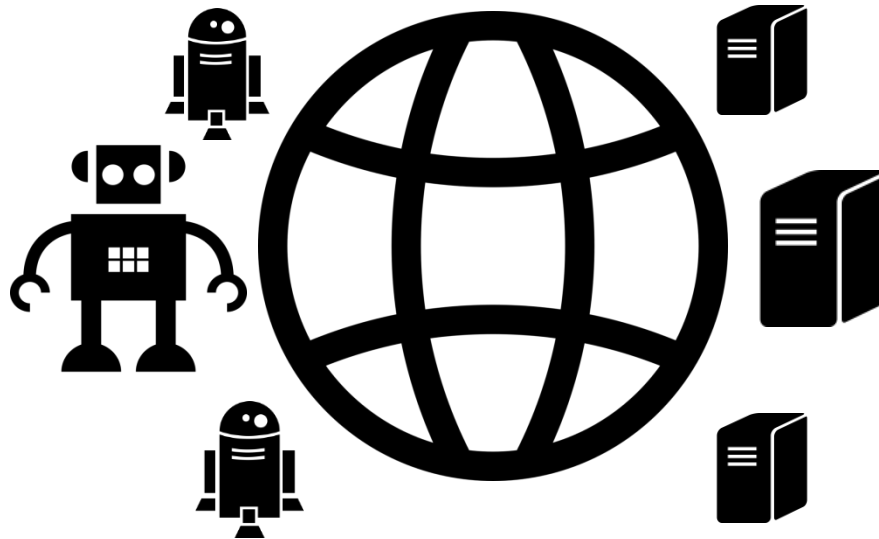
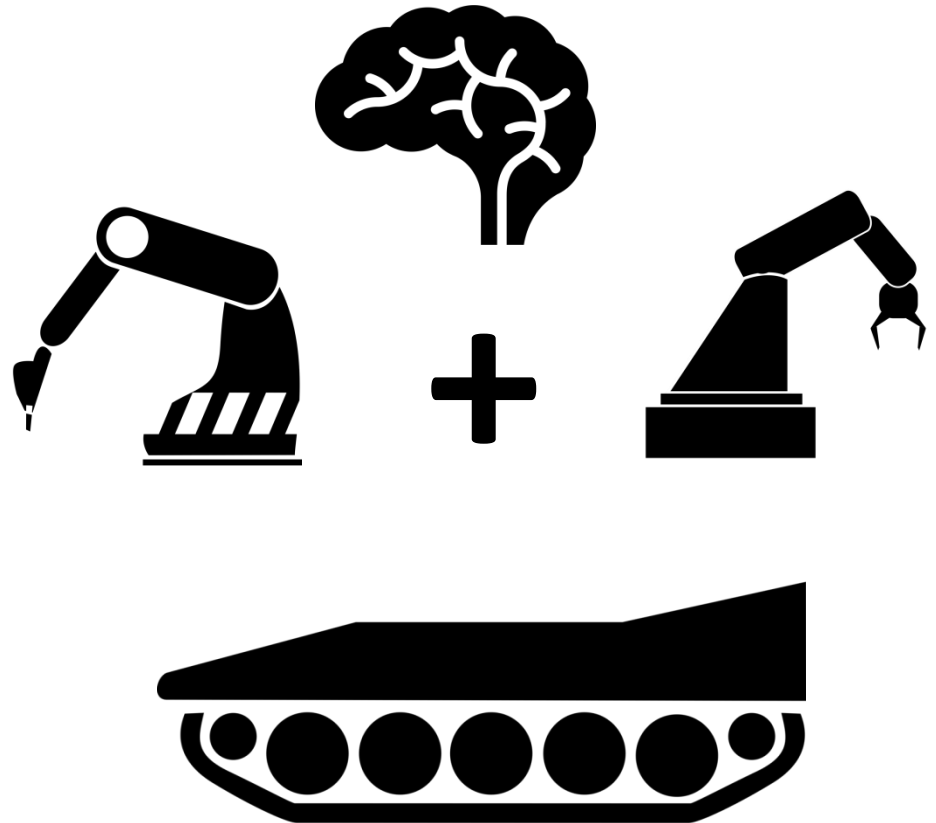
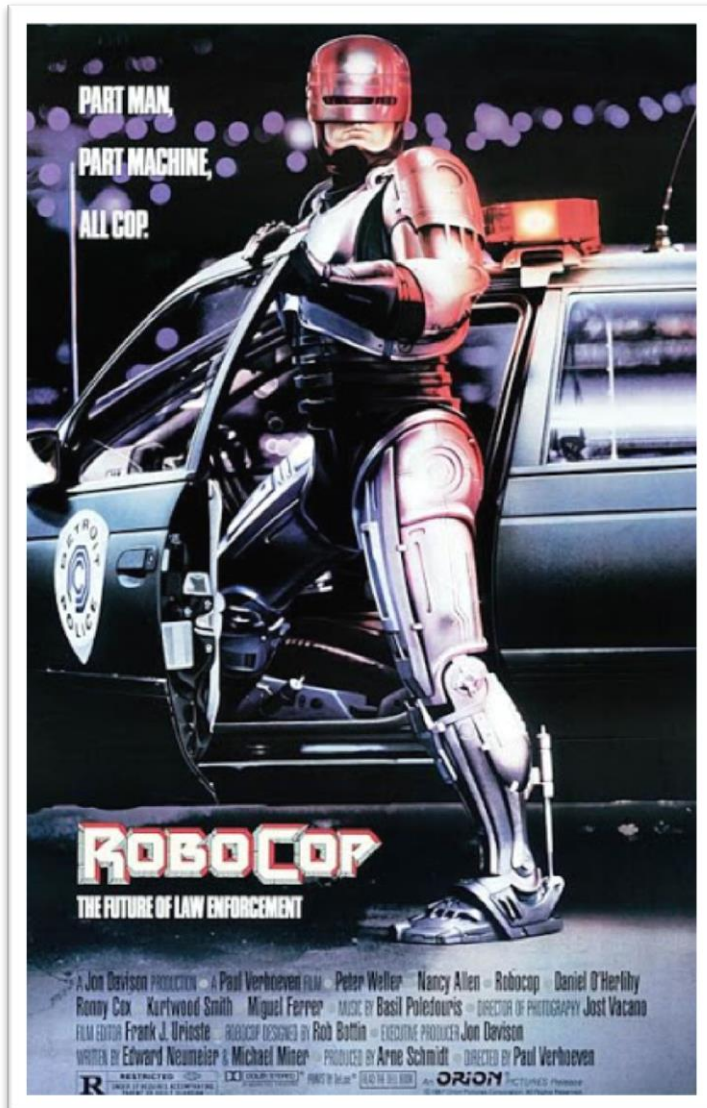


Robotic Middlewares

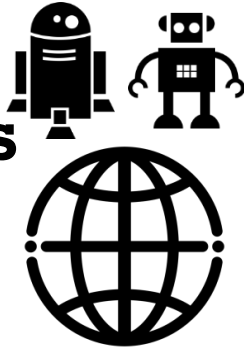




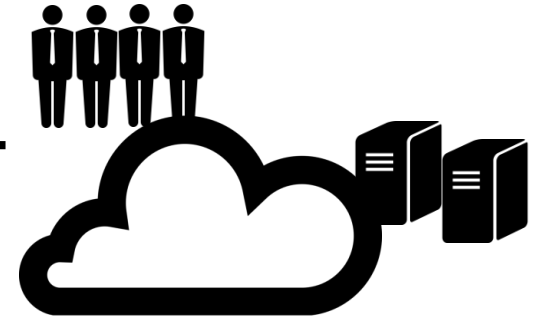
What is Robotic today ?



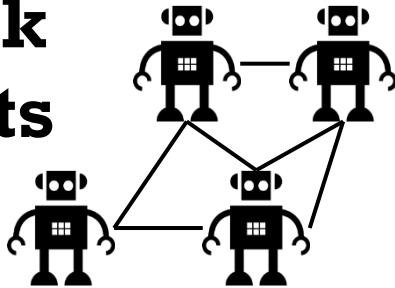
**Internet
for robots**



**Crowd
sourcing
Cloud
Computing**



**Network
of robots**

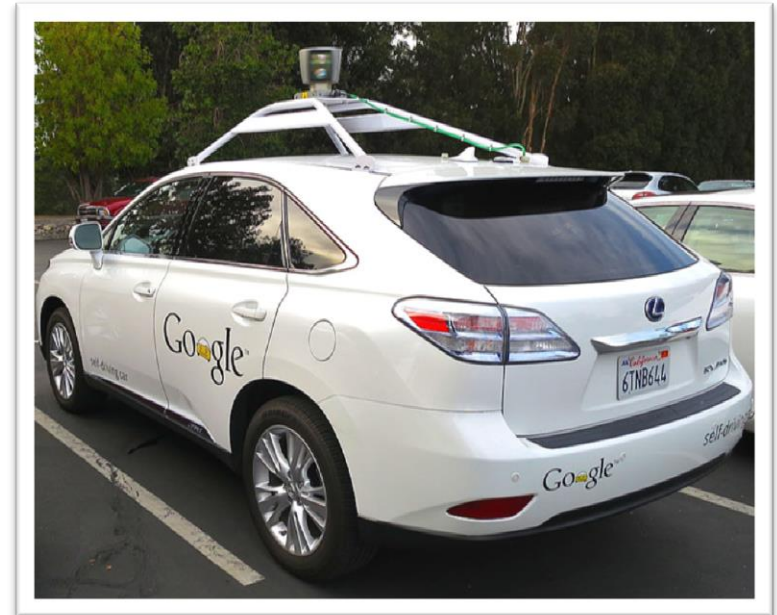


Distributed Intelligence

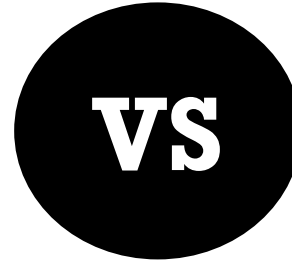
Industrial

VS

Service



Industrial



Service

- Known environment
- High speed design
- Intelligence close to the hardware
- Limited robot cooperation
- Limited (forbidden ?) human interactions

- Unknown environment
- Need to deal with uncertain
- Higher level of intelligence
- Strong human interactions
- Possibilité of robot to robot, human to robot cooperation



Current Robotic Needs

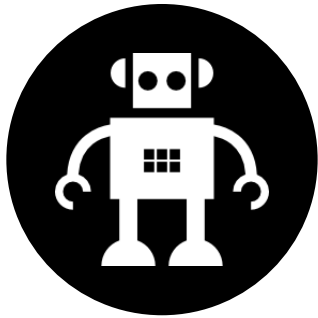


- Specific implementation
- Heterogeneous communication
- Centralized communication
- Re-Inventing the Wheel Syndrom



Develop Robotic Framework

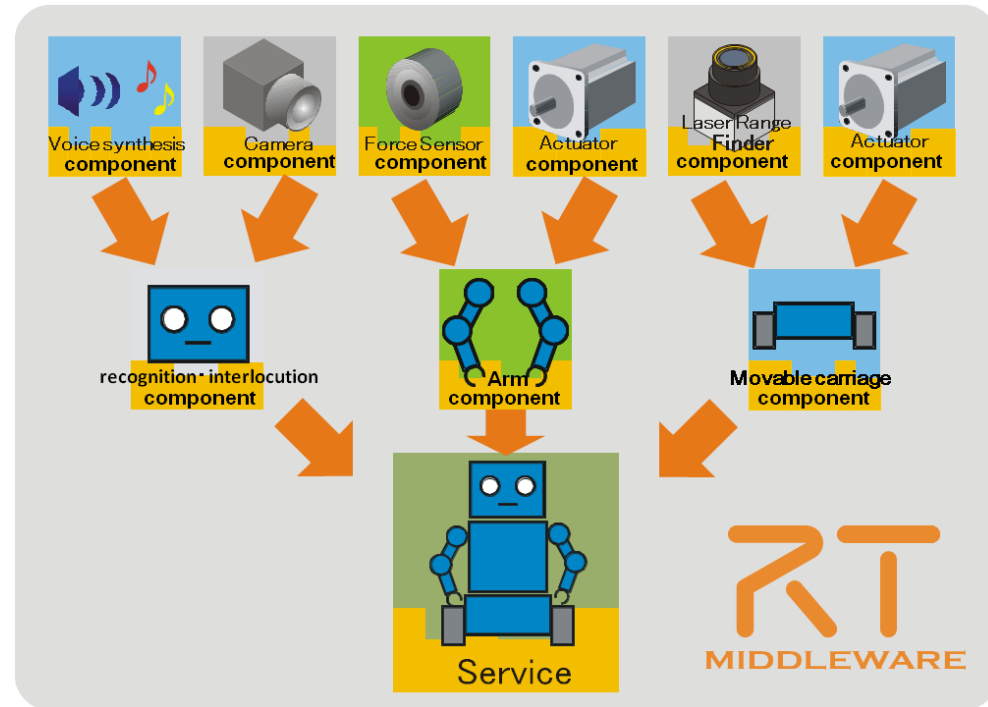
- Naoqi
- Urbi
- RT-Middleware
- ROS



RT Middleware: Robot Technology Middleware

What is RT Middleware ?

- A Robotic Middleware
- A common platform to assist development of robots
- A Modular Design
- OpenRTM-aist implementation based on Corba
- Under LGPL License



Who provides RT-MiddleWare ?



Develop an implementation of RT-Middleware called OpenRTM-aist



Defines / discusses RT-component specification/ standard

Who uses RT-MiddleWare ?



VSTONE



Segway Blackship



Kanto Auto Works,
Patrafour



Mobilierobots
PeopleBot



Choromet 2



G-ROBOTS GR001



HRP-2



Segway RMP

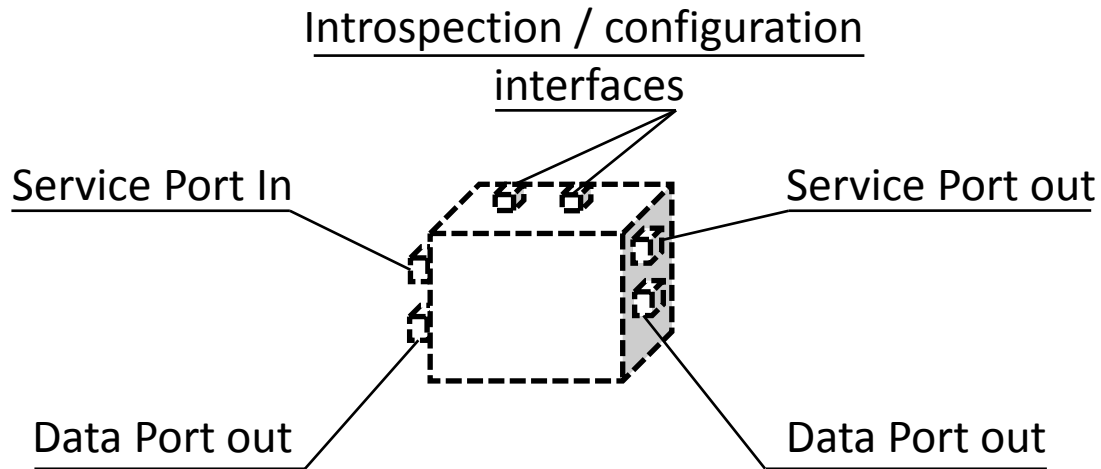
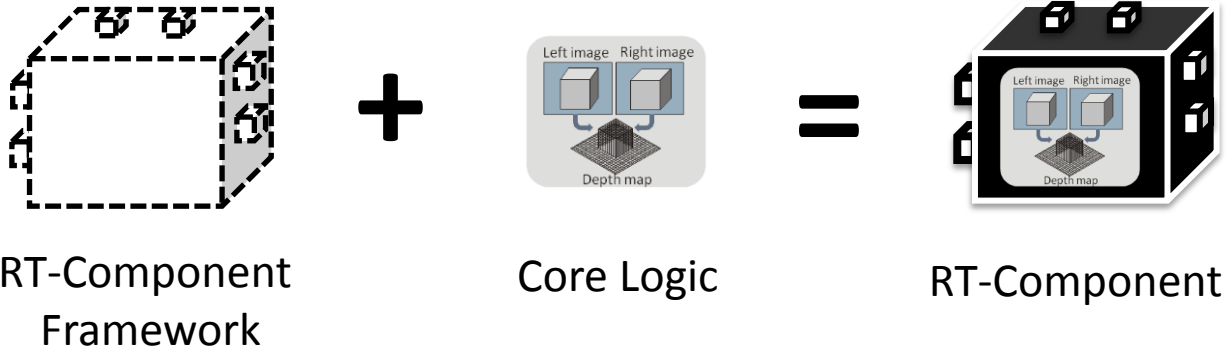
Copyright Leber, Jumel, Saraydaryan 2015

Who uses RT-MiddleWare ?

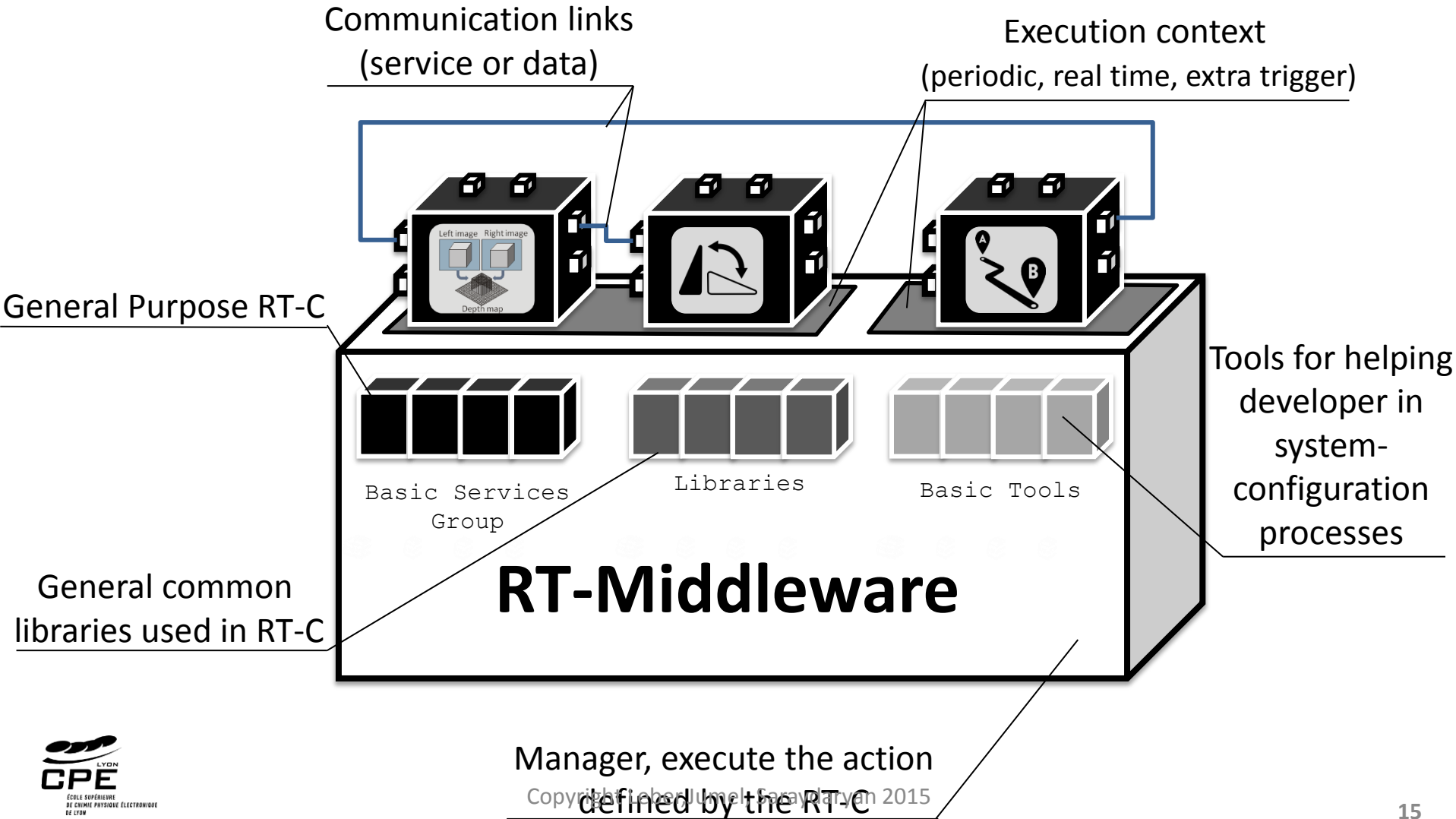


Kawada Industry Humanoid Robot - HIRO

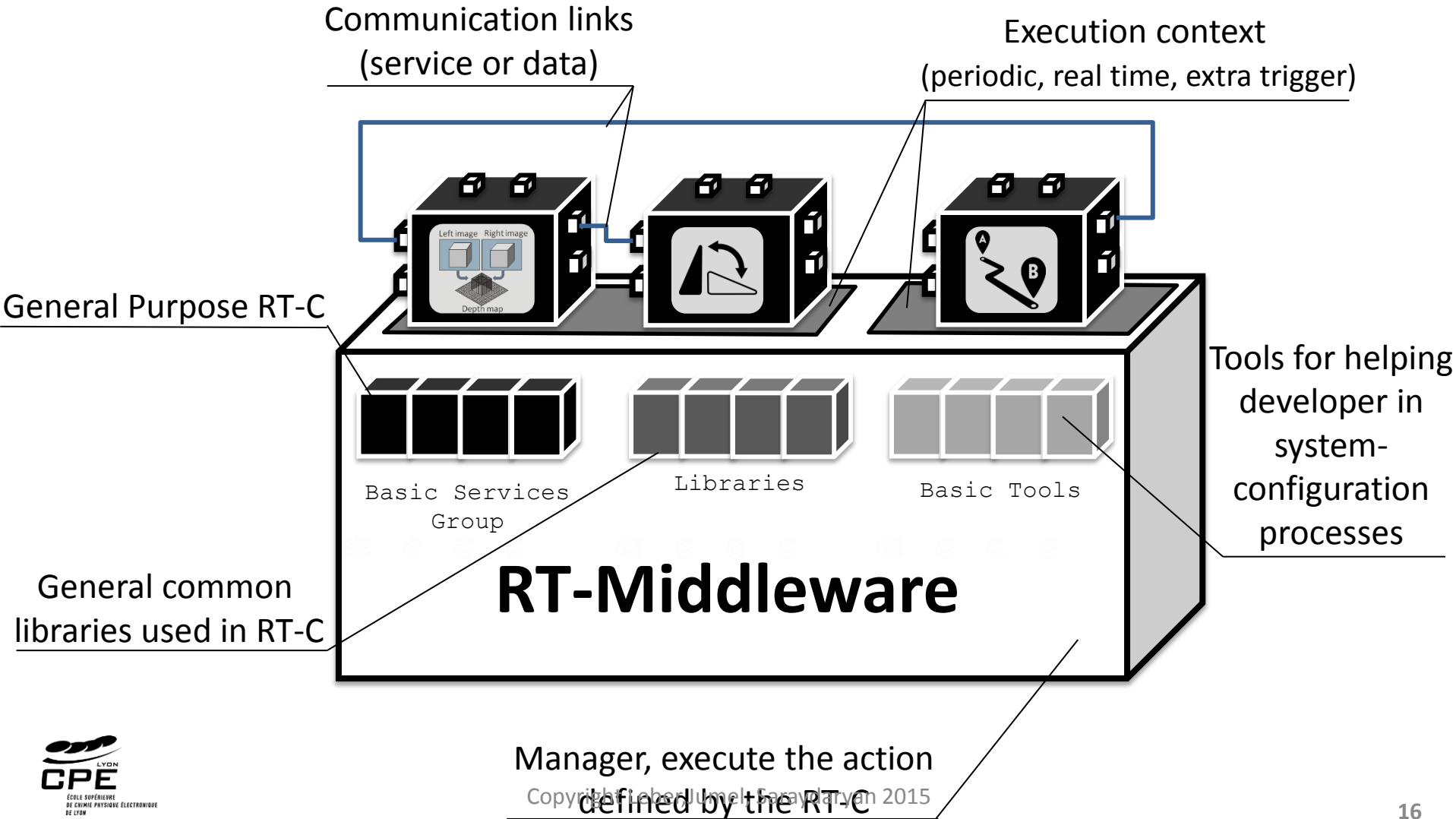
How does it work ?



