LARM - Logiciel et Architecture pour la Robotique Mobile

Software and Architecture for Mobile Robots

An Introduction

Guillaume Lozenguez

IMT Lille Douai
École Mines-Télécom
IMT-Université de Lille

What is a Robot ?

What is a Robot ?

On Wikipedia:

en

"A robot is a machine—especially one programmable by a computer—capable of carrying out a complex series of actions automatically."

fr

"Un robot est un dispositif mécatronique (alliant mécanique, électronique et informatique) conçue pour accomplir automatiquement des tâches imitant ou reproduisant, dans un domaine précis, des actions humaines."

What is a Robot ?

From my point of view

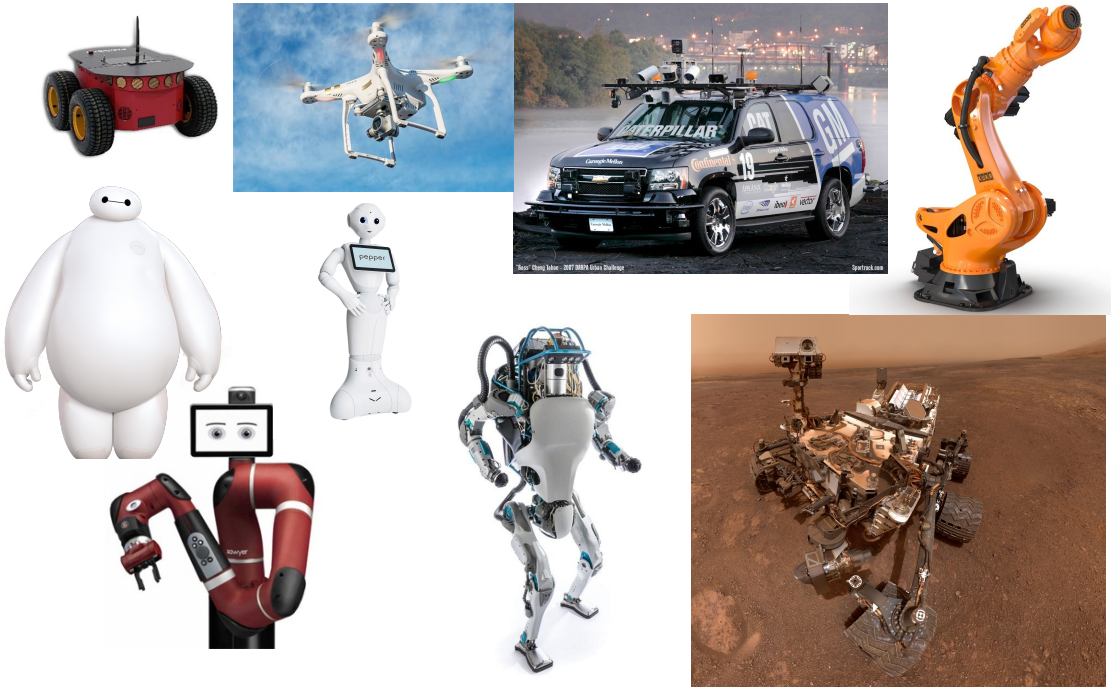
"A **robot** is a **mechatronics** machine capable of autonomously acting in a real environment."

- perceives with *sensors*
- models its environment and adapt its behavior
- acts with *actuators*

generally involves Artificail-Intelligence:

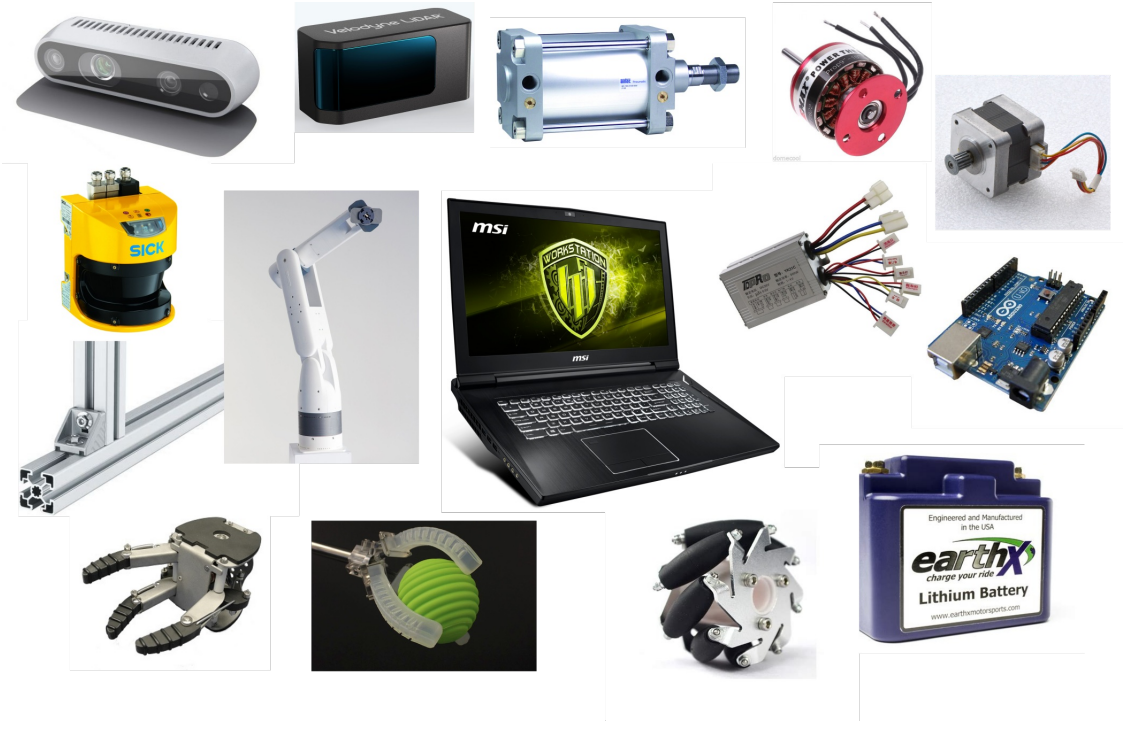
- capable to mimic natural (human, animal, insect,...) intelligence

Some examples



Macro: a large variety of robots

Some examples



Micro: a large variety of components.

From a mechanic point of view

Focus on:

- Resistance
- Weight
- Distortion
- Vibration absorption
- Machining, Assembly

for different robots:

- Fast
- Precise
- Strong
- resistant (dust, water,...)
- safe
- less expensive

From an electronic point of view

Focus on sensors, motor, energy systems and hardware.

From a automation point of view

Focus on:

- Physics science
- Signal processing
- Control system

by manipulating

- Times series, torques
- Vector, Matrices

From a software point of view

Focus on:

- Algorythms
- Knoledge representation
- Artificial intelligence
- Software architecture

Robots are complex and singular systems
which require modular computer programs.

Schedule

- Introduction
- Presentation of the UV
- Today: First contact with **Linux** and **ROS**

UV-LARM

Software and Architecture for Mobile Robots

Mostly about: autonomous navigation.

- Communicate with robot components
- Control robot movements (nonholonomic robot)
- Perception of the local environment (laser, vision)
- SLAM (Simultaneous Localization and Mapping)
- Path finding and navigation.

UV-LARM - Schedule

1st week: Introduction to notions with tutorials.

2d week: Chalenge kickoff and some complementary notions.

3d week: Chalenge as your projet.

4th week: Evaluation through the code you provide.

Always *9* to *12* and *14* to *18*.

In *Develter* or *3130*.

With or without a teacher.

Why ROS:

ROS: The Robot Operating System (ROS) is a set of *software libraries* and *tools* that help you build robot applications.

- The number one Robotic Middle used in academic
- Open and oriented toward its *many contributors*
- Supported by most of the professionals

It permits thinking robot programs in a modular way as independent program's *nodes* working together by communicating through *topics*.

It comes with useful functionality like *frame* management and *transform*

Why Ubuntu Linux:

Because

- We love *GNU*
- ROS natively support Ubuntu
- Linux is open and well documented

Today:

wiki ROS Beginner tutorials:

- Create a ROS project (catkin)
- Implement communicating nodes (publisher and subscribers)

But first : Installation and configuration of Ubuntu:

- Setup tutorial on gitbook: https://ceri-num.gitbook.io/uv-larm/

Before to go :

Reminder on Linux Terminal (or Shell)

- **ls**: list directories elements
- **cd**: change the directory
- **rm**: permanently remove a file
- **man**: open the manual on a command
- **sudo**: act as the super-user
- **find**, **egrep**, **cat**, **top**, **ps**, **apropos**...

And tabulation is your best friend.