人工
智能

The
ShanghAI
Lectures

上海

授课

The Shanghai Lectures 2022

Natural and Artificial Intelligence in Embodied Physical Agents

November 10th, 2022

From Zagreb, Croatia

Today's program (CET)

08:30   sites begin connecting

08:55   all sites are ready

09:00 (Fabio) Welcome

09:05 <u>The Role of Embodiment in Intelligent Systems vs. Practical Robotics in 2022</u>

10:00 Break

10:10 Guest Lecture by Enrica Zereik, CNR-INM and Heron@CNR Joint Lab, Genoa, Italy:

<u>Controlling 'uncontrollable' stuff: Playing with the H2Arm</u>

11:00 Wrap-up

Today's Guest Lecture

10:10 CET Enrica Zereik, CNR-INM and Heron@CNR Joint Lab, Genoa, Italy

«Controlling 'uncontrollable' stuff: Playing with the H2Arm»

Stay tuned!

Lecture 2

The Role of Embodiment in Intelligent Systems
vs.
Practical Robotics in 2022

Fabio Bonsignorio
Professor, ERA CHAIR in AI for Robotics

University of Zagreb
Faculty of Electrical Engineering and Computing
Laboratory for Autonomous Systems and Mobile Robotics

www.heronrobots.com

Today's topics

- **Practical Robotics in 2022**

- **The classical approach: Cognition as computation**

- **Successes and failures of the classical approach**

- **Some problems of the classical approach**
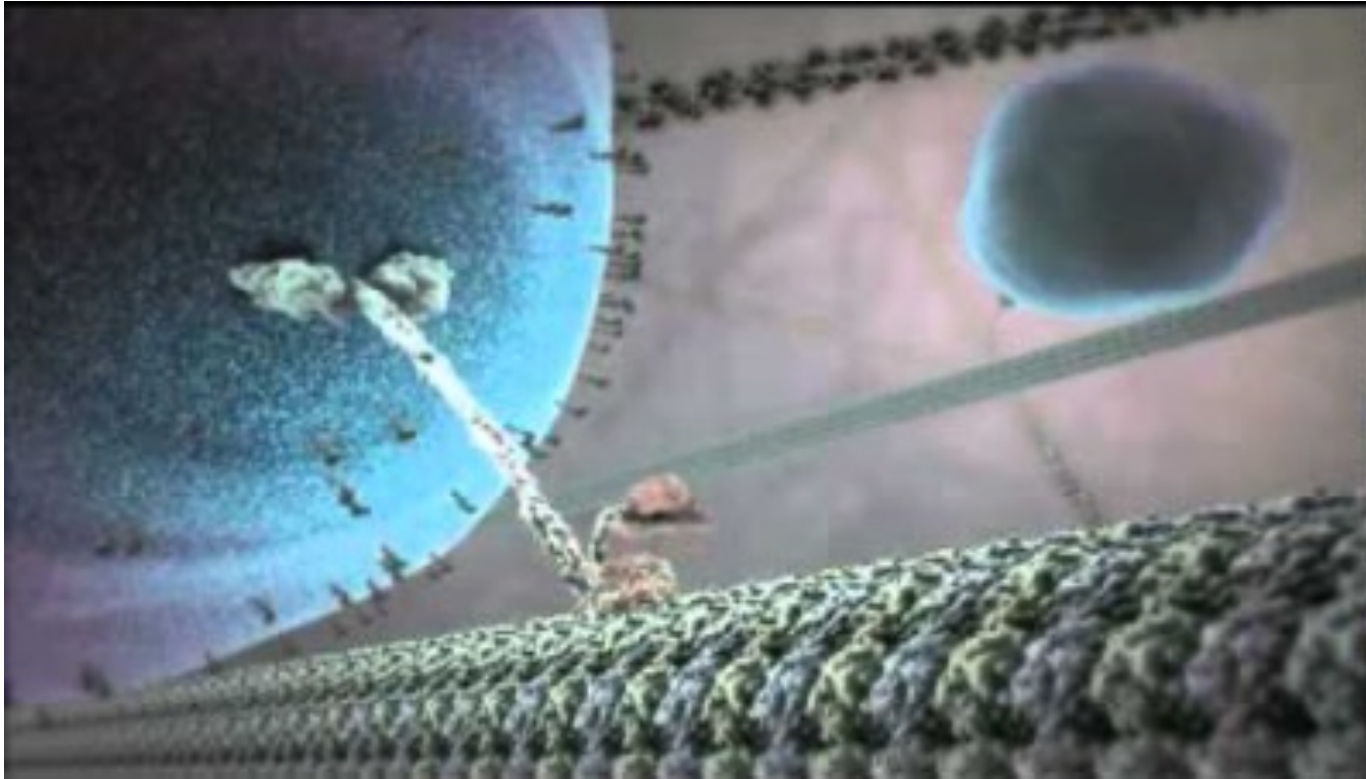
- **The need for an embodied approach**

Today's topics

- **Practical Robotics in 2022**

- The classical approach: Cognition as computation

- Successes and failures of the classical approach

- Some problems of the classical approach

- The need for an embodied approach

Issues to think about:
an unfair comparison

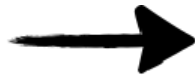# Issues to think about:
an unfair comparison

The synthetic methodology

Slogan:

"Understanding by building"

**modeling behavior of interest
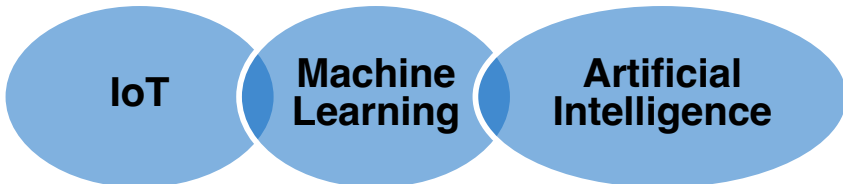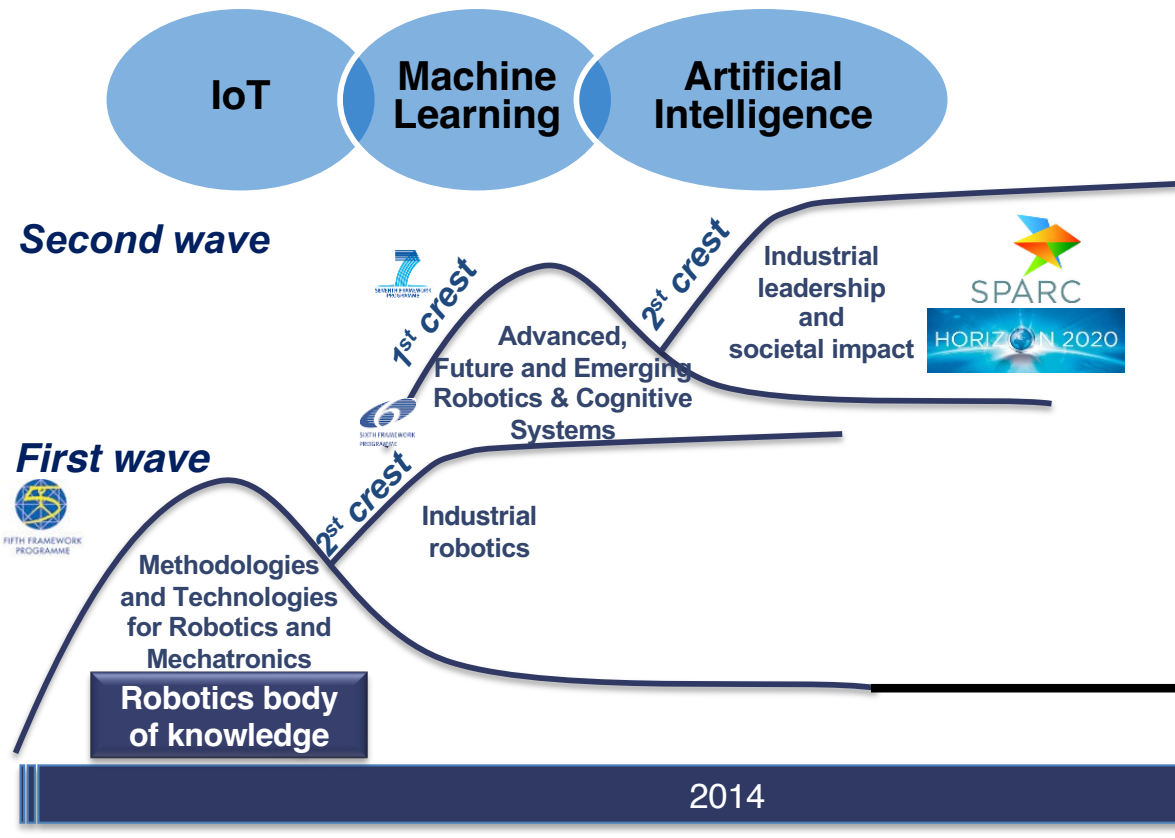abstraction of principles**

→

**robots as tools for scientific
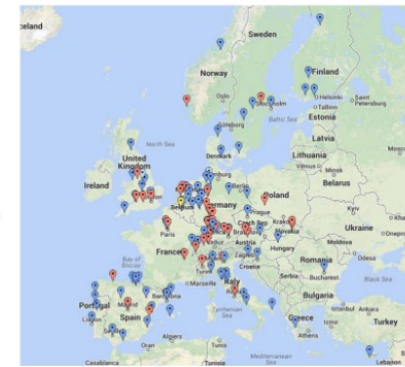investigation**

Many examples during ShanghAI lectures

The second wave

we are here!

