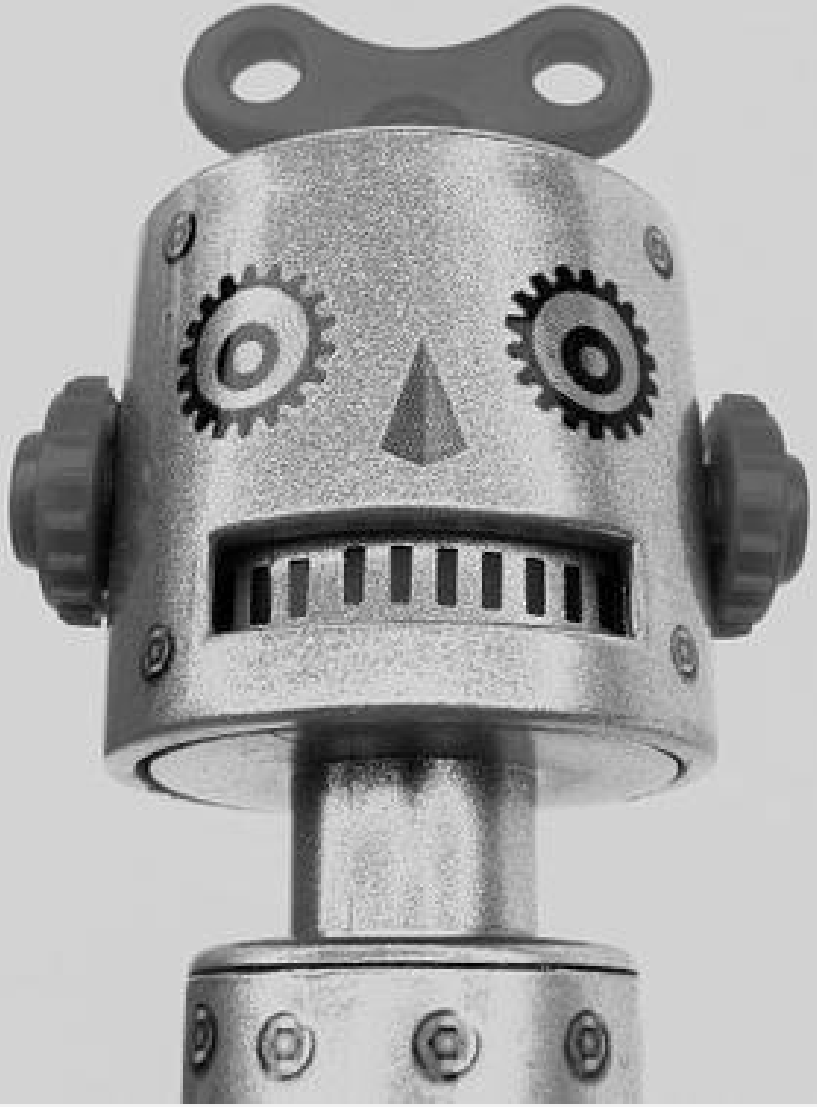
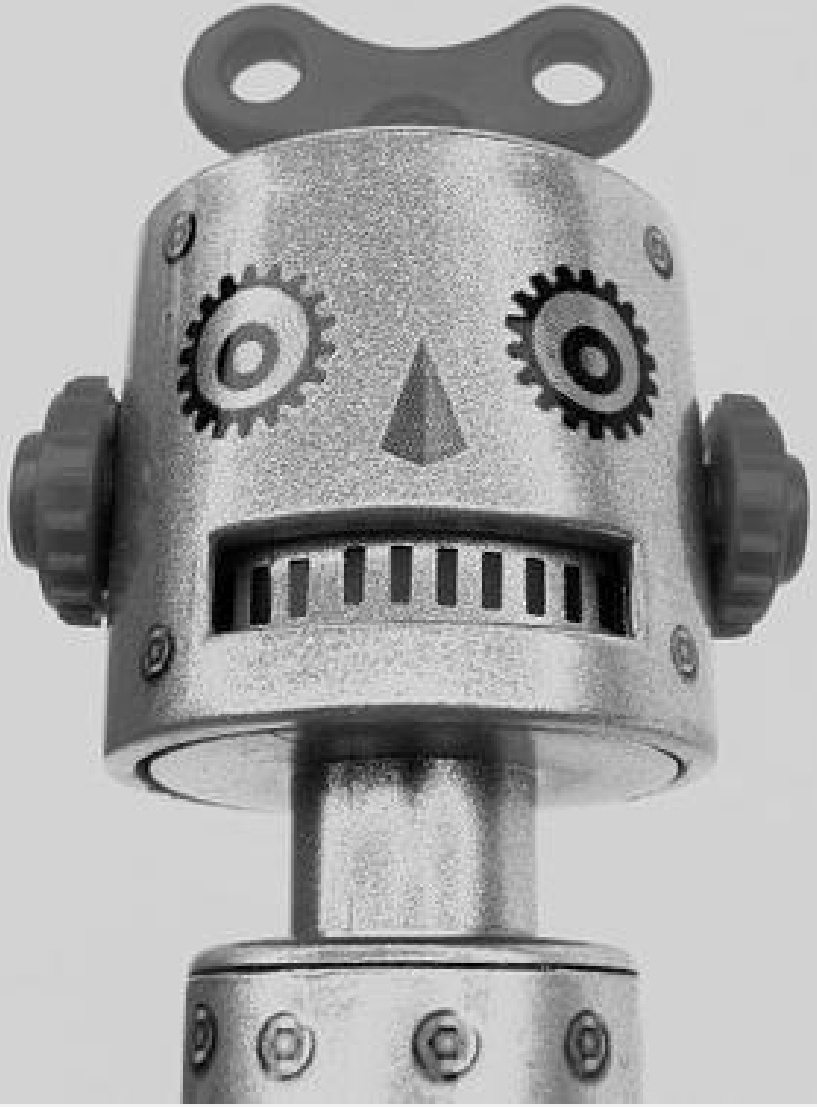
ROBOTIC PROGRAMMING

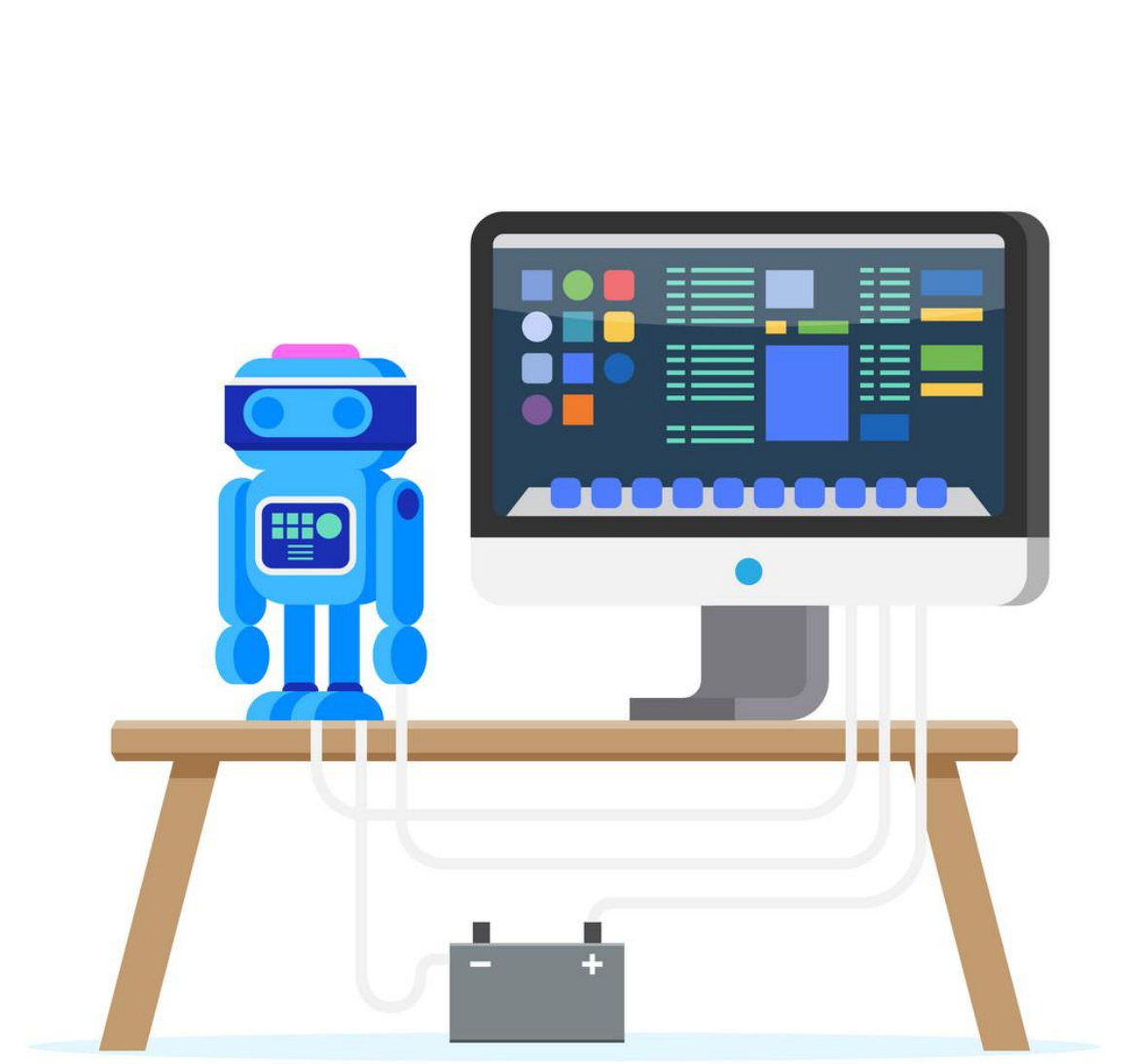


A simple task for humans can be complex for robots.



