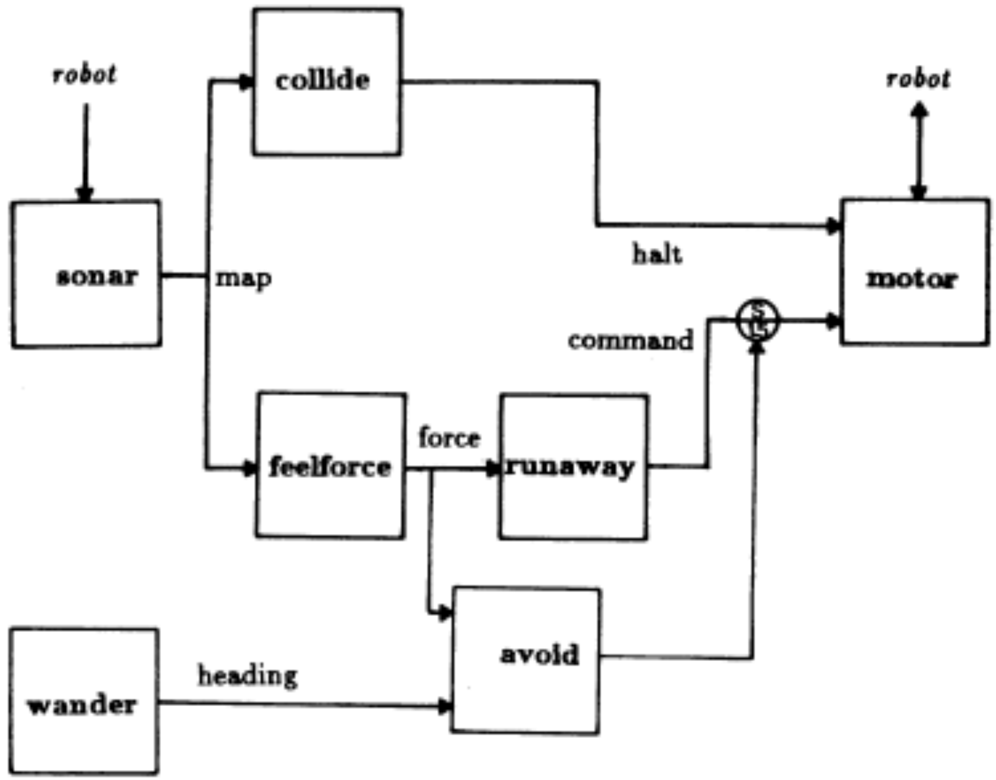


Figure 6. The level *0* control system augmented with the level *1* system.