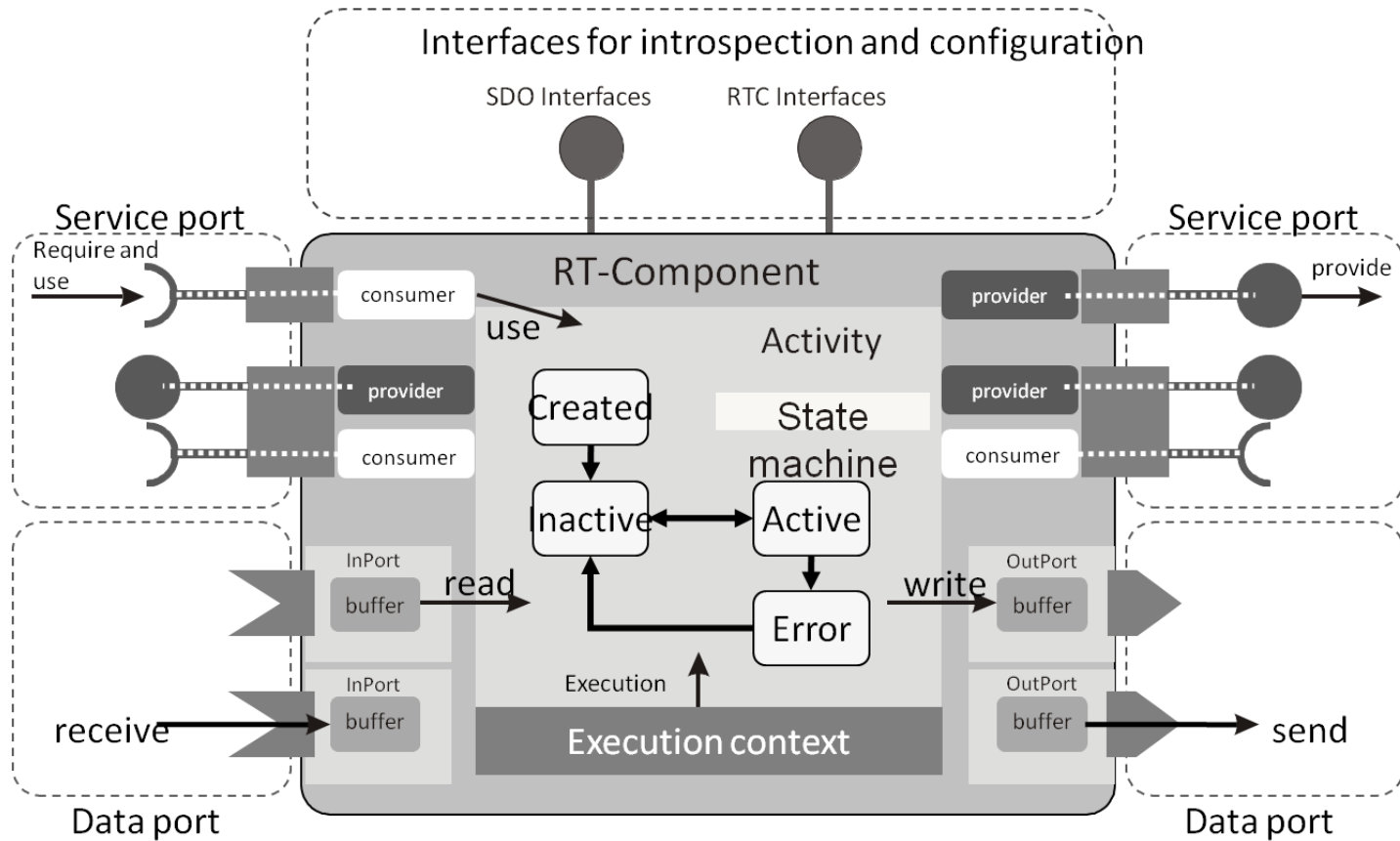
How does it work ?



How does it work ?



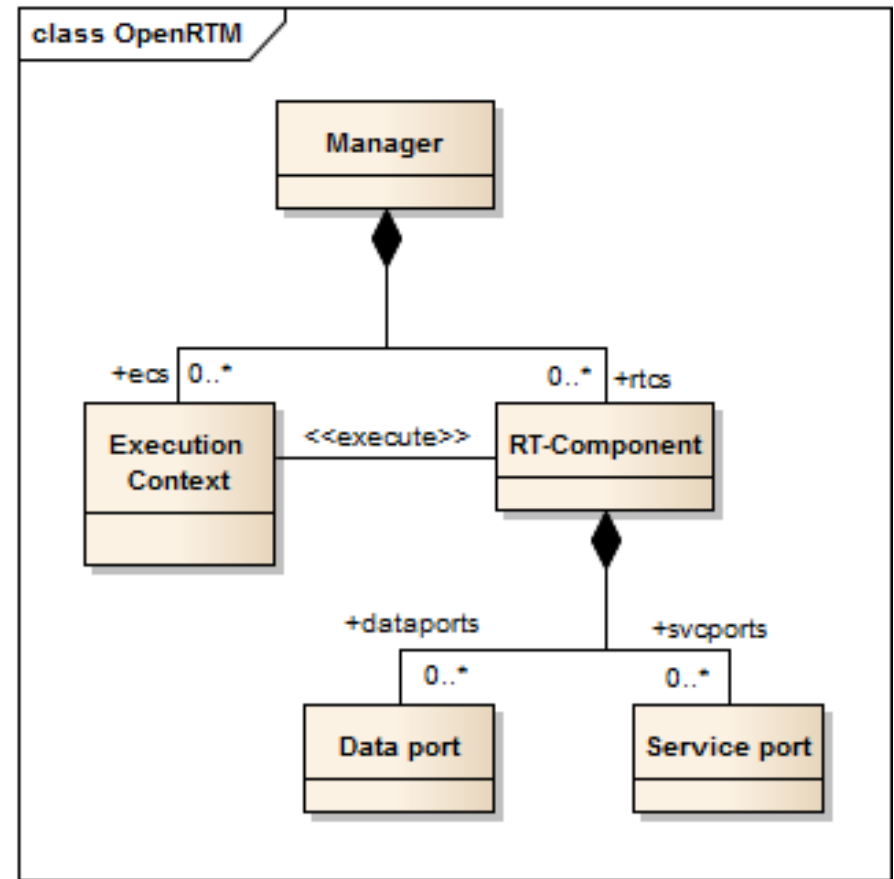
How does it work ?

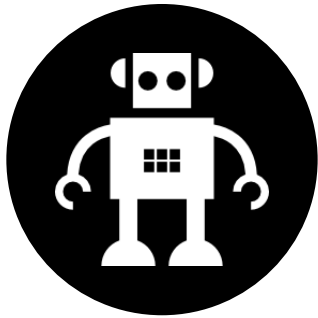


http://www.openrtm.org/openrtm/sites/default/files/440/RTCArchitecture_en.png

How To develop ?

1. Define the specification of a component.
2. Generate the templates of the component by using a cord generator.
3. Integrate libraries etc. you have in a specific location in the template.
4. Compile, test and debug.
5. Integrate into a system.

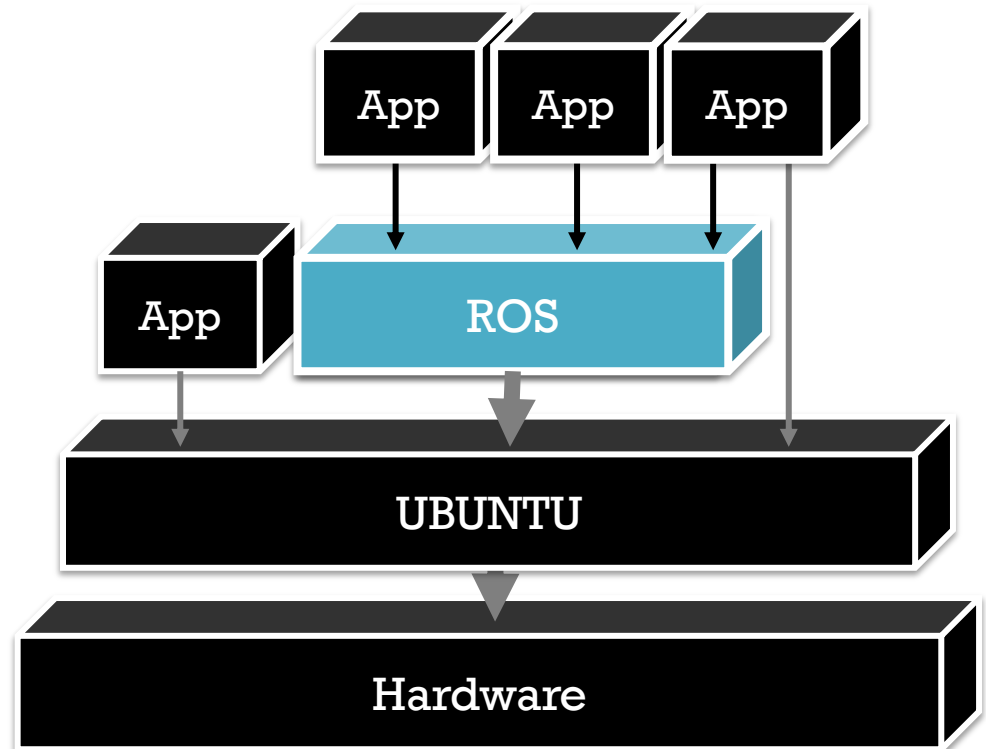




ROS: Robot Operating System

What is ROS ?

- A Robotic Middleware
- A set of libraries
- A distributed Modular Design
- Free and OpenSource



What does ROS Provide?

Applications

Fetching beer,
scraping the seafloor,
play football,
manufacture objects

Capabilities

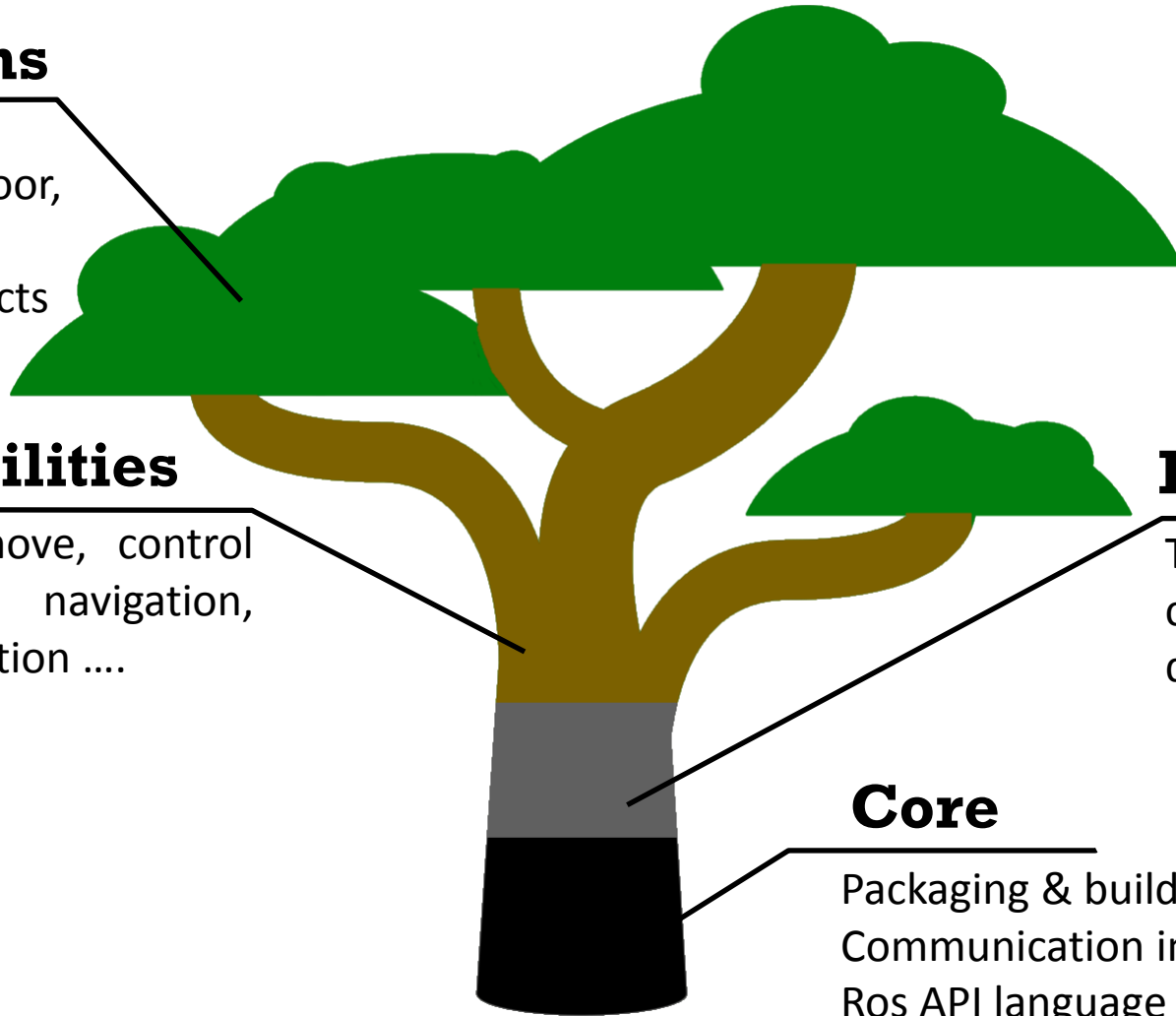
Grasping, move, control
execution, navigation,
object detection

Libraries

Tf, opencv, pcl, kdl,
cisst, simulation,
drivers ...

Core

Packaging & build tools,
Communication infra
Ros API language binding
Introspection tools ...



Who provides ROS ?

An Active Community



A great history



May 1, 2007 — September 1, 2007

Switchyard at Stanford

Before the ideas fully coalesced to become ROS, several robotics software frameworks were prototyped in research projects at Stanford, including the STanford Artificial Intelligence Robot (STAIR) and the Personal Robotics (PR) program.



NOVEMBER 1, 2007
ROS at Willow Garage



Who uses ROS ?



Fraunhofer IPA
Care-O-bot



Videre Erratic



TurtleBot



Aldebaran Nao



Lego NXT



Shadow
Hand



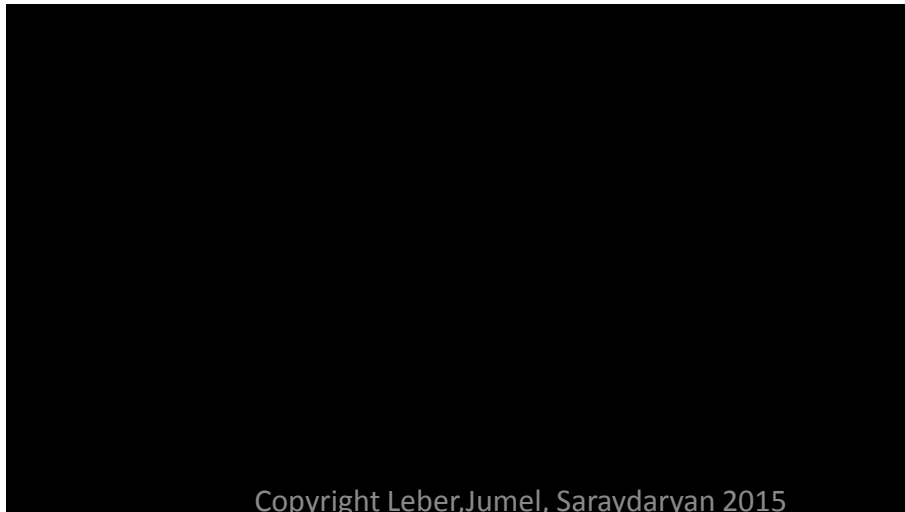
Willow Garage PR2



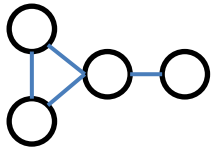
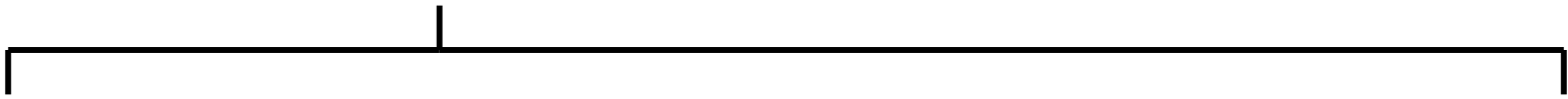
iRobot
Roomba



Robotnik
Guardian



How does it work ?