ROBOTIC PROGRAMMING

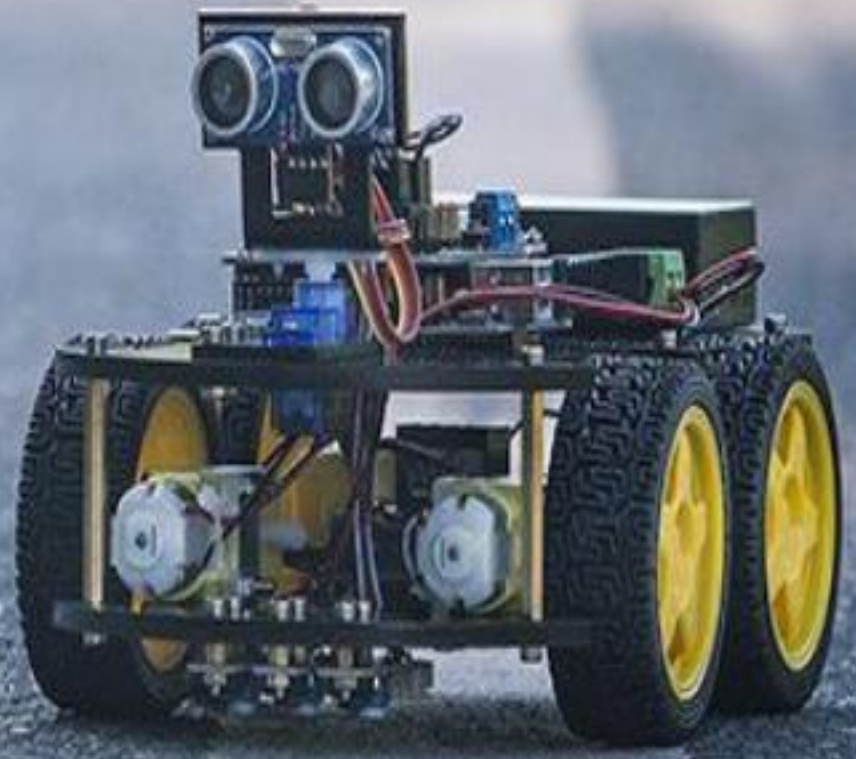
- Sensors
- Actuators
- Computing Unit

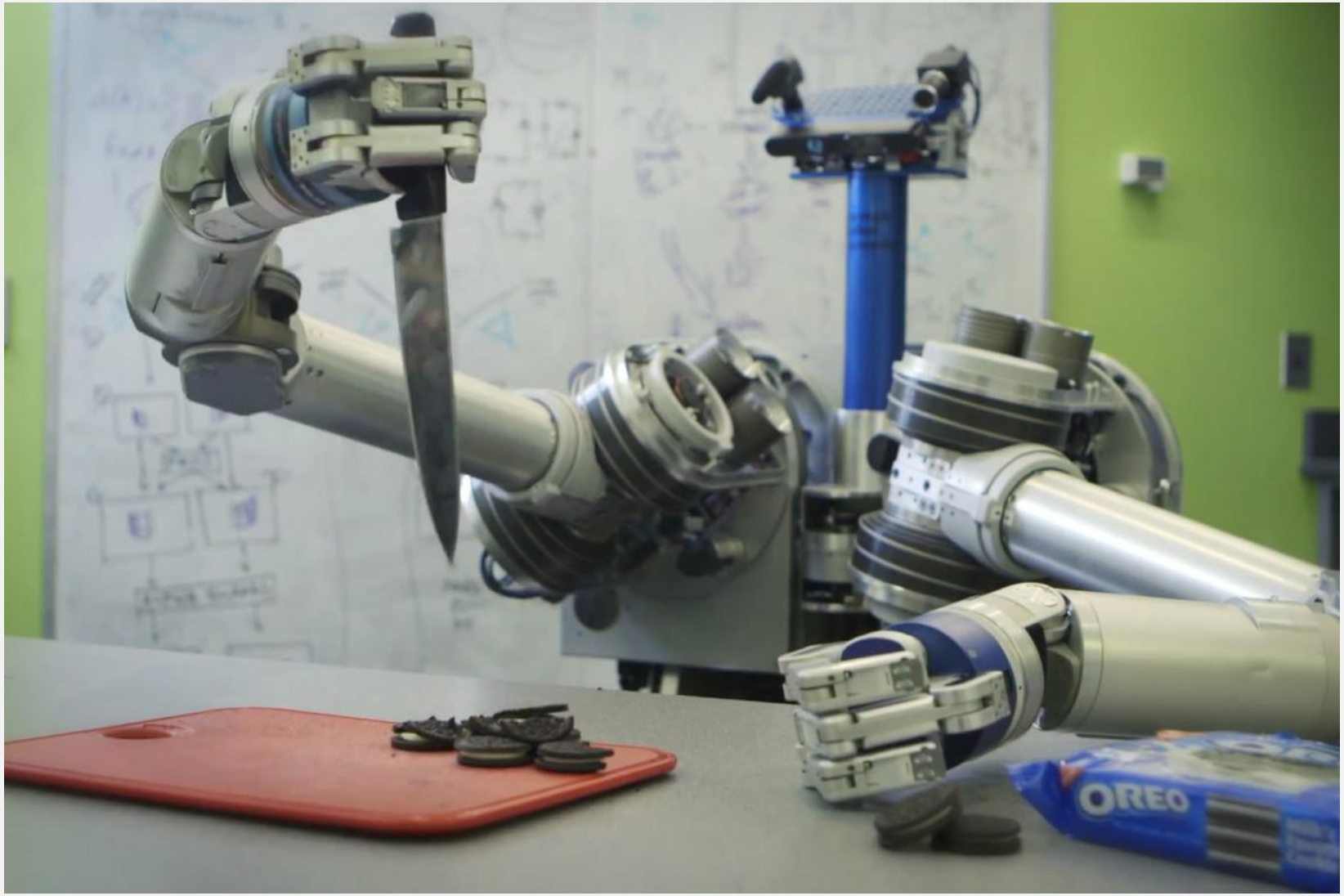


Randomly Moving Bot

Obstacle Avoiding Bot

Simultaneous Localization and Mapping(SLAM)