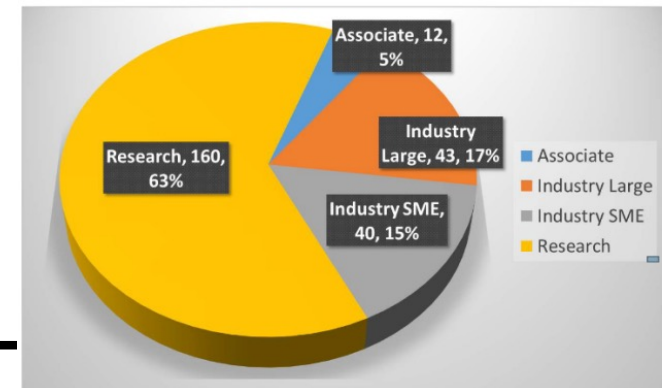IoT   Machine Learning   Artificial Intelligence

Second wave

First wave

Industrial leadership and societal impact

Advanced, Future and Emerging Robotics & Cognitive Systems

1st crest

2st crest

2st crest

Industrial robotics

Methodologies and Technologies for Robotics and Mechatronics

Robotics body of knowledge

SPARC

HORIZON 2020

Membership development

280 member organisations

Legend:
- Industry
- Research
- Associate
- euRobotics AISBL

Associate, 12, 5%
Industry Large, 43, 17%
Research, 160, 63%
Industry SME, 40, 15%

- Associate
- Industry Large
- Industry SME
- Research

2014

2020

Important Remark

We need a lot of code and system engineering even for simple tasks....

How to reduce the effort?

Robot Operating System (ros.org)

From Opensource Robotics Foundation (formerly from Willow Garage) (openrobotics.org)

- ROS is a 'middleware' (a layer in between the OS and your robot application specific system software)

- It provides messages passing between processes, code, libraries organized into 'packages' (filled with Python and C++ code mainly), to implement as 'blackboxes' miryads of complex function to perform a wide range of tasks (including the examples we have seen before)

- ROS is a network distributed system relaying on tcp-ip where processes, called nodes, are responsible for individual sets of tasks. Nodes communicate with each other using messages passing via logical channels called topics. Each node can send or get data from other nodes using a publish/subscribe model. r.

- ROS allows massive code reuse in robotics research and development and is perfect for rapid prototyping of new systems and applications, in particular where efficiency is not a main constraint (although being rather cumbersome and inefficient and having a steep learning curve, you will see ☺).

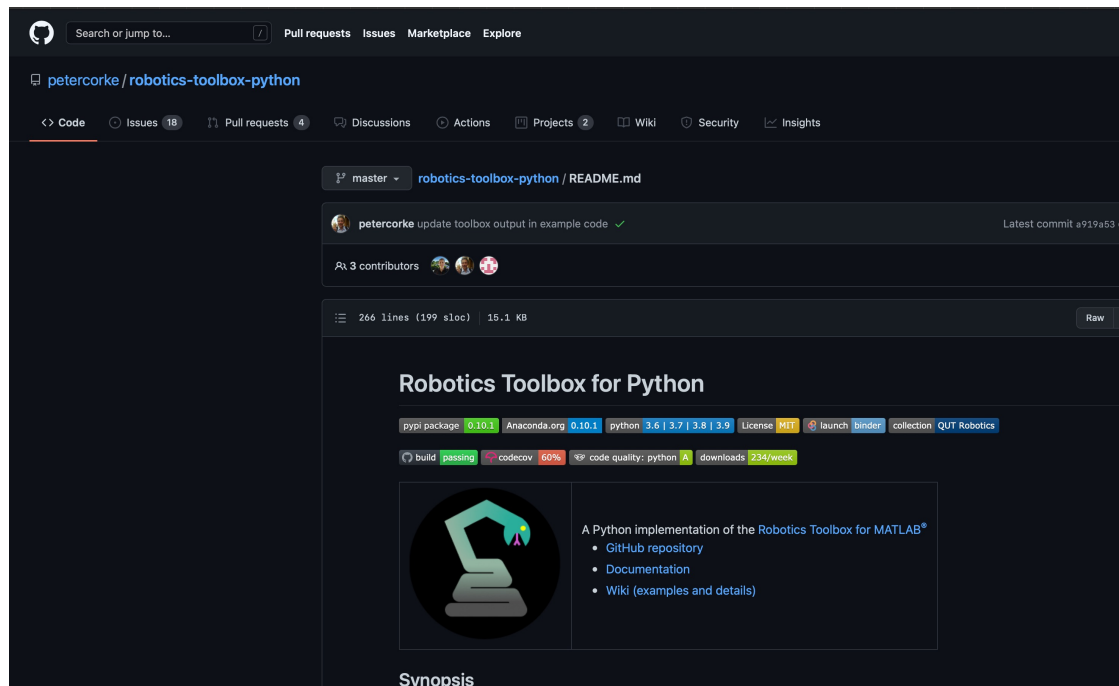- You can add your own code packages
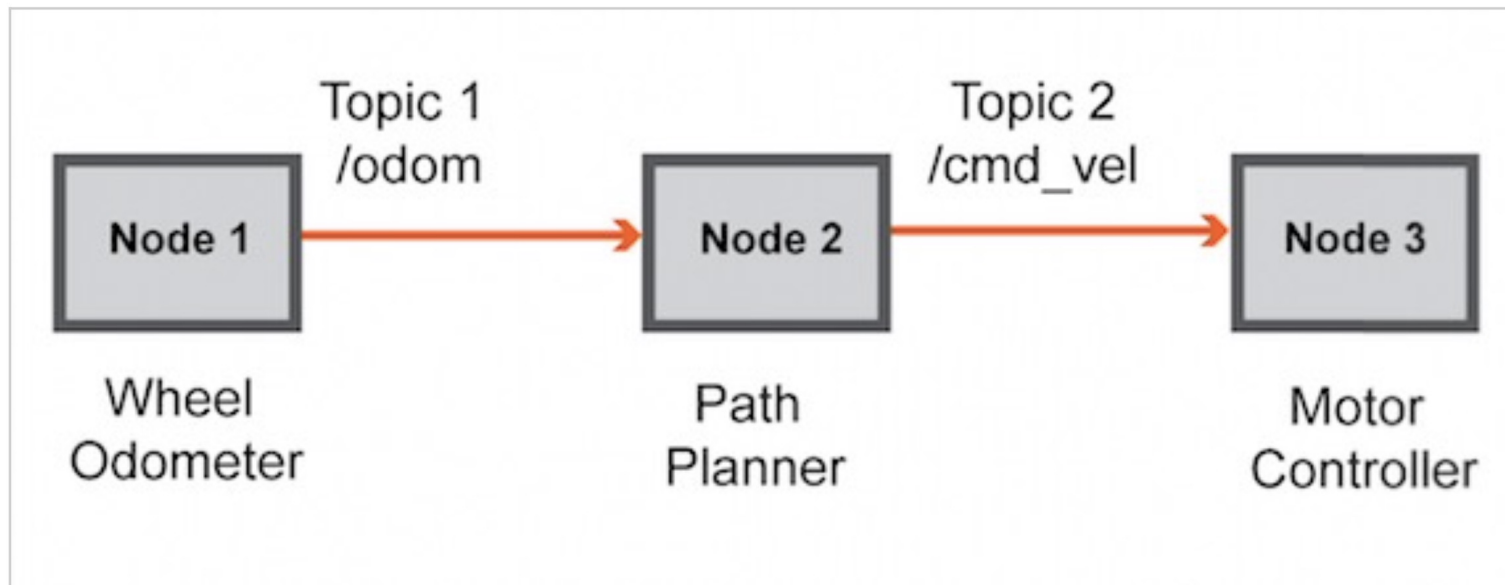
Nodes and topics. Image adapted from a tutorial from Justin Huang.

And easier toolboxes

- Matlab Robotics Toolbox, see Mathworks Matlab doc

- Python Robotics Toolbox (still in development), see
  https://github.com/petercorke/robotics-toolbox-python/blob/master/README.md

1) http://wiki.ros.org/turtlebot/Tutorials/indigo/Turtlebot%20Installation

2) http://wiki.ros.org/turtlebot_gazebo/Tutorials/indigo/Gazebo%20Bringup%20Guide

TurtleBot

TurtleBot is a low-cost, personal robot kit with open-source software. TurtleBot was created at Willow Garage by Melonee Wise and Tully Foote in November 2010. With TurtleBot, you'll be able to build a robot that can drive around your house, see in 3D, and have enough horsepower to create exciting applications.

The TurtleBot kit consists of a mobile base, 2D/3D distance sensor, laptop computer or SBC(Single Board Computer), and the TurtleBot mounting hardware kit. In addition to the TurtleBot kit, users can download the TurtleBot SDK from the ROS wiki. TurtleBot is designed to be easy to buy, build, and assemble, using off the shelf consumer products and parts that easily can be created from standard materials. As an entry level mobile robotics platform, TurtleBot has many of the same capabilities of the company's larger robotics platforms, like PR2.

