Program/simulation tools
ROS/Gazebo/OpenHRP

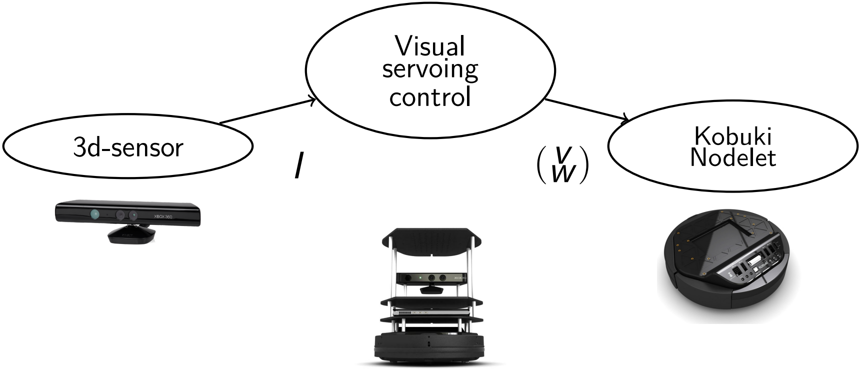
Robotics Principia

24th January 2019, Sophia-Antipolis, France

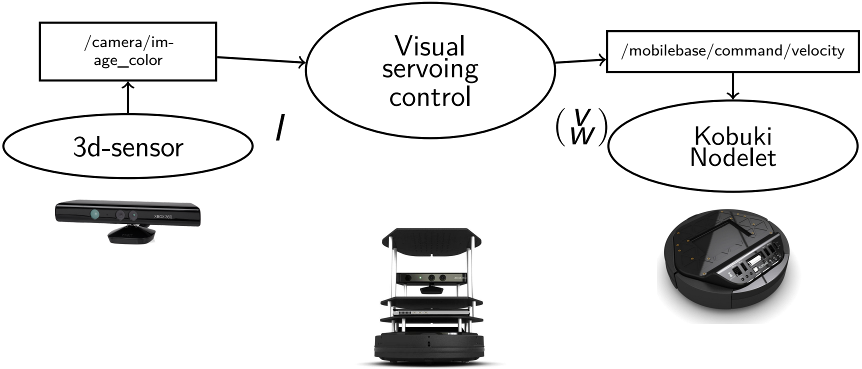
O. Stasse, Gepetto,
LAAS-CNRS

Table of Contents

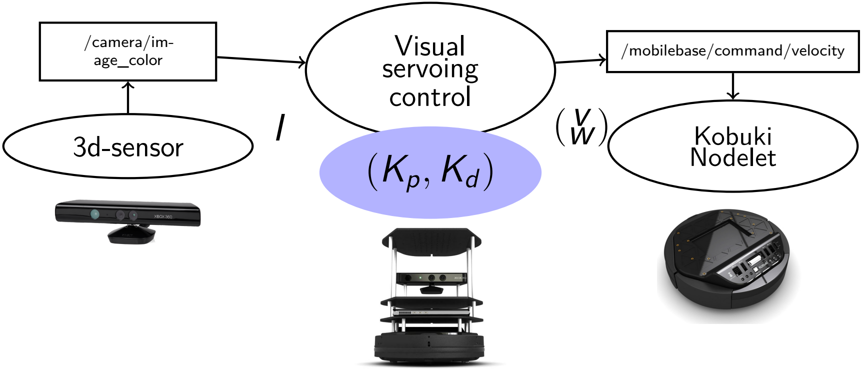
Exemple: Asservissement visuel



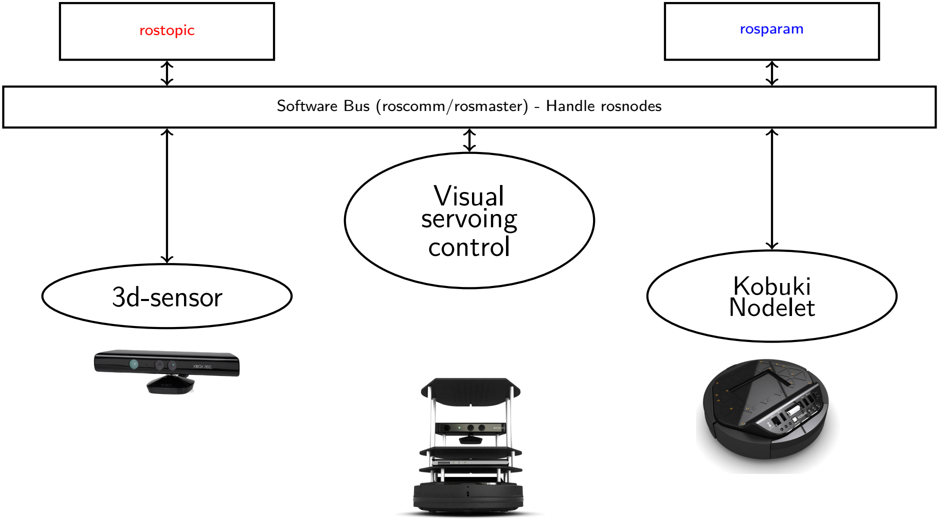
Exemple: Asservissement visuel - Topics



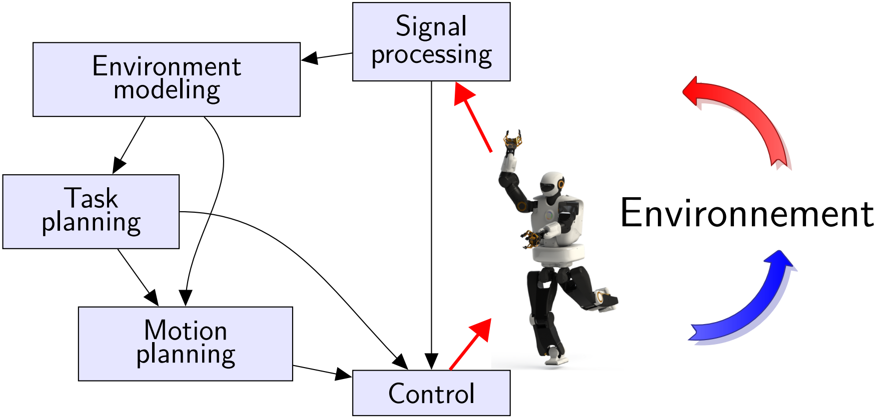
Exemple: Asservissement visuel - Topics - Params