Level 0, 1, 2

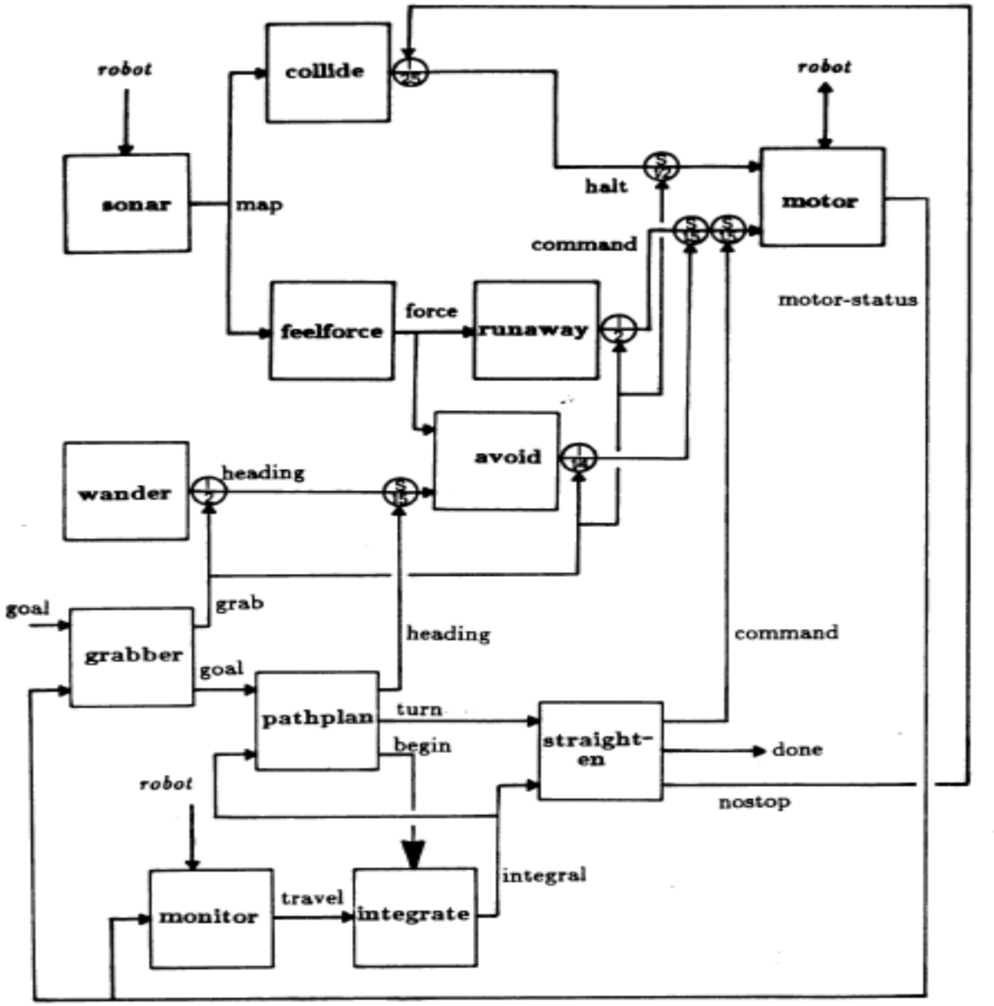
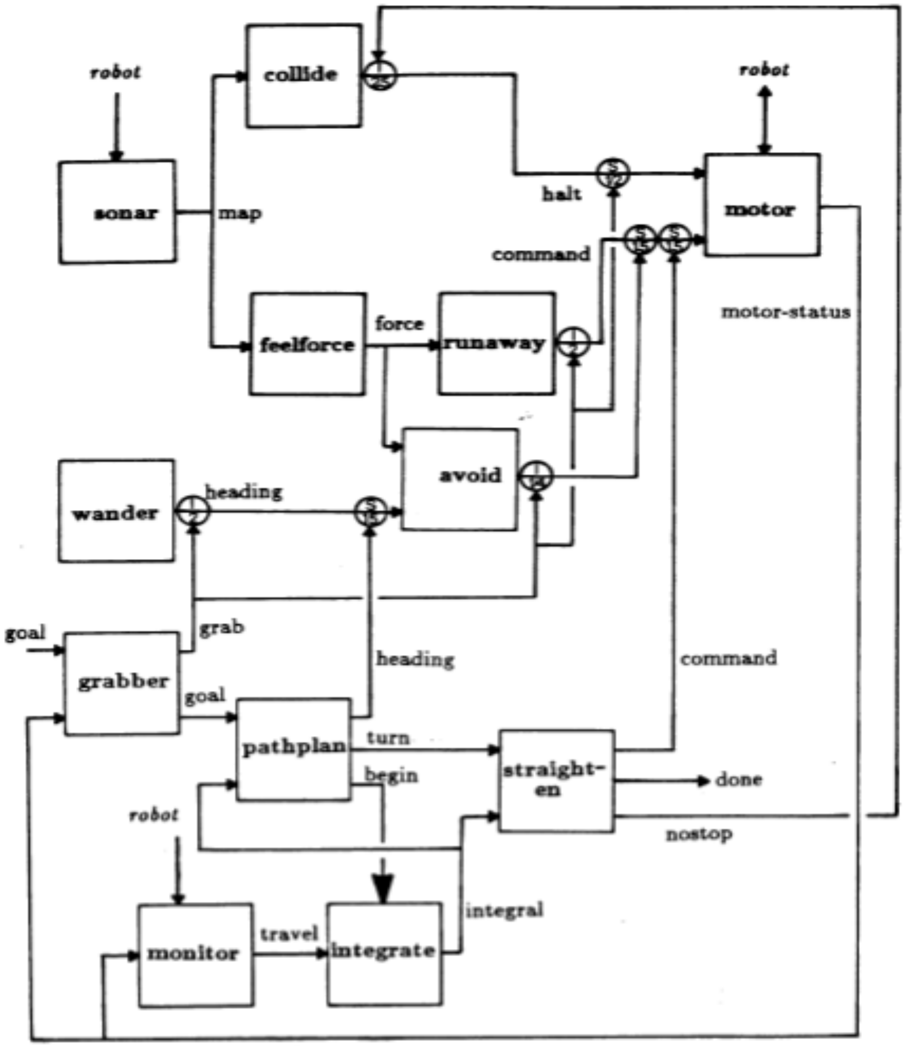