For more information about hardware and software, please see http://www.turtlebot.com

ROS 2 Documentation

The ROS Wiki is for ROS 1. Are you using ROS 2 (Dashing/Foxy/Rolling)? Check out the ROS 2 Documentation

Wiki

Distributions

ROS/Installation

ROS/Tutorials

RecentChanges

Robots/TurtleBot

Page

Immutable Page

Info

Attachments

More Actions:

Raw Text

Do

User

Login

About | Support | Discussion Forum | Service Status | Q&A answers.ros.org

Search:

Submit

Documentation   Browse Software   News   Download

ROS.org

turtlebot/ **Tutorials**/ **indigo**/ **Turtlebot Installation**

Note: This tutorial assumes that you have completed the previous tutorials: Interacting with your Turtlebot.

💡 Please ask about problems and questions regarding this tutorial on 🌐 answers.ros.org. Don't forget to include in your question the link to this page, the versions of your OS & ROS, and also add appropriate tags.

Turtlebot Installation

Description: Installing software onto the turtlebot.

Keywords: turtlebot installation

Tutorial Level: BEGINNER

Next Tutorial: PC Installation

Contents

ROS 2 Documentation

The ROS Wiki is for ROS 1. Are you using ROS 2 (Dashing/Foxy/Rolling)? Check out the ROS 2 Documentation

Wiki

Distributions

ROS/Installation

ROS/Tutorials

RecentChanges

Turtlebot Installation

Page

Immutable Page

Info

Attachments

More Actions:

Raw Text ▾

Do

User

Login

ROS.org

Search: [   ] [Submit]

| Documentation | Browse Software | News | Download |

turtlebot_gazebo/ **Tutorials**/ indigo/ **Gazebo Bringup Guide**

Note: This tutorial assumes that you have completed the previous tutorials: **TurtleBot Installation**.

💡 Please ask about problems and questions regarding this tutorial on 🌐answers.ros.org. Don't forget to include in your question the link to this page, the versions of your OS & ROS, and also add appropriate tags.

Gazebo Bringup Guide

Description: See the simulated turtlebot in Gazebo.

Keywords: gazebo, simulator

Tutorial Level: BEGINNER

Next Tutorial: Explore the Gazebo world

Bringup Turtlebot.

Contents
1. Overview
2. Execution
3. Notes
4. What Next?

1. Overview

In this tutorial, we will bring up the turtlebot in the Gazebo simulator. This tutorial assumes you have completed TurtleBot Installation.

2. Execution

The following command starts the turtlebot in the simulated world

```
$ roslaunch turtlebot_gazebo turtlebot_world.launch
```

even more complex…

ROS.org

Search: [   ] Submit

| Documentation | Browse Software | News | Download |

Robots/ REEM-C/ Tutorials

1. Tutorials Installation

1. **Installing Ubuntu with ROS**
   This tutorial describes the steps needed to get a proper Unbuntu and ROS installation to have a system up and running for the REEM-C tutorials.
2. **Installing REEM-C Simulation**
   A brief summary of commands to install the required packages for REEM-C simulation

2. Tutorials

1. **Launching a REEM-C Gazebo simulation**
   This tutorial describes how to open a Gazebo simulation for the REEM-C robot.
2. **Play pre-recorded motions on REEM-C**
   This tutorial will show you how to play pre-recorded motions on the REEM-C robot.
3. **Use MoveIt! with REEM-C**
   This tutorial will show you how to play with Move It ! with the REEM-C robot.

Create a new tutorial:

[   ] Enter tutorial name

Except where otherwise noted, the ROS wiki is licensed under the
Creative Commons Attribution 3.0

ROS 2 Documentation

The ROS Wiki is for ROS 1. Are you using ROS 2 (Dashing/Foxy/Rolling)?
Check out the ROS 2 Documentation

Wiki
Distributions
ROS/Installation
ROS/Tutorials
RecentChanges
Robots/REEM-C/Tutorials

Page
Immutable Page
Info
Attachments
More Actions:
Raw Text ▼
Do
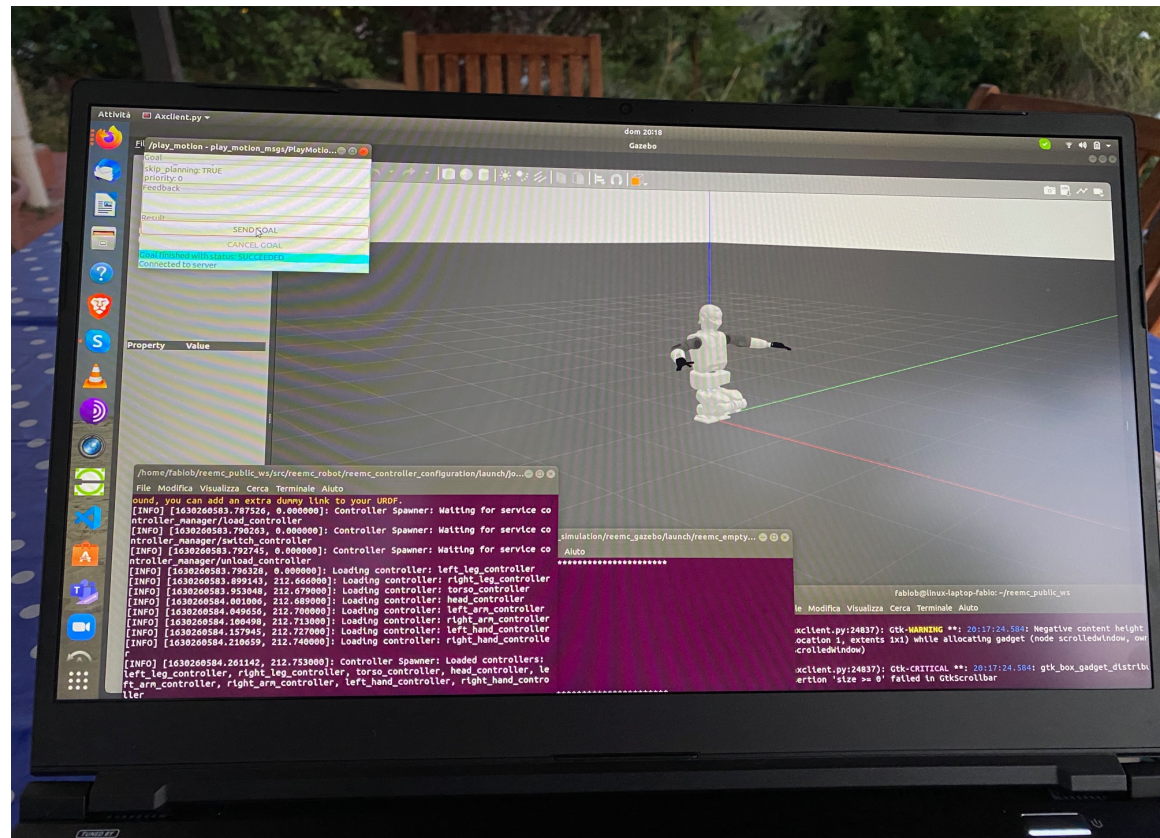
User
Login

Wiki: Robots/REEM-C/Tutorials (last edited 2018-07-05 10:59:14 by VictorLopez)

Brought to you by: Open Robotics

even more complex…
http://wiki.ros.org/Robots/REEM-C/Tutorials

The most? complex…

Artificial Intelligence & Autopilot

We develop and deploy autonomy at scale in vehicles, robots and more. We believe that an approach based on advanced AI for vision and planning, supported by efficient use of inference hardware, is the only way to achieve a general solution for full self-driving and beyond.
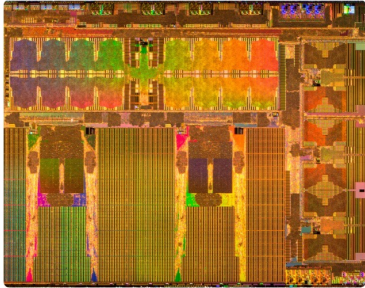

Hardware

Build silicon chips that power our full self-driving software from the ground up, taking every small architectural and micro-architectural improvement into account while pushing hard to squeeze maximum silicon performance-per-watt. Perform floor-planning, timing and power analyses on the design. Write robust, randomized tests and scoreboards to verify functionality and performance. Implement compilers and drivers to program and communicate with the chip, with a strong focus on performance optimization and power savings. Finally, validate the silicon chip and bring it to mass production.

Neural Networks

Apply cutting-edge research to train deep neural networks on problems ranging from perception to control. Our per-camera networks analyze raw images to perform semantic segmentation, object detection and monocular depth estimation. Our birds-eye-view networks take video from all cameras to output the road layout, static infrastructure and 3D objects directly in the top-down view. Our networks learn from the most complicated and diverse scenarios in the world, iteratively sourced from our fleet of nearly 1M vehicles in real time. A full build of Autopilot neural networks involves 48 networks that take 70,000 GPU hours to train 🔥. Together, they output 1,000 distinct tensors (predictions) at each timestep.