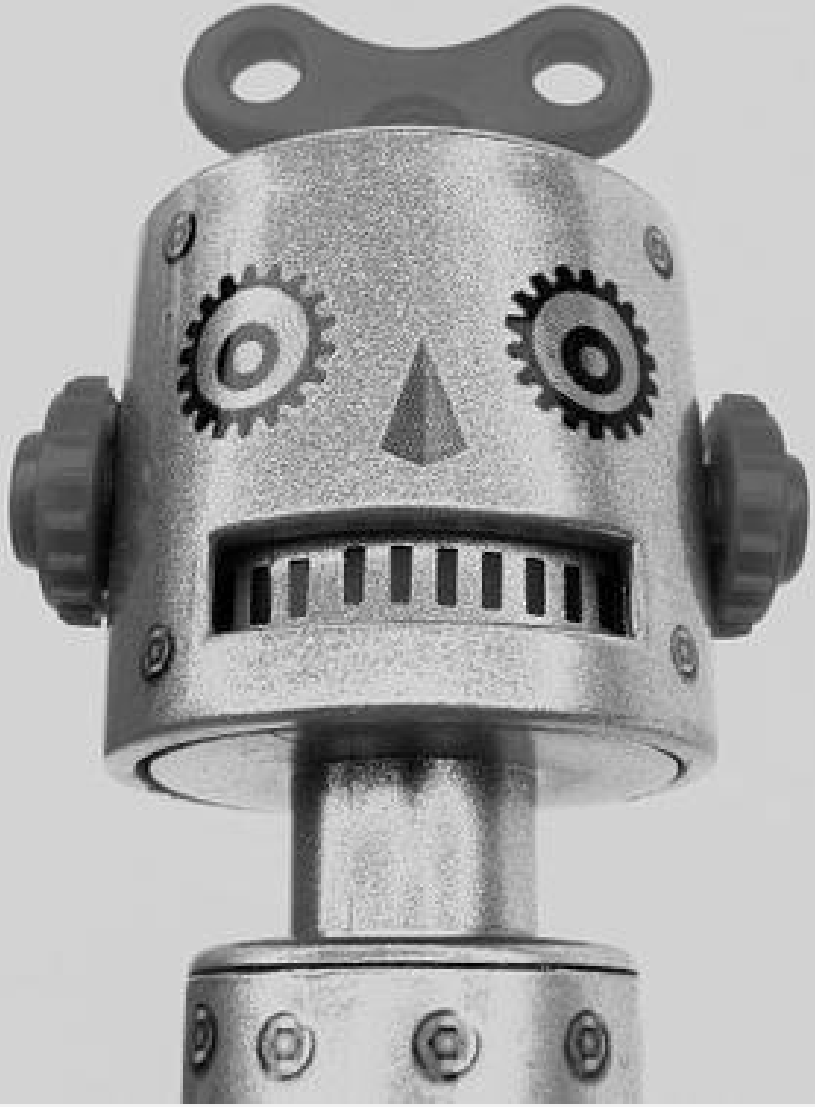




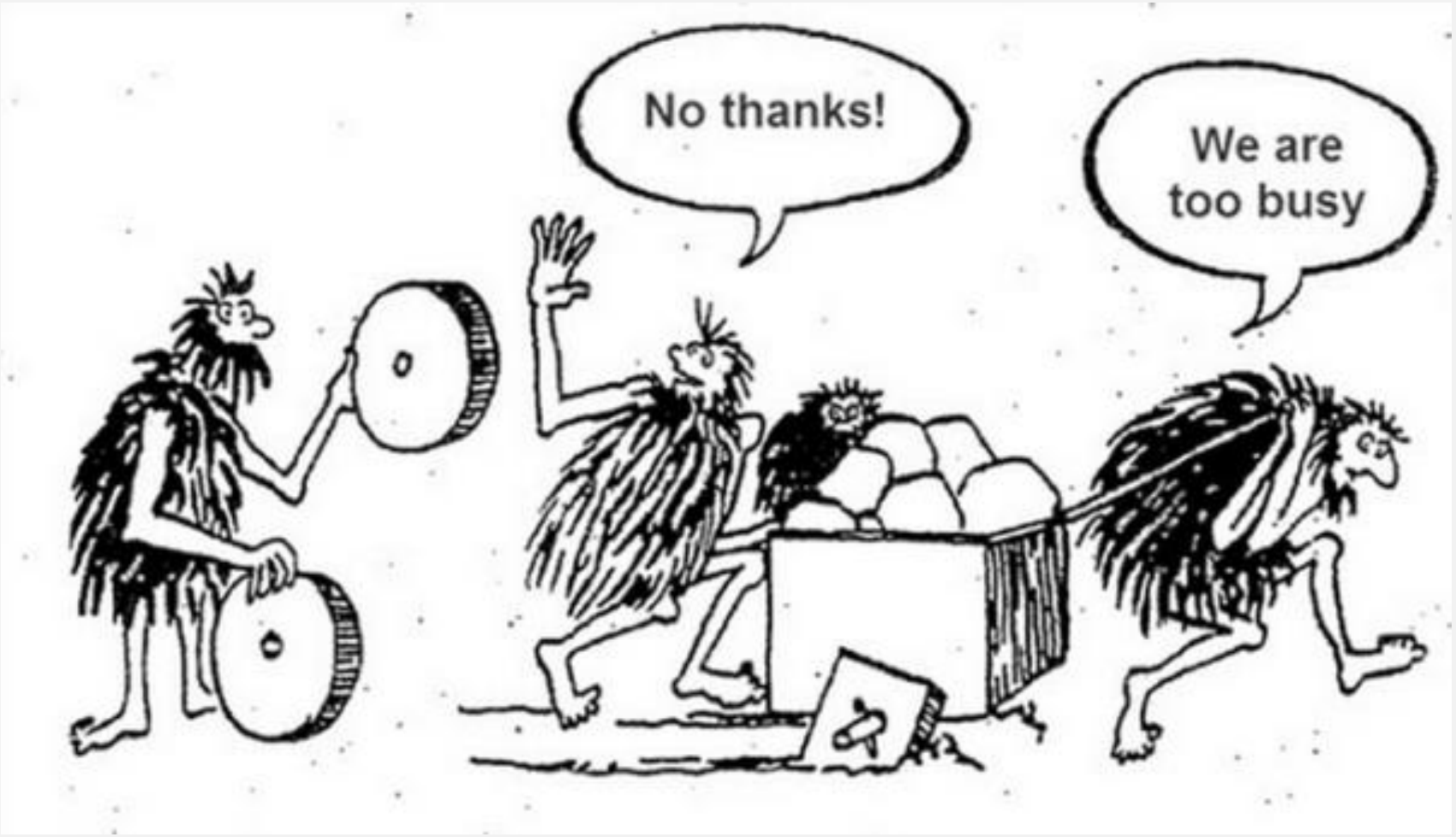
But Robotics is hard to learn, & it takes time to develop a good software for a robot.

ROBOTIC PROGRAMMING

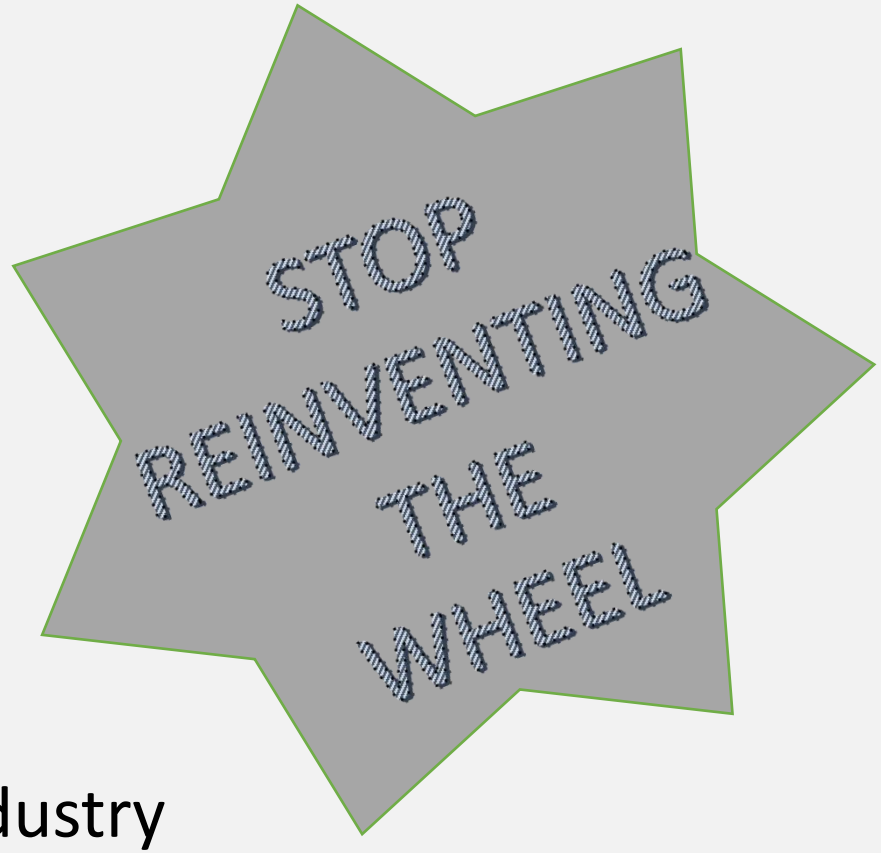
So how do we optimize it?



STOP RE-INVENTING THE WHEEL



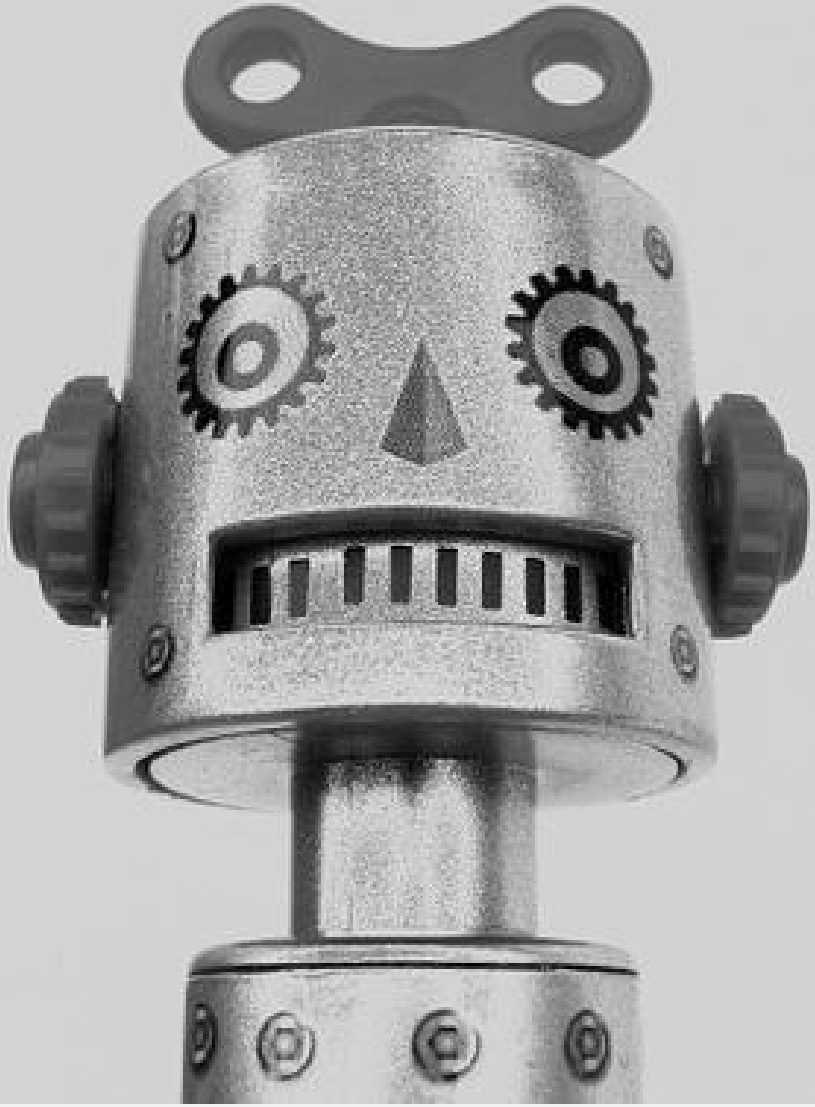
How do we optimize it?



