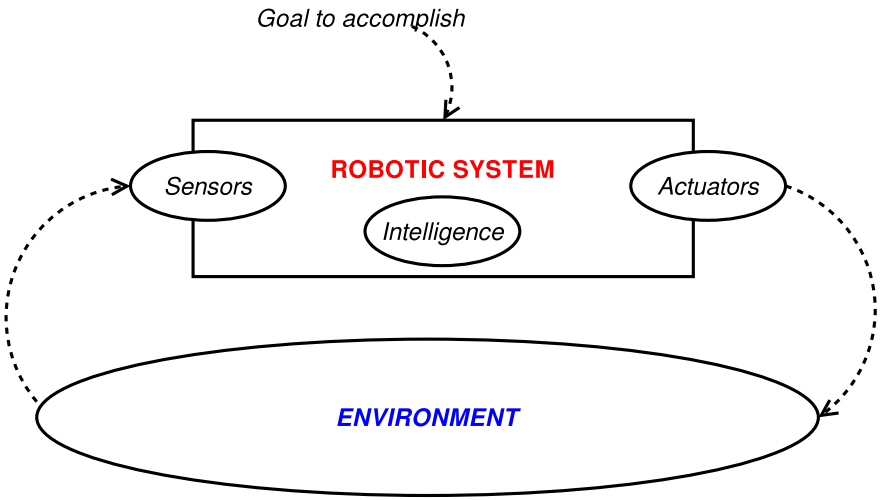
Basics of A Robotic System

Corrado Santoro

ARSLAB - Autonomous and Robotic Systems Laboratory
Dipartimento di Matematica e Informatica - Università di Catania, Italy
santoro@dmi.unict.it

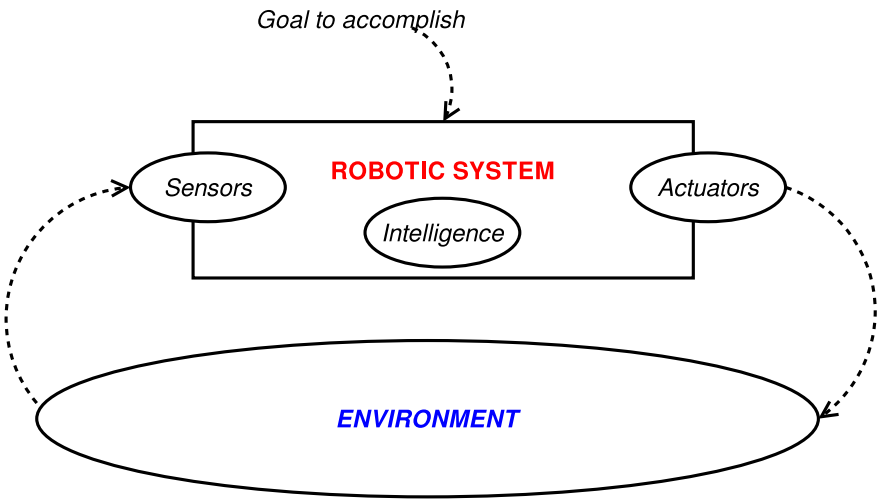
Programmazione Sistemi Robotici

What is a robot?



A **Robot** is a mechanic/electronic/software system able to achieving a **well defined goal** in a complete **autonomy**.

Basics Components/Properties



1. A robot lives in a reference **environment** (physical or logical/virtual)
2. It has some **sensors** to understand what is happening in the environment
3. Using data from sensors, the “intelligence” elaborates the **actions** to be done
4. Executes actions onto the environment through proper **actuators**

Examples or Robots

- Your washing machine (yes! it's a robot)
- Your microwave oven (yes! it's a robot)
- The Roomba dust cleaner robot
- Mechanical arms in assembly lines of industries
- Semi-self-guided cars
- Unmanned Ground and Aerial Vehicles
- Humanoids
- ...