Hardware

Build silicon chips that power our full self-driving software from the ground up, taking every small architectural and micro-architectural improvement into account while pushing hard to squeeze maximum silicon performance-per-watt. Perform floor-planning, timing and power analyses on the design. Write robust, randomized tests and scoreboards to verify functionality and performance. Implement compilers and drivers to program and communicate with the chip, with a strong focus on performance optimization and power savings. Finally, validate the silicon chip and bring it to mass production.

Neural Networks

Apply cutting-edge research to train deep neural networks on problems ranging from perception to control. Our per-camera networks analyze raw images to perform semantic segmentation, object detection and monocular depth estimation. Our birds-eye-view networks take video from all cameras to output the road layout, static infrastructure and 3D objects directly in the top-down view. Our networks learn from the most complicated and diverse scenarios in the world, iteratively sourced from our fleet of nearly 1M vehicles in real time. A full build of Autopilot neural networks involves 48 networks that take 70,000 GPU hours to train 🔥. Together, they output 1,000 distinct tensors (predictions) at each timestep.


Autonomy Algorithms

Develop the core algorithms that drive the car by creating a high-fidelity representation of the world and planning trajectories in that space. In order to train the neural networks to predict such representations, algorithmically create accurate and large-scale ground truth data by combining information from the car's sensors across space and time. Use state-of-the-art techniques to build a robust planning and decision-making system that operates in complicated real-world situations under uncertainty. Evaluate your algorithms at the scale of the entire Tesla fleet.

The most? complex…

Autonomy Algorithms

Develop the core algorithms that drive the car by creating a high-fidelity representation of the world and planning trajectories in that space. In order to train the neural networks to predict such representations, algorithmically create accurate and large-scale ground truth data by combining information from the car's sensors across space and time. Use state-of-the-art techniques to build a robust planning and decision-making system that operates in complicated real-world situations under uncertainty. Evaluate your algorithms at the scale of the entire Tesla fleet.
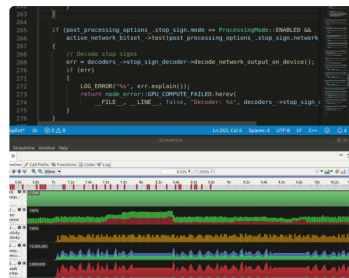
Code Foundations

Throughput, latency, correctness and determinism are the main metrics we optimize our code for. Build the Autopilot software foundations up from the lowest levels of the stack, tightly integrating with our custom hardware. Implement super-reliable bootloaders with support for over-the-air updates and bring up customized Linux kernels. Write fast, memory-efficient low-level code to capture high-frequency, high-volume data from our sensors, and to share it with multiple consumer processes— without impacting central memory access latency or starving critical functional code from CPU cycles. Squeeze and pipeline compute across a variety of hardware processing units, distributed across multiple system-on-chips.

Evaluation Infrastructure

Build open- and closed-loop, hardware-in-the-loop evaluation tools and infrastructure at scale, to accelerate the pace of innovation, track performance improvements and prevent regressions. Leverage anonymized characteristic clips from our fleet and integrate them into large suites of test cases. Write code simulating our real-world environment, producing highly realistic graphics and other sensor data that feed our Autopilot software for live debugging or automated testing.

The most? complex…(so far, IFF ☺)

Tesla Bot

Develop the next generation of automation, including a general purpose, bi-pedal, humanoid robot capable of performing tasks that are unsafe, repetitive or boring. We're seeking mechanical, electrical, controls and software engineers to help us leverage our AI expertise beyond our vehicle fleet.

DRC partial vindication? :-) ANA Avatar XPRIZE

NimbRo Team University of Bonn

Today's topics

"Birth" of AI, 1956 Meeting at Dartmouth College, Hanover, New Hampshire

1 Septemper 1955: a proposal for a "2 month, 10 man study of artificial intelligence" gave birth to a word that would define a field set on forever changin

George A. Miller, Psychologist
"The Magical Number Seven Plus or Minus Two"

John McCarthy, Computer Scientist
Initiator of Artificial Intelligence

Herbert Simon
and Allen Newell
The "Logic Theorist"

Noam Chomsky,
Linguist
"Syntactic Structures"

50 Years later, 2006 Meeting on Monte Verità, Ascona, Switzerland

Information Theory and Quantification

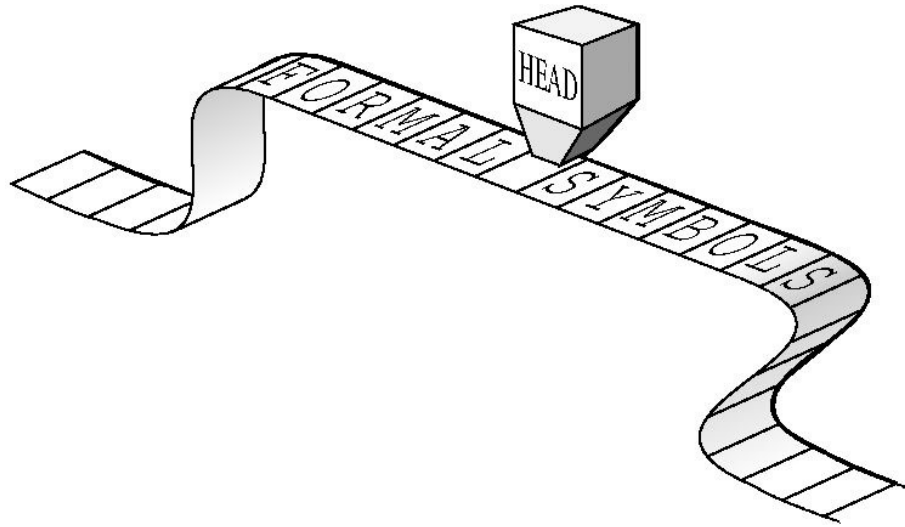# 50 Years later, 2006 Meeting on Monte Verità, Ascona, Switzerland

Monte Verità: "il luogo dove la nostra fronte sfiora il cielo..."

Nel diciannovesimo secolo e nei primi anni del ventesimo secolo il Ticino, repubblica e cantone dal 1803, diventò un passaggio verso sud e destinazione privilegiata di un gruppo di solitari anticonvenzionali, i quali trovarono nella regione, con la sua atmosfera meridionale, terreno fertile in cui piantare quei semi dell'utopia che non erano riusciti a coltivare a nord.

Il Ticino venne a rappresentare l'antitesi del nord urbanizzato e industrializzato, un santuario per qualsiasi tipo di idealista. Dal 1900 in poi il monte Monescia sopra Ascona diventò un polo di attrazione per chi cercava una vita "alternativa". Questi riformatori, i quali cercavano una terza via tra il blocco capitalista e quello comunista, finirono col trovare casa nella regione.
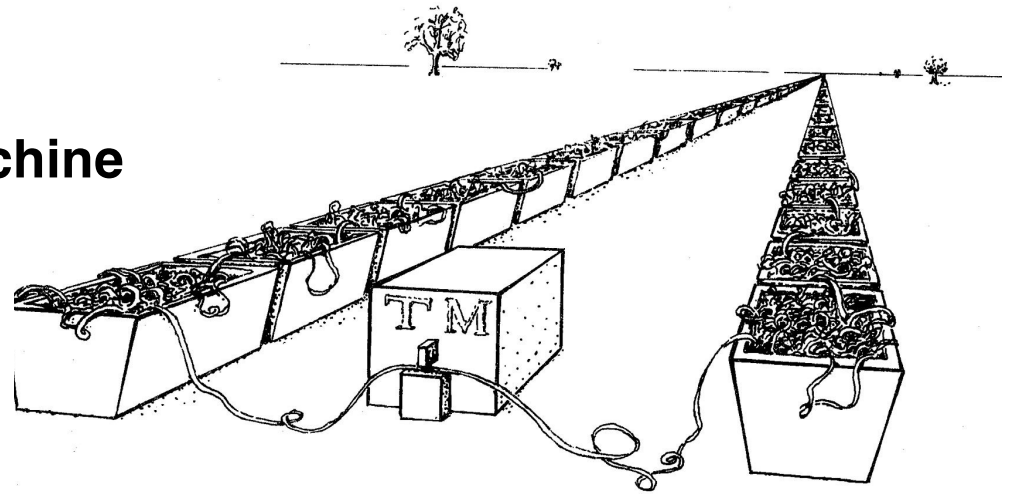
Turing Machine (1)

