The ShanghAI Lectures

上海授课

人工智能

The Shanghai Lectures 2023

Natural and Artificial Intelligence in Embodied Physical Agents

November 2nd, 2023

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 2023</u>
10:00 Break
10:15 Krzysztof Pomorski, Technical University of Lodz, Institute of Physics, Poland:
<u>Thermodynamics in Stochastic Conway’s Game of Life</u>
11:00 Wrap-up

Today’s Guest Lecture

10:15 Krzysztof Pomorski, Technical University of Lodz, Institute of Physics, Poland

«Thermodynamics in Stochastic Conway’s Game of Life »

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

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No. 952275

www.heronrobots.com

Today’s topics

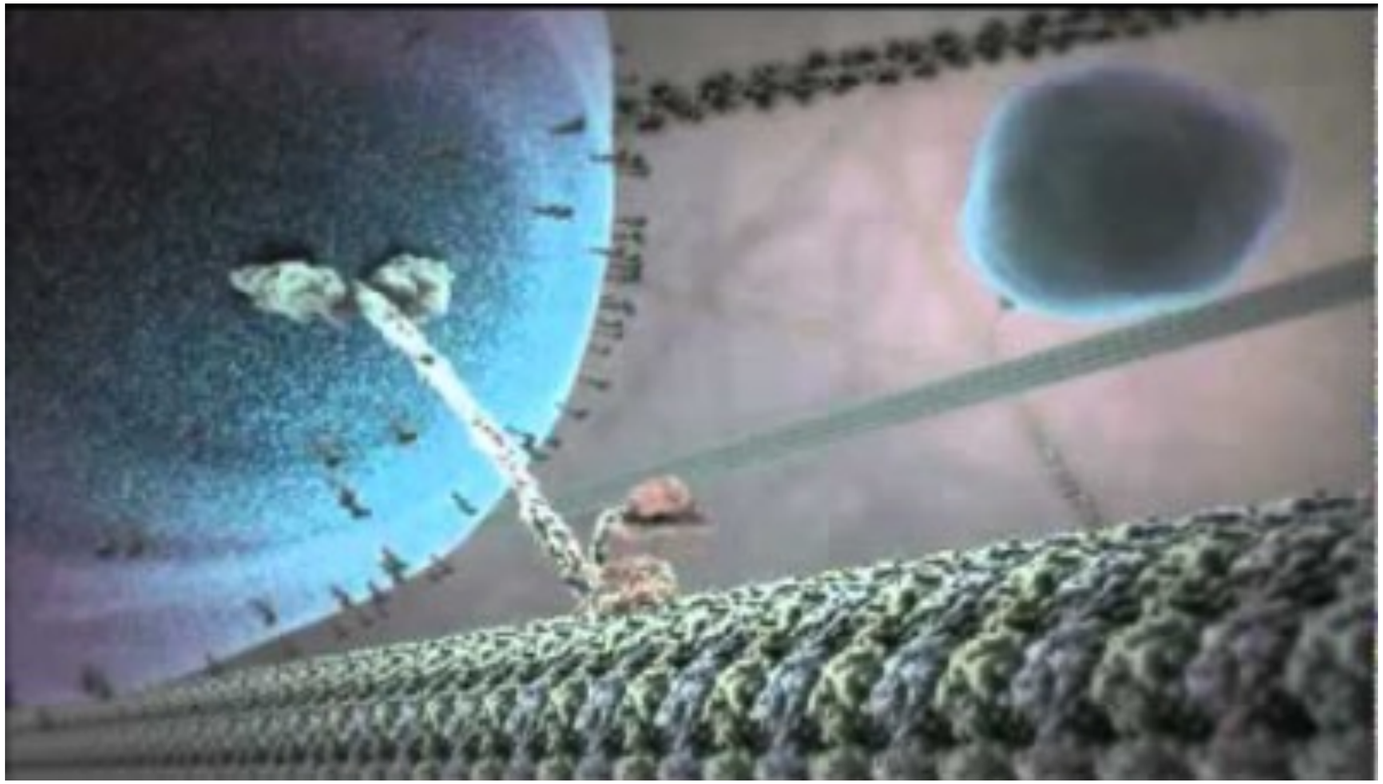
- Practical Robotics in 2022
- The classical approach: Cognition as computation
- Successes and failures of the classical approach
- Some problems of the classical approach
- The need for an embodied approach

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

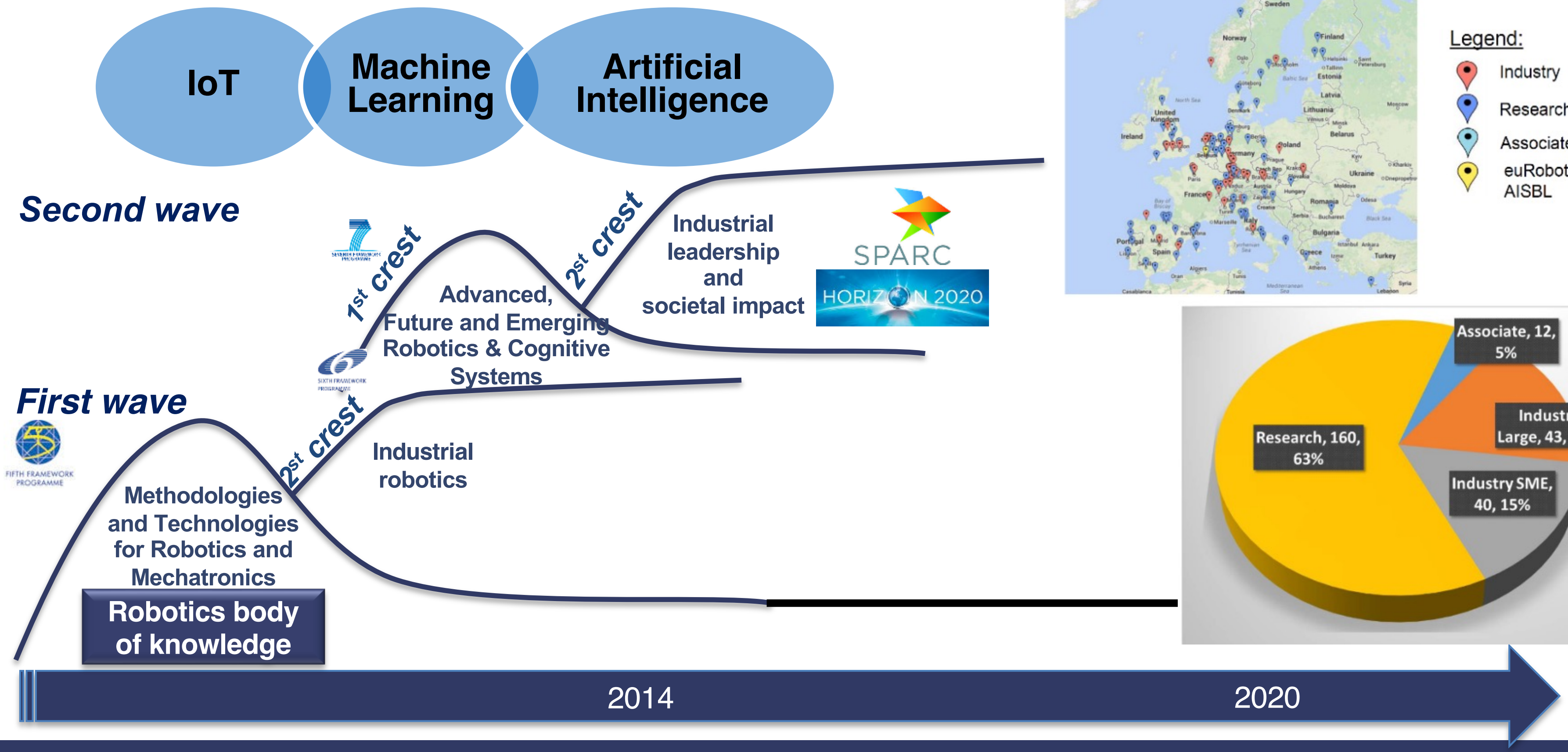
$\longrightarrow$

Many examples during ShanghAI lectures

robots as tools for scientific investigation

The second wave

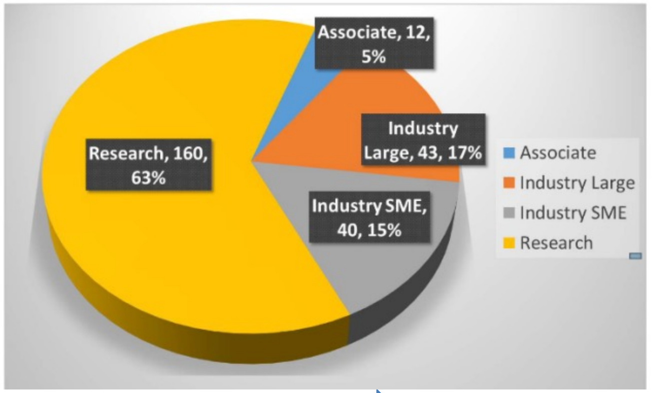
we are here!