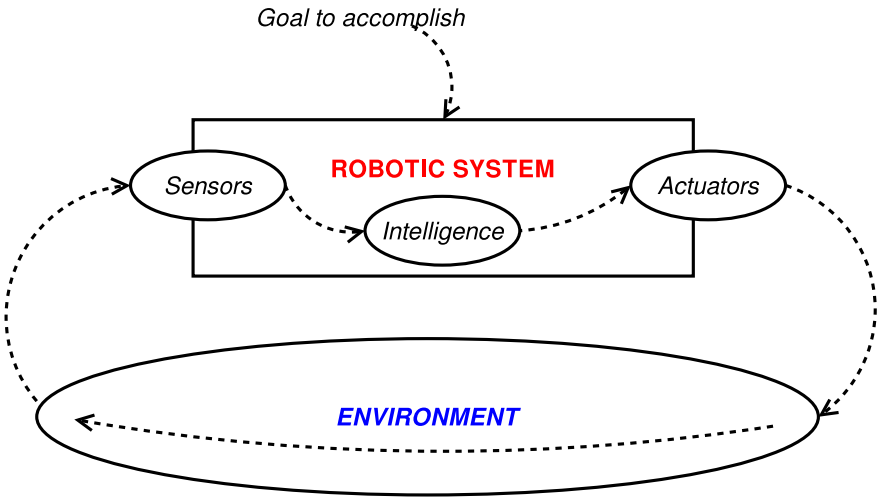
(same sample videos!)

Robotics, is it only a matter of engineers?

Is "Robotics" worth for computer scientists?

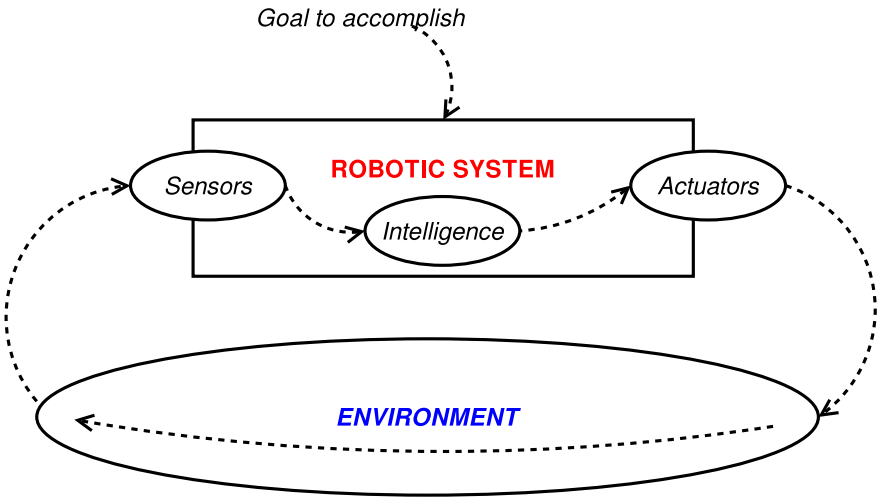
- Robot complexity is increasing fast
- Robots (in more or less "intelligent" form) are destined to pervade our lives
- Today **any robot** has at least one microprocessor
- **All tasks**, sensing, actuator driving, action planning, "reasoning", are performed in software
- Software running in robots is now highly sophisticated, and includes not only low-level tasks, but also AI techniques, and web/cloud-computing aspects
- Software **is** a matter of computer scientists!

The Concept of “Feedback”



1. A robot system is a **feed-back** system
2. Any **action** performed by the robot affects the environment
3. The environment is **sensed** by the robot
4. So the results of **the action done by the robot is sensed again by the robot itself**
5. In some sense, the robot **feeds** itself **back**
6. It is a **data loop**
7. And many **data loops** exist in any robotic systems

The Importance of the Feedback



- A robotic system lives in **physical world** where actions:
  - do not happen **immediately**
  - have a latency (i.e. reaction time) which can be variable and sometimes not identifiable
  - May fail due to several reason
  - May provoke unpredictable results
- **We must not implicitly assume that the execution of an action succeeds!**

A “classical” (working) non-robotic software:

```
...
FILE * f = fopen(...);

fwrite(&mydata, sizeof(mydata), 1, f);
// ^^^ we do not care of a problem here (e.g. disk failure);
// indeed, the write goes to buffer cache and the failure is
// detected as soon as the cache is really written to the disk
// by the operating system

fclose(f);
...
```

A non-working robotic software:

```
// Go to deposit and pick a box

go_to_deposit();
// Is the path towards the deposit clear?
// Are we sure that an obstacle is not present?
// Is the obstacle temporary or persistent?

pick_the_box();
// Is the box present?
// Could I pick the box?
// Did the box fall?
...
```