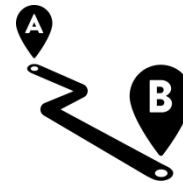
Set of Nodes

- Application
- Services
- Modules



Name Service

- Node registration
- Node Directory
- Communication orchestrator



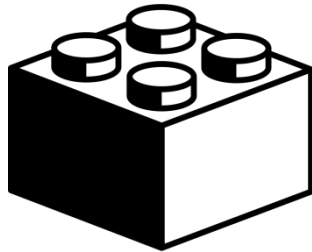
Communication

- Asynchrone (topics)
- Synchrones (services)



Build Tools

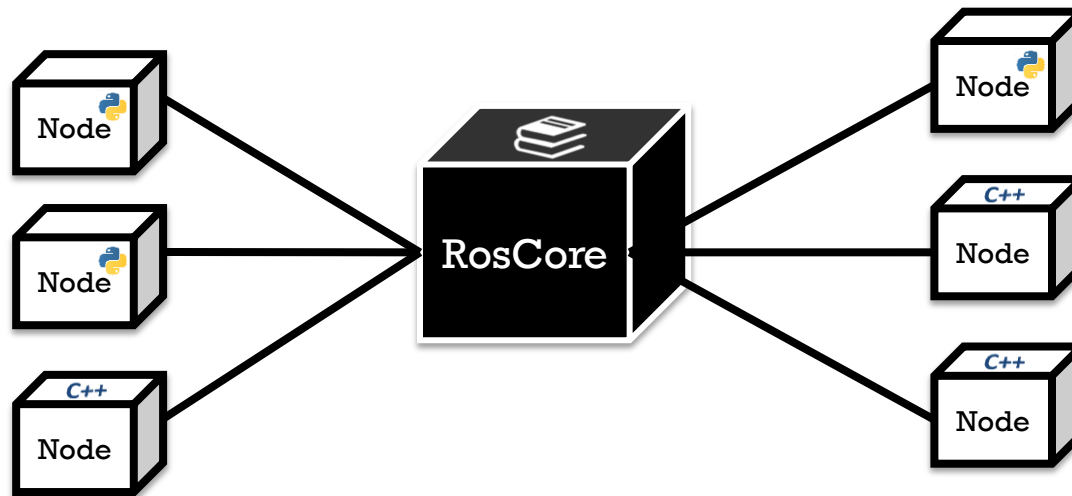
- Compilation,
- Packaging
- Dependency management



ROS: The Basics

ROS: Node

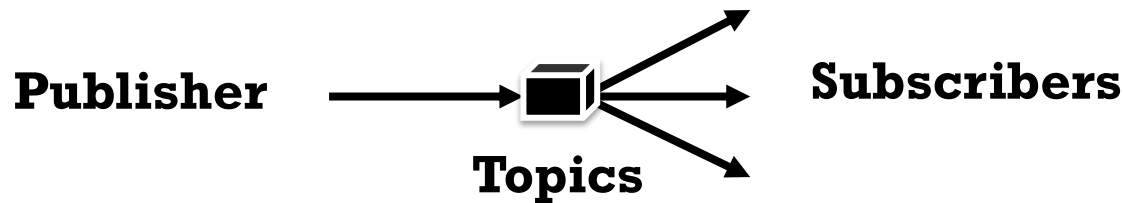
- Process that uses ROS framework
- Can be Remote
- Are known via Roscore
- Can be written into different languages : Python, C++



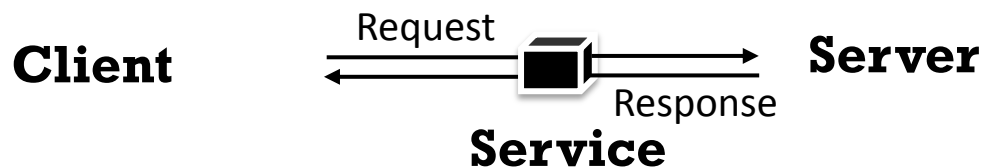
- Ros Core acts primarily as a name serveur
- Provide communication architecture
- Hold global parameters

ROS: Communication

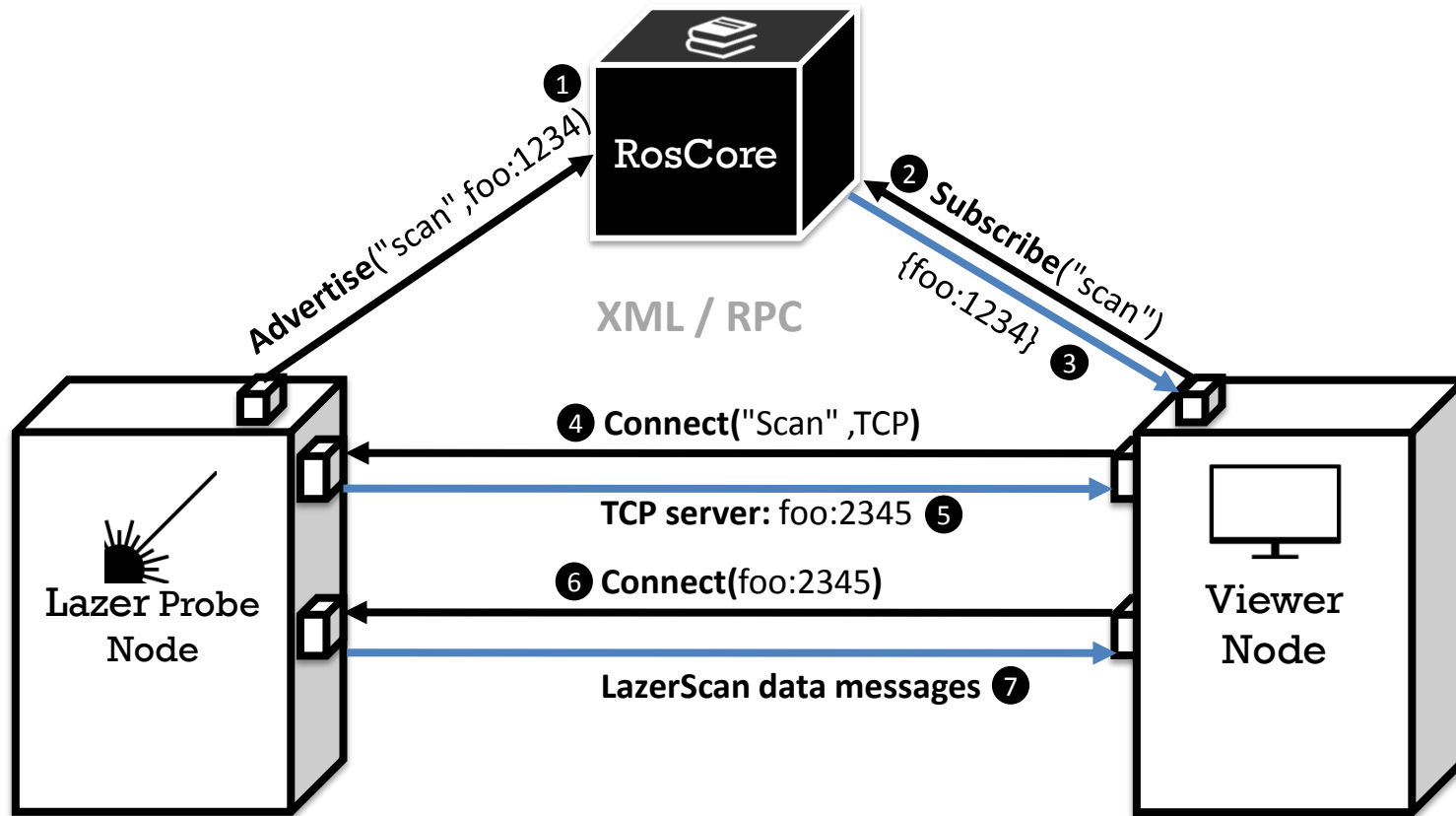
- Topics
 - Asynchronous
 - Many to Many
 - Publish/subscribe anonymous message passing



- Services
 - Synchronous
 - RPC request /reply interaction



ROS: Communication



ROS: Command line

<code>roscore</code>	Start the ros Master
<code>roslaunch</code>	Start a set of ros node
<code>rostopic</code>	Interact with ros topic
<code>rosservice</code>	Interact with ros service
<code>rosparam</code>	Getting and setting ros parameters
<code>rosviz</code>	Register/replay ros topic
<code>rosclear</code>	Display information about ros node
<code>rostopic</code>	Display message data structure

...

ROS: Command line

roscore

- Dfn:
Start the ros Master
- Usage
roscore
roscore -p 4523

```

roscore http://astroros-VirtualBox:11311/
astroros@astroros-VirtualBox:~$ roscore
... logging to /home/astroros/.ros/log/893ca7c0-49fc-11e4-88c9-08002799def9/ros1
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://astroros-VirtualBox:56033/
ros_comm version 1.11.3

SUMMARY
=====

PARAMETERS
* /rosdistro: <...>
* /rosversion: <...>

NODES

auto-starting new master
process[master]: started with pid [30879]
ROS_MASTER_URI=http://astroros-VirtualBox:11311/

setting /run_id to 893ca7c0-49fc-11e4-88c9-08002799def9
process[rosout-1]: started with pid [30892]
started core service [/rosout]
    
```

ROS: Command line

roslaunch

- Dfn:
Start the a ros node

- Usage

```
roslaunch <package> <executable>  
roslaunch vision facedetection
```

```
astroros@astroros-VirtualBox: ~  
astroros@astroros-VirtualBox:~$ roslaunch turtlesim turtlesim_node  
[ INFO] [1412232629.757742090]: Starting turtlesim with node name /t  
urtlesim  
[ INFO] [1412232629.831810387]: Spawning turtle [turtle1] at x=[5,54  
4445], y=[5,544445], theta=[0,000000]
```

ROS: Command line

roslaunch

- Dfn:
Start the multiple ros node

- Usage
`roslaunch <package> <file.launch>`
`roslaunch vision facedetection`

```

/home/astros/ros_ws/src/turtle_sample/launch/turtle.launch http://localh
astros@astros-VirtualBox:~$ roslaunch turtle_sample turtle.launch
... logging to /home/astros/.ros/log/1f28151e-4a00-11e4-9096-080027
99def9/roslaunch-astros-VirtualBox-1172.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://astros-VirtualBox:49129/

SUMMARY
=====
PARAMETERS
* /rostdistro: <...>
* /rosversion: <...>

NODES
/
  mimic (turtlesim/mimic)
/turtlesim1/
  sim (turtlesim/turtlesim_node)
/turtlesim2/
  sim (turtlesim/turtlesim_node)

ROS_MASTER_URI=http://localhost:11311

core service [/rosout] found
process[turtlesim1/sim-1]: started with pid [1190]
process[turtlesim2/sim-2]: started with pid [1192]
process[mimic-3]: started with pid [1212]
    
```


ROS: Command line

roslaunch

- Dfn:
Start the multiple ros node
- Usage

```
roslaunch <package> <file.launch>
rosrun turtle_sample turtle.launch
```

turtle.launch

```
<launch>
  <node pkg="turtlesim" name="operator" type="turtle_teleop_key">
    <remap from="/turtle1/cmd_vel"
           to="/turtlesim1/turtle1/cmd_vel"/>
  </node>
  <group ns="turtlesim1">
    <node pkg="turtlesim" name="sim" type="turtlesim_node"/>
  </group>
  <group ns="turtlesim2">
    <node pkg="turtlesim" name="sim" type="turtlesim_node"/>
  </group>
  <node pkg="turtlesim" name="mimic" type="mimic">
    <remap from="input" to="turtlesim1/turtle1"/>
    <remap from="output" to="turtlesim2/turtle1"/>
  </node>
</launch>
```

ROS: Command line

rostopic

- Dfn:
Interact with published topics

- Usage
 - roslaunch list
 - rostopic echo
 - rostopic pub
 - rostopic type

```

astroros@astroros-VirtualBox: ~
astroros@astroros-VirtualBox:~$ rostopic list
/rosout
/rosout_agg
/turtle1/cmd_vel
/turtle1/color_sensor
/turtle1/pose
    
```

```

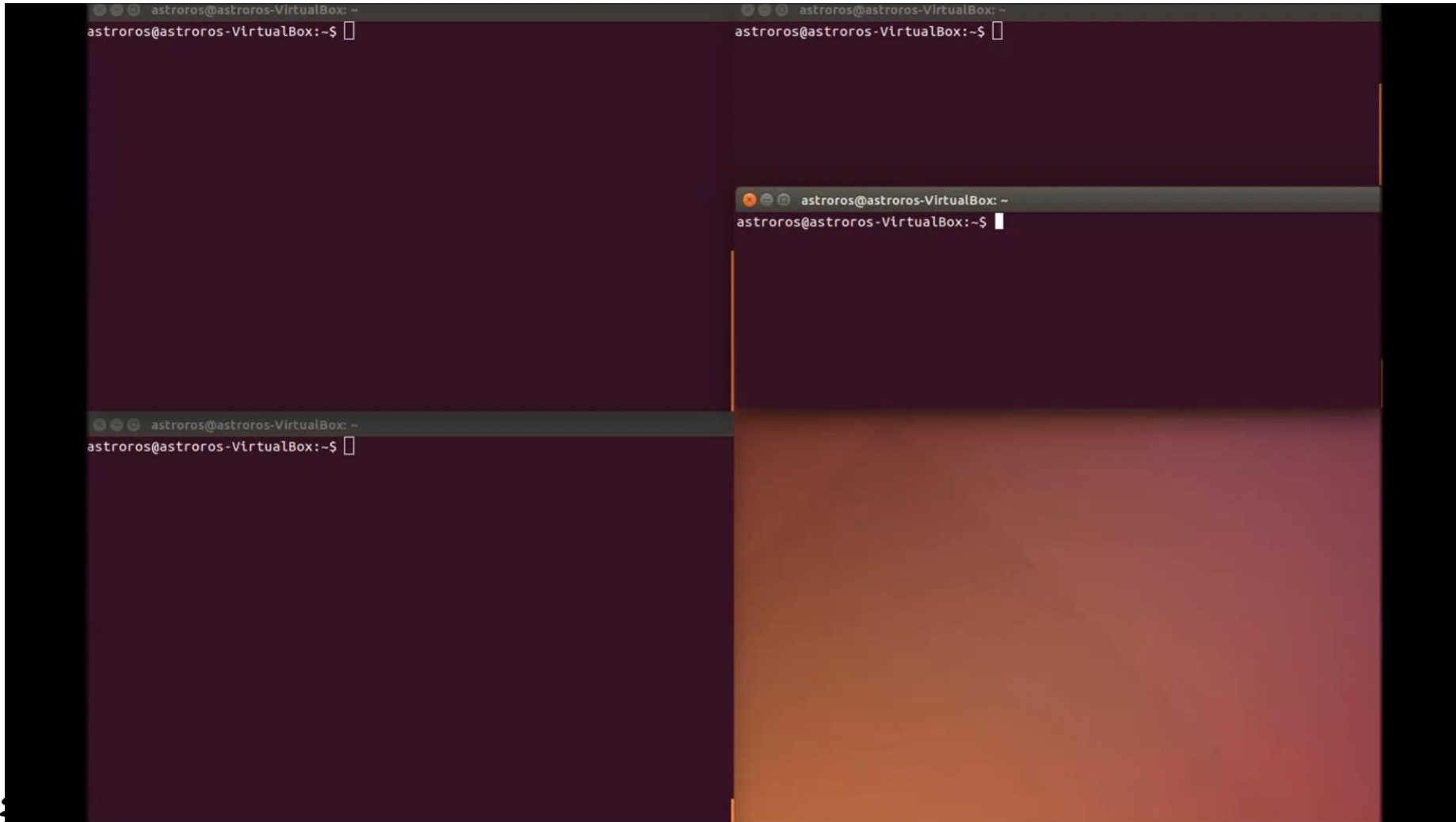
astroros@astroros-VirtualBox: ~
astroros@astroros-VirtualBox:~$ rostopic echo /turtle1/color_sensor
r: 69
g: 86
b: 255
---
r: 69
g: 86
    
```

```

astroros@astroros-VirtualBox: ~
astroros@astroros-VirtualBox:~$ rostopic pub /turtle1/color_sensor turtlesim/Color
{r:10,g:0,b:12}
publishing and latching message. Press ctrl-C to terminate

astroros@astroros-VirtualBox:~$ rostopic pub /turtle1/cmd_vel geometry_msgs/Twist '
{linear: {x: -5, y: 0.0, z: 0.0}, angular: {x: 0.0,y: 0.0,z: 0.0}}'
publishing and latching message. Press ctrl-C to terminate
    
```

ROS: Command line



ROS: Sample Python publisher

```
#!/usr/bin/env python
# license removed for brevity
import rospy
from geometry_msgs.msg import Twist

def talker():
    pub = rospy.Publisher('/turtle1/cmd_vel', Twist)
    rospy.init_node('Turtle_Manager', anonymous=True)
    r = rospy.Rate(10) # 10hz
    while not rospy.is_shutdown():
        twist = Twist()
        twist.linear.x = 0.1;
        rospy.loginfo(twist)
        pub.publish(twist)
        r.sleep()

if __name__ == '__main__':
    try:
        talker()
    except rospy.ROSInterruptException: pass
```

ROS: Sample C++ publisher

```
#include "ros/ros.h"
#include <geometry_msgs/Twist.h>
#include <sstream>

int main(int argc, char **argv)
{
    ros::init(argc, argv, "Turtle_Manager");
    ros::NodeHandle n;
    ros::Publisher turtle_pub = n.advertise<geometry_msgs::Twist>("/turtle1/cmd_vel", 1000);
    ros::Rate loop_rate(10);

    int count = 0;
    while (ros::ok())
    {
        geometry_msgs::Twist base_cmd;
        base_cmd.linear.x = 0.1;
        msg.data = ss.str();
        ROS_INFO("%s", base_cmd.str());

        turtle_pub.publish(base_cmd);

        ros::spinOnce();
        loop_rate.sleep();
        ++count;
    }
    return 0;
}
```

ROS: Distribution

- Package

Contains nodes and libraries that logically constitutes a useful module

- Composition

/src	cpp source files
/scripts	python source files
/msg	message definition files
/srv	service definition files
/launch	launch files
package.xml	package manifests (metadata about package)
CMakeLists.txt	build and installation instruction

```

├── CMakeLists.txt
├── launch
│   ├── turtle-cmd.launch
│   └── turtle.launch
├── msg
│   └── coord.msg
├── package.xml
├── scripts
│   └── turtleMgt.py
├── src
│   └── turtleMgt.cpp
├── srv
│   └── sum.srv
└──
    
```

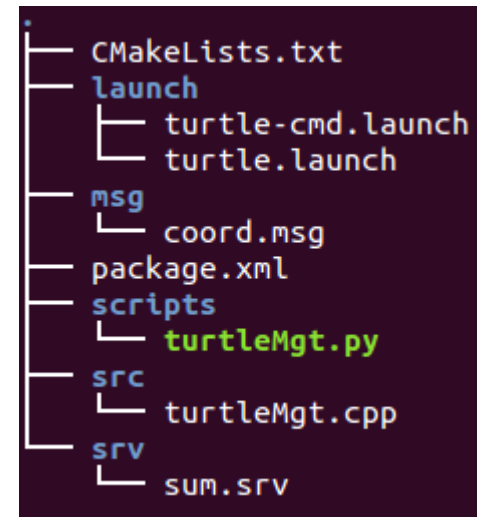
ROS: Distribution

CMakeLists.txt (1/2)

```

cmake_minimum_required(VERSION 2.8.3)
project(turtle_sample)

find_package(catkin REQUIRED COMPONENTS
  roscpp
  rospy
  std_msgs
  message_generation
)
## Generate messages in the 'msg' folder
add_message_files(
  FILES
  coord.msg
)
## Generate services in the 'srv' folder
add_service_files(
  FILES
  sum.srv
)
## Generate added messages and services with any dependencies listed here
generate_messages(
  DEPENDENCIES
  std_msgs
)
    
```

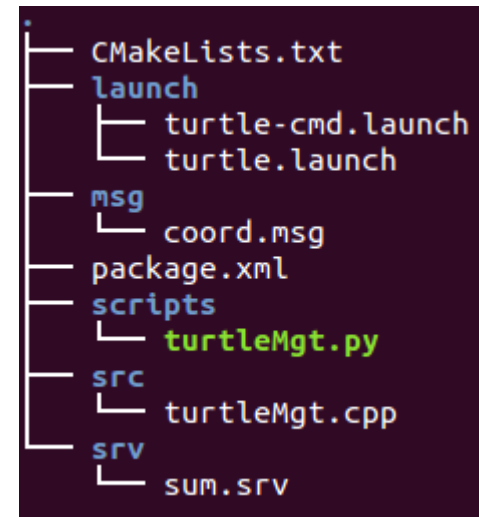


ROS: Distribution

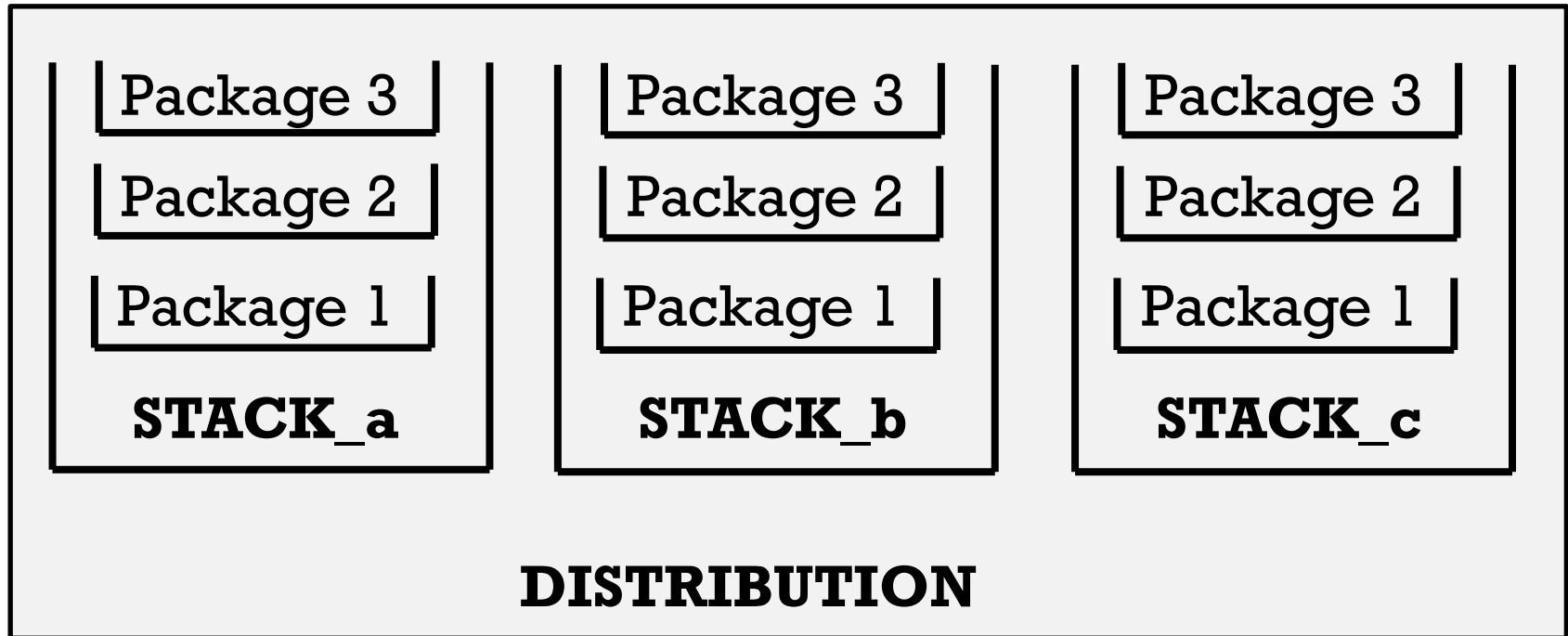
CMakeLists.txt (2/2)

```

catkin_package(
  CATKIN_DEPENDS roscpp
  rospy
  std_msgs
  message_runtime
)
include_directories(
  ${catkin_INCLUDE_DIRS}
)
## Declare a cpp executable
add_executable(turtleMgt src/turtleMgt.cpp)
## Add cmake target dependencies of the executable/library
add_dependencies(turtleMgt
  turtle_sample_generate_messages_cpp)
## Specify libraries to link a library or executable target
against
target_link_libraries(turtleMgt
  ${catkin_LIBRARIES}
)
    
```