Turing Machine (2)

move tape R

Intructions on the tape

HEAD

FORMAL SYMBOLS

Read/write on tape

move tape L
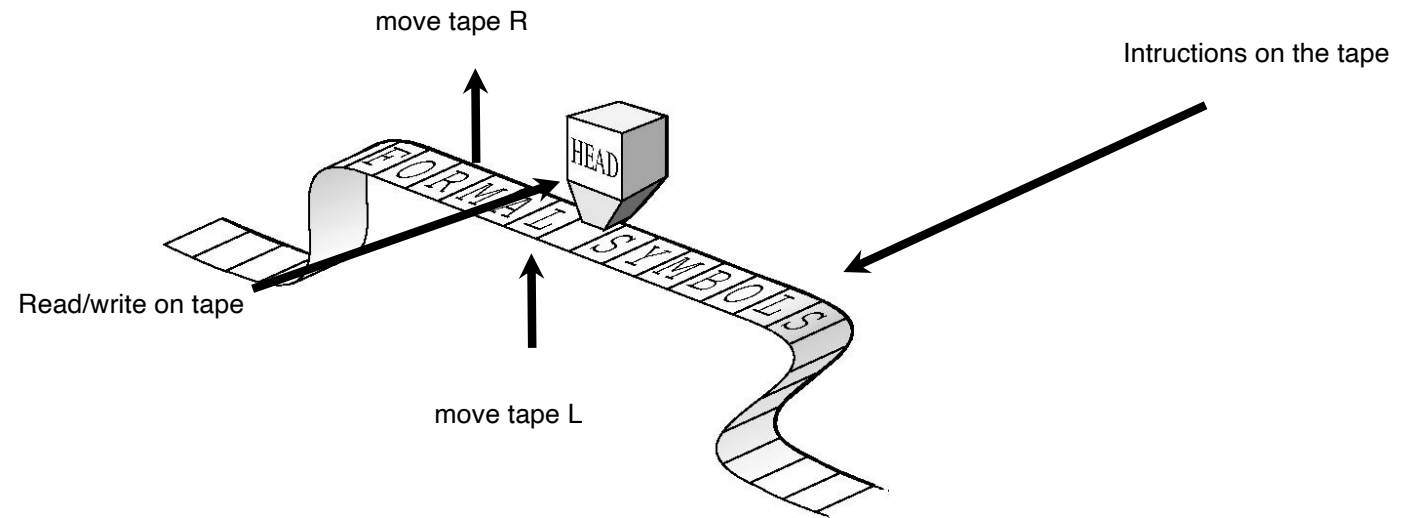
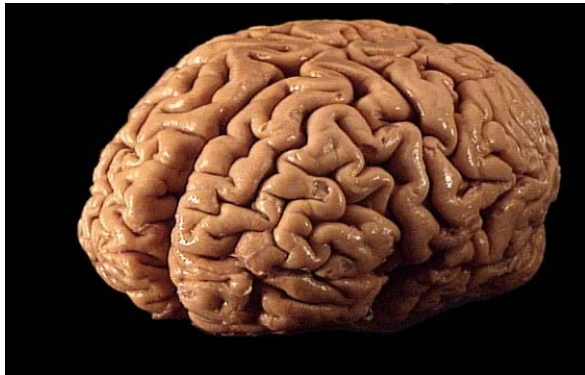| input from tape	1	2	state of read/write head
_	_R2	HALT	
A	AL1	BR2	
B	BL1	AR2	
C	CL1	CR2	

The Universal Turing Machine

Turing Machine (5)

an "embodied" Turing Machine

Cartoon by Roger Penrose

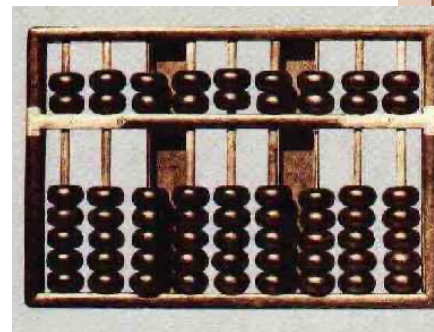# Functionalism and the "Physical Symbol Systems Hypothesis"
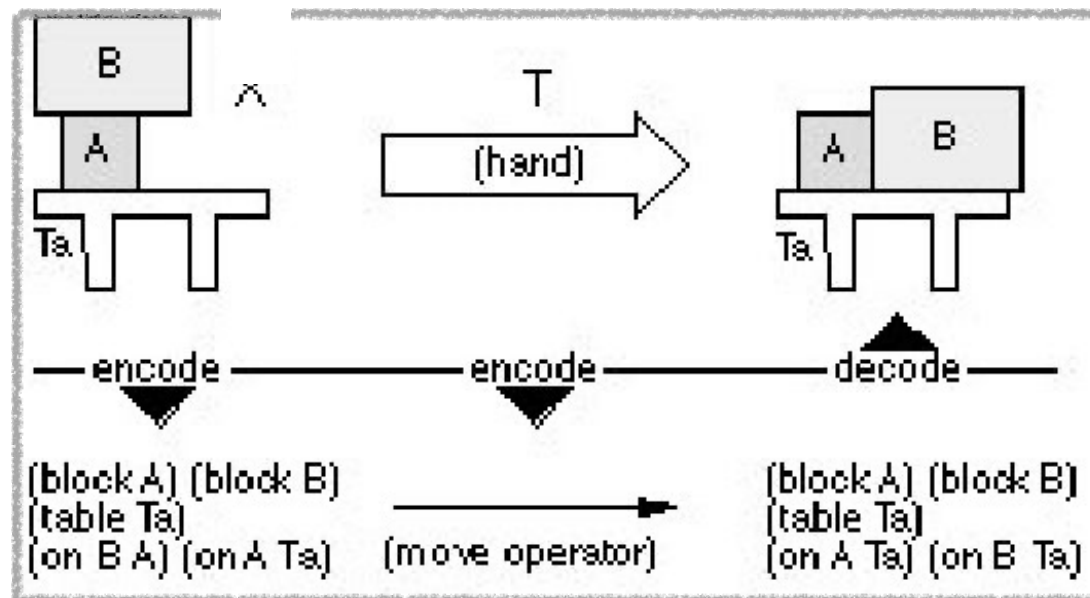
biological
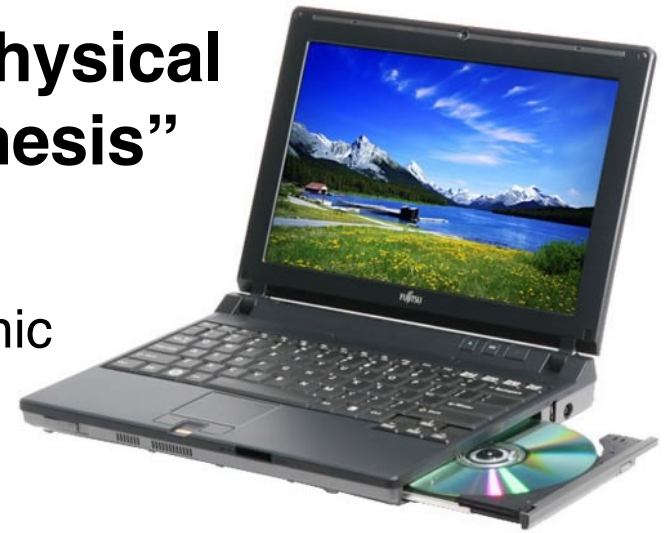
electronic

Swiss cheese
Hilary Putnam
(American
Philosopher)

mechanical

Functionalism and the "Physical Symbol Systems" Hypothesis

Model/Representation:

GOFAI

G

O

F

A

I

(John Haugeland, Philosopher)

Classical AI: Research areas

- **problem solving**

- **knowledge representation and reasoning**

- **acting logically**

- **uncertain knowledge and reasoning**

- **learning and memory**

- **communicating, perceiving and acting**

- **(adapted from Russell/Norvig: Artificial intelligence, a modern approach)**

Today's topics

- Practical Robotics in 2022

- The classical approach: Cognition as computation

- **Successes and failures of the classical approach**

- Some problems of the classical approach

- The need for an embodied approach

- The "frame-of-reference" problem

Classical AI: Successes

- **search engines**

- **formal games (chess!)**

- **text processing systems/translation**

- **data mining systems**

- **restricted natural language systems**

- **appliances**

- **manufacturing**

- **control systems**

Chess: New York, 1997

Garry Kasparov

The best player in the world shows no signs of slowing down

Deep Blue

This 1.4 ton 8-year-old sure plays a mean game of chess

1 win   3 draws   2 wins

Classical AI: Failures

- recognizing a face in the crowd

- vision/perception in the real world

- common sense

- movement, manipulation of objects

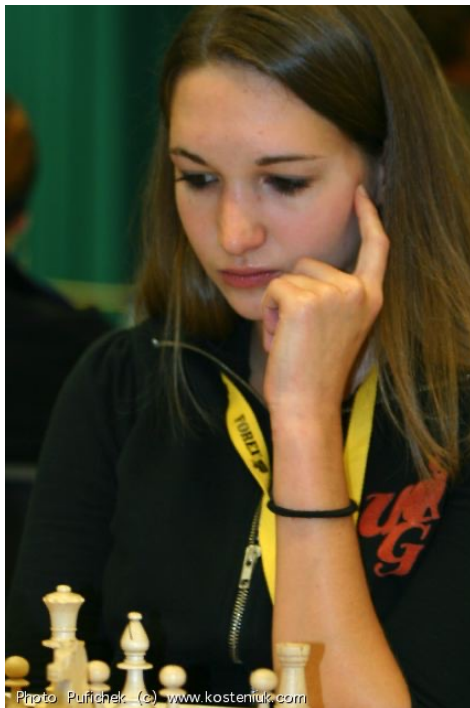- walking, running, swimming, flying

- speech (everyday natural language)
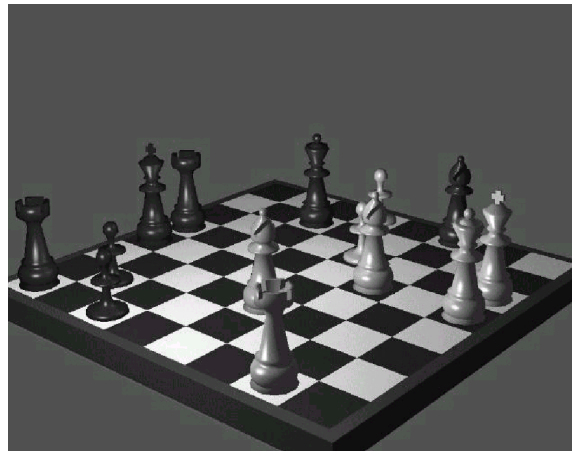
Why is perception hard?