IoT
Machine Learning
Artificial Intelligence
Second wave
First wave
FIFTH FRAMEWORK PROGRAMME
Methodologies and Technologies for Robotics and Mechatronics
Robotics body of knowledge
2st crest
Industrial robotics
1st crest
Advanced, Future and Emerging Robotics & Cognitive Systems
2st crest
Industrial leadership and societal impact
SPARC
HORIZON 2020
2014
2020

Membership development

280 member organisations

Legend:
Industry
Research
Associate
euRobotics AISBL



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)

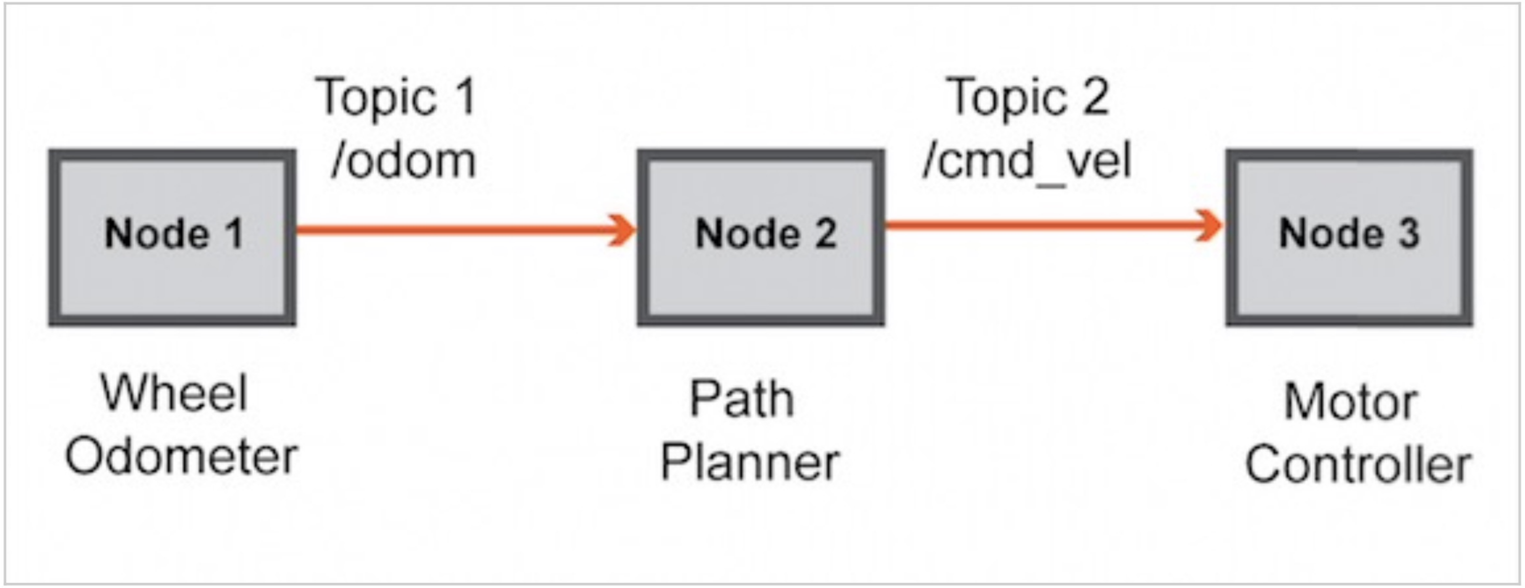
From Opensource Robotics Foundation (formerly from Willow Garage)
(openrobotics.org)

- 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.

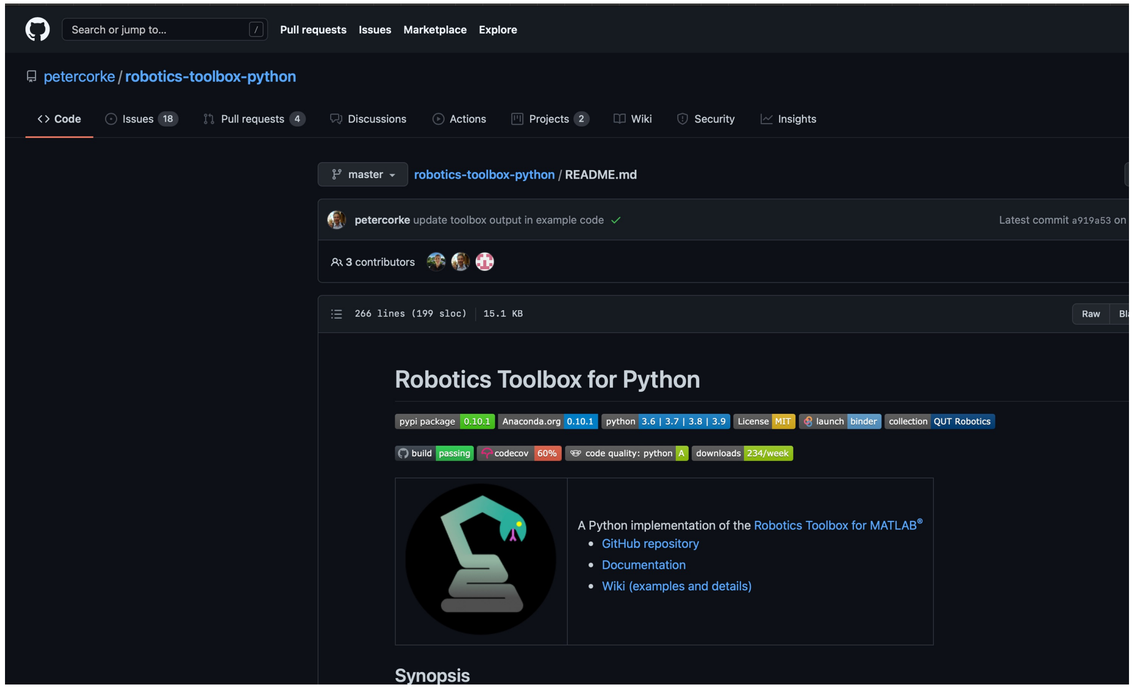
- ROS allows massive code reuse in robotics research and development and is perfect for rapid prototyping of new systems and applications, in particular where efficiency is not a main constraint (although being rather cumbersome and inefficient and having a steep learning curve, you will see ☺).
- You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


Search or jump to... Pull requests Issues Marketplace Explore

petercorke / robotics-toolbox-python

Code Issues 18 Pull requests 4 Discussions Actions Projects 2 Wiki Security Insights

master robotics-toolbox-python / README.md

petercorke update toolbox output in example code ✓ Latest commit a919a53 on

3 contributors

266 lines (199 sloc) 15.1 KB Raw

Robotics Toolbox for Python

pypi package 0.10.1 | Anaconda.org 0.10.1 | python 3.6 | 3.7 | 3.8 | 3.9 | License MIT | launch binder | collection QUT Robotics