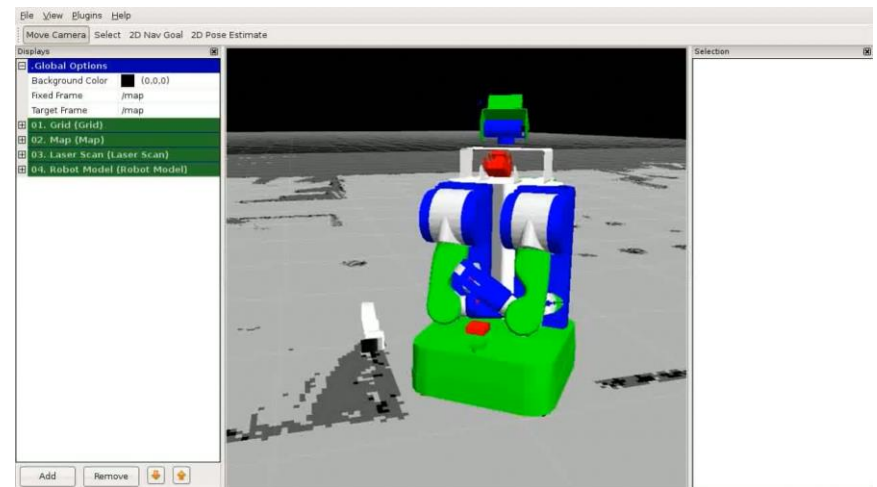
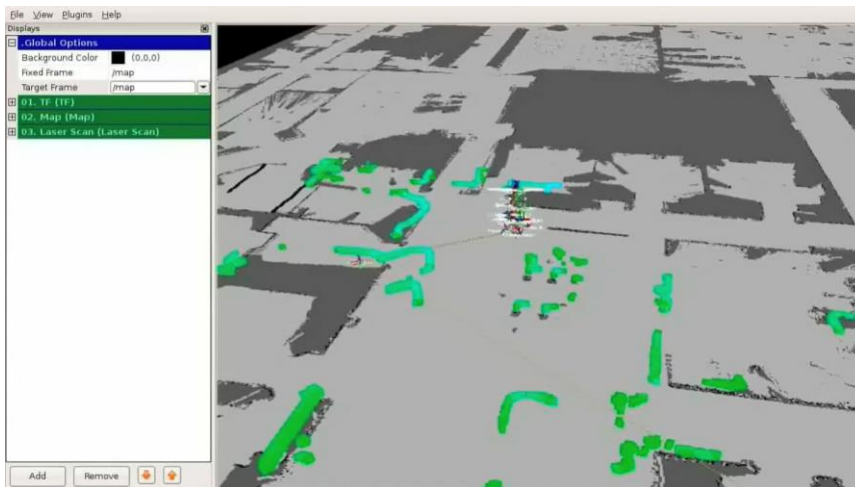
ROS: Distribution



ROS: Toolbox

RVIZ

```
$ rosrn rviz rviz
```



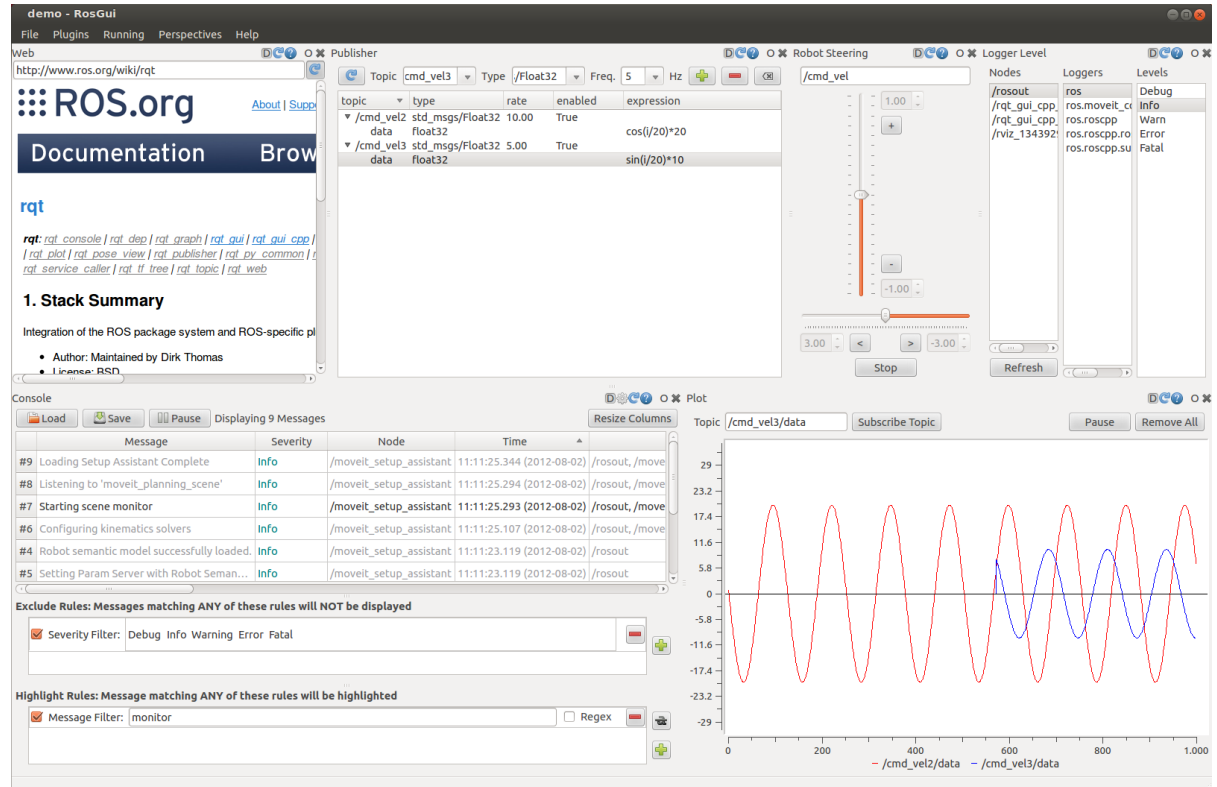


ROS: Toolbox

RQT Software framework implementing various GUI tools in the form of plug-in

Usage

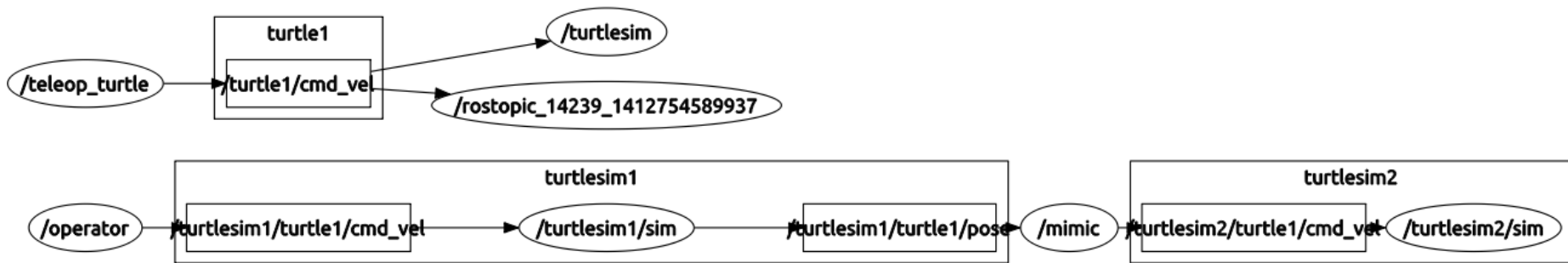
```
rqt_graph
rqt_image_view
rqt_logger_level
rosrun rqt_gui rqt_gui
```



ROS: Toolbox

- Usage

<code>rqt_graph</code>	display a graph of ros Node and topics
<code>rqt_image_view</code>	display available image topics
<code>rqt_logger_level</code>	set Ros log level per ros node
<code>roslaunch rqt_gui rqt_gui</code>	gui container for ros helper tools

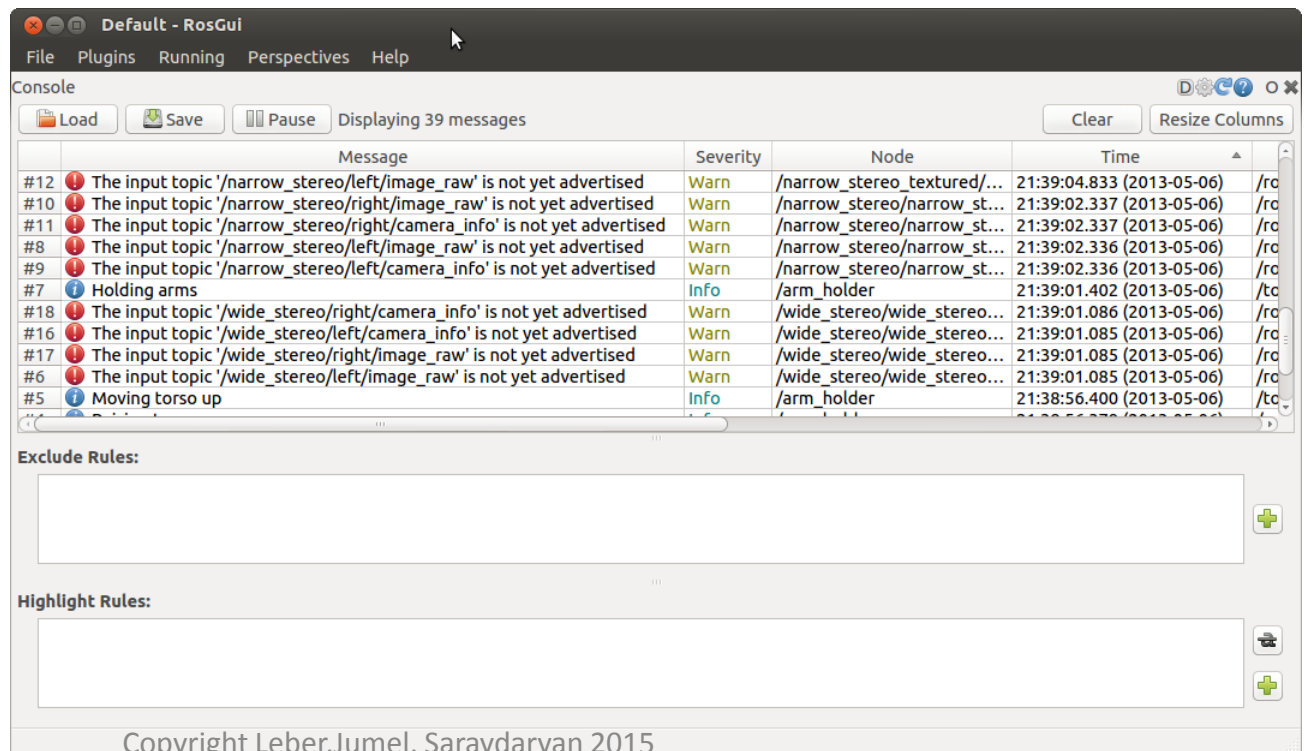
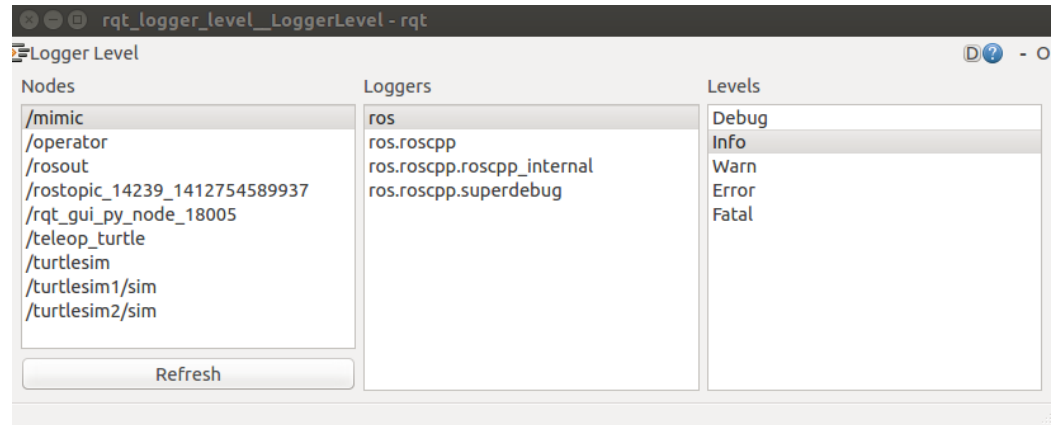




ROS: Toolbox

- Usage for debug

```
rqt_graph
rqt_logger_level
rqt_console
```





ROS: Libraries

About Arm and Gripper Manipulation with ROS; **TF** and **Move it**



KR 30-3

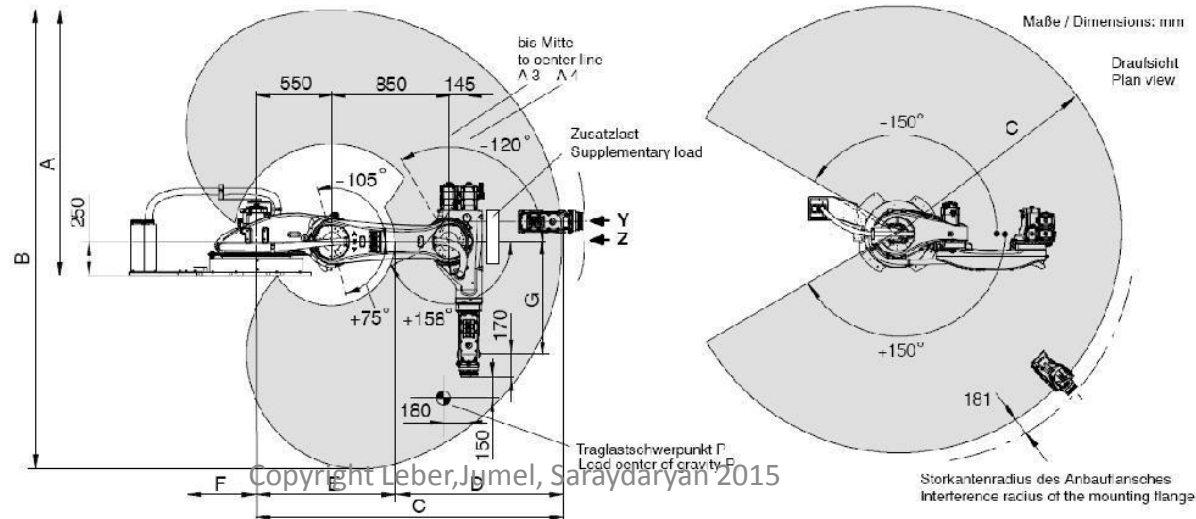
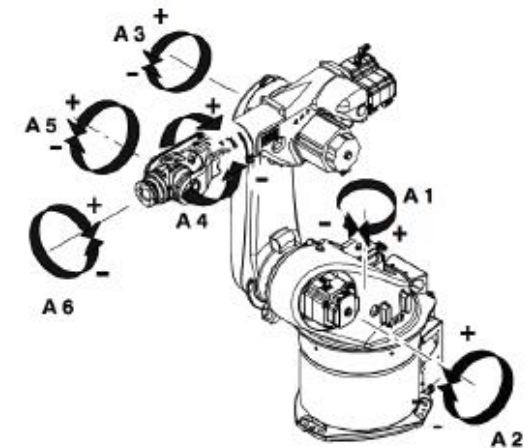
Le KR 30-3, un artiste avec une enveloppe d'évolution sphérique, est l'idéal quand il s'agit d'espace et d'investissements limités.

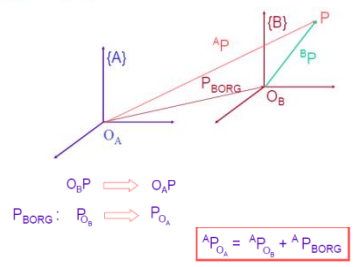
Charges	
Charge admissible	30 kg
Charge supplémentaire	35 kg

Enveloppe d'évolution	
Portée max.	2033 mm

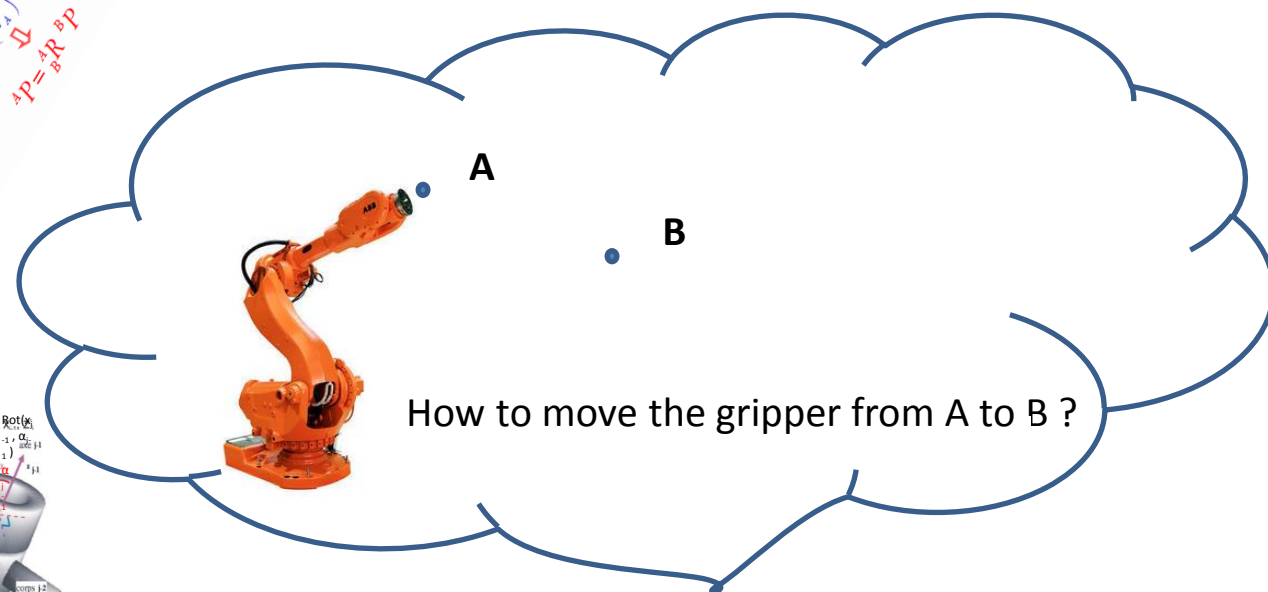
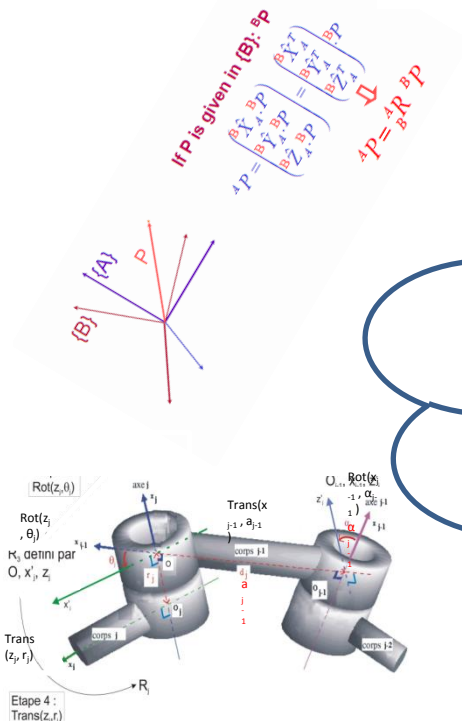
Autres caractéristiques et versions

Nombre des axes	6
Répétabilité	<±0,06 mm
Poids	665 kg
Positions de montage	Sol, plafond
Commande	KR C4





i	α_{i-1}	a_{i-1}	θ_i	r_i
1	0	0	θ_1	0
2	$-\pi/2$	0	$-\pi/2$	r2
3	$-\pi/2$	L_2	θ_3	0
4	$\pi/2$	0	θ_4	0
5	0	L_4	0	L_5



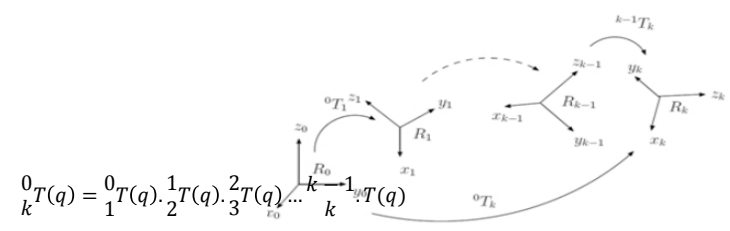
$$q = f^{-1}(x)$$

$$J(q) = \frac{\partial x}{\partial q}$$

$$T_{i-1, i} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_{i-1} & -\sin \alpha_{i-1} & 0 \\ 0 & \sin \alpha_{i-1} & \cos \alpha_{i-1} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{R(z_{i-1}, \alpha_{i-1})} \underbrace{\begin{pmatrix} 1 & 0 & 0 & a_{i-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\text{translation de } a_{i-1} \text{ le } z_{i-1}} \underbrace{\begin{pmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{R(z_i, \theta_i)} \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r_i \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\text{translation de } r_i \text{ le } z_i}$$