Today's topics

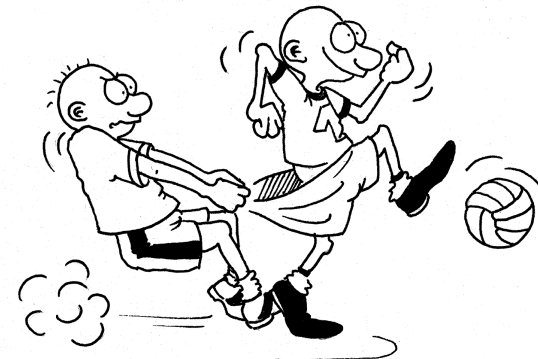# Fundamental problems of the classical approach

Monika Seps, chess master
former master student
AI Lab, Zurich

virtual, formal world

real world

Differences
real vs. virtual worlds

Successes and failures of the classical approach

Successes

applications (e.g. Google)

chess

manufacturing

(applications:"controlled"artificial worlds)

Failures

foundations of behavior

natural forms of intelligence

interaction with real world

(scientific: "real worlds")

Industrial environments vs. real world

industrial environments

environment well-known

little uncertainty

predictability

("controlled"artificial worlds)

real world environment

limited knowledge and predictability
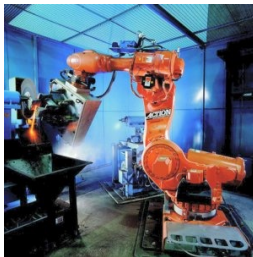
rapidly changing

high-level of uncertainty

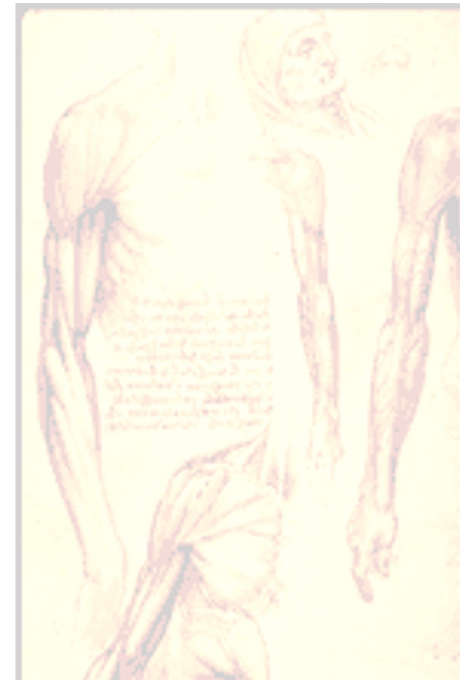("real" worlds)

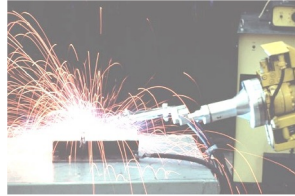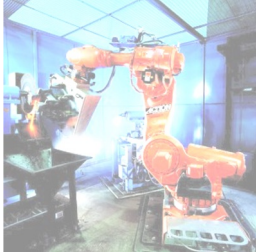# Industrial robots vs. natural systems

principles:
- **strong, precise, fast motors**
- **centralized control**
- **computing power**
- **optimization**

Industrial robots

Industrial robots vs. natural systems

humans

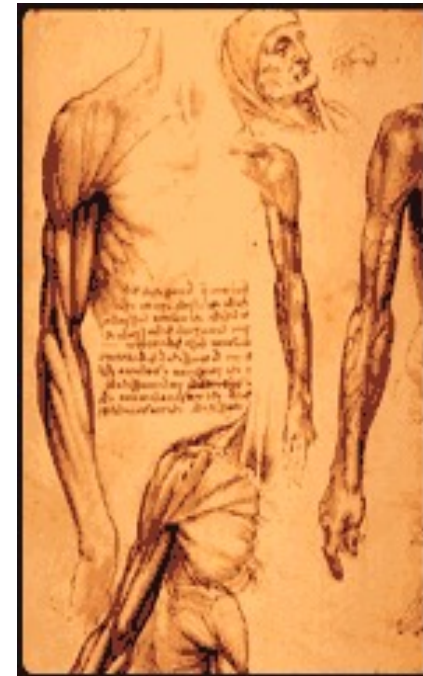principles:
- low precision
- compliant
- reactive
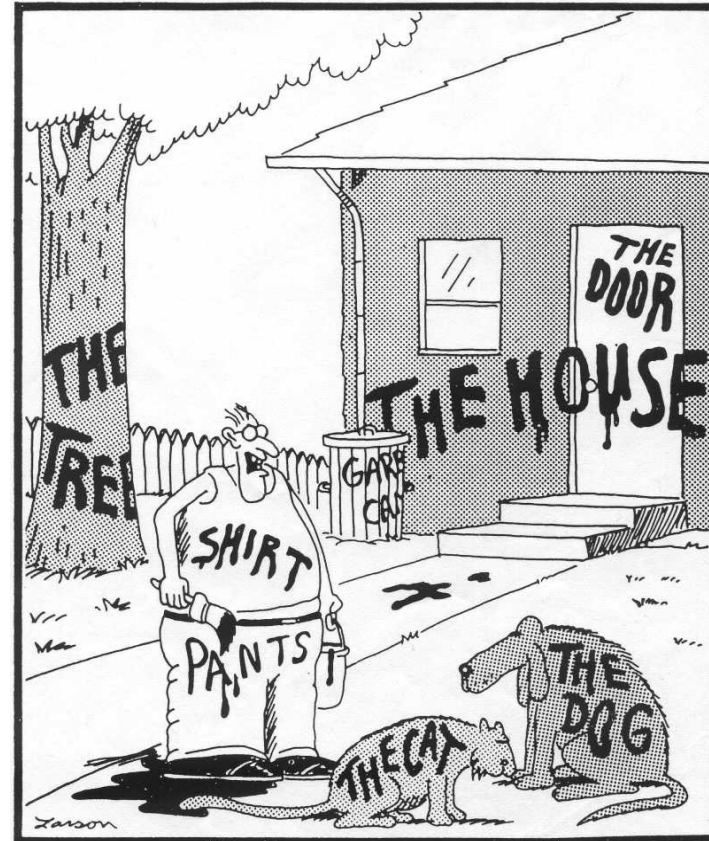- coping with uncertainty

no direct transfer of methods

Fundamental problems of classical approach

- **"symbol grounding problem"**

- **"frame problem"**

- **"homunculus problem"**

The "symbol grounding" problem

**real world:
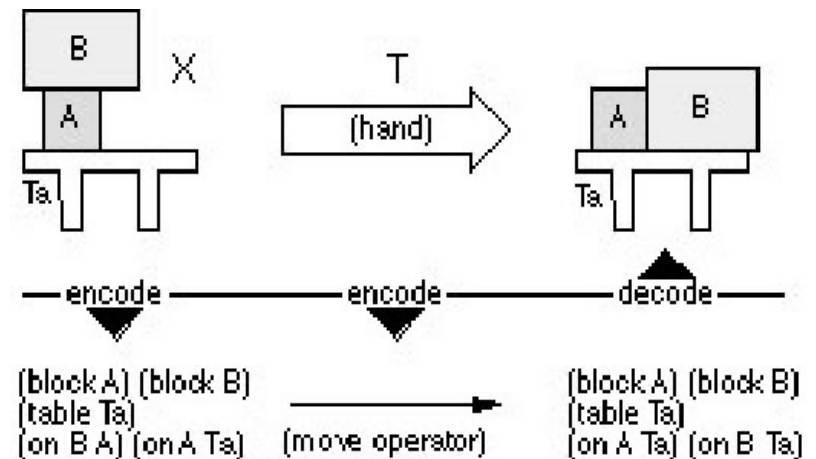doesn't come
with labels ...**

"Now! ... *That* should clear up a few things around here!"

Gary Larson

The "frame problem"
Maintaining model of real world

- the more detailed the harder

- information acquisition

- most changes: irrelevant to current situation

Today's topics

- Practical Robotics in 2022

- The classical approach: Cognition as computation

- Successes and failures of the classical approach

- Some problems of the classical approach

- **The need for an embodied approach**

Two views of intelligence

classical:

cognition as computation

embodiment:

cognition emergent from sensory-motor and interaction processes

The need for an embodied perspective

- "failures" of classical AI

- fundamental problems of classical approach

- Wolpert's quote:

The need for an embodied perspective

"Why do plants not have brains?"

# The need for an embodied perspective

"Why do plants not have brains?