build passing | codecov 60% | code quality: python A | downloads 234/week

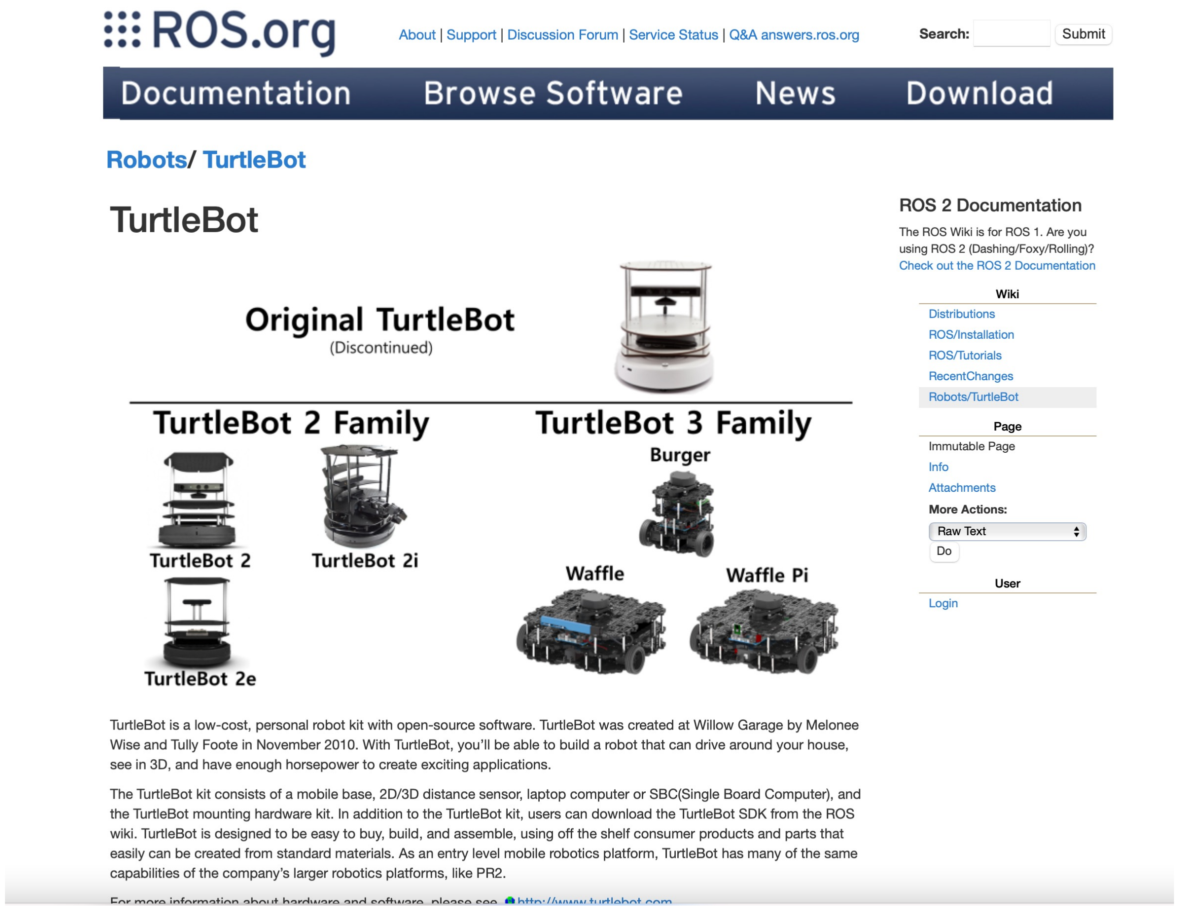
A Python implementation of the Robotics Toolbox for MATLAB®
- GitHub repository
- Documentation
- Wiki (examples and details)

Synopsis

ROS

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

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



TurtleBot

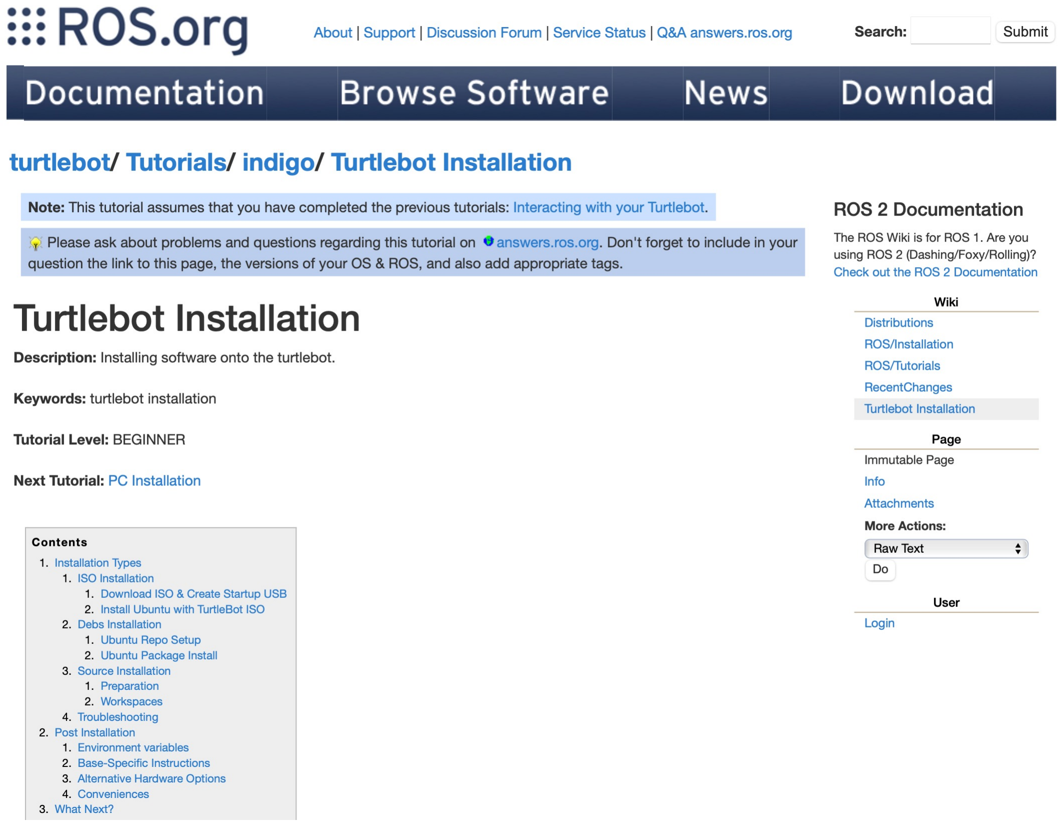
TurtleBot is a low-cost, personal robot kit with open-source software. TurtleBot was created at Willow Garage by Melonee Wise and Tully Foote in November 2010. With TurtleBot, you'll be able to build a robot that can drive around your house, see in 3D, and have enough horsepower to create exciting applications.

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


ROS.org
About | Support | Discussion Forum | Service Status | Q&A answers.ros.org
Search: Submit
Documentation
Browse Software
News
Download