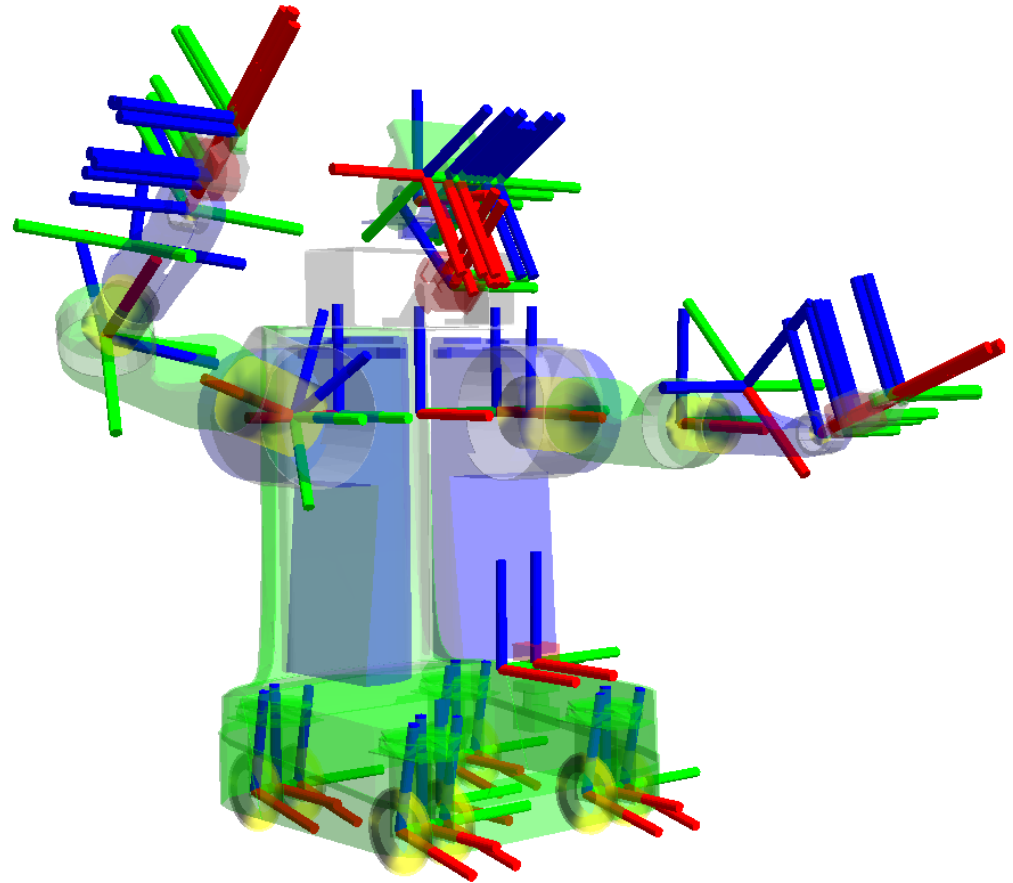
soit :

$$T_{i-1, i} = \begin{pmatrix} \cos \theta_i & -\sin \theta_i & 0 & a_{i-1} \\ \cos \alpha_{i-1} \sin \theta_i & \cos \alpha_{i-1} \cos \theta_i & -\sin \alpha_{i-1} & -r_i \sin \alpha_{i-1} \\ \sin \alpha_{i-1} \sin \theta_i & \sin \alpha_{i-1} \cos \theta_i & \cos \alpha_{i-1} & r_i \cos \alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.1)$$



TF

- Definition:
Move a robot arm to a given position
- Properties
 - register object coordiantes,
 - transform data from one landmark to another
 - distributed architecture
 - timestamp coordinate frame



TF

The screenshot shows a ROS GUI window with a 3D grid visualization and a configuration panel. The configuration panel is titled "Displays" and contains the following settings:


Section	Property	Value
.Global Options	Background Color	(0,0,0)
	Fixed Frame	/base_link
	Target Frame	/base_link
01. Grid (Grid)		

At the bottom of the configuration panel, there are buttons for "Add", "Remove", and two small icons. The 3D grid is a perspective view of a white grid on a black background. At the bottom of the window, there is a "Time" section with the following data:

Time	Value
Wall Time:	1253821171.876786
Wall Elapsed:	21.204925
ROS Time:	1253583718.837125
ROS Elapsed:	20.370129


A "Reset" button is located to the right of the ROS Elapsed time field.

URDF; Unified Robot Description Format



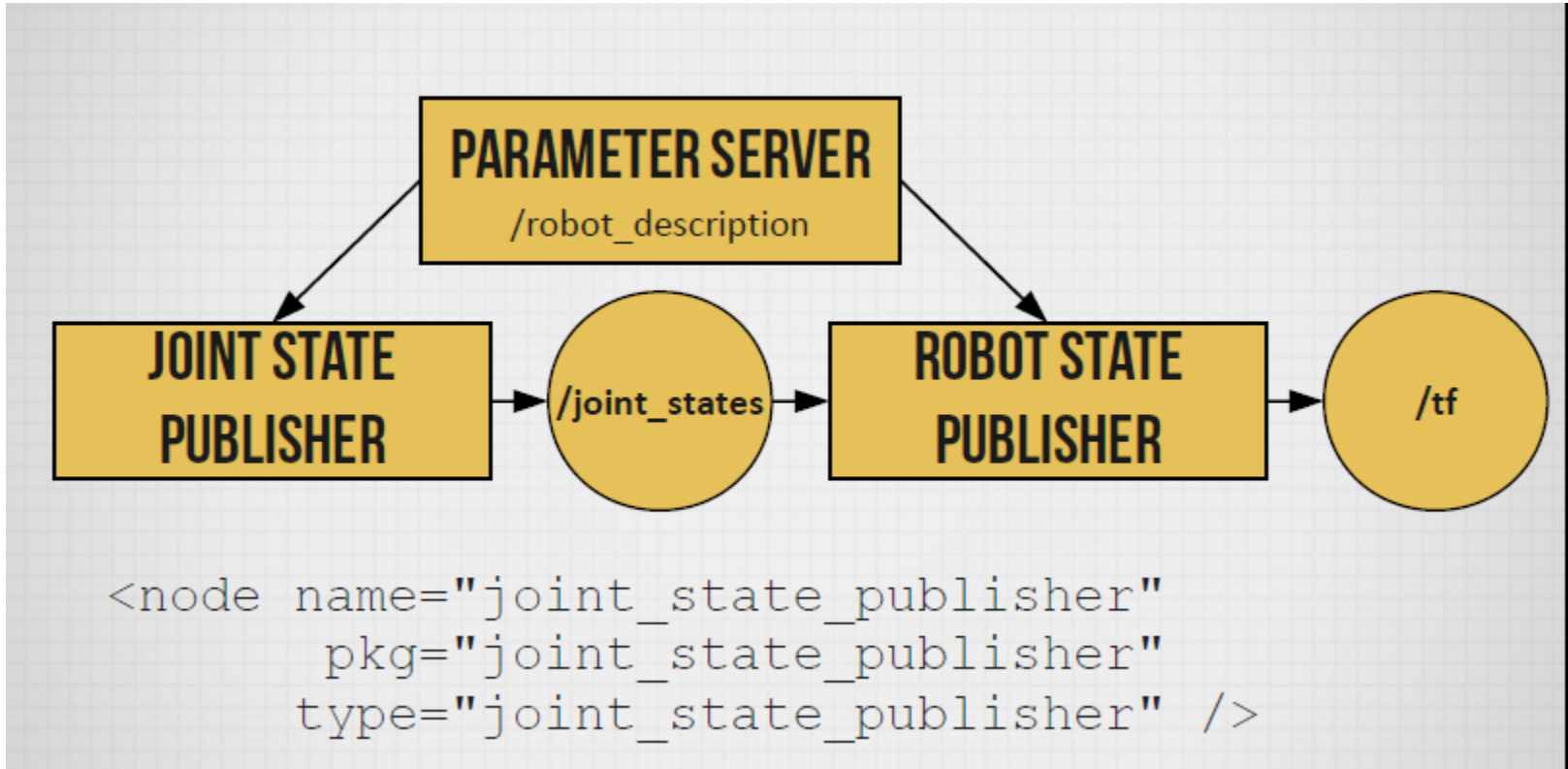
```

<?xml version="1.0" ?>
<robot name="lewis">
  <link name="base">
    <inertial>
      <mass value="1"/>
      <inertia ixx="100" ixy="0"
      <origin/>
    </inertial>
    <visual>
      <origin xyz="0 0 .15"/>
      <geometry>
        <cylinder length="0.3"
        </geometry>
        <material name="lewisred">
          <color rgba="1 0 0 1"/>
        </material>
      </visual>
      <collision>
        <origin xyz="0 0 .15"/>
        <geometry>
          <cylinder length="0.3"
          </geometry>
        </collision>
      </link>
      <link name="body">
        <inertial>
          <mass value="1"/>
    
```

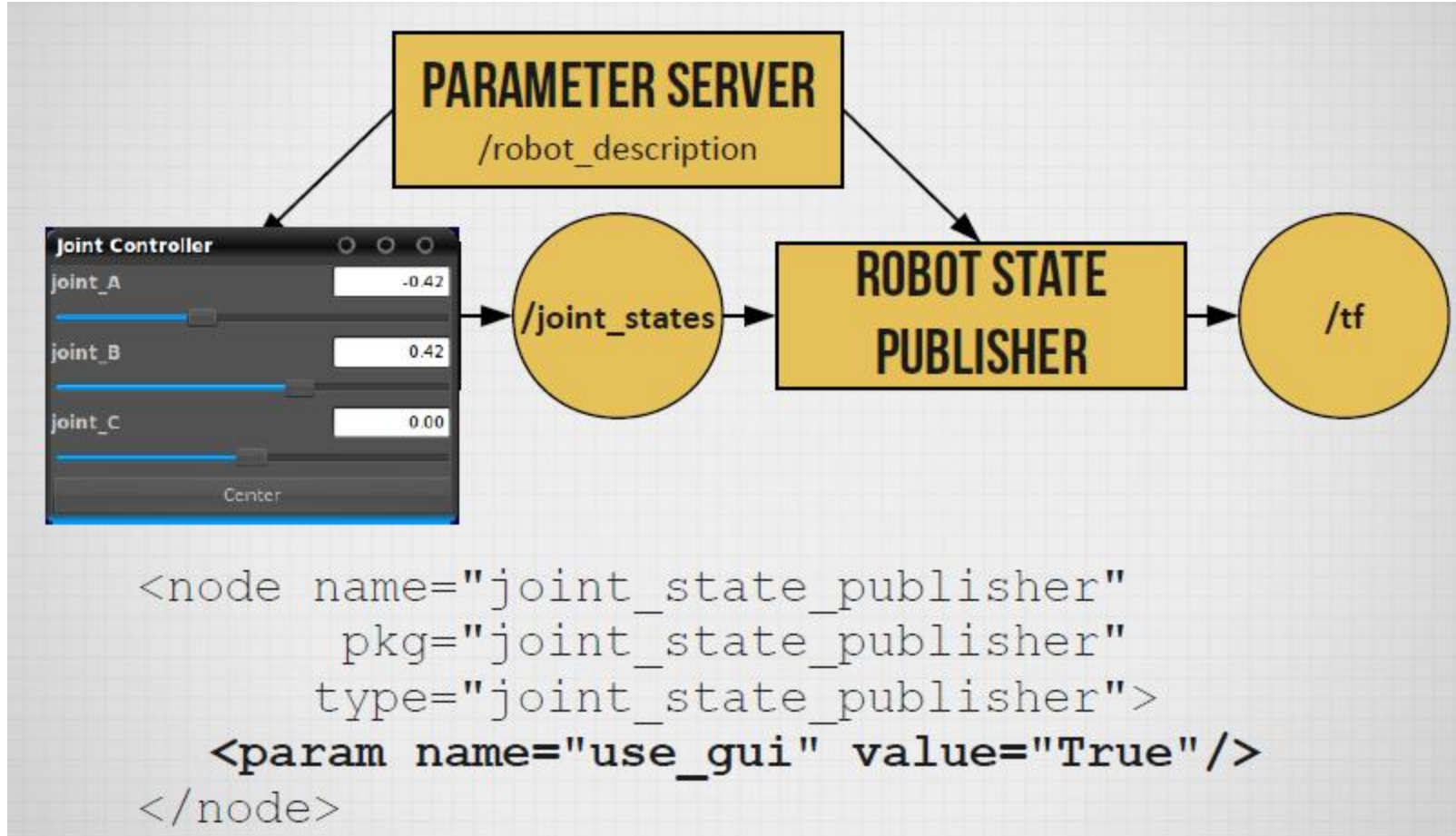


<http://www.ros.org/wiki/urdf>

State Publisher

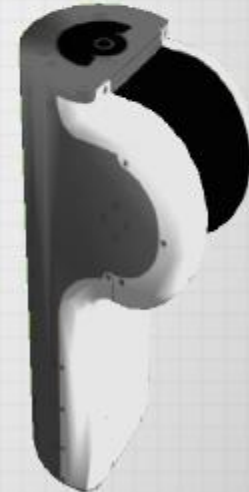
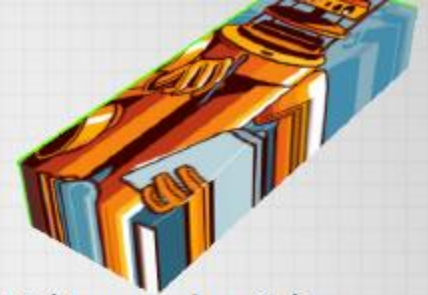


Generated Gui Controller



Model Representation

```
<link name="box">
  <visual>
    <geometry>
      <box size="1.5 .5 .25"/>
    </geometry>
    <material name="pattern">
      <texture filename="package://roscon_urdf/logo.jpg"/>
    </material>
  </visual>
</link>
<link name="mesh">
  <visual>
    <geometry>
      <mesh filename=
        "package://roscon_urdf/shoulder_pan.dae" />
    </geometry>
  </visual>
</link>
```

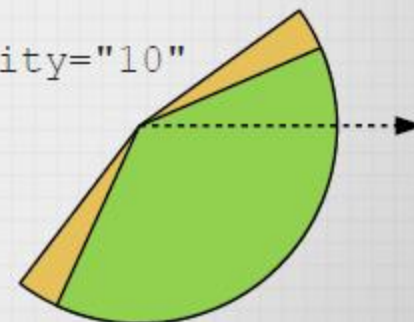


Model Simulation

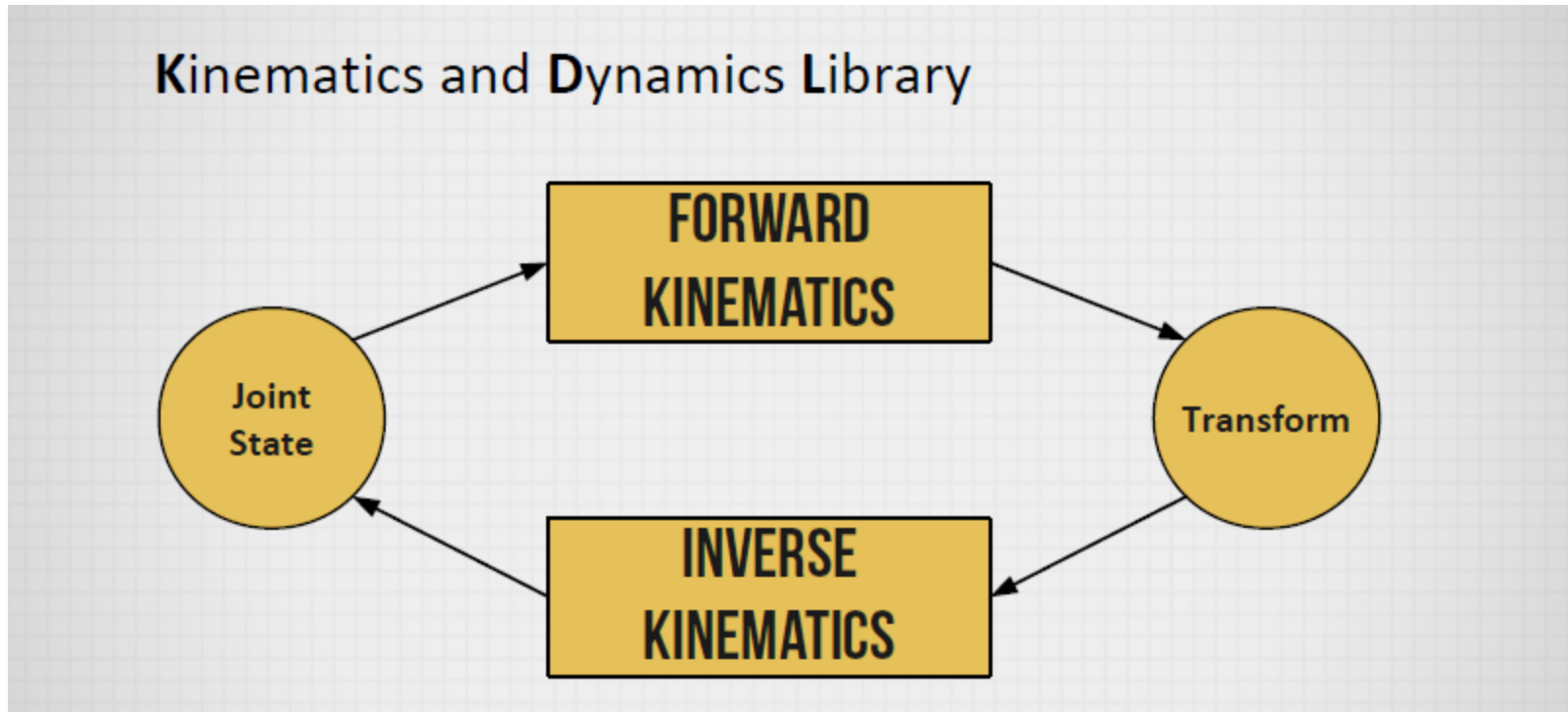
```

<link name="r_shoulder_pan_link">
  <inertial>
    <mass value="25.799322"/>
    <origin xyz="-0.001 .024 -0.098"/>
    <inertia ixx=".866" ixy="-0.061" ixz="-0.121"
             iyy=".874" iyz="-0.059" izz=" .274"/>
  </inertial>
  ...
</link>
<joint name="r_shoulder_pan_joint" type="revolute">
  ...
  <limit effort="30" velocity="2.088"
         lower="-2.28" upper=".71" />
  <safety_controller k_position="100" k_velocity="10"
                    soft_lower_limit="-2.14
                    soft_upper_limit="0.56"/>
  <dynamics damping="10.0"/>
  <calibration rising="-0.785"/>
</joint>
    
```

$$\mathbf{I} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}$$



How move gripper from A to B ?

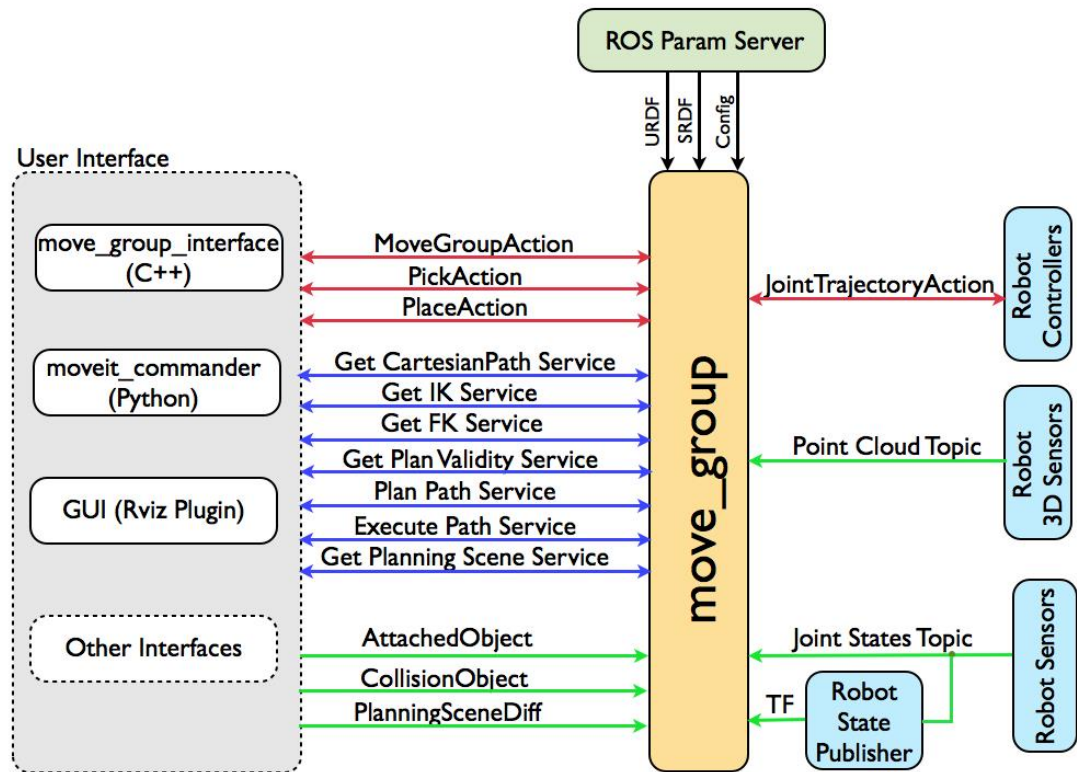


Move it !

- Definition:
Move a robot arm to a given position

- Properties

Inverse kinematics
 motion planning,
 manipulation,
 3D perception,
 kinematics,
 control and navigation
 Collision avoidance



Move it !