- Parallel research in academia as well as industry
- Efficiency in research and practices
- Collaborative apporach towards work
- Reusability of resources
- Cost effectiveness
- **Existing robotics software framework support**

ROS

Robotic Operating System



ROS

Robotic Operating System

- The ROS project started at Stanford University in 2007, led by roboticist Morgan Quigly. In the beginning it was a group of software developed for robots at Stanford.
- Later in 2007, a robotics research startup called Willow Garage took over the project and coined the name ROS, which stands for Robot Operating System.
- In 2012, the Open Source Robotics Foundation (OSRF) takes over the ROS project.

ROS

Robotic Operating System

- Robot Operating System, despite its name, is not an operating system. Nor it is really a framework, but rather a **meta operating system**.
- A meta operating system is built on top of the operating system and allows different processes (nodes) to communicate with each other at runtime.
- ROS is more of a middleware, something like a low-level “framework” based on an existing operating system.

ROS

Robotic Operating System

Robot Operating System is mainly composed of :

- A core (middleware) with communication tools
- A set of plug & play libraries

The ROS Equation

Plumbing + Tools + Capabilities + Ecosystem

= ROS

Chosing between languages for robotics programming.

+

Trade off between performance and development time.

ROS

Rospy & Roscpp

“ If your goal is research and academia, I would not recommend learning ROS with C++ even if you mastered it, because in academia, speed in testing hypothesis is more important than speed of execution. Hence Python would be your choice. “

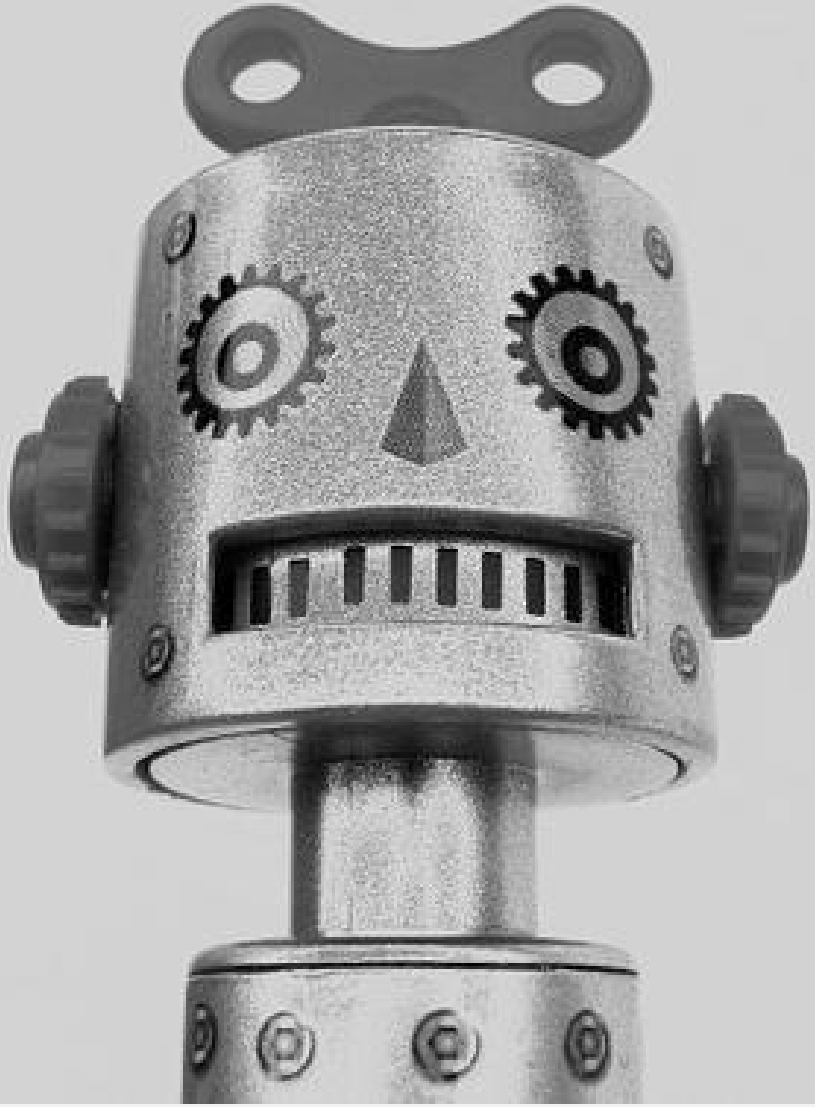
Main languages in ROS repos by popularity

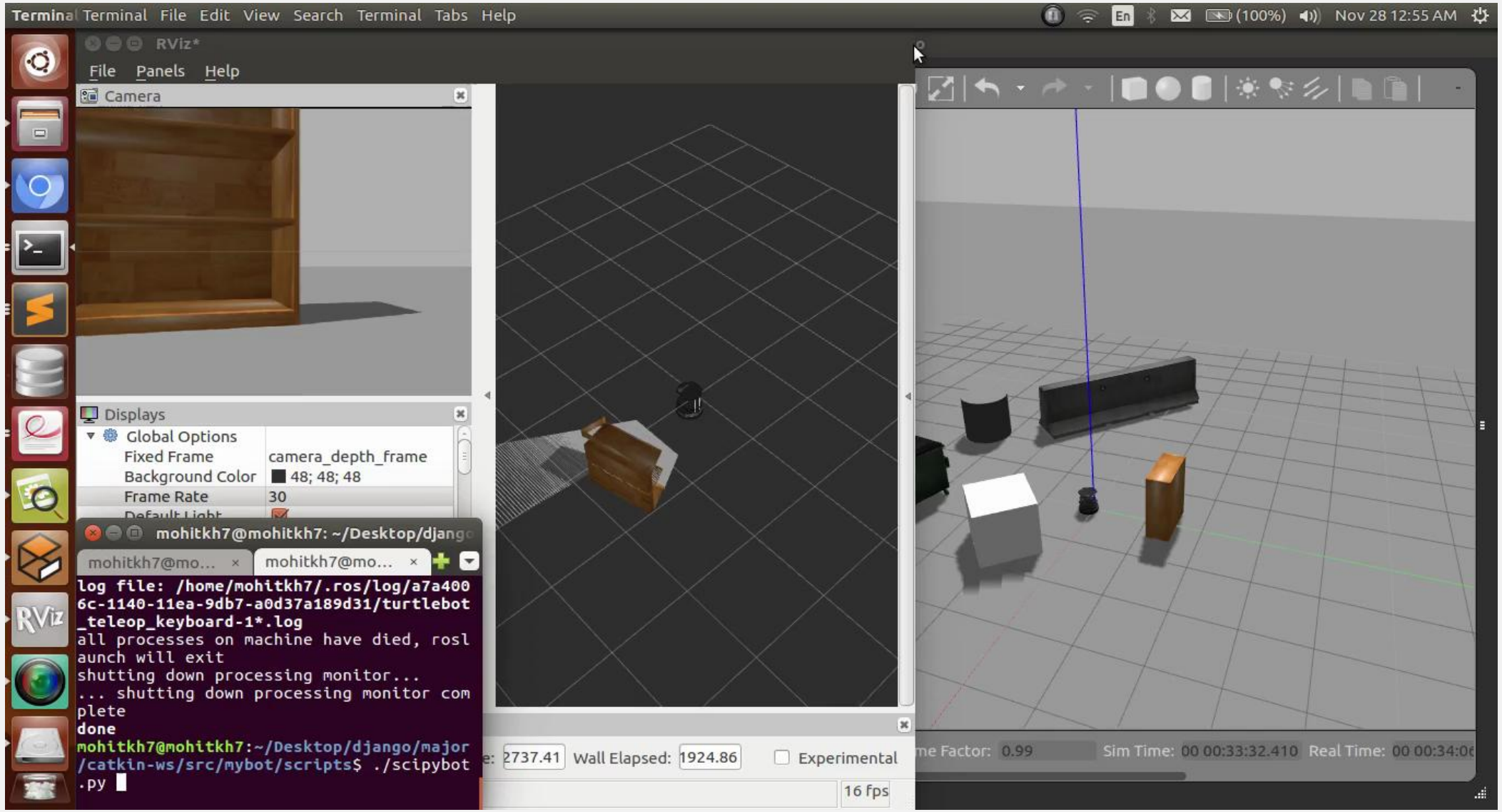
rank	Language	repos	percent
1.	C++	350	55.0%
2.	Python	158	24.8%
3.	CMake	82	12.7%
4.	C	15	2.4%
5.	Common Lisp	8	1.3%
6.	None	7	1.1%
7.	Java	4	0.6%
8.	EmberScript	3	0.5%
9.	Shell	2	0.3%
10.	HTML	2	0.3%
11.	Arduino	1	0.2%
12.	Emacs Lisp	1	0.2%
13.	Lua	1	0.2%
14.	Protocol Buffer	1	0.2%
15.	C#	1	0.2%