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

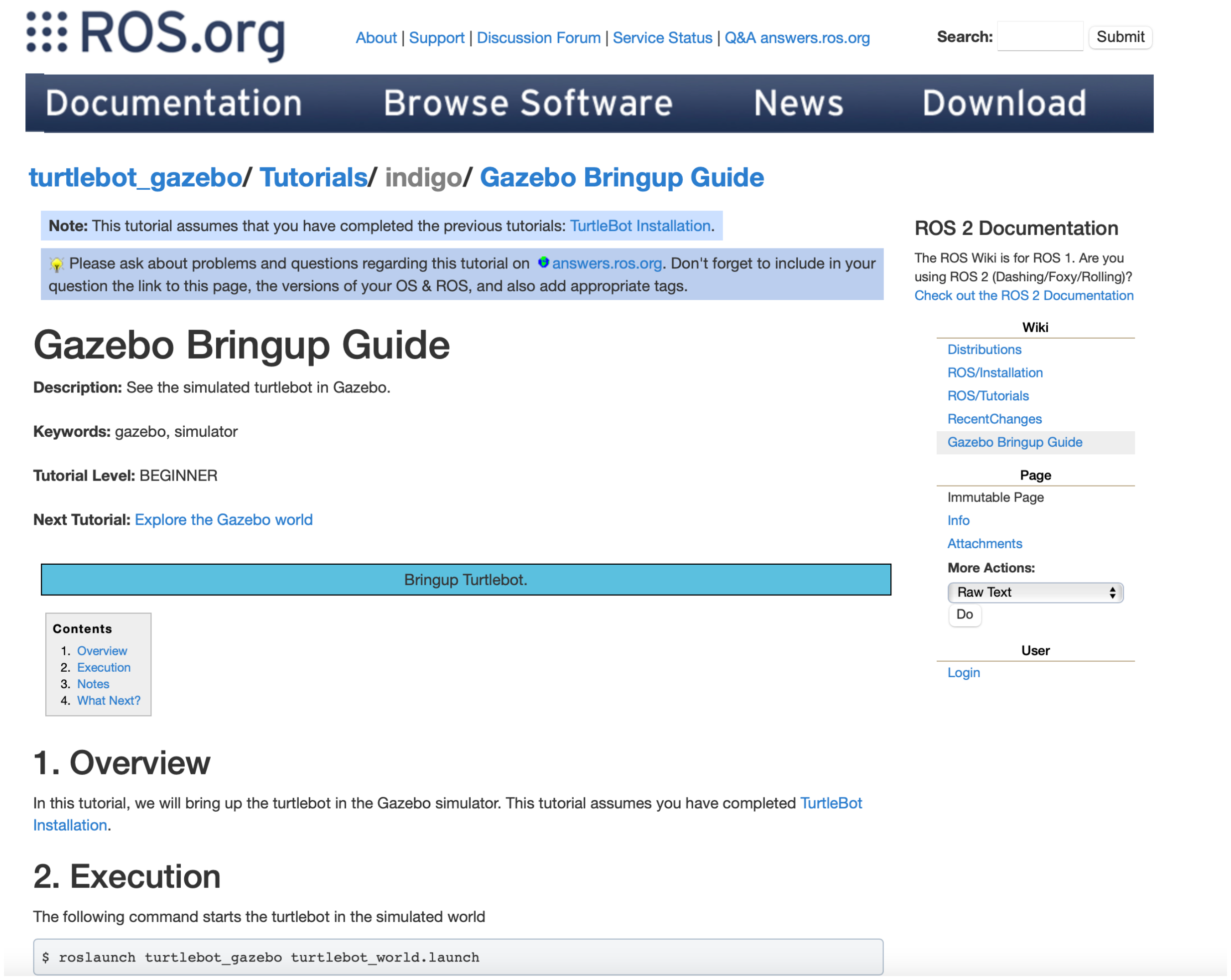
Tutorial Level: BEGINNER

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


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

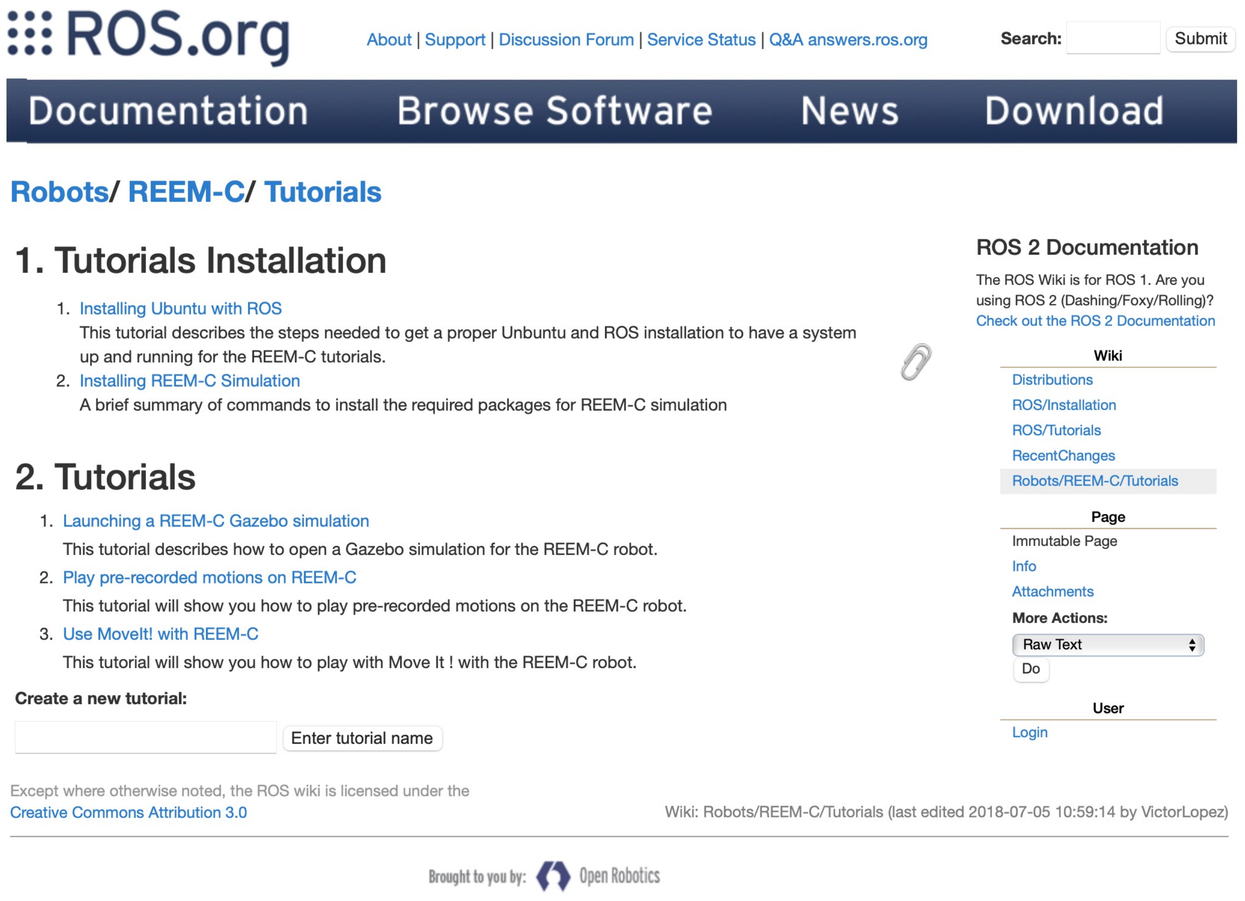
The following command starts the turtlebot in the simulated world

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

ROS 2 Documentation

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

even more complex…



ROS.org

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

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

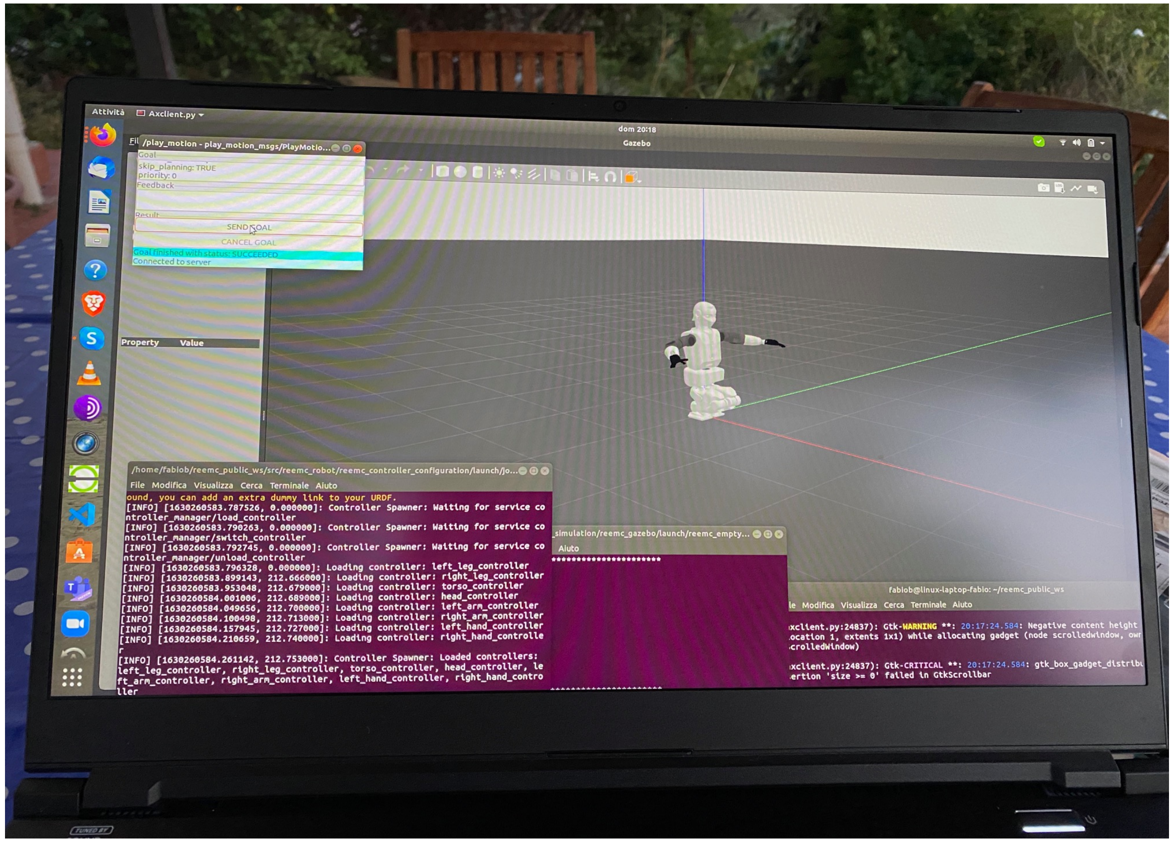
Except where otherwise noted, the ROS wiki is licensed under the Creative Commons Attribution 3.0

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


Attività
Axclient.py
dom 20:18
Gazebo
/play_motion - play_motion_msgs/PlayMotio...
Goal
Feedback
Result
SEND GOAL
CANCEL GOAL
Property
Value
File Modifica Visualizza Cerca Terminale Aiuto

skip_planning: TRUE
priority: 0

Goal finished with status: SUCCEEDED
Connected to server

/home/fabiob/reemc_public_ws/src/reemc_robot/reemc_controller_configuration/launch/jo...

```
ound, you can add an extra dummy link to your URDF.
[INFO] [1630260583.787526, 0.000000]: Controller Spawner: Waiting for service co
ntroller_manager/load_controller
[INFO] [1630260583.790263, 0.000000]: Controller Spawner: Waiting for service co
ntroller_manager/switch_controller
[INFO] [1630260583.792745, 0.000000]: Controller Spawner: Waiting for service co
ntroller_manager/unload_controller
[INFO] [1630260583.796328, 0.000000]: Loading controller: left_leg_controller
[INFO] [1630260583.899143, 212.666000]: Loading controller: right_leg_controller
[INFO] [1630260583.953048, 212.679000]: Loading controller: torso_controller
[INFO] [1630260584.001006, 212.689000]: Loading controller: head_controller
[INFO] [1630260584.049656, 212.700000]: Loading controller: left_arm_controller
[INFO] [1630260584.100498, 212.713000]: Loading controller: right_arm_controller
[INFO] [1630260584.157945, 212.727000]: Loading controller: left_hand_controller
[INFO] [1630260584.210659, 212.740000]: Loading controller: right_hand_controlle
r
[INFO] [1630260584.261142, 212.753000]: Controller Spawner: Loaded controllers:
left_leg_controller, right_leg_controller, torso_controller, head_controller, le
ft_arm_controller, right_arm_controller, left_hand_controller, right_hand_contro
ller
```