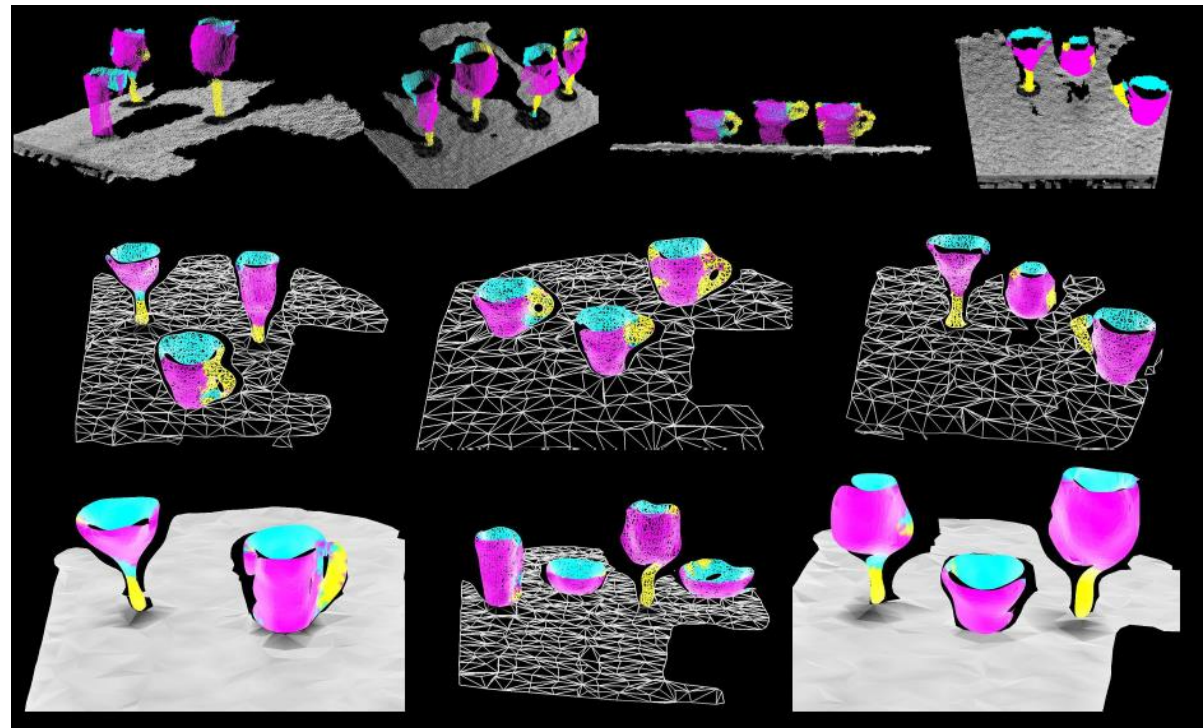


MONTAGE 2013

PCL Point Cloud Library

- Definition:
 - 2D-3D point cloud processing

- Properties
 - Filtering
 - Feature estimation
 - Surface reconstruction
 - Model fitting
 - Segmentation ...



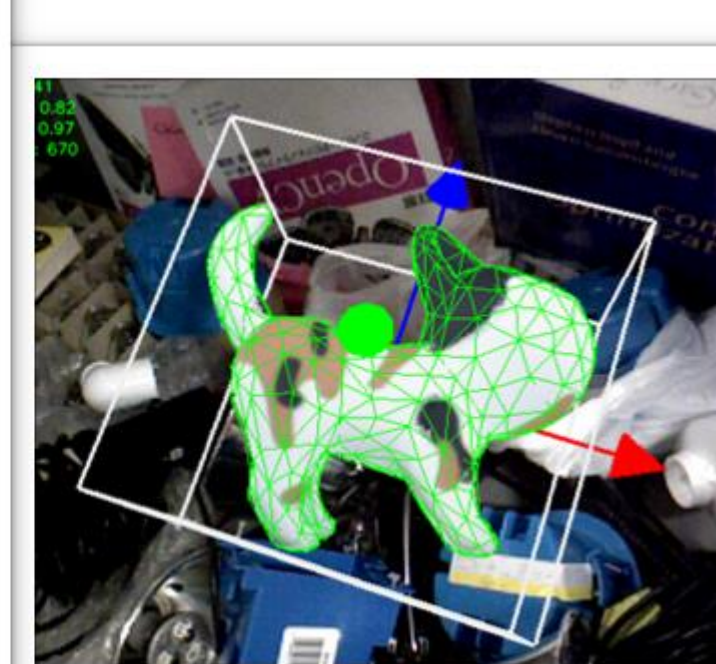
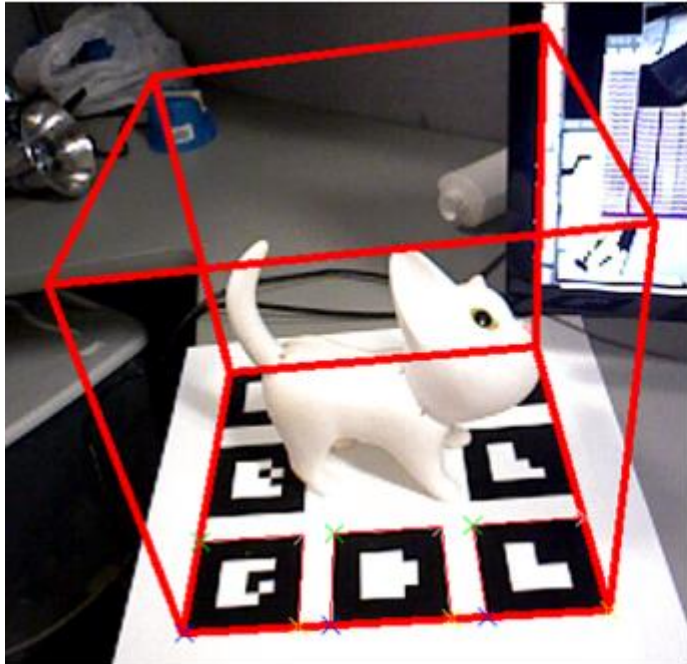
PCL Point Cloud Library



OpenCV



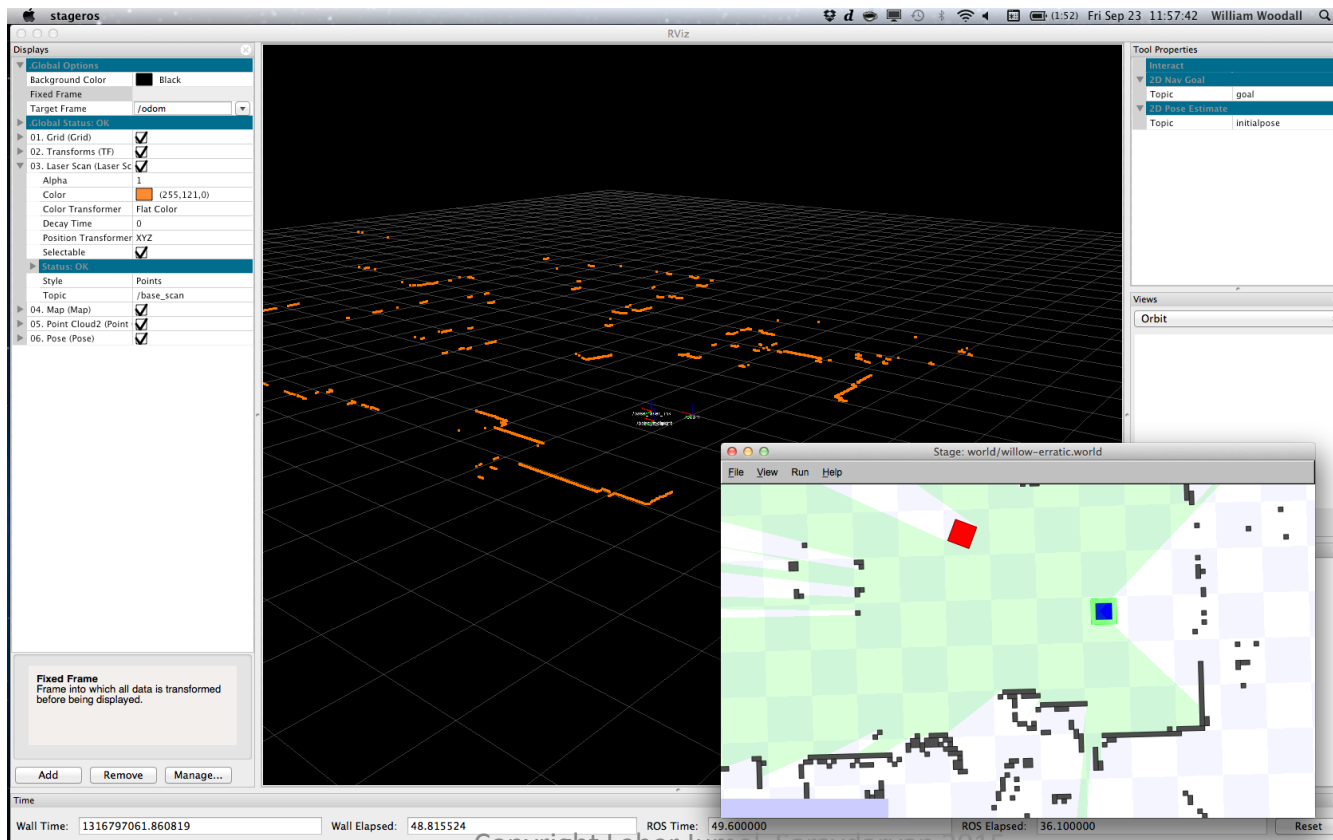
- Definition :
an Open source Computer Vision and machine learning software library



Stage simulator

- Definition :

Stage is a robot simulator. It provides a virtual world populated by mobile robots and sensors, along with various objects for the robots to sense and manipulate.



Gazebo simulator

- Definition:
Gazebo is a simulator of complex robots, probe and environment simulator



Gazebo simulator



Our ROS Feedback

Ros Usage



Robocup@work 2014



Pick Action



Precision place
Action

Smart Robotic Room



People activity follow up



Ros Usage





Pros

- Free and open-source
- Lot of robots and equipments compliant
- Tools box impressive
- Very Active community (lots of examples and new tools)
- More and more mature (stability, easy to use, customization)



Cons

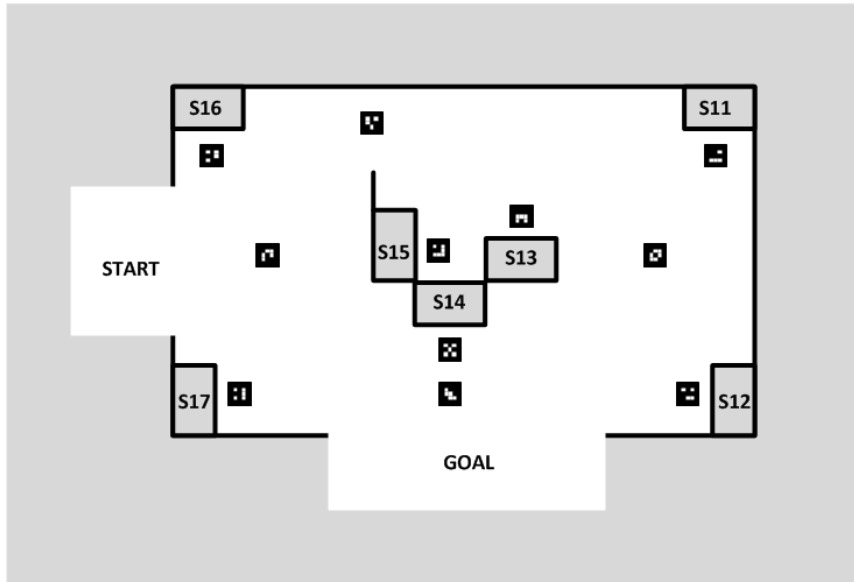
- Inequal contribution maturity
- Currently lot of difficulty to use package between different versions
- Design for a single core on a single computer (need to synchronise time between computers, no official multi-core support)
- Security and reliability need to be evaluated

Robocup@work Usecase

KUKA Youbot Platform



- omni-directional mobile platform
- 5-degree-of-freedom manipulator
- 2-finger gripper
- real-time EtherCAT communication
- open interfaces (ROS compliant)
- basic control software
- arm and platform can be used independently



Working area

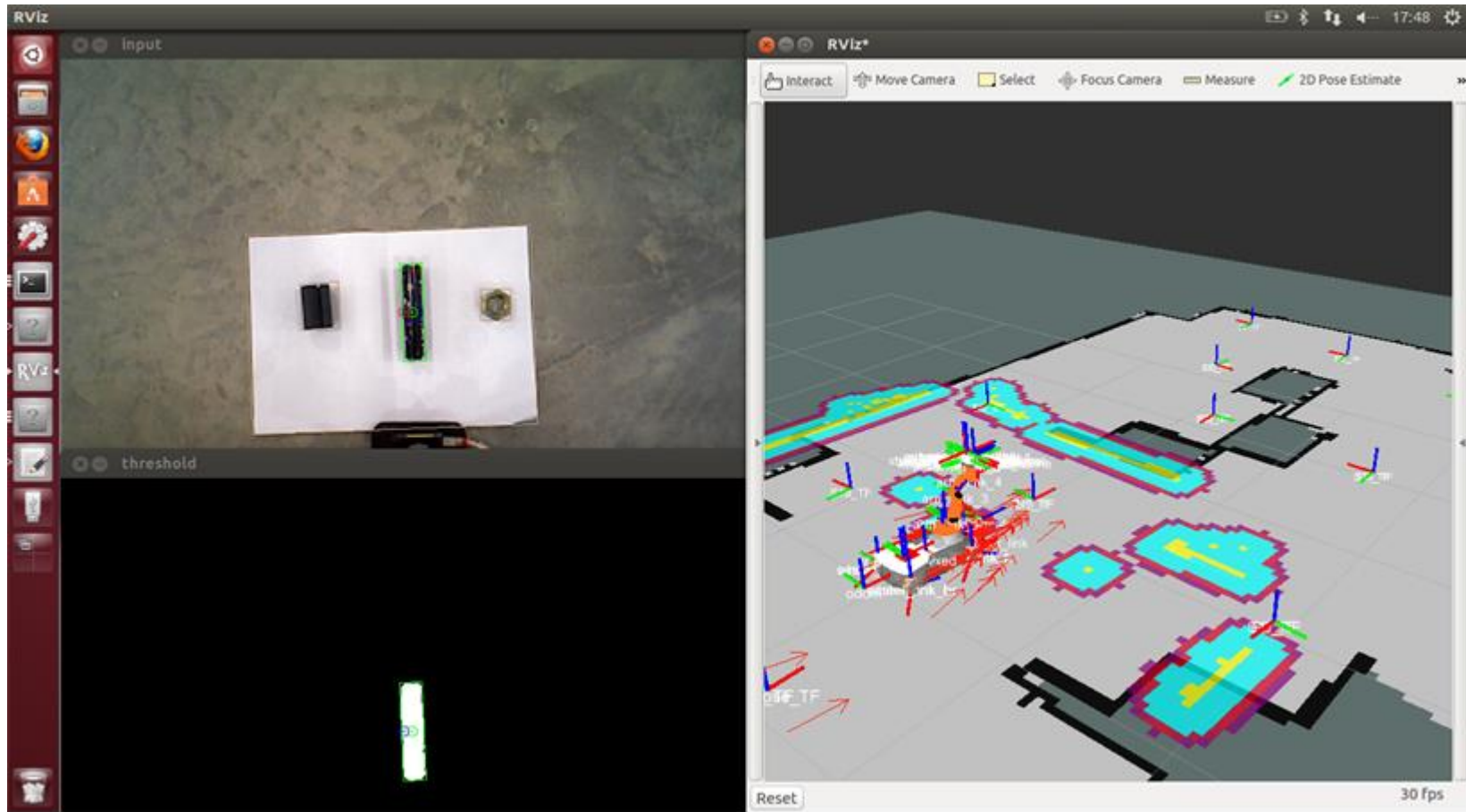


Pick Action

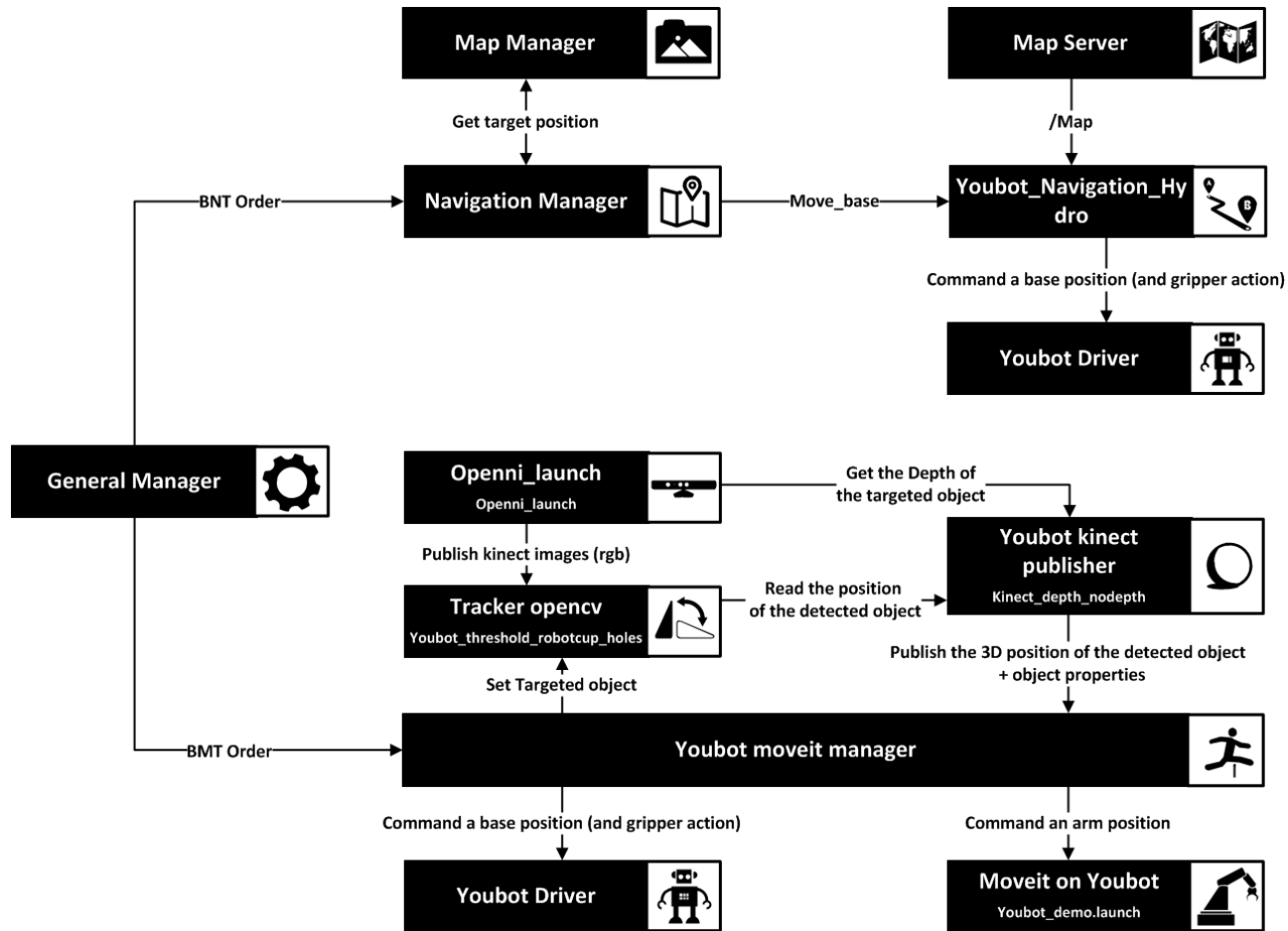


Precision place Action

Robocup@work Usecase



Robocup@work Usecase





References

References

<http://www.ros.org/history/>
<http://wiki.ros.org/Robots>
<http://www.ros.org/about-ros/>
<http://wiki.ros.org/ROS/Tutorials>
<http://wiki.ros.org/rqt>
<http://wiki.ros.org/tf>
<http://wiki.ros.org/ROS/Introduction>
<http://wiki.ros.org/ROS/CommandLineTools>
<http://wiki.ros.org/opencv2>
<http://osrf.github.io/www.ros.org/features/>
<http://www.cnblogs.com/kekec/archive/2011/06/20/2085035.html>
<http://moveit.ros.org/documentation/concepts/>
<http://code.opencv.org/projects/gsoc2013/wiki>
<http://playerstage.sourceforge.net/doc/Stage-3.2.1/>

- Ros Workshop , Lorenz Mösenlechner, Technische Universität München, 2012
- Advanced Techniques for Mobile Robotics ROS, Wolfram Burgard, Cyrill Stachniss, Kai Arras, Maren Bennewitz
- Summer course on ROS framework 2013 (Instituto Superior Técnico, Lisbon)

Images

<http://alligatographe.blogspot.fr/2012/12/robocop-verhoeven.html>

<http://www.theoldrobots.org/images3/manufacturing4.JPG>

<http://www.cnblogs.com/kekec/archive/2011/06/20/2085035.html>

<http://williamjwoodall.files.wordpress.com/2011/09/screen-shot-2011-09-23-at-11-57-36-am.png>

<http://wiki.ros.org/tf>

<http://moveit.ros.org/documentation/concepts/>

<http://answers.ros.org/upfiles/13855650235971625.png>

<http://osrf.github.io/www.ros.org/img/rqt.png>

Videos

Ros 5 years

<https://www.youtube.com/watch?v=PGaXiLZD2KQ>

TF

<http://www.youtube.com/watch?v=dQylpsMUQSw>

Gazebo

<https://www.youtube.com/watch?v=RvfuKP5m0w0>

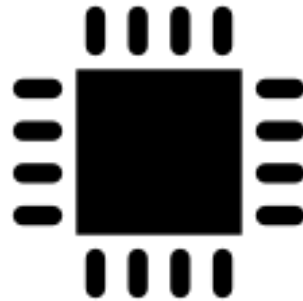
PCL

<https://www.youtube.com/watch?v=7J-9pN96mBY>

moveit

<https://www.youtube.com/watch?v=dbICGZzeUqs>

Hardware solutions for robotics of service



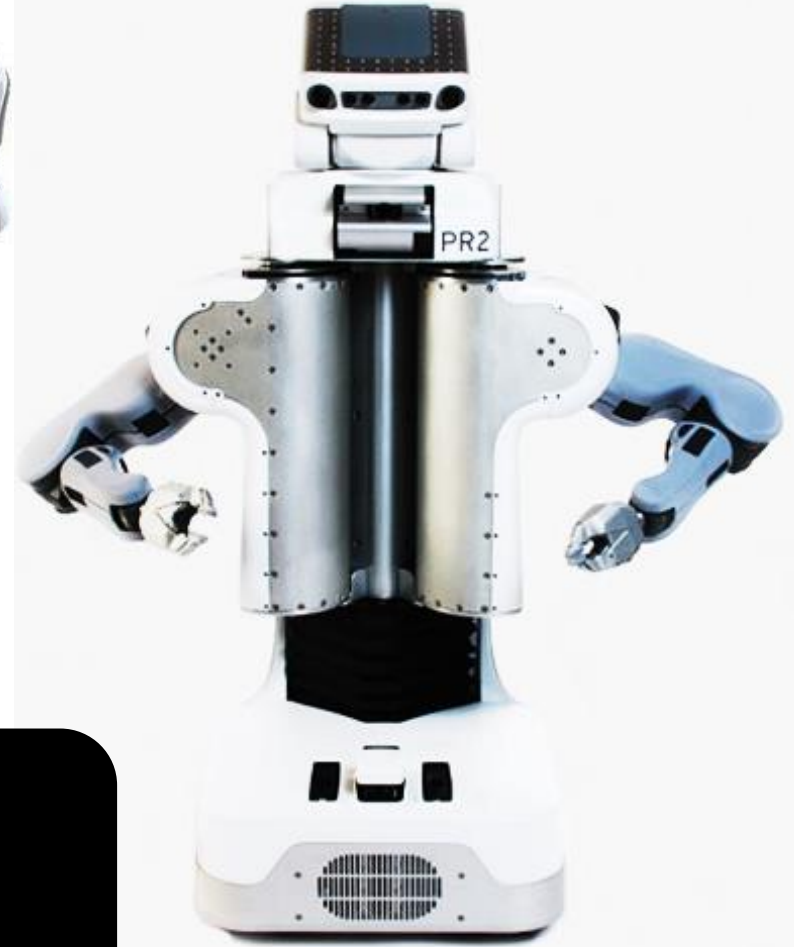


Emblematic Robots

Nao



PR2



Neato



2

1

3

Nao



Nao

Proprioceptive sensors ?