A (almost) **working** robotic software:

```
// Go to deposit and pick a box

retry:
go_to_deposit(); // this action is asynchronous

while (true) {
  if (motion_successful())
    break;
  if (obstacle_present()) {
    stop_motion();
    sleep(5); // wait 5 seconds and retry
    goto retry;
  }
}

...
```

The Importance of the Dynamics

- A robotic system lives in **physical world**
- Actions, since are relevant to physical “things”, require a certain **reaction time**
- This **reaction time** has to be identified and considered in developing algorithms
- Sometimes this reaction time is not good for our application (too slow? too fast) and proper countermeasures must be performed
- This is commonly understood as **(system) dynamics**

The Closed World Assumption

- A robotic system lives in physical environment but it has a **partial representation** of the environment itself
- The representation is provided by its sensors and by a map implemented in the intelligence
- But:
  - each sensor is **highly specialised**, it “senses” only a specific information
  - each sensor is affected by a measure error that could be **non-negligible**
- And:
  - the map of the world in the intelligence considers only “things” that are relevant to robot tasks
  - any other “thing” present or “event” happening in the environment is **not understood** by the robot

The Closed World Assumption

- A robot designed to pick boxes: what about if we give it a sphere?
- A robot designed to pick *light* boxes: what about if we give it a heavy one?
- A robot designed to pick *red* boxes: what about if we give it a pink or orange one?
- A wheeled robot driving on a smooth floor: what about if there are small sticks on the way?

In the Closed World Assumption, anything that is "known" is true, and (conversely) anything that is "unknown" is false.

This is **unrealistic** and anything that is "unknown" could provoke unpredictable results in robot behaviour.

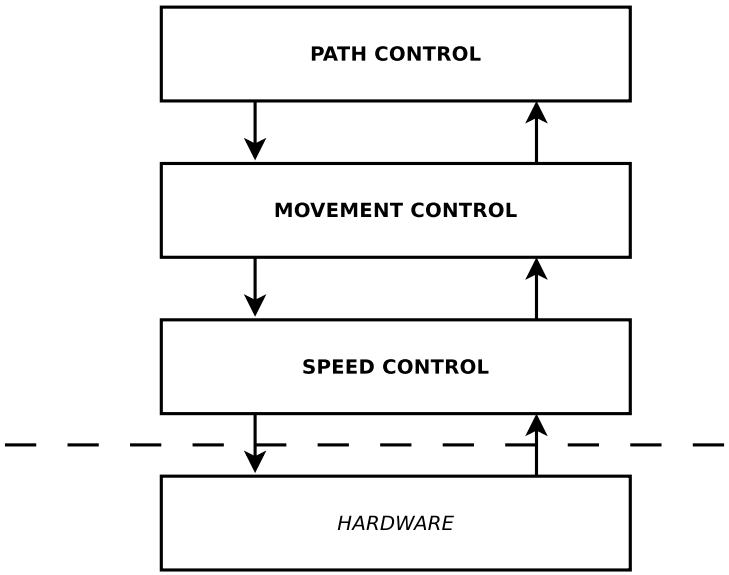
A Case-Study
Eurobot 2015

TANO: Parts

- **Motion Control:** how the robot moves in the playing arena
- **Arm Control:** how arms are driven
- **Arm Automation:** how and when arms are activated and in what sequence of actions
- **Opponent Recognition:** where is my opponent?
- **Intelligence:** what is my strategy?

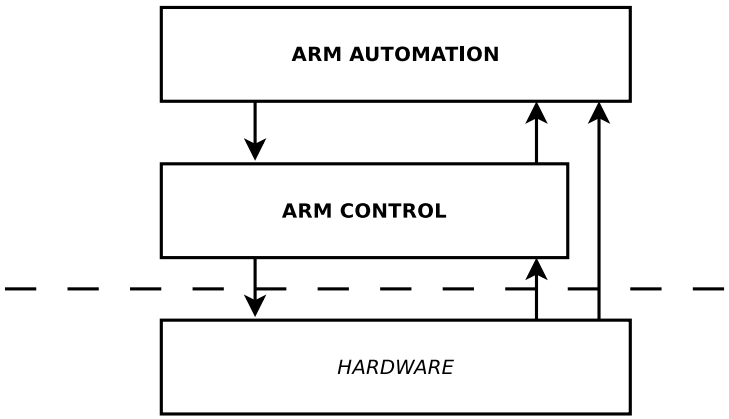
Motion Control

- **Speed Control:** how do we assure a certain wheel speed?
- **Movement Control:** how do we obtain a straight or circular movement?
- **Path Control:** how do we reach a certain point in the playing field?