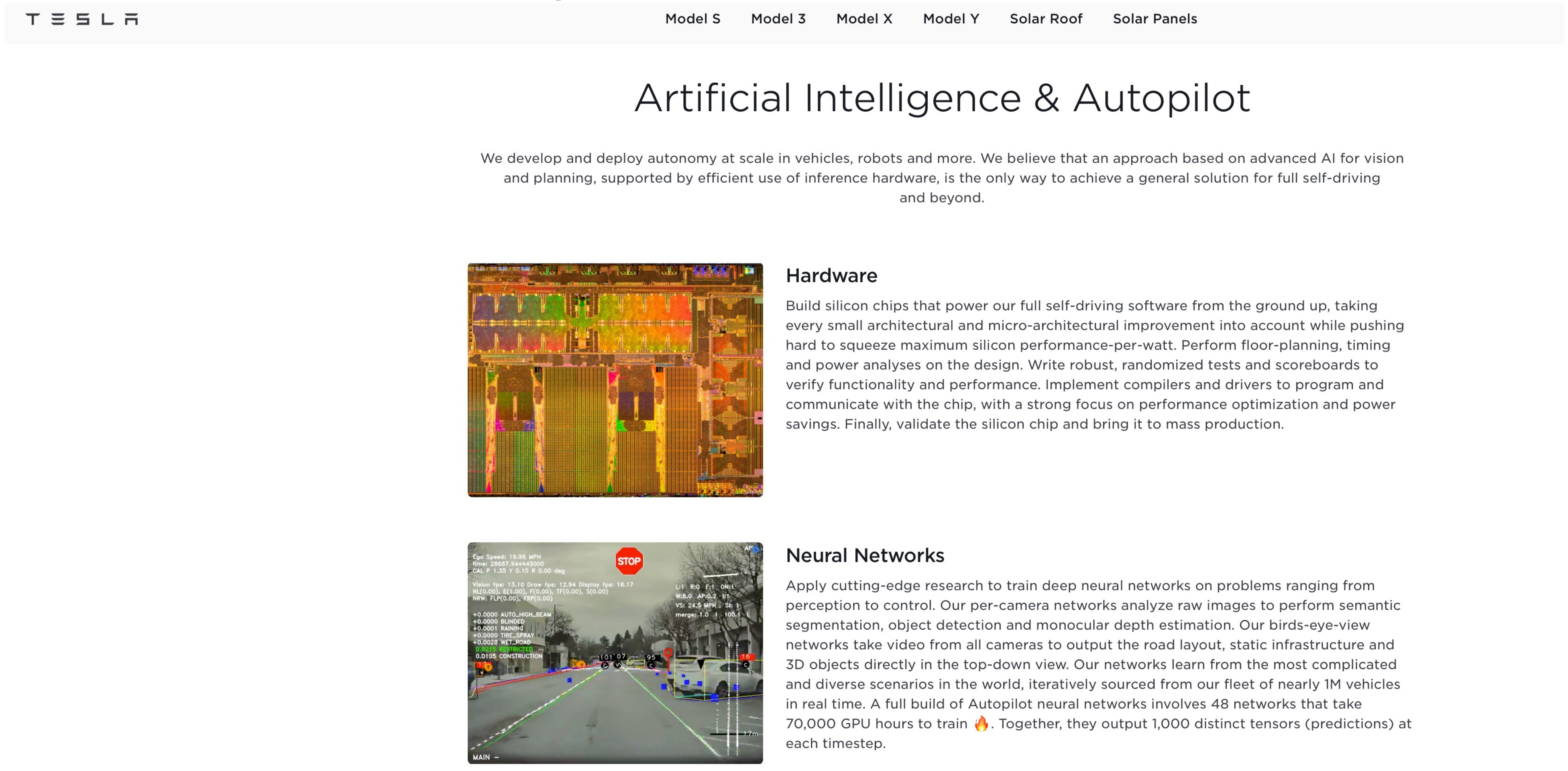
_simulation/reemc_gazebo/launch/reemc_empty...

```
************************
```

fabiob@linux-laptop-fabio: ~/reemc_public_ws

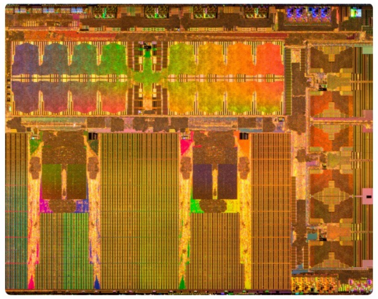
```
xclient.py:24837): Gtk-WARNING **: 20:17:24.584: Negative content height
ocation 1, extents 1x1) while allocating gadget (node scrolledwindow, owr
crolledWindow)

xclient.py:24837): Gtk-CRITICAL **: 20:17:24.584: gtk_box_gadget_distribu
ertion 'size >= 0' failed in GtkScrollbar
```

The most? complex…



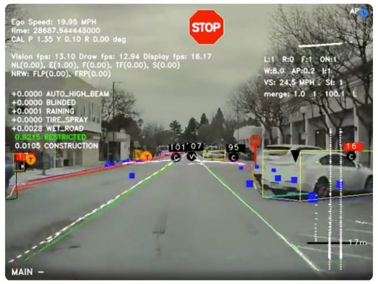
Artificial Intelligence & Autopilot

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.



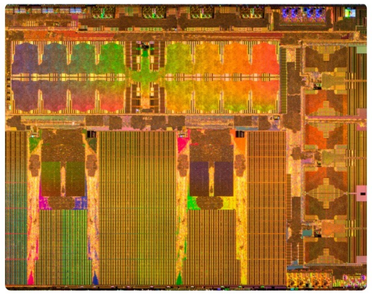
Hardware

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.



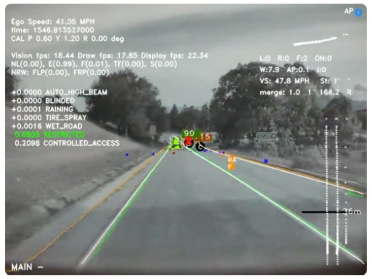
Neural Networks

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.

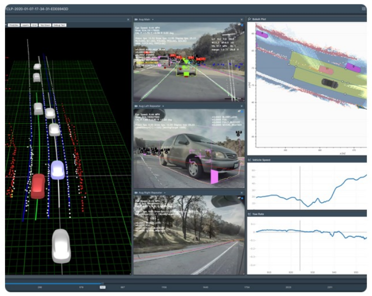


```
Ego Speed: 41.05 MPH
time: 1546.813527000
CAL P 0.60 Y 1.20 R 0.00 deg

Vision fps: 18.44 Draw fps: 17.85 Display fps: 22.34
NL(0.00), E(0.99), F(0.01), TF(0.00), S(0.00)
NRW: FLP(0.00), FRP(0.00)

+0.0000 AUTO_HIGH_BEAM
+0.0000 BLINDED
+0.0001 RAINING
+0.0000 TIRE_SPRAY
+0.0016 WET_ROAD
 0.6805 RESTRICTED
 0.2098 CONTROLLED_ACCESS
```

```
L:0  R:0  F:2  ON:0
W:7.9  AP:0.1  I:0
VS: 47.8 MPH   St: 1
merge: 1.0  1  168.2  R
```

Neural Networks

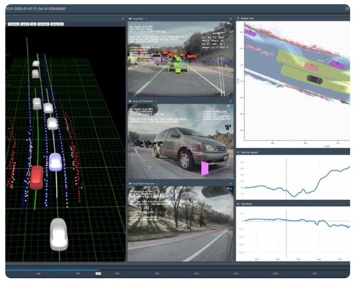
Apply cutting-edge research to train deep neural networks on problems ranging from perception to control. Our per-camera networks analyze raw images to perform semantic segmentation, object detection and monocular depth estimation. Our birds-eye-view networks take video from all cameras to output the 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

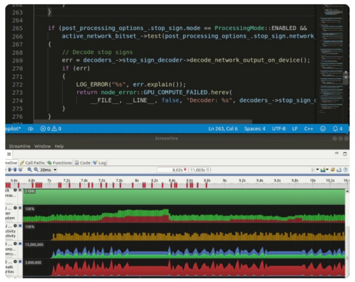
Develop the core algorithms that drive the car by creating a high-fidelity representation of the world and planning trajectories in that space. In order to train the neural networks to predict such representations, algorithmically create accurate and large-scale ground truth data by combining information 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.



```
    }

    if (post_processing_options_.stop_sign.mode == ProcessingMode::ENABLED &&
        active_network_bitset_->test(post_processing_options_.stop_sign.network_
    {
        // Decode stop signs
        err = decoders_->stop_sign_decoder->decode_network_output_on_device();
        if (err)
        {
            LOG_ERROR("%s", err.explain());
            return node_error::GPU_COMPUTE_FAILED.herev(
                __FILE__, __LINE__, false, "Decoder: %s", decoders_->stop_sign_
        }
    }
```