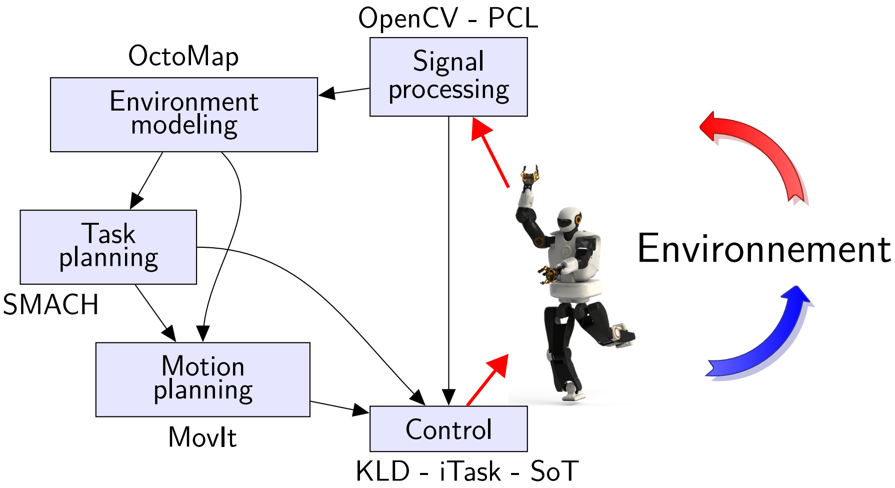
Exemple: Asservissement visuel - Software bus



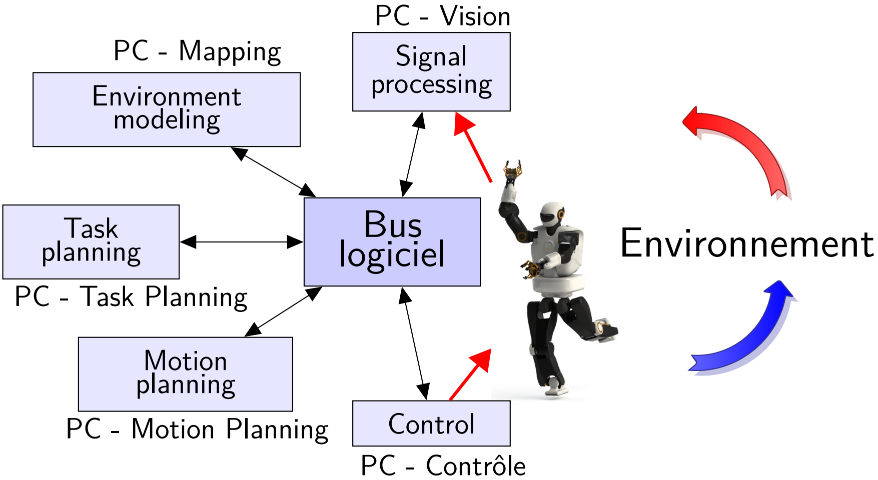
Motivations - Robotics application



Motivations - Robotics application - Tools



Motivations - Robotics application distributed

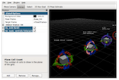


Motivations

The robotics community must *collaborate* to create robotics systems of the same quality than in the computer science and physics community.
ROS is the tool that allowed to scale-up.