Actuators ?

Exteroceptive sensors ?

Proprioceptive sensors

- MRE (Magnetic Rotary Encoder)
- Temperature sensor
- Current sensors



Exteroceptive sensors

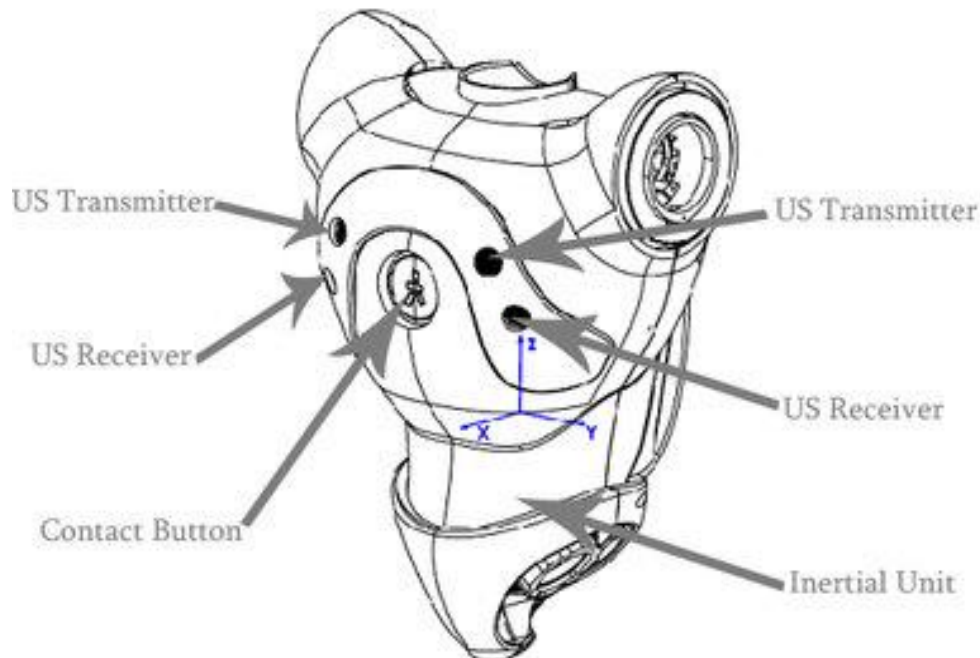
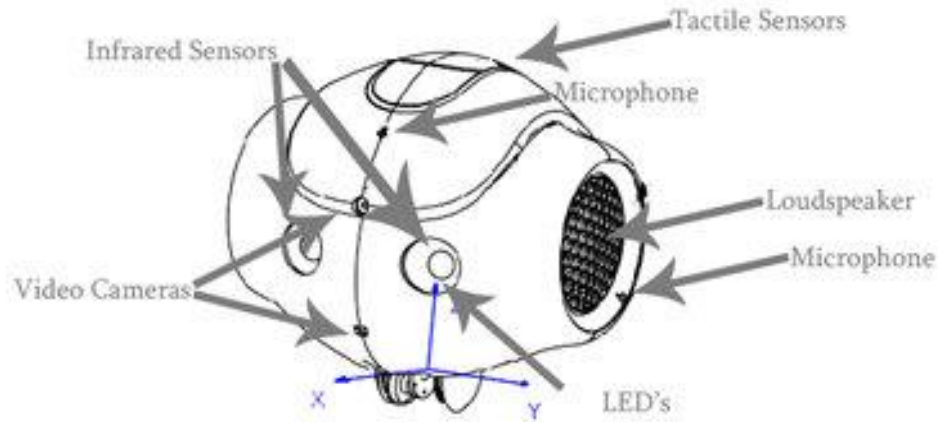
- FSR (Force Sensing Resistor)
- Bumpers
- Inertial Unit (accelero, gyro)
- Sonar
- User button
- Cameras x2
- Microphones x4
- Infrared receiver
- Capacitive sensors



Actuators

- Motors
- LEDs (visible colors)
- LEDs (infrared)
- Loudspeaker

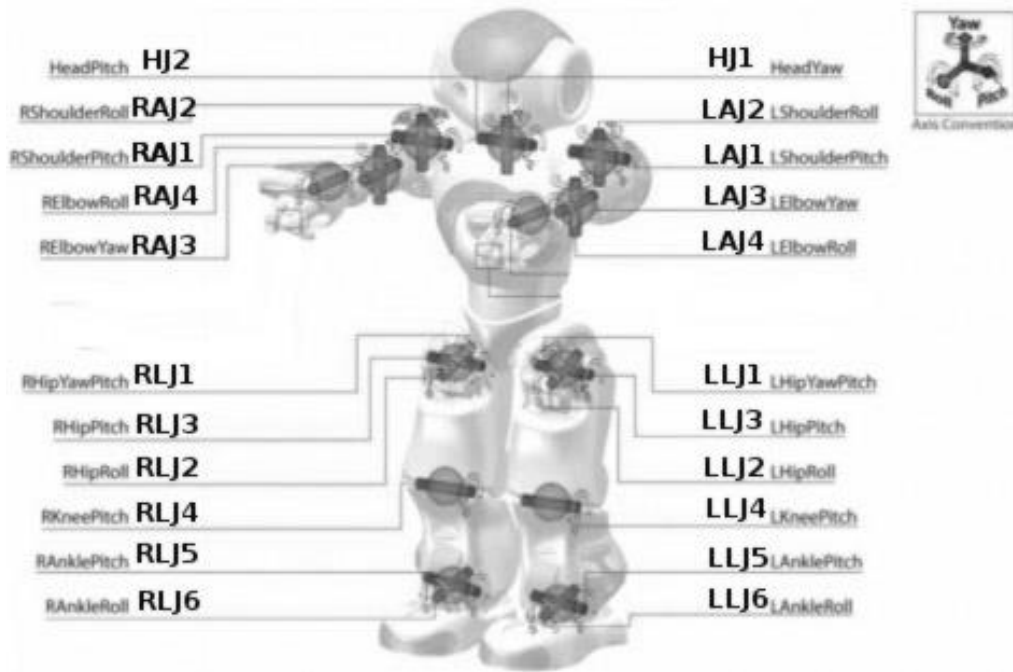




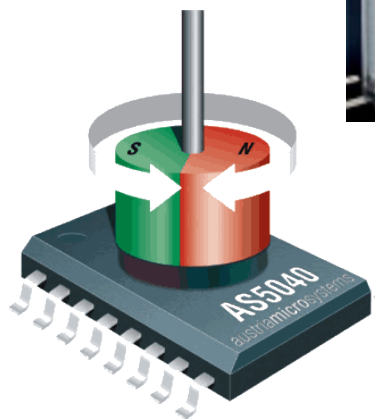
Copyright Leber, Jumel, Saraydaryan 2015

Motors + MRE subset

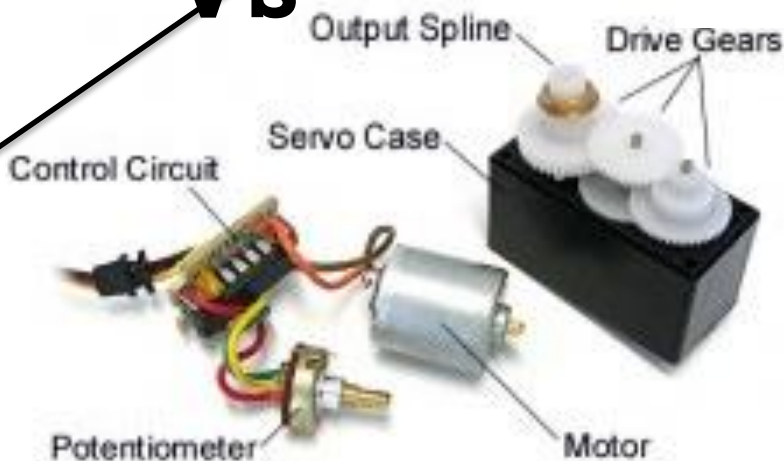
- Nao has 25 motors
- Nao has 34 MRE
- Very good cost/performance/size solution



Motors + MRE subset

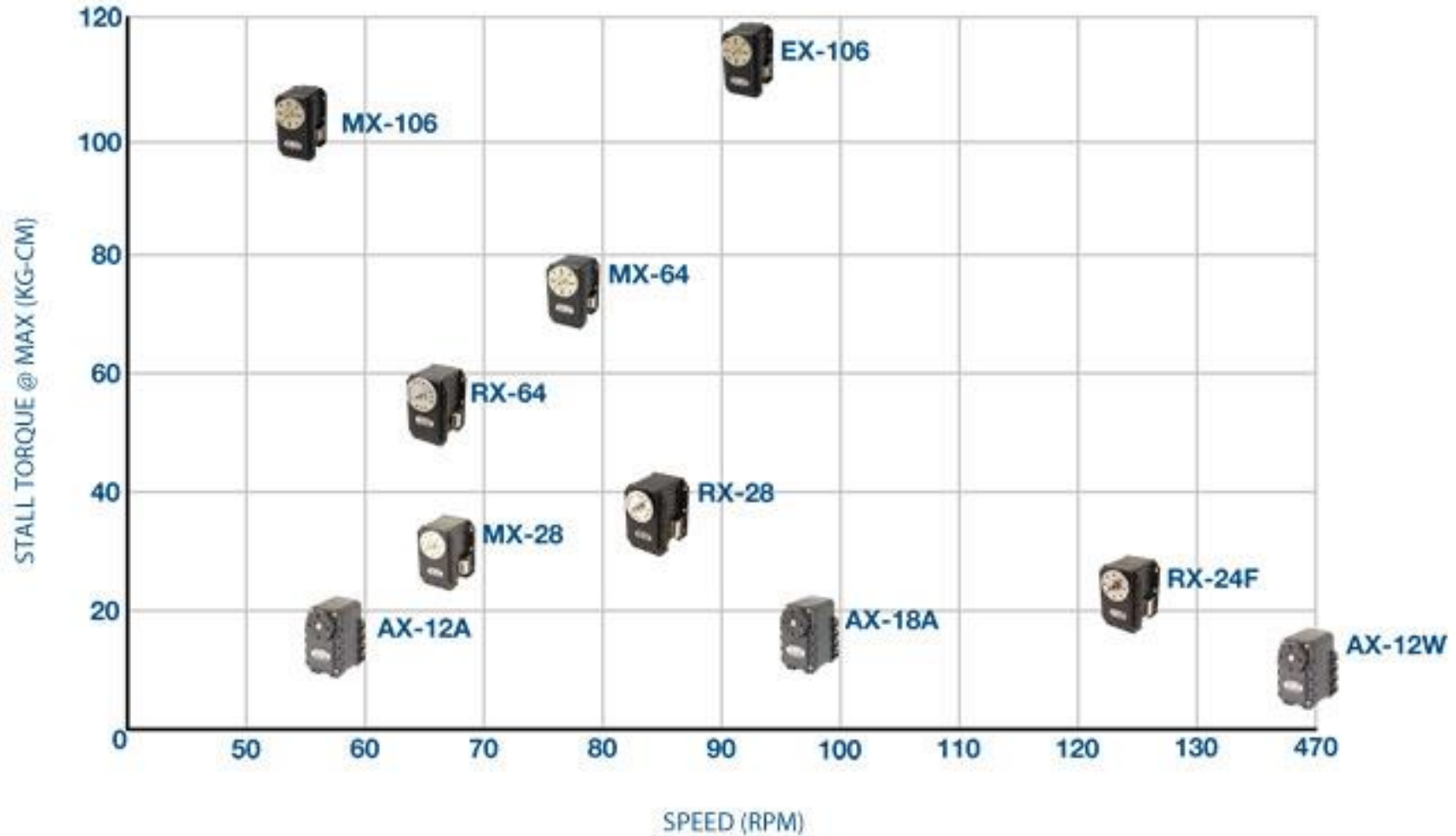


vs



Classical servo motor

Example of dynamixel motors





Motor recommendations in mobile robotics

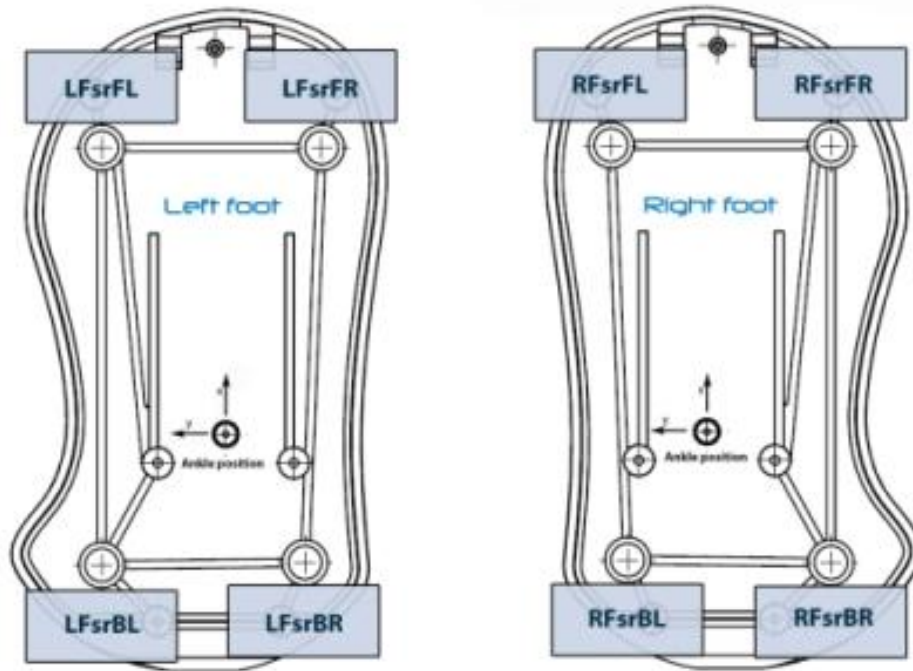
- Brushed (permanent magnets) or Brushless DC motors
 - Optimize size/integration
 - Join effectiveness and simplicity of control
- Consider thermal dissipation
 - If your robot ...
 - ... makes recurrent movements
 - ... makes lots of accelerations
 - ... is position loop-controlled with a « significant » torque to stand
 - Electrothermal model → Choose right product / Use as state observer

Coupling and cables ?

- Moving parts and cables... big issue!
- Two main solutions with many variants:
 - Loop/loops with cables
 - Use of harmonic drives (rather for industrial arm)
- Recommendations:
 - Chose appropriate cables
 - Make endurance tests

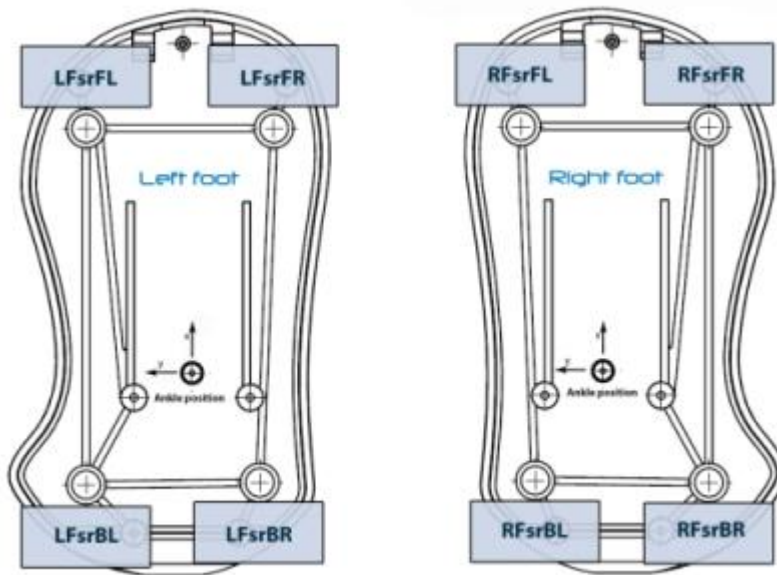


FSR (Force Sensing Resistor)



FSR (Force Sensing Resistor)

- Nao has 4 FSRs per foot
- FSR is the most accessible technologie to measure pressure
- Capacitive technology more appropriate but expensive



Neato



Proprioceptive sensors

- Optical wheel encoders
- Battery « manager »



Exteroceptive sensors

- Sharp (infrared ranger)
- Bumper
- User buttons
- Magnetic band sensor
- Lidar

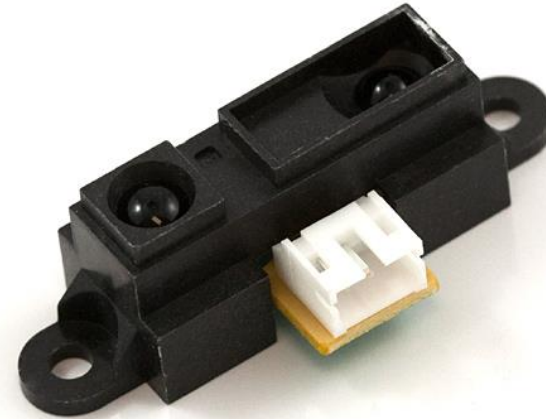
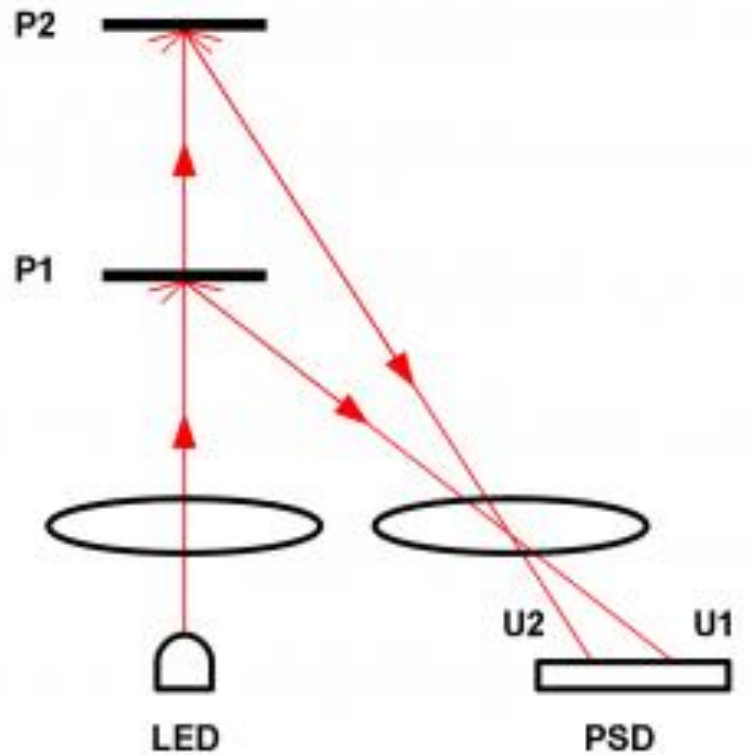


Actuators

- Motors (2 wheels, fan, brush)
- LEDs (visible colors)
- Display

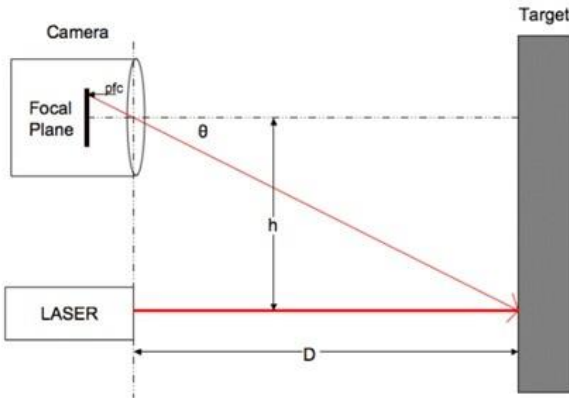


Sharp infrared ranger

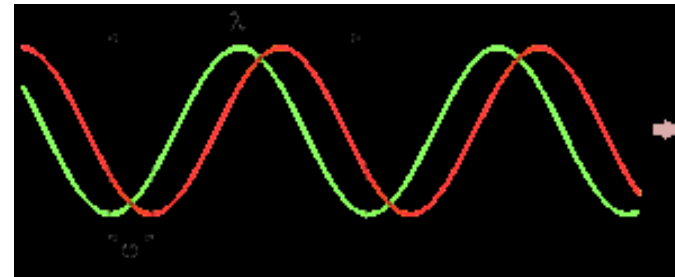


Position sensitive photo-transistors

Lidar (sharp principle)