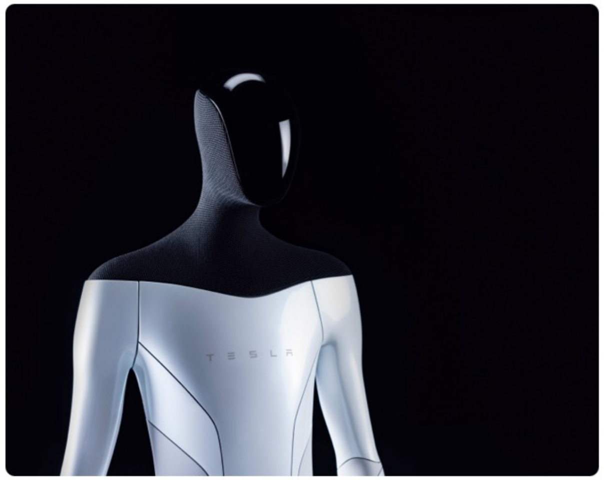
Code Foundations

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.

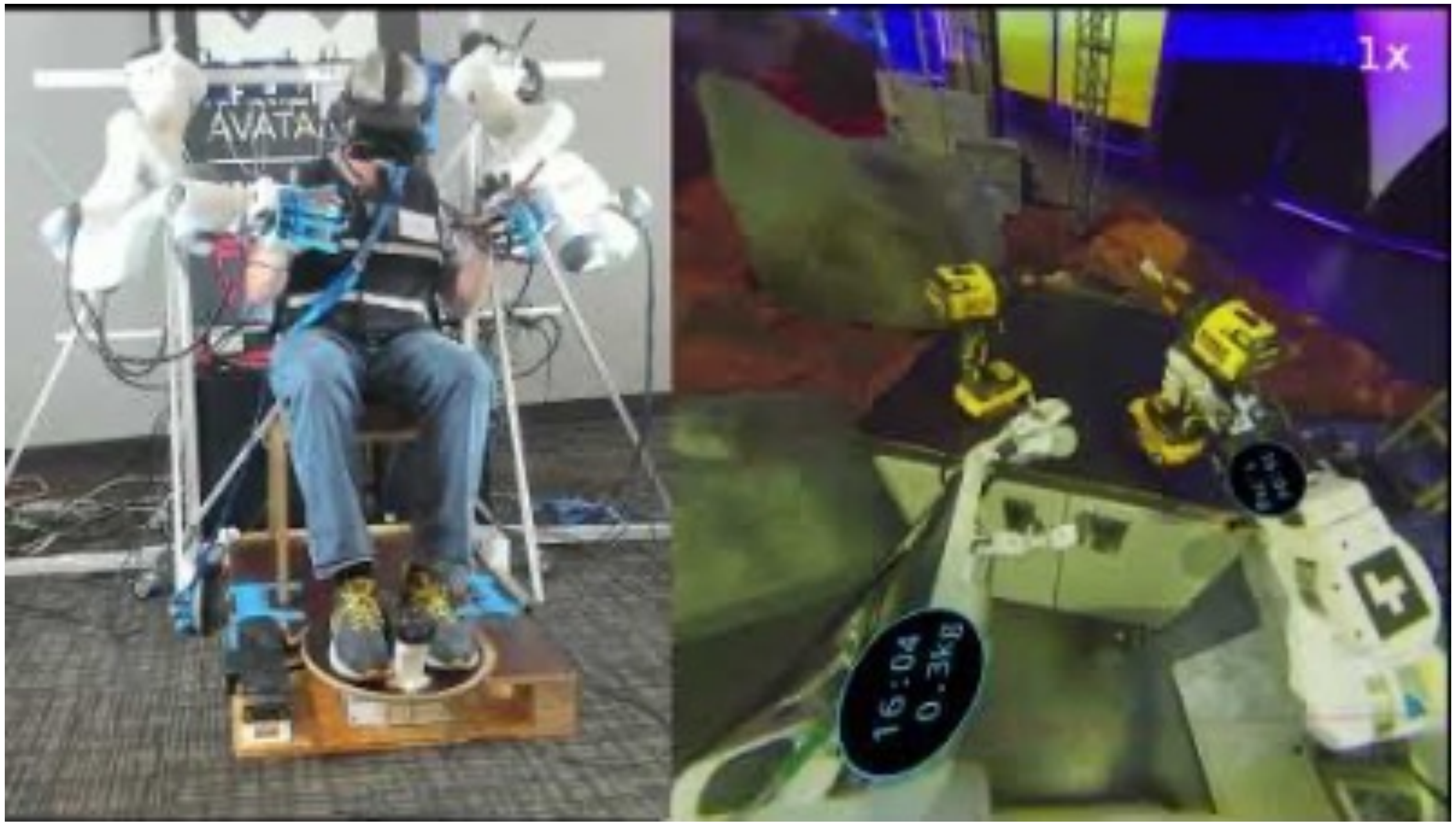
The most? complex…(so far, IFF ☺)



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



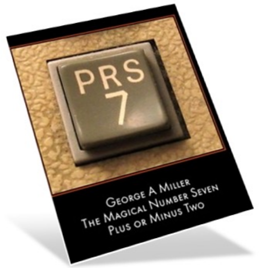
NimbRo Team University of Bonn

Today’s topics

- 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

"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



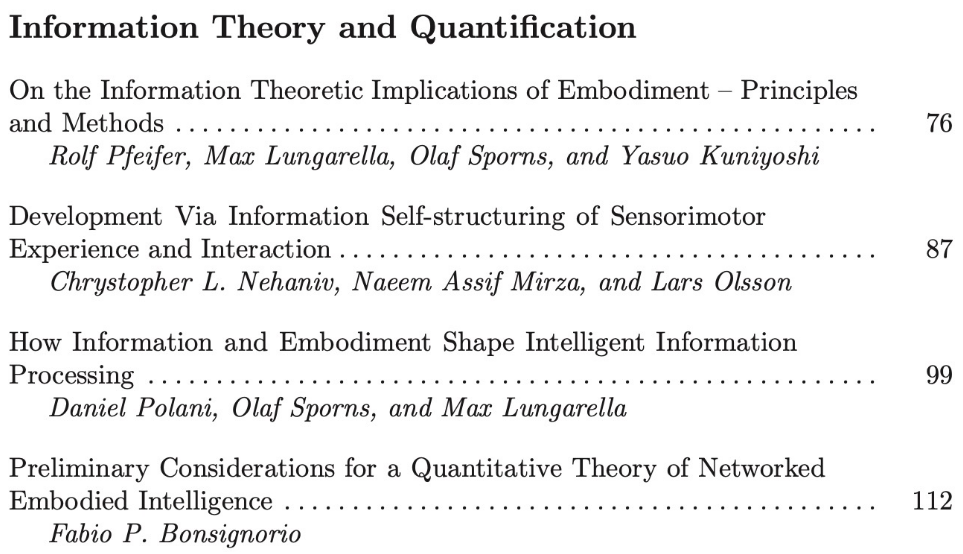
George A. Miller, Psychologist
"The Magical Number Seven Plus or Minus Two"

John McCarthy, Computer Scientist
Initiator of Artificial Intelligence

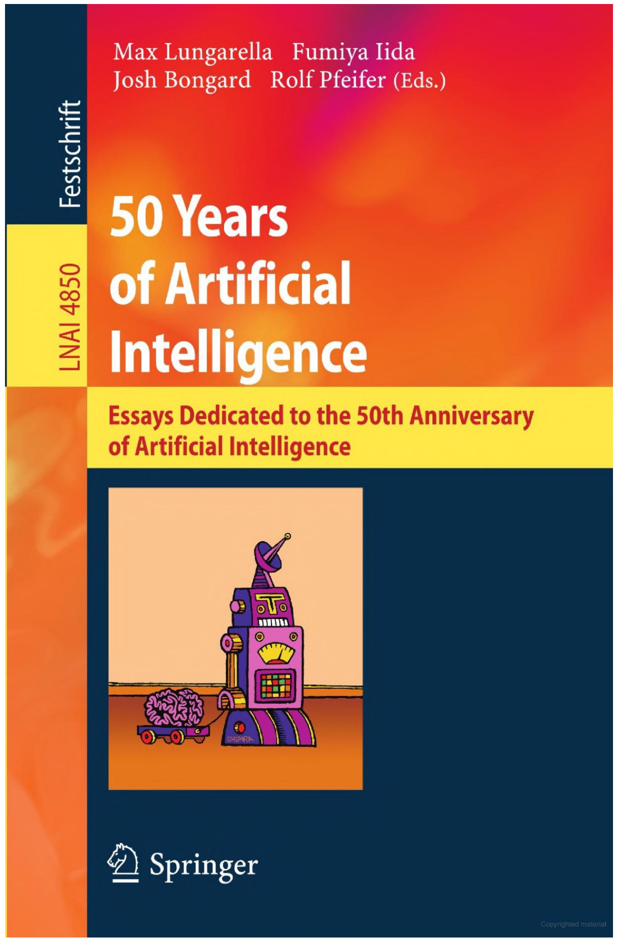
Herbert Simon
and Allen Newell
The "Logic Theorist"

Noam Chomsky,
Linguist
"Syntactic Structures"

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



Information Theory and Quantification



Max Lungarella Fumiya Iida
Josh Bongard Rolf Pfeifer (Eds.)