ROS Distributions

Distro	Release date	Poster	*Tuturtle*, turtle in tutorial	EOL date
ROS Noetic Ninjemys	May, 2020 (planned, see Upcoming Releases)	TBA	TBA	May, 2025 (planned)
ROS Melodic Morenia (**Recommended**)	May 23rd, 2018			May, 2023 (Bionic EOL)
ROS Lunar Loggerhead	May 23rd, 2017			May, 2019
ROS Kinetic Kame	May 23rd, 2016			April, 2021 (Xenial EOL)
ROS Jade Turtle	May 23rd, 2015			May, 2017

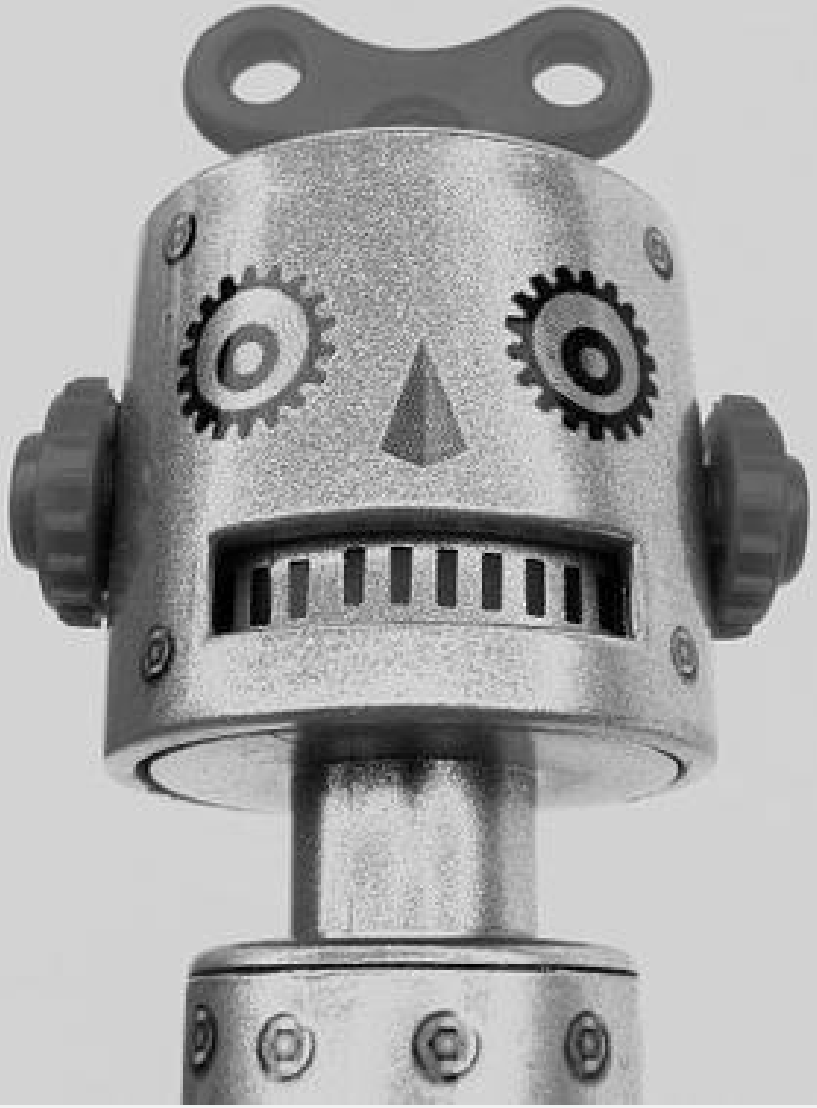
Agenda of WORKSHOP



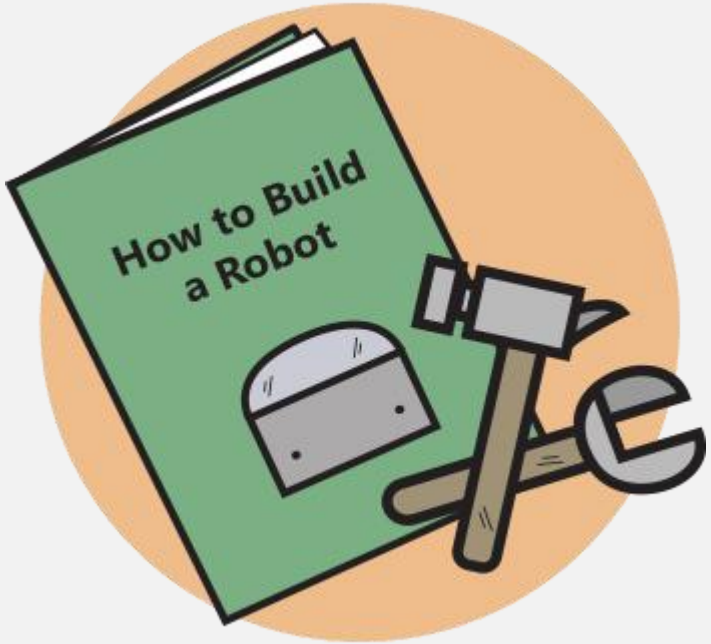

Terminal File Edit View Search Terminal Tabs Help
Nov 28 12:55 AM
RViz*
File Panels Help
Camera
Displays
Global Options
Fixed Frame camera_depth_frame
Background Color 48; 48; 48
Frame Rate 30
mohitkh7@mohitkh7: ~/Desktop/dango
log file: /home/mohitkh7/.ros/log/a7a4006c-1140-11ea-9db7-a0d37a189d31/turtlebot_teleop_keyboard-1*.log
all processes on machine have died, roslaunch will exit
shutting down processing monitor...
... shutting down processing monitor complete
done
mohitkh7@mohitkh7:~/Desktop/django/major/catkin-ws/src/mybot/scripts$ ./scipybot.py
2737.41 Wall Elapsed: 1924.86 Experimental
16 fps
me Factor: 0.99 Sim Time: 00 00:33:32.410 Real Time: 00 00:34:06

LEARNING PATH

- Tools to be used
- ROS Fundamentals
- Hands-on
- Building a ROBOT
- ROS Discussion
- Practical Implementations
- Future with ROS
- Closure

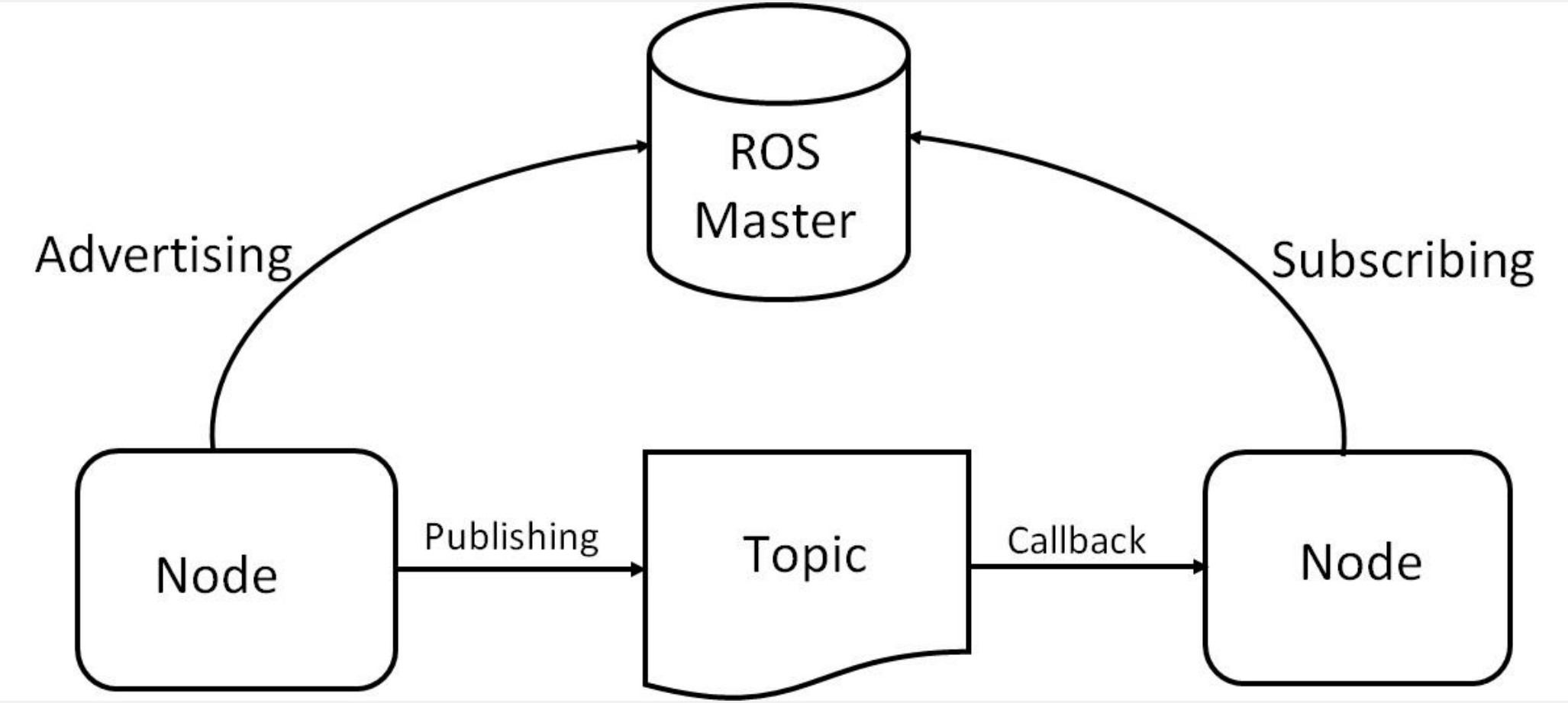


Tools to be used:



- Gazebo(simulator with dynamic and kinematic physics)
- TurtleBot(ROS Compatible robot)
- Rviz(sensor data visualisation tool)

ROS Architecture



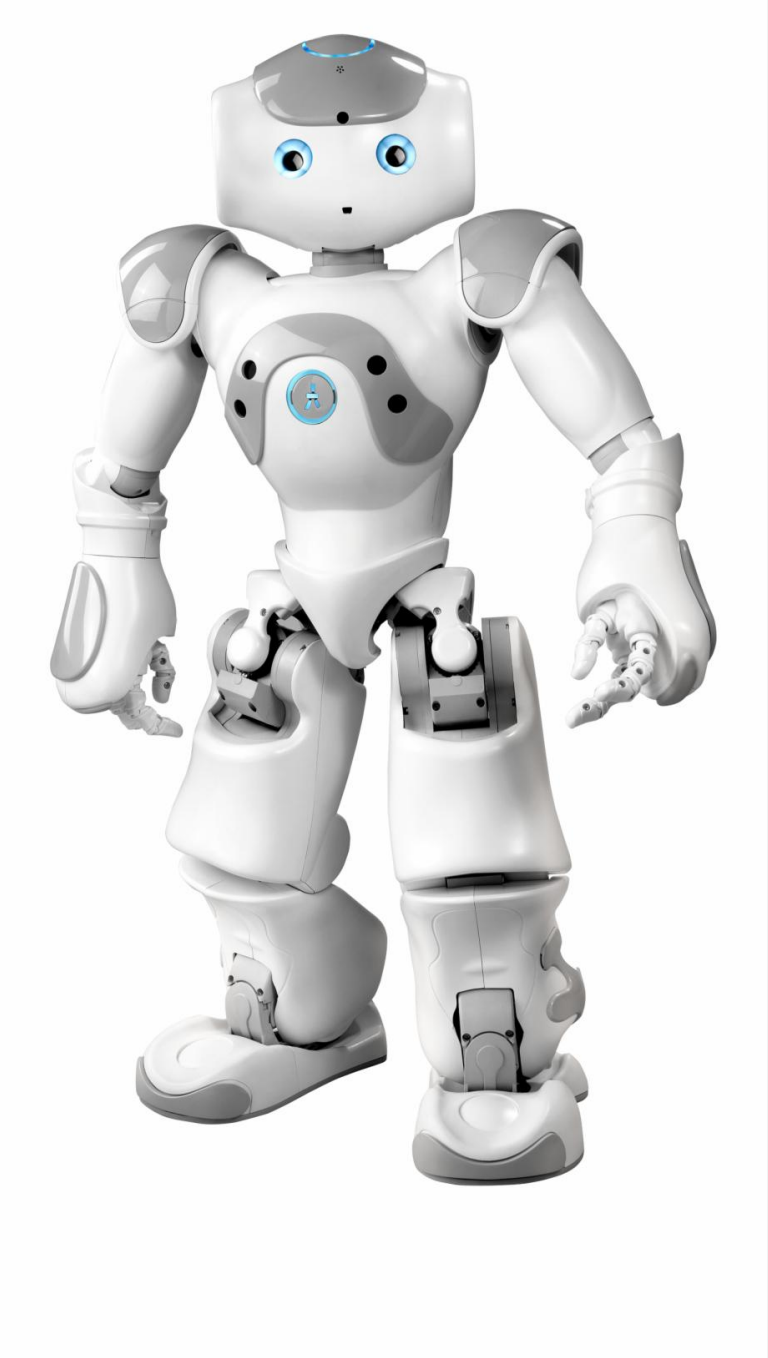
RosPy

How is robotics programming done with ROS

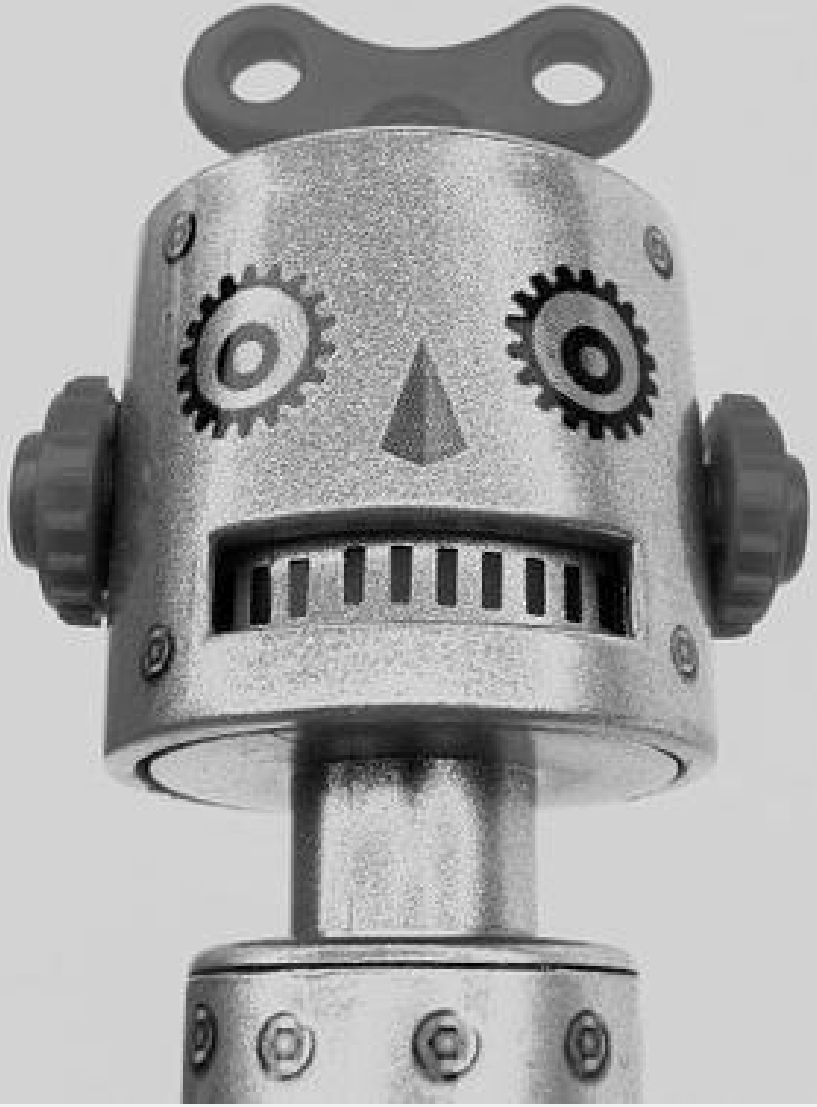
- Roscore
- Nodes (Publisher/ Subscriber)
- Topics (commuication)
- Messages
- Services

ROS type	Serialization	C++ type	Python type	Notes
`bool`	Unsigned 8-bit integer	`uint8_t`	`bool`	
`int8`	Signed 8-bit integer	`int8_t`	`int`	
`uint8`	Unsigned 8-bit integer	`uint8_t`	`int`	`uint8[]` is treated as a `string` in Python
`int16`	Signed 16-bit integer	`int16_t`	`int`	
`uint16`	Unsigned 16-bit integer	`uint16_t`	`int`	
`int32`	Signed 32-bit integer	`int32_t`	`int`	
`uint32`	Unsigned 32-bit integer	`uint32_t`	`int`	
`int64`	Signed 64-bit integer	`int64_t`	`long`	
`uint64`	Unsigned 64-bit integer	`uint64_t`	`long`	
`float32`	32-bit IEEE float	`float`	`float`	
`float64`	64-bit IEEE float	`double`	`float`	
`string`	ASCII string	`std::string`	`string`	ROS does not support Unicode strings; use a UTF-8 encoding
`time`	`secs/nsecs` unsigned 32-bit ints	`ros::Time`	`rospy.Time`	`duration`

Now let's
keep calm
and
build a ROBOT !