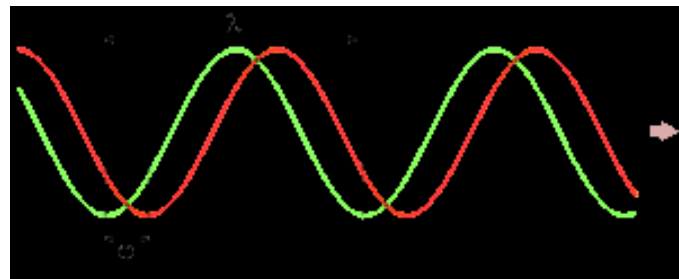
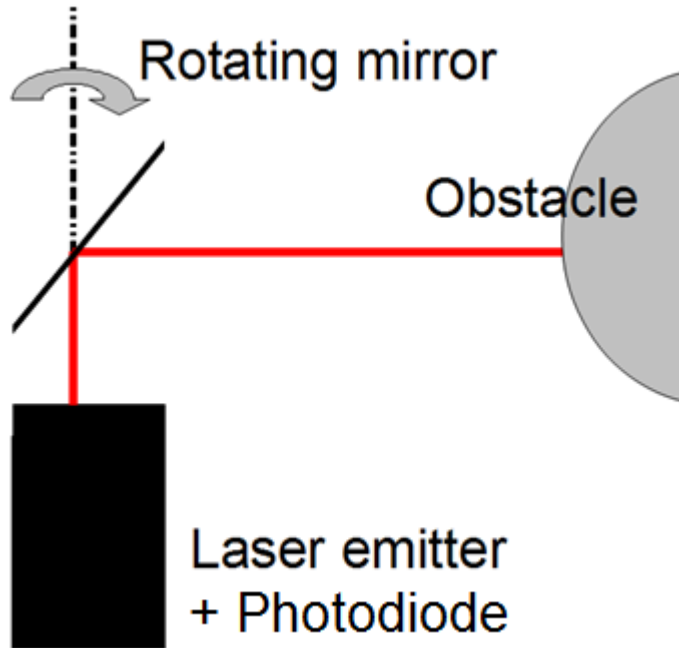


Vs



Lidar (phase difference)

Lidar (phase difference)



$$S(t) = A \cdot \sin(\omega \cdot t)$$

$$R(t) = B \cdot \sin(\omega \cdot (t - dt))$$

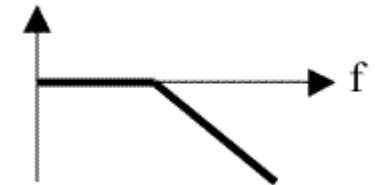
$$M(t) = S(t) \cdot R(t)$$

$$M(t) = A \cdot B \cdot \frac{\cos(\omega \cdot dt) - \cos(2 \cdot \omega \cdot t - \omega \cdot dt)}{2}$$

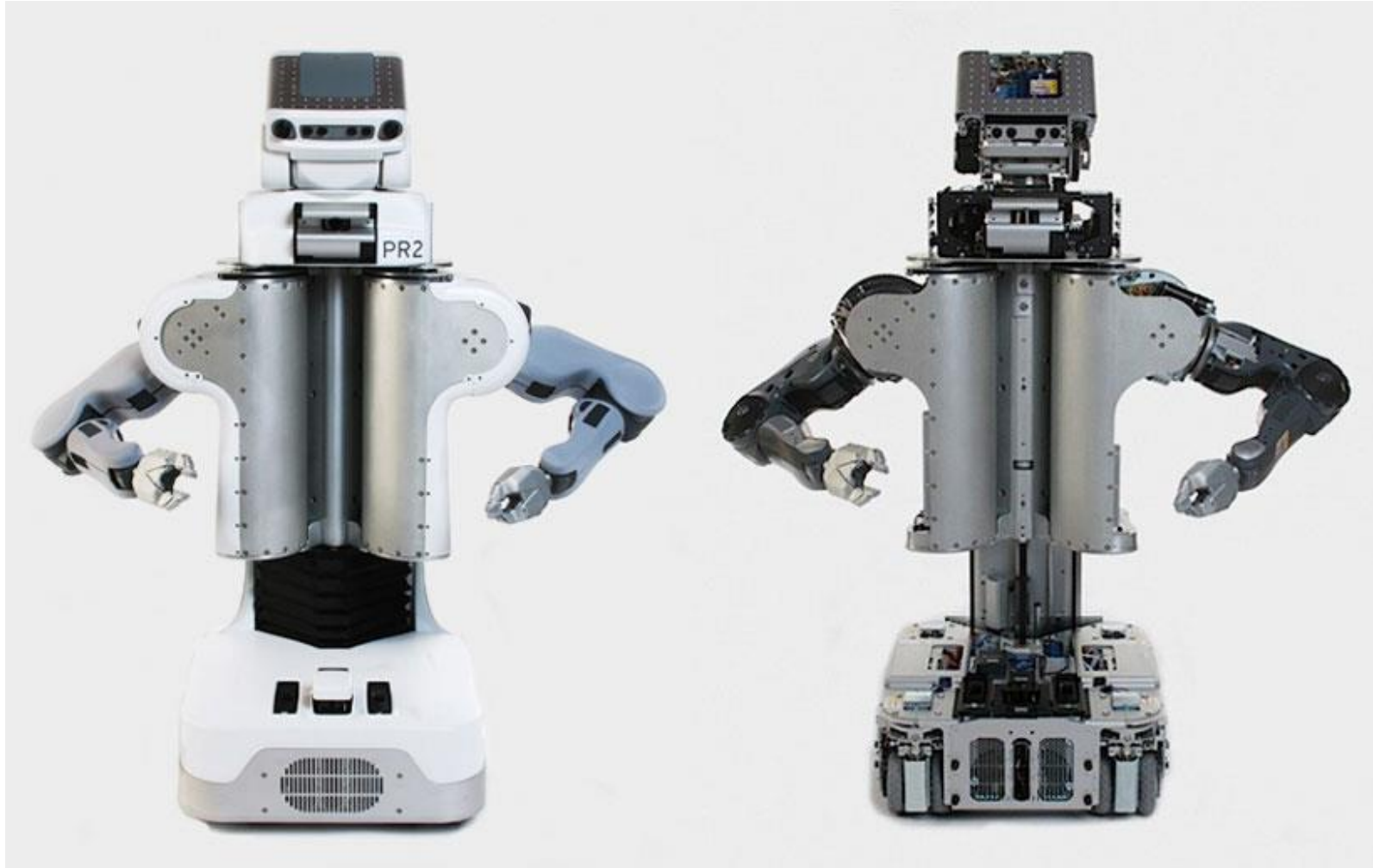
Constant for a certain distance

High Frequency

Extract value with low-pass filter



PR2





Cloth Grasp Point Detection based on Multiple-View Geometric Cues with Application to Robotic Towel Folding

Jeremy Maitin-Shepard
Marco Cusumano-Towner
Jinna Lei
Pieter Abbeel

Department of Electrical Engineering and Computer Science
University of California, Berkeley

International Conference on Robotics and Automation, 2010

3D Cameras

- Stereo camera



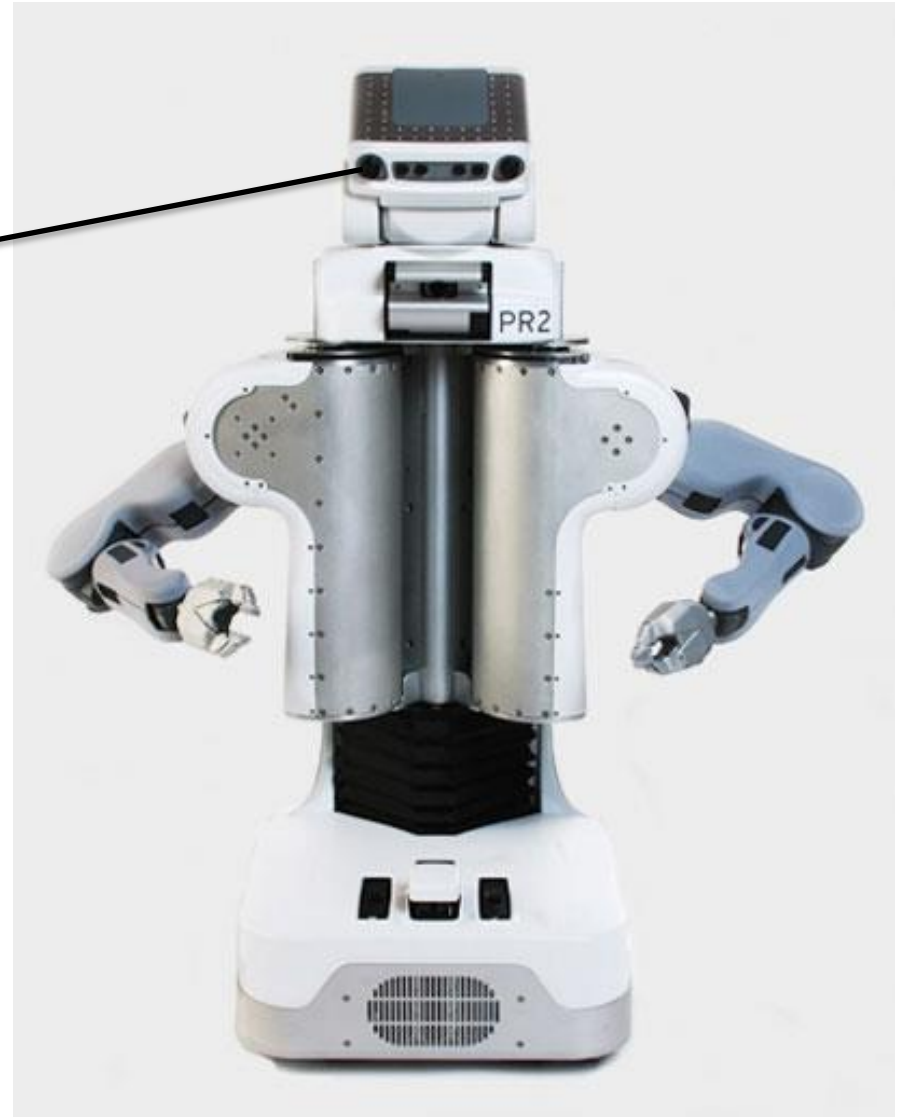
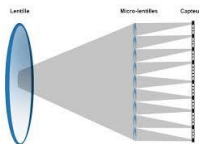
- Structured light camera



- Time of Flight (ToF) camera



- Plenoptic camera



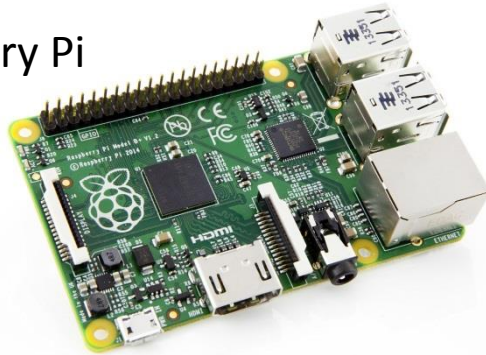
3D Cameras



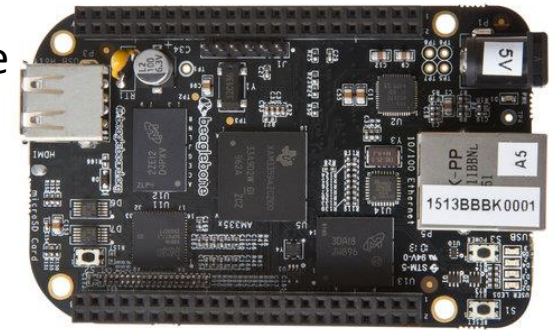
	Range	Width Resolution	Depth Resolution	Outside	Price
Stereo	0<--->0 ~50m	~cm	~cm		
Structured light	~10m	~cm	~cm		
ToF	~50m	~mm	~cm		
Plenoptic	~1m	~mm	->10cm (not linear)		

Robotic process units (prototyping case study)

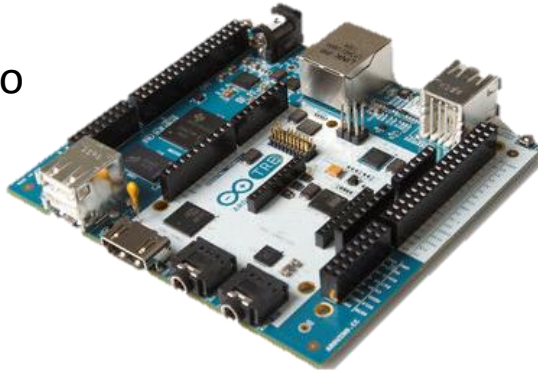
- Raspberry Pi
B+



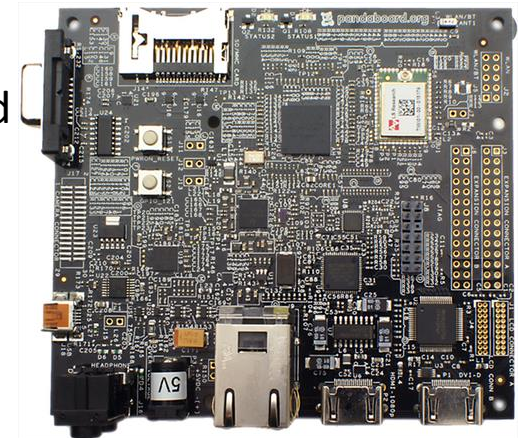
- BeagleBone
Black



- Arduino
Tre

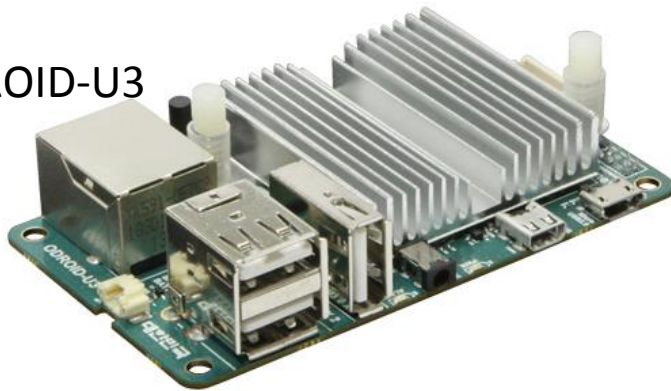


- Panda Board
ES



Robotic process units (prototyping case study)

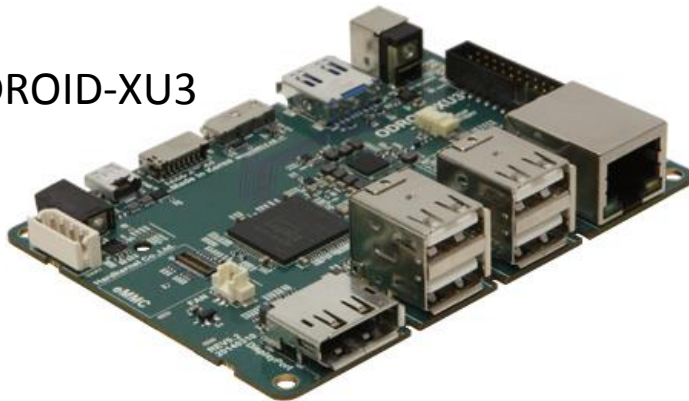
▪ ODROID-U3



▪ UDOO



▪ ODROID-XU3



▪ WandBoard





Robotic process units (prototyping case study)

	Raspberry Pi – B+	Arduino Tree	BeagleBone Black	Panda Board ES	UDOO	Wand Board	ODROID U3	ODROID XU3
CPU	ARM710 700 Mhz	A-8 1 Ghz	A-8 1 Ghz	A-9 1,2Ghz	A-9 Quad 1Ghz	A-9 Quad 1Ghz	A-9 Quad 1,7Ghz	A-15 Quad 2Ghz
RAM	512 MB	512 MB	512 MB		1 GB	2 GB	2 GB	2 GB
Con.	+	++	++	+++	++++	++++	+	++++
Price	32 \$	Coming soon	55 \$	185 \$	135 \$	129 \$	69 \$	179 \$

SEE OUR RANGE OF RESEARCH ROBOTS

FROM **£1,500** INC VAT

Find Out More

Check out the latest arrivals in our **BOOKSTORE**

Books & Magazines >

View our **FAQs & How-to-Do** Guides

Read Now >

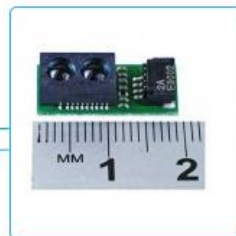
Everything you need for **3D PRINTING**

3D PRINTERS >

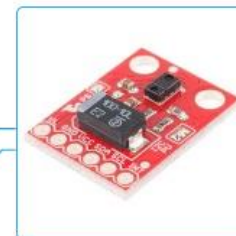
New In Most Popular Special Offers Clearance



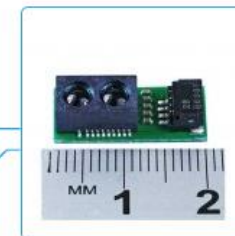
Pressure Sensor Breakout
MS5803-14BA



Sharp GP2Y0E02A
Product Code: GP2Y0E02A



RGB and Gesture Sensor
Product Code: SEN-12787



Sharp GP2Y0E02B
Product Code: GP2Y0E02B



(0) MY CART

- [Products](#) ▾
- [Robot Apps](#) ▾
- [Community](#) ▾
- [Services](#) ▾

Home

RobotShop : The World's Leading Robot Store → [Here's why !](#)

- ### Categories
- [Robots to Build & Experiment](#) >
 - [Robots for the House](#) >
 - [Professional & Service Robots](#) >
 - [Wearable Technology](#) >
 - [RobotShop App Store](#) >

Best Sellers

Choose ▾

On Sale

Choose ▾

New Products

Choose ▾



- Robot Parts**
 - ▶ Microcontrollers
 - ▶ Motors
 - ▶ Sensors[More >](#)
- Robots & Kits**
 - ▶ Robot Construction Kits
 - ▶ Robotic Arms
 - ▶ Wheeled Development Platforms[More >](#)
- Personal & Domestic Robots**
 - ▶ Robot Vacuums
 - ▶ Robot Pet Care
 - ▶ Robot Window Cleaners
- Professional & Service Robots**
 - ▶ Defense, Security & Surveillance Robots
 - ▶ Guide & Telepresence Robots
 - ▶ Entertainment Robots





Génération Robots | HumaRobotics | Blog

Contact us +33 5.56.39.37.05 | Youtube | Facebook | Twitter | Google+

The European personal programmable robot specialist

My account | Subscribe

£ \$ € CHF

Ex: Mindstorm, Kit Gadgeteer, Capteur...

Génération ROBOTS

CART : 0 | Products are in stock!

Consumer Robots | Programmable robots for Education | Electronics and Robotics | Robots for Research | Brands

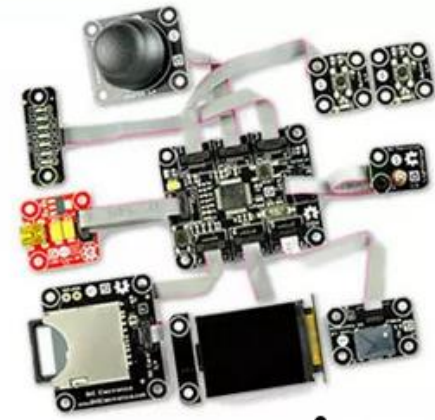
MAKERBOT 3D PRINTERS

What are you going to print today ?



FEZ CERBERUS TINKER KIT

8 module types - 100% open source



ROBOTICS SEMINAR

THE NEW CHALLENGES FOR TOMORROW'S ROBOTS



NEWSLETTER

Stay in touch with robotics and its development. Sign up for our newsletter!

your e-mail...



References

Images

<http://www.intorobotics.com/inside-nao-next-gen-designed-for-future-home-robotic-applications/>

http://home.roboticlab.eu/en/examples/sensor/ir_distance

<http://www.generationrobots.com/en/content/52-localize-with-a-hokuyo-laser-range-finder>

http://www.sensorsportal.com/HTML/DIGEST/april_06/Rotary_Encoder.htm

<http://brooklynhobbies.com/hs-5735mg-digital-servo-hitec-35735>

<http://robocup.bowdoin.edu/public/software/nao/NaoQi/1.10/nao->

[1.10/doc/site_en/reddoc/hardware/FSR.html](http://robocup.bowdoin.edu/public/software/nao/NaoQi/1.10/nao-1.10/doc/site_en/reddoc/hardware/FSR.html)

Videos

<https://www.youtube.com/watch?v=gy5g33S0Gzo>



Check Mark by aguycalledgary from The Noun Project
Gauge by David Waschbüsch from The Noun Project
Cocktail by Okan Benn from The Noun Project
Tools by Mister Pixel from The Noun Project
Lego by jon trillana from The Noun Project
book by Les vieux garçons from The Noun Project
Tree by Grant Taylor from The Noun Project
Caution by Jardson A. from The Noun Project
Globe by Richard Schumann from The Noun Project
Cloud by Edward Boatman from The Noun Project
Question by Jardson A. from The Noun Project
Robot by jon trillana from The Noun Project
Robot by Dylan Thompson from The Noun Project
Robot by Ricardo Moreira from The Noun Project
Robot by William Hollowell from The Noun Project
Sustainable Energy by Simple Icons from The Noun Project
Quote by irene hoffman from The Noun Project



Raphael Leber

raphael.leber@cpe.fr



Fabrice Jumel

fabrice.jumel@cpe.fr



Jacques Saraydaryan

Jacques.saraydaryan@cpe.fr