The answer is actually quite simple — they don't have to move." Lewis Wolpert, UCL

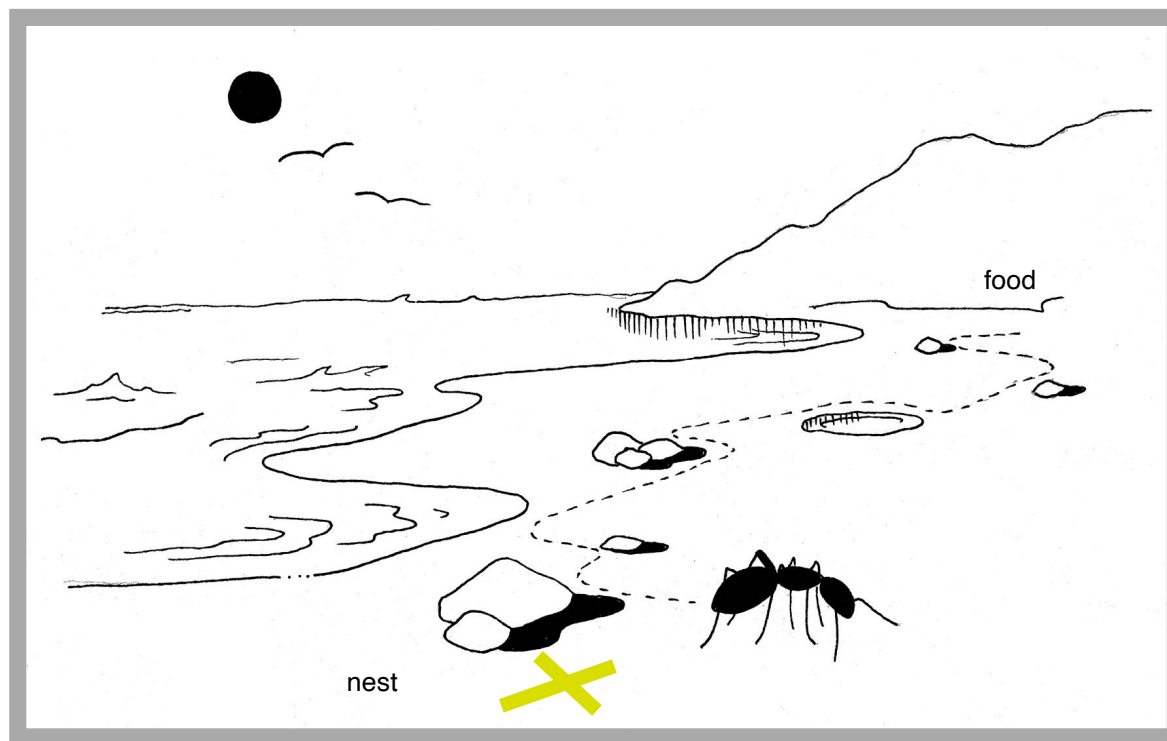# The need for an embodied perspective

- "failures" of classical AI

- fundamental problems of classical approach

- Wolpert's quote: Why do plants not …?

- Interaction with environment: always mediated by body

"Frame-of-reference"
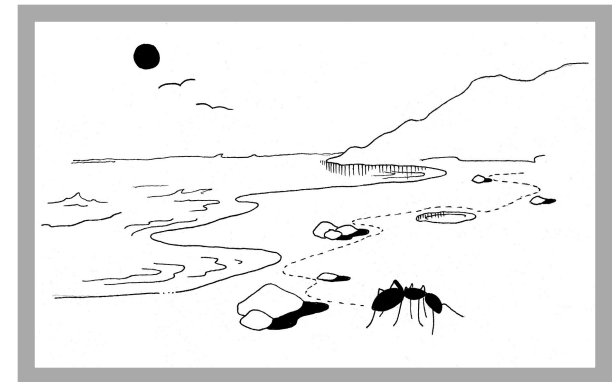# Simon's ant on the beach
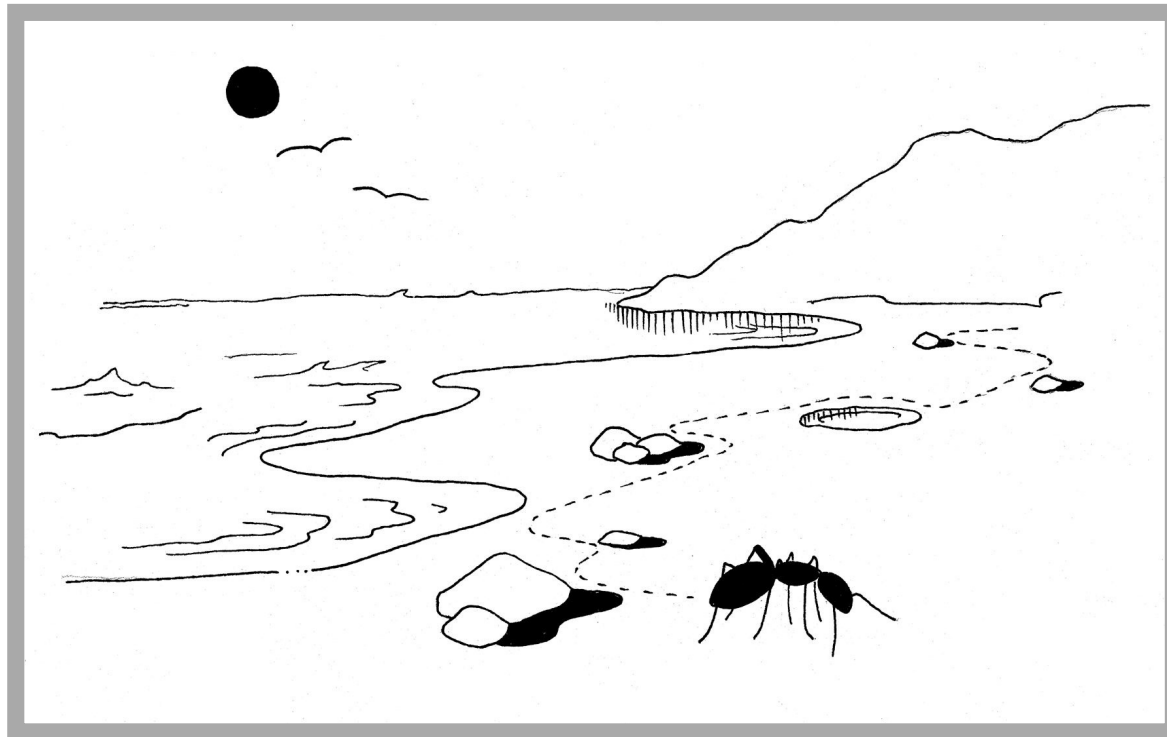
"Frame-of-reference"
Simon's ant on the beach
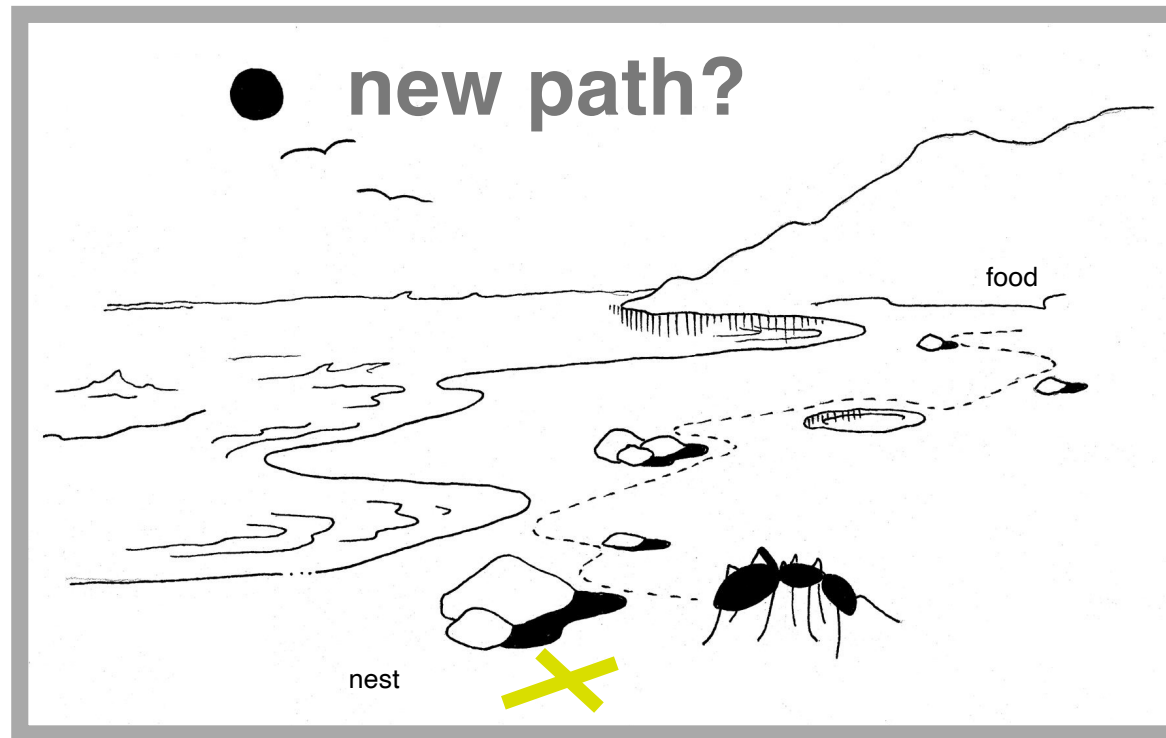
"Frame-of-reference"
Simon's ant on the beach