Festschrift

LNAI 4850

50 Years of Artificial Intelligence

Essays Dedicated to the 50th Anniversary of Artificial Intelligence

Springer

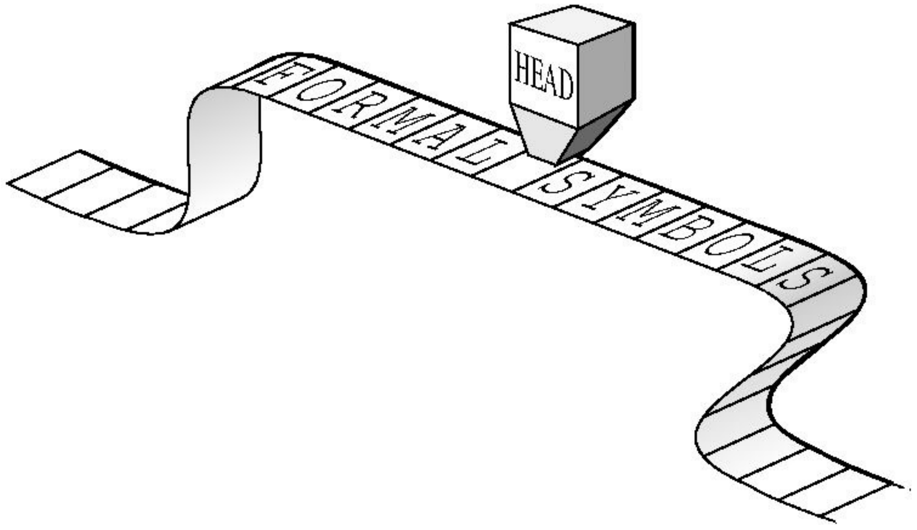
50 Years later, 2006 Meeting on Monte Verità, Ascona, Switzerland

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

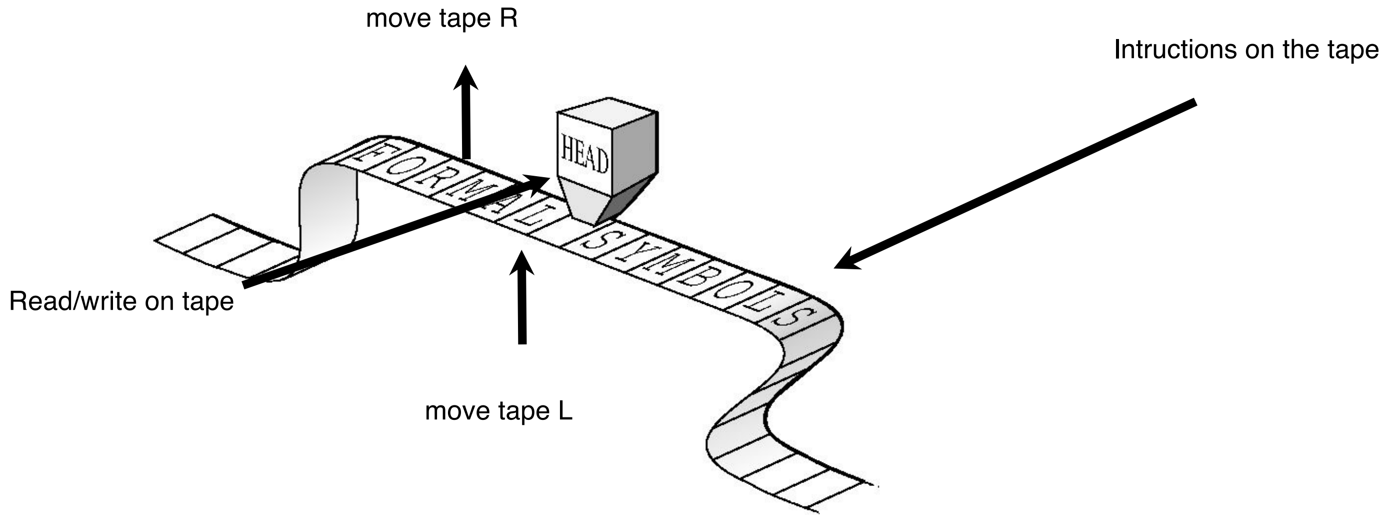
Nel diciannovesimo secolo e nei primi anni del ventesimo secolo il Ticino, repubblica e cantone dal 1803, diventò un passaggio verso sud e destinazione privilegiata di un gruppo di solitari anticonvenzionali, i quali trovarono nella regione, con la sua atmosfera meridionale, terreno fertile in cui piantare 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)

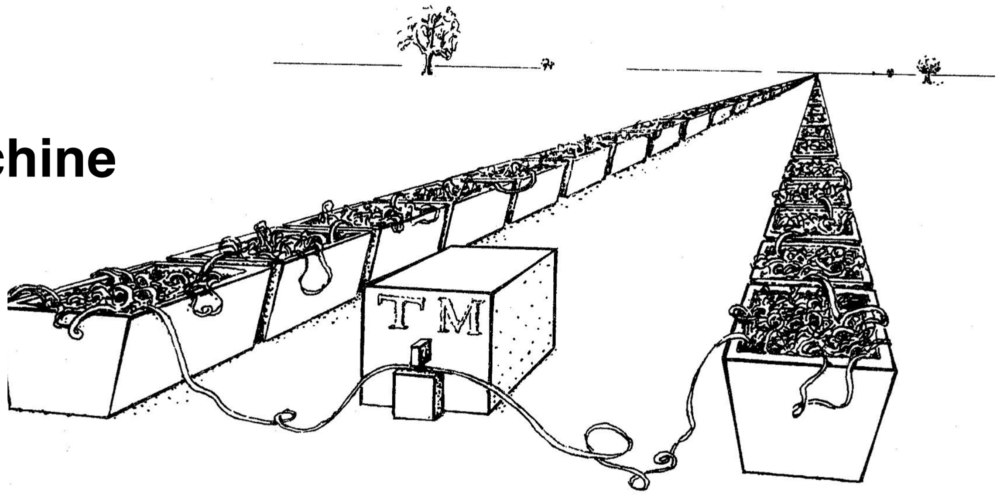


input from tape	1	2	state of read/write head
_	_R2	HALT	
A	AL1	BR2	
B	BL1	AR2	
C	CL1	CR2	

The Universal Turing Machine

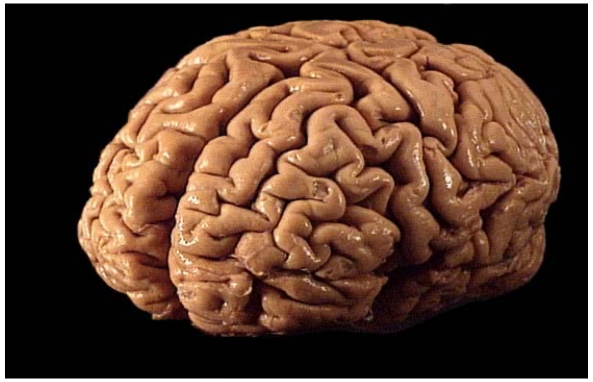
Turing Machine (5)

an “embodied” Turing Machine



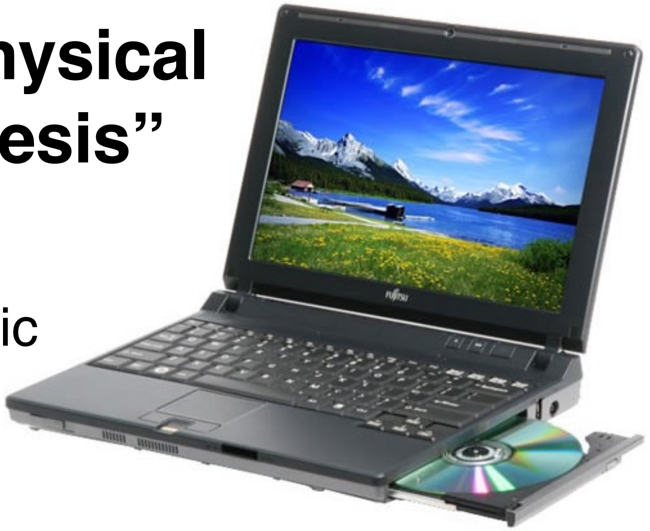
Cartoon by Roger Penrose

Functionalism and the "Physical Symbol Systems Hypothesis"