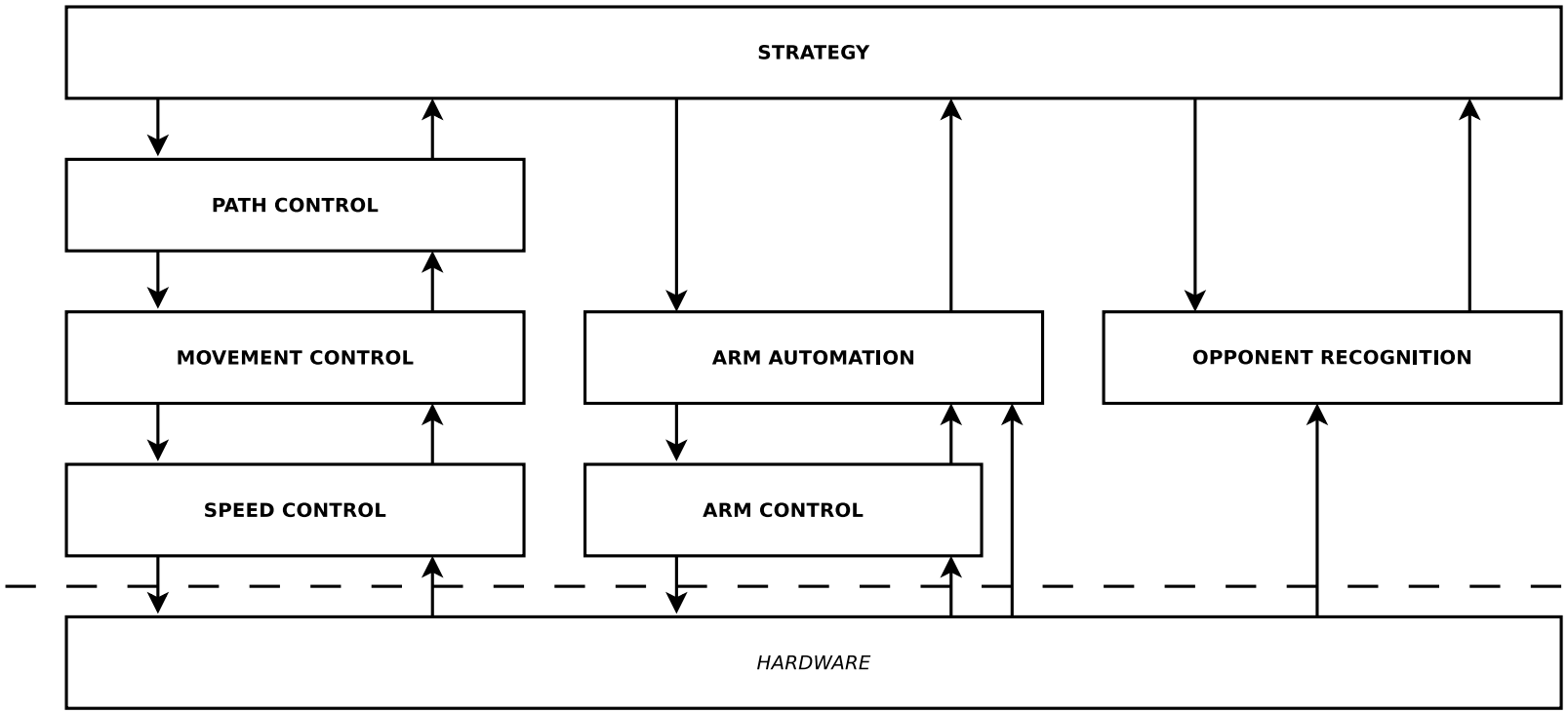
Arm Control and Automation

- **Arm Control:** how do we assure that an arm has gone in a certain position?
- **Arm Automation:** how do we perform a **reactive** task such as: *sense* the cylinder, *close the grip*, *lift the grip up*



Overall Software Structure

- A **layered architecture** is present in a robot
- Each layer/block has a specific role and uses different programming models and abstractions
- **We'll analyse and study all of them! :-)**



Basics of A Robotic System

Corrado Santoro

ARSLAB - Autonomous and Robotic Systems Laboratory
Dipartimento di Matematica e Informatica - Università di Catania, Italy
santoro@dmi.unict.it

Programmazione Sistemi Robotici