- **simple behavioral rules**

- **complexity in interaction,
  not — necessarily — in brain**

- thought experiment:
  **increase body by factor of 1000
  everything else the same**

"Frame-of-reference"
Simon's ant on the beach

"Frame-of-reference"
F-O-R

- perspectives issue

- behavior vs. mechanism issue

- complexity issue

Intelligence:
Hard to agree on definitions, arguments

- **necessary and sufficient conditions?**

- **are robots, ants, humans intelligent?**

more productive question:

"Given a behavior of interest, how to implement it?"
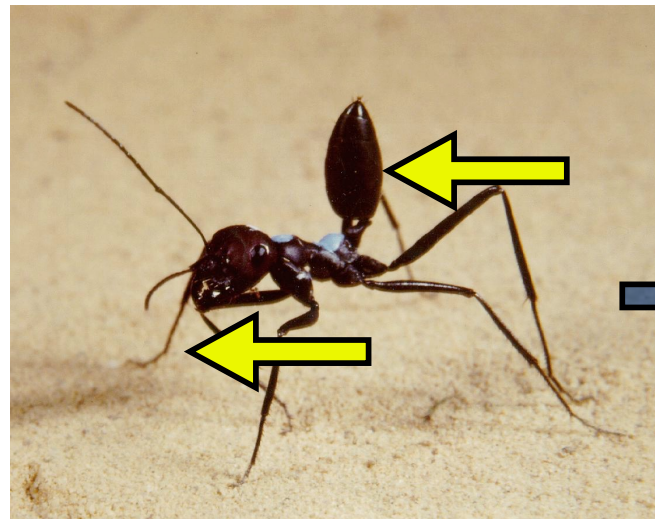

does it come about?"

Communication through interaction with environment

exploitation of interaction with environment

angle sensors in joints

→

simpler neural circuits

Emergence of behavior: the quadruped "Puppy"

- **simple control (oscillations of "hip" joints)**

- **spring-like material properties ("under-actuated" system)**

- **self-stabilization, no sensors**

- **"outsourcing" of functionality**

actuated: oscillation

springs

passive

morphological computation
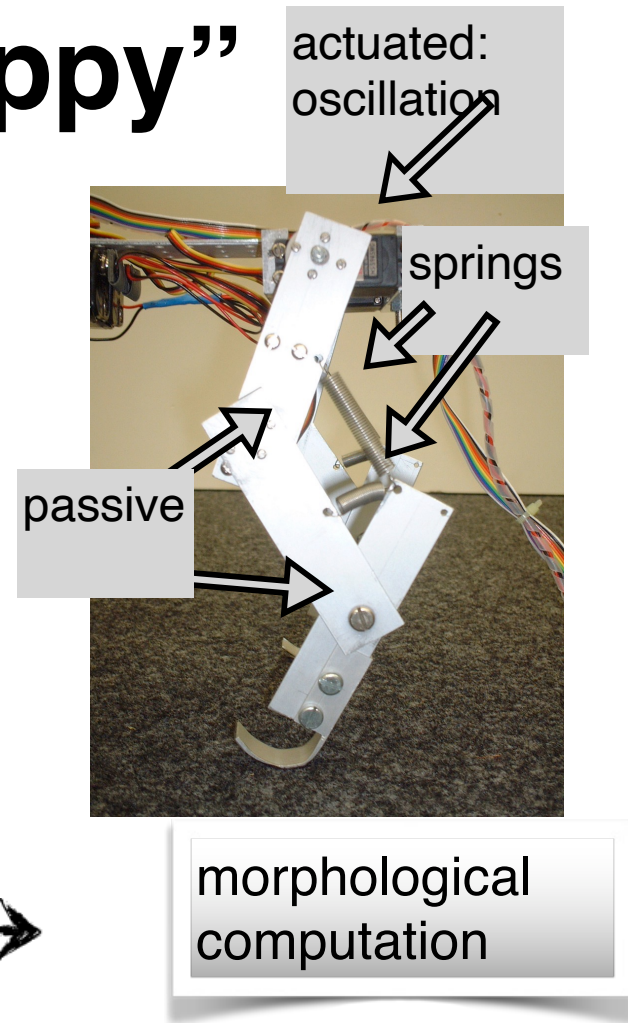
Implications of embodiment
Self-stabilization
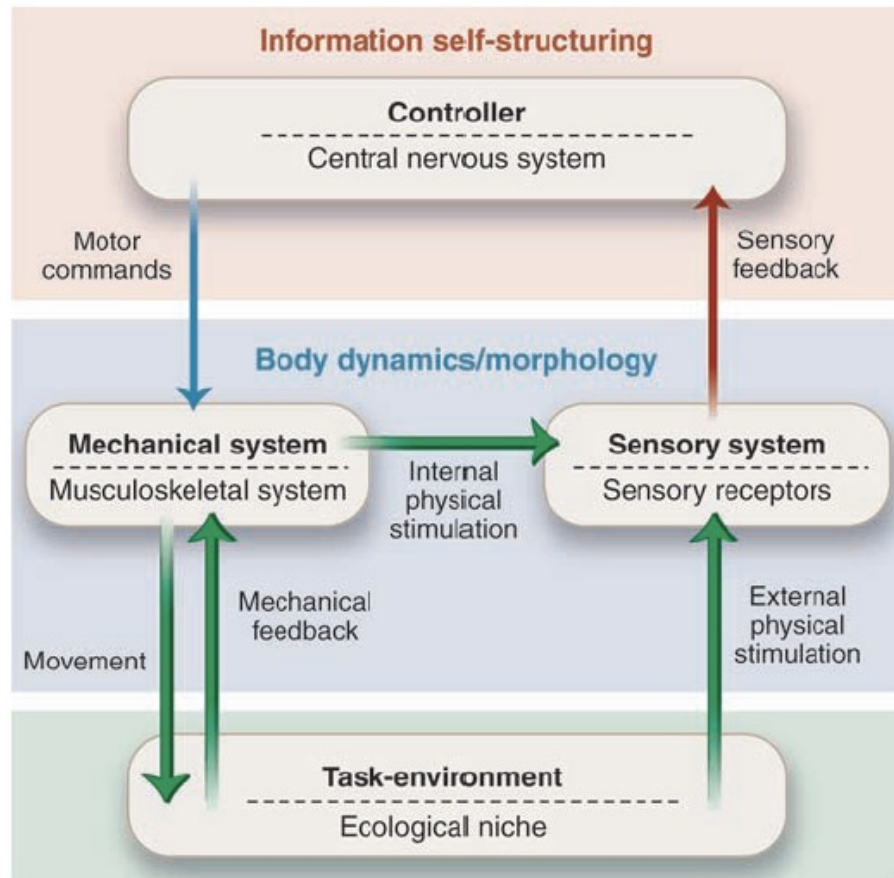
"Puppy", but…
   also Cruse's Ants!!!

Pfeifer et al.,Science,
16 Nov. 2007

Today's Guest Lecture

10:10 CET Enrica Zereik, CNR-INM and Heron@CNR Joint Lab, Genoa, Italy

«Controlling 'uncontrollable' stuff: Playing with the H2Arm»

Stay tuned!

End of lecture 2

Thank you for your attention!

stay tuned for lecture 3

"Emerging Intelligence: Embodiment, Cognition from Interaction, Development and Evolution

<u>Short Bio</u>

The ShanghAI Lectures 2013-

Prof. Fabio Bonsignorio is **ERA Chair in AI for Robotics** at FER, University of Zagreb, Croatia. He is **Founder and CEO of Heron Robots (advanced robotics solutions), see www.heronrobots.com**. He has been visiting professor at the **Biorobotic Institute of the Scuola Superiore Sant'Anna in Pisa from 2014 to 2019**. He has been a professor in the Department of System Enginering and Automation at the **University Carlos III of Madrid until 2014**. In 2009 he got the **Santander Chair of Excellence in Robotics** at the same university. alla stessa università. He has been working for some 20 years in the high tech industry before joining the research community.

He is a **pioneer and has introduced the topic of Reproducibility of results in Robotics and AI**. He is a **pioneer in the application of the blockchain to robotics and IA (smart cities, smart land, smart logistics, circular economy. He coordinates Topic Group of euRobotics about Experiment Replication, Benchmarking, Challenges and Competitions. He is co-chair IEEE Robotics & Automation Society (RAS) Technical Commitee, TC-PEBRAS (PErformance and Benchmarking of Robotics and Autonomous Systems)**.

He is **Distinguished Lecturer per la IEEE Robotics and Automation Society.**' Senior Member of IEEE e member of the Order of the Engineers of Genoa, Italy.

He coordinates the task force robotics, in the G2net,an EU network studying the application of **Machine Learning and Deep Learning (Apprendimento Profondo) to Gravitational wave research, la Geophysics and Robotics**.

Has given invited seminars and talks in many places: MIT Media Lab, Max Planck Institute, Imperial College, Politecnico di Milano in Shenzhen, London, Madrid, Warsaw, San Petersbourg, Seoul, Rio Grande do Sul.…

Thank you!

fabio.bonsignorio@fer.hr
fabio.bonsignorio@heronrobots.com
fabio.bonsignorio@gmail.com

University of Zagreb
Faculty of Electrical Engineering and Computing
Laboratory for Autonomous Systems and Mobile Robotics

www.heronrobots.com