Figure 7. The level 0 and 1 control systems augmented with the level 2 system.

Levels 0, 1, and 2 Control Systems



Steels' Mars explorer system, using the subsumption architecture, achieves near-optimal cooperative performance in simulated 'rock gathering on Mars' domain:

The objective is to explore a distant planet, and in particular, to collect sample of a precious rock. The location of the samples is not known in advance, but it is *known that they tend to be clustered.*

For individual (non-cooperative) agents, the lowest-level behavior, (and hence the behavior with the highest "priority") is obstacle avoidance:

if detect an obstacle *then* change direction (1)

Any samples carried by agents are dropped back at the mother-ship:

if carrying samples *and* at the base
then drop samples (2)

Agents carrying samples will return to the mother-ship:

if carrying samples and *not* at the base
then travel up gradient (3)

Agents will collect samples they find:

if detect a sample *then* pick sample up (4)

An agent with “nothing better to do” will explore randomly:

if true *then* move randomly (5)

Le regole si suppongono immesse nella gerarchia

1<……<5 (la precondizione di 1 se true interrompe tutte le altre regole e il robot cambia direzione etc..)

Macchina a stati finiti

Limiti e soluzioni per il Mars Explorer

Fatto: solitamente i campioni di roccia si trovano in mucchi, sarebbe il caso quindi di utilizzare più agenti capaci di comunicare fra di loro in modo da rendere noto che in un posto già esplorato si trovano altri campioni, ma questo non è permesso dalla subsumption architecture.

Trucco: mettere la comunicazione "nell' ambiente", cioè togliere regole introdurre altre azioni e conseguenti regole (stigmergic):

Togliere 5.3 e sostituirla con:

> *if* carrying samples *and* *not* at the base *then* drop 2 crumbs and travel up gradient (7)
>
> If sense crumbs then pick up 1 crumbs and travel down gradient (8)

La gerarchia diventa :

1<2<7<4<8 <5

Advantages of Reactive Agents

Simplicity

Computational tractability

Robustness against failure

Elegance

Limitations of Reactive Agents

Agents without environment models must have sufficient information available from local environment

If decisions are based on local environment, how does it take into account non-local information (i.e., it has a "short-term" view)

Difficult to make reactive agents that learn

Since behavior emerges from component interactions plus environment, it is hard to see how to engineer specific agents (no principled methodology exists)

It is hard to engineer agents with large numbers of behaviors (dynamics of interactions become too complex to understand)