Why to learn **ROS ?**

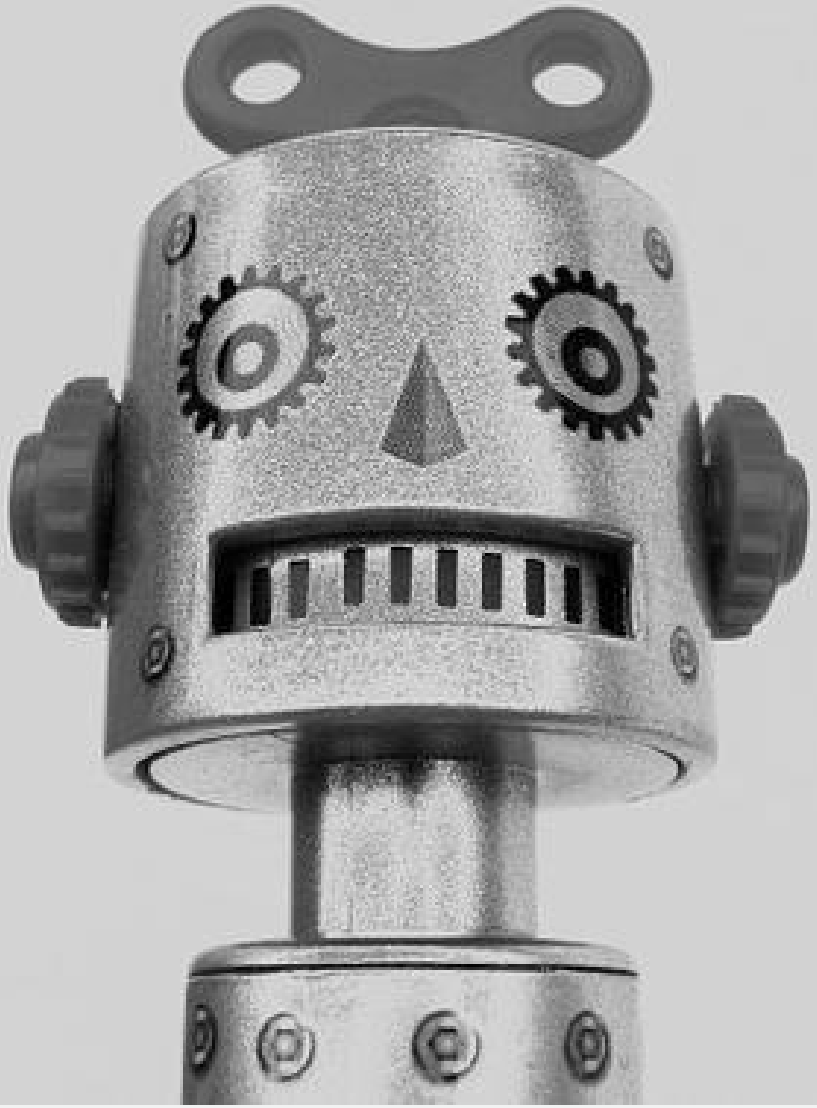


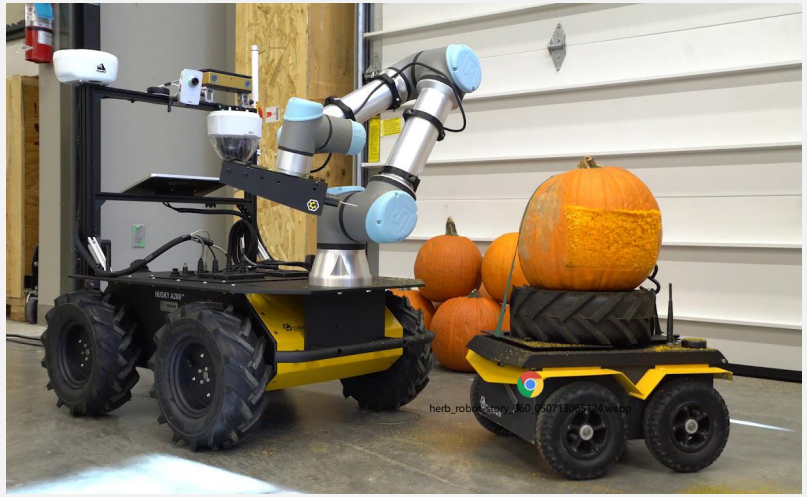
Features

Why RosPy is better for academia and research

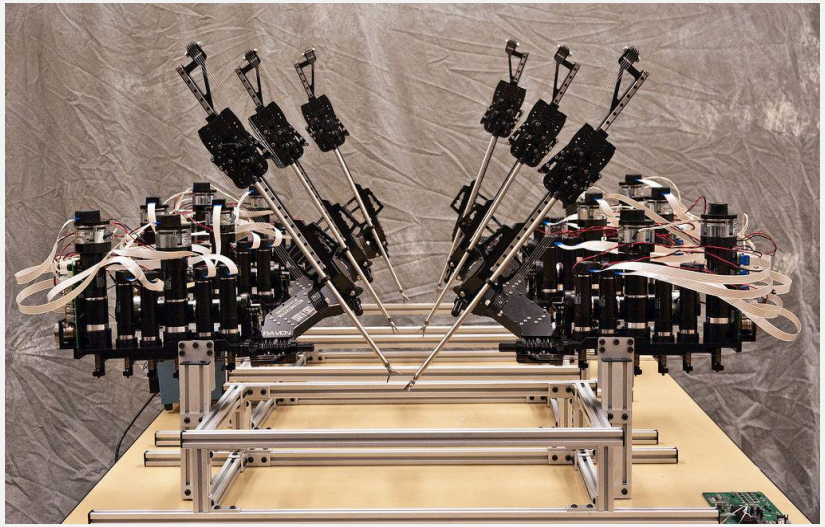
- Simulation + real world application
- Community support
- Prebuild library
- Cost effective (for simulation)
- Popularity
- Tools
- Customization

Are serious things done with ROS in **real life**?



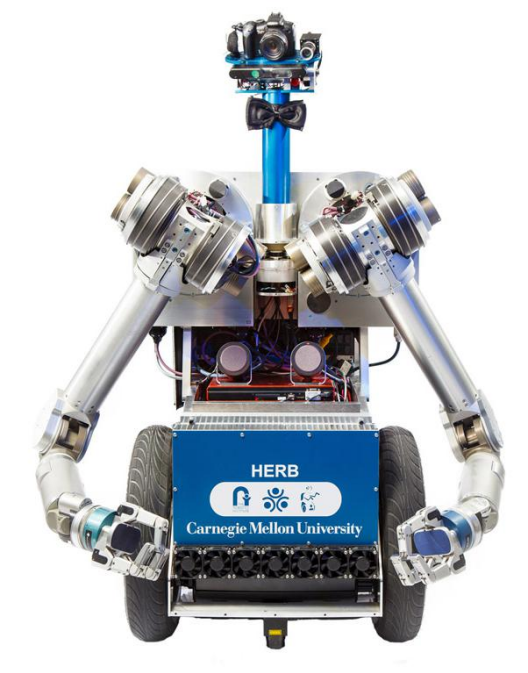


HUSKY (a medium sized robotic development platform by ClearPathRobotics)



Raven II Surgical Robotic Research Platform

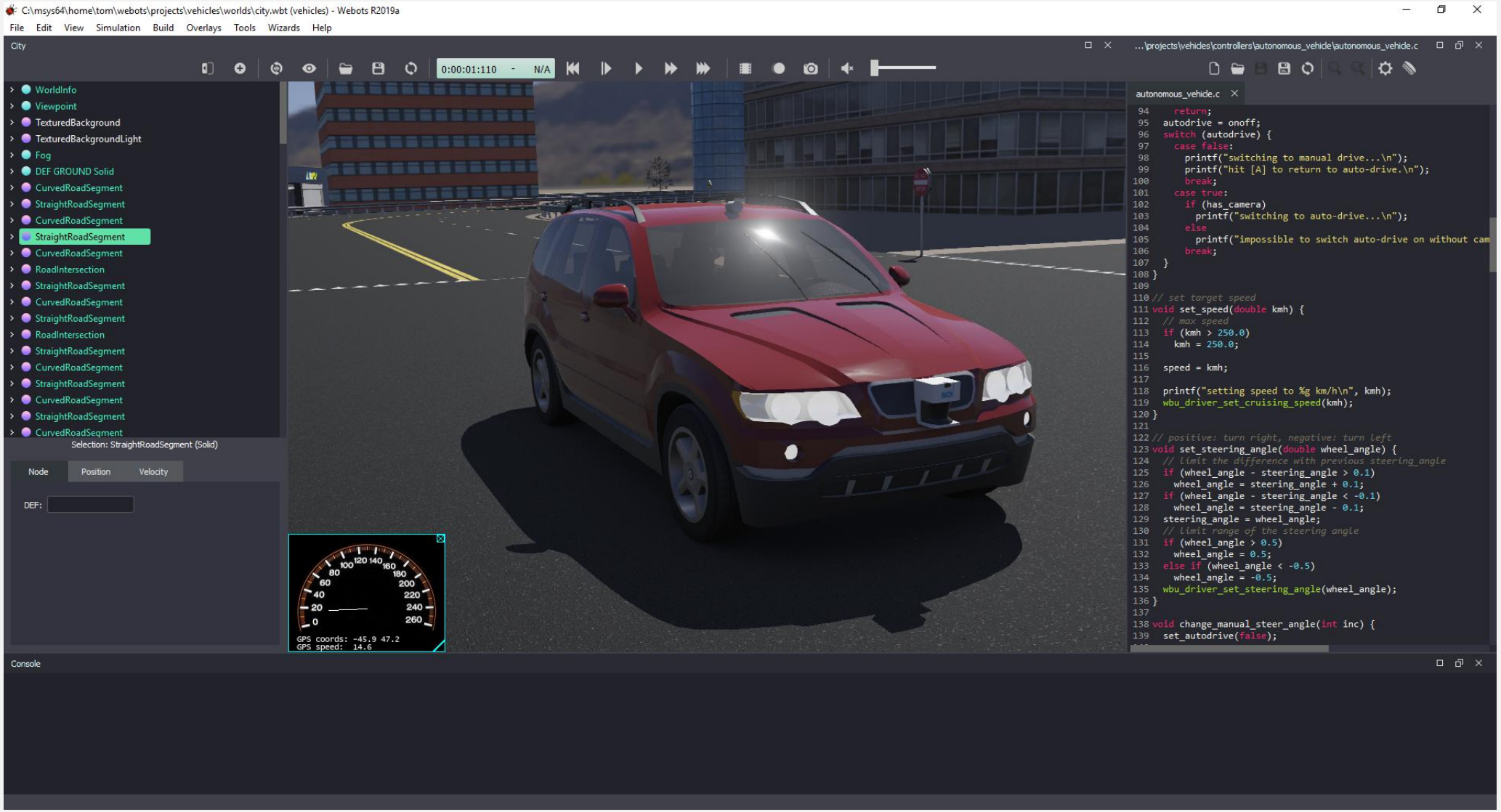
Robonaut 2: A NASA robot designed to automate various tasks on the International Space Station.