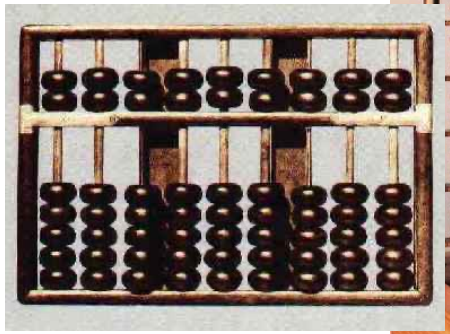
biological

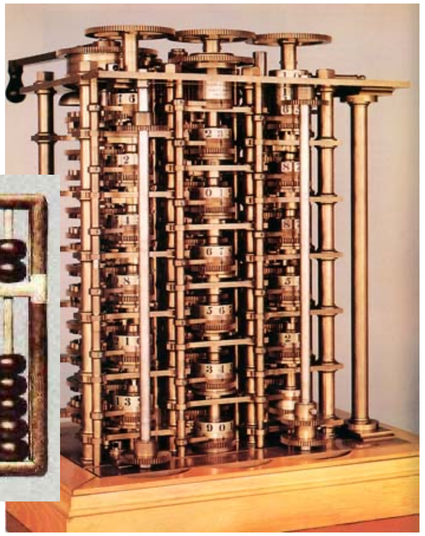
electronic



Swiss cheese
Hilary Putnam
(American
Philosopher)

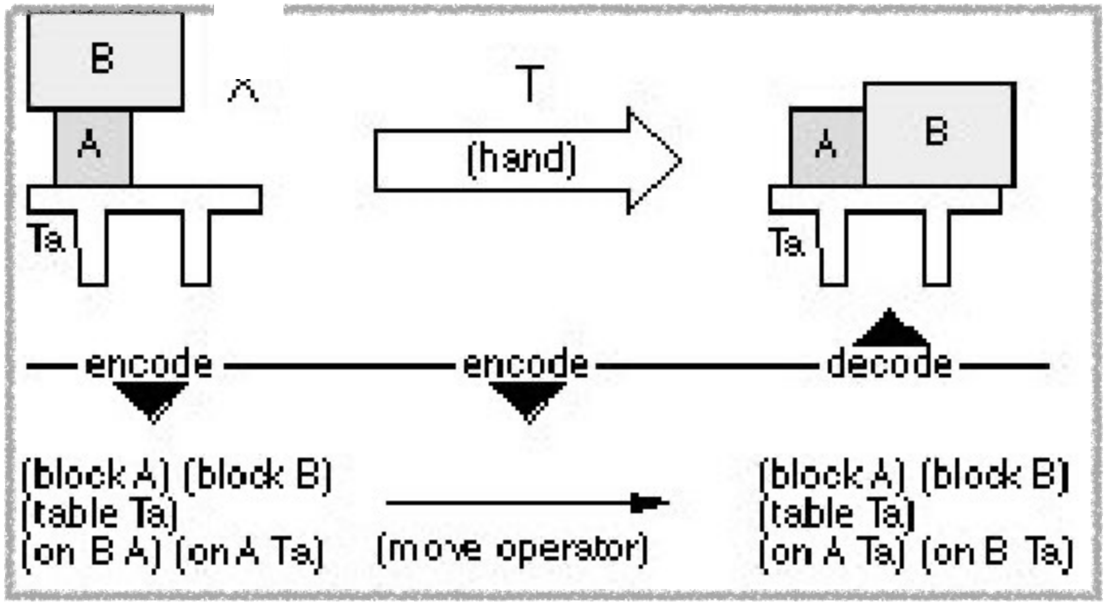
mechanical





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)

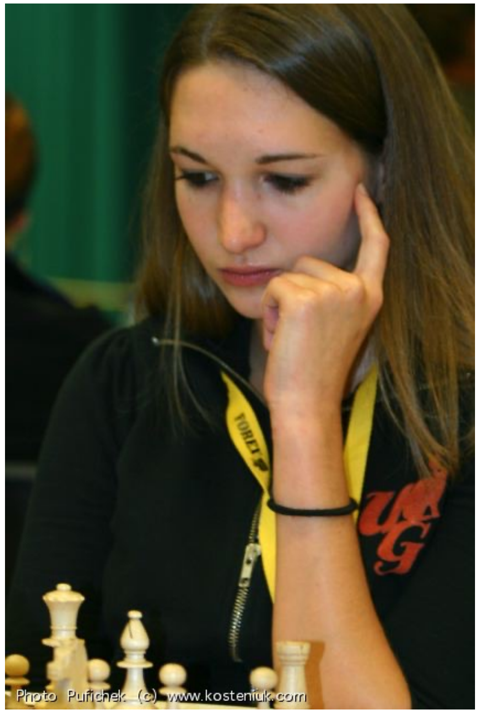
Why is perception hard?

Today’s topics

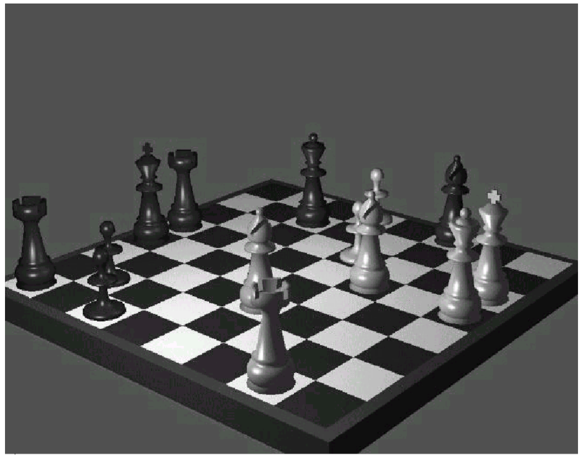
- Practical Robotics in 2022
- The classical approach: Cognition as computation
- Successes and failures of the classical approach
- **Some problems of the classical approach**
- The need for an embodied approach

Fundamental problems of the classical approach

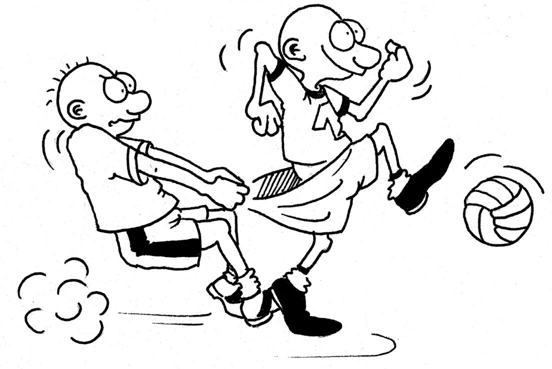
Monika Seps, chess master
former master student
AI Lab, Zurich



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:“control led”artificial worlds)

Failures

foundations of behavior

natural forms of intelligence

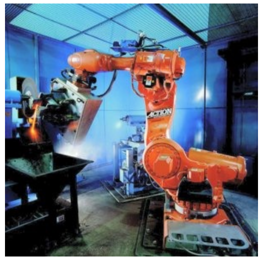
interaction with real world

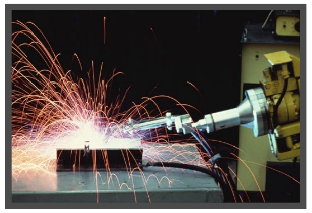
(scientific: “real worlds”)

Industrial environments vs. real world

industrial environments	real world environment
environment well-known	**limited knowledge and predictability**
little uncertainty	**rapidly changing**
predictability	**high-level of uncertainty**
(“controlled”artificial worlds)	(“real” worlds)

Industrial robots vs. natural systems

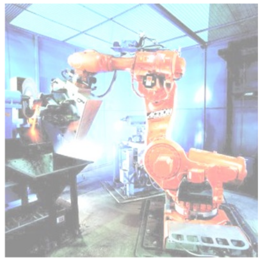
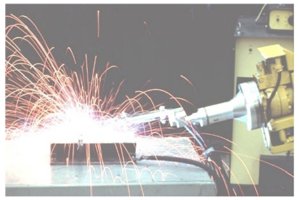




principles:
- strong, precise, fast motors
- centralized control
- computing power
- optimization

Industrial robots

Industrial robots vs. natural systems

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

humans

no direct transfer of methods

→

Short Bio

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 a **Distinguished Lecturer** for **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 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

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No. 952275

www.heronrobots.com