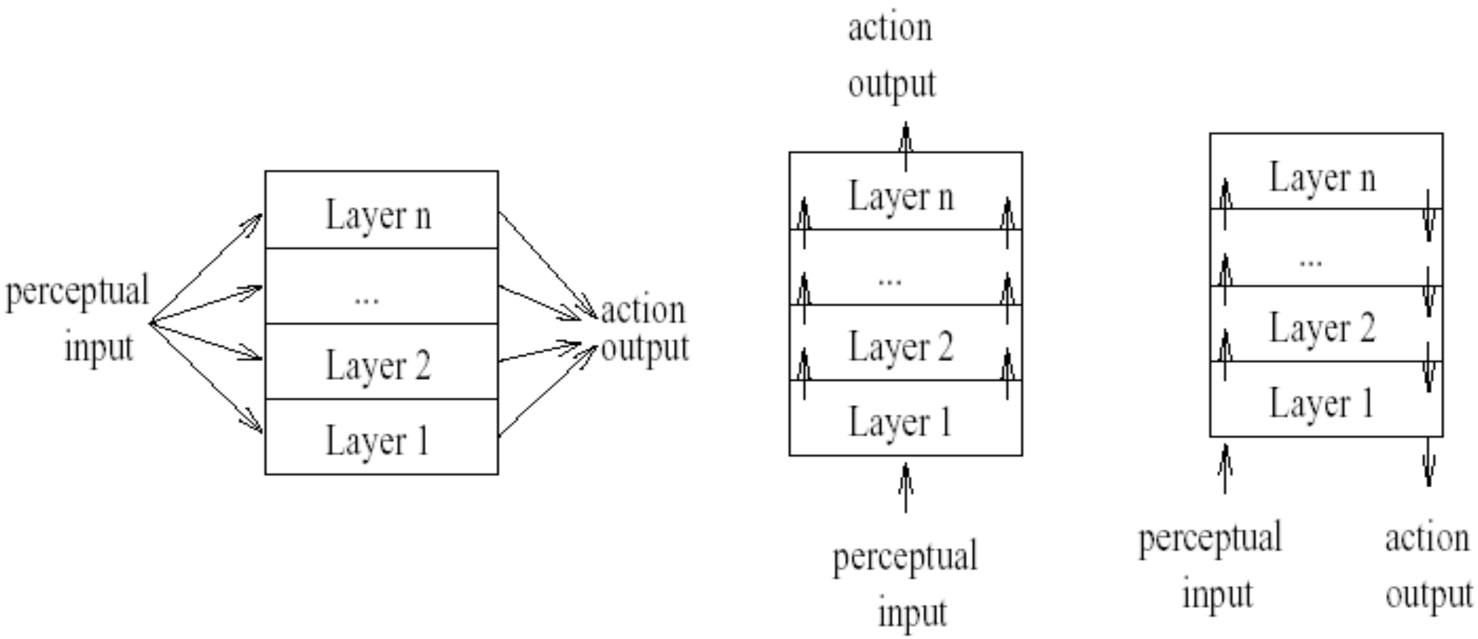
Hybrid Architectures

- Many researchers have argued that neither a completely deliberative nor completely reactive approach is suitable for building agents
- They have suggested using *hybrid* systems, which attempt to marry classical and alternative approaches
- An obvious approach is to build an agent out of two (or more) subsystems:
  - a *deliberative* one, containing a symbolic world model, which develops plans and makes decisions in the way proposed by symbolic AI
  - a *reactive* one, which is capable of reacting to events without complex reasoning

Hybrid Architectures

- A key problem in such architectures is what kind of control framework to embed the agent's subsystems in, to manage the interactions between the various layers
- *Horizontal layering*
  Layers are each directly connected to the sensory input and action output.
  In effect, each layer itself acts like an agent, producing suggestions as to what action to perform.
- *Vertical layering*
  Sensory input and action output are each dealt with by at most one layer each

Hybrid Architectures



(a) Horizontal layering

(b) Vertical layering (One pass control)

(c) Vertical layering (Two pass control)