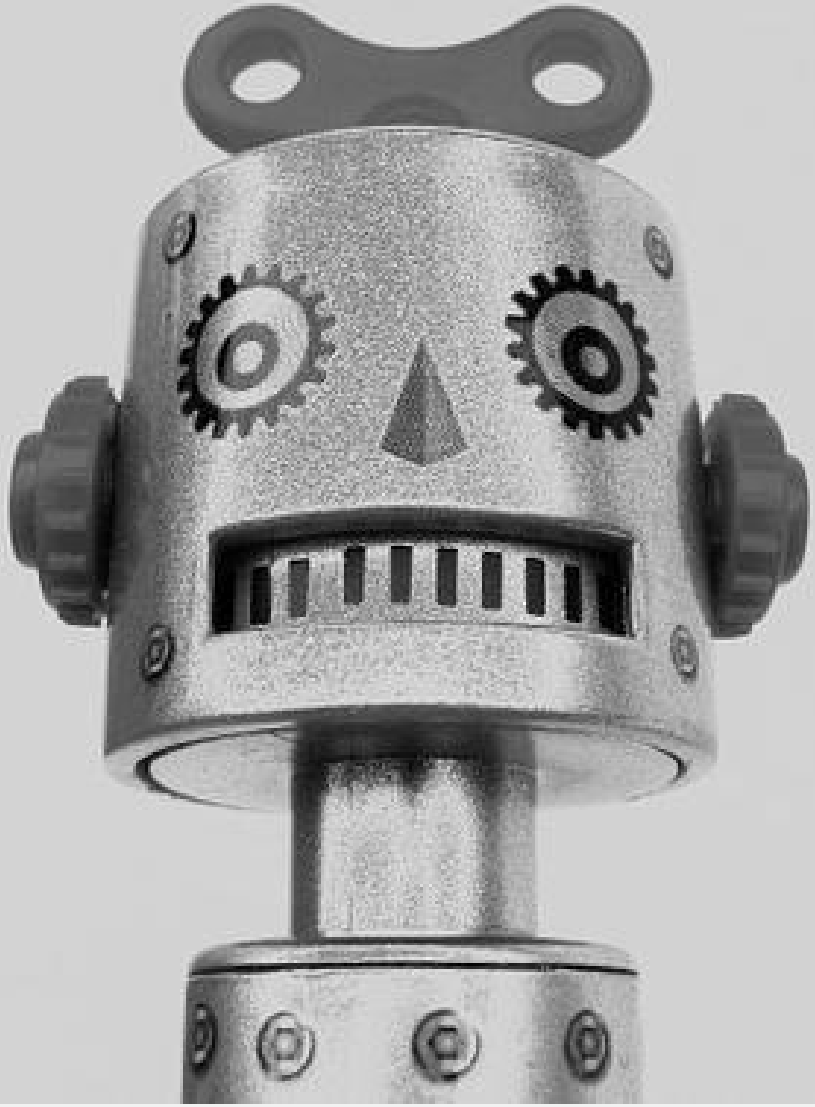
HERB, developed at Carnegie Mellon University in Intel's personal robotics program

A full-size humanoid robot that is mainly used for research purposes

WEBOTS Interface (a free and open-source 3D robot simulator used in industry, education and research.)



How to learn **ROS ?**



How to learn **ROS ?**

Books

- Programming Robots with ROS
- Robot Operating System (ROS) for Absolute Beginners

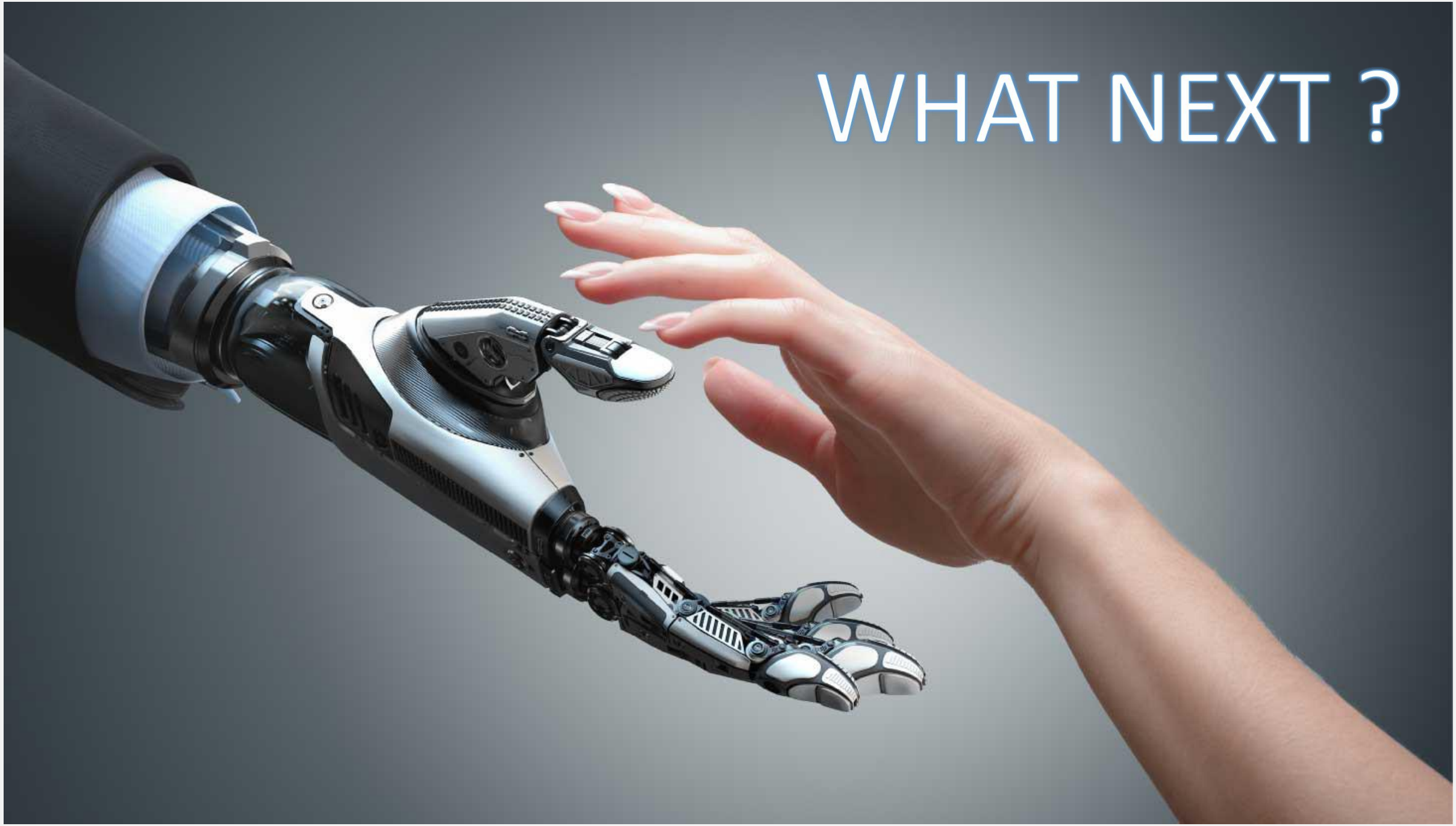
Websites

- ROS Wiki
- Robotigniteacademy

Videos

- The construct
- Robotics System Lab

WHAT NEXT ?



Feel free to <u>DISTURB</u> !

- Mohit Khandelwal
  mohitkh7@gmail.com

- Akshita Kanojia
  akshitakanojia786@gmail.com