=

+



+

+

Plumbery   Tools   Capacities   Ecosystem

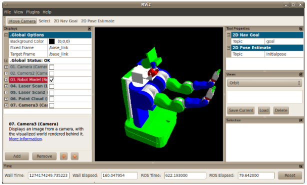
Motivations - Plumbery

Middleware

- Publication/subscription transmission on anonymous messages (Message Passing)
- Record and playback messages
- Requests/answers using remote procedure calls
- Distributed parameter server

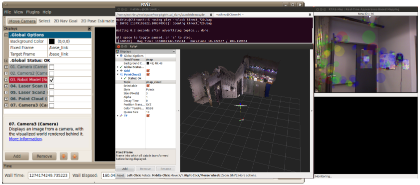
Motivations - Tools

rviz	3D graphical interface
rosbag	Record and data visualizing
rxplot	Online viewing quantities
rxgraph	Graph application structure display
rqt	Incremental Graphical User Interface
catkin	Compiling and packages management system



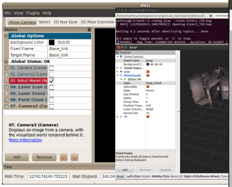
Motivations - Tools

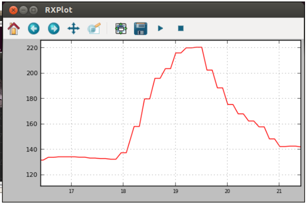
rviz	3D graphical interface
rosbag	Record and data visualizing
rxplot	Online viewing quantities
rxgraph	Graph application structure display
rqt	Incremental Graphical User Interface
catkin	Compiling and packages management system



Motivations - Tools

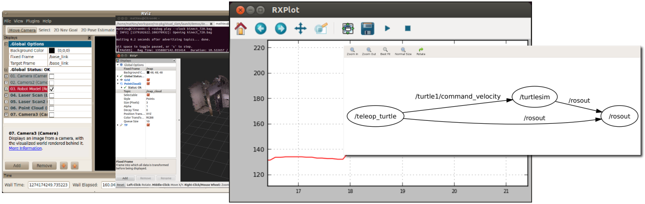
rviz	3D graphical interface
rosbag	Record and data visualizing
rxplot	Online viewing quantities
rxgraph	Graph application structure display
rqt	Incremental Graphical User Interface
catkin	Compiling and packages management system





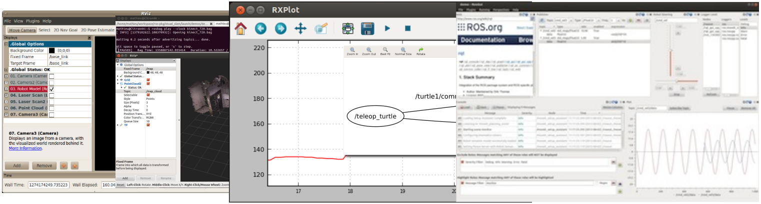
Motivations - Tools

rviz	3D graphical interface
rosbag	Record and data visualizing
rxplot	Online viewing quantities
rxgraph	Graph application structure display
rqt	Incremental Graphical User Interface
catkin	Compiling and packages management system



Motivations - Tools

rviz	3D graphical interface
rosbag	Record and data visualizing
rxplot	Online viewing quantities
rxgraph	Graph application structure display
rqt	Incremental Graphical User Interface
catkin	Compiling and packages management system



Motivations - Ecosystem

- Messages standardization for the robots
- Model description language (even if imperfect)
- Geometry - Dynamical libraries for robots
- Preemptable remote procedure calls
- Diagnostics
- Pose estimation
- Localization
- Map building
- Navigation

Motivations - Ecosystem

Operating Systems

Robots

Packages

Support : Ubuntu

Experimental :

Ubuntu ARM | OS X (Homebrew) | OS X (MacPorts)

OpenEmbedded/Yocto | Debian | Arch Linux

Windows | Ångström | UDOO

Motivations - Ecosystem

Operating Systems

Robots

Packages

01/05/2014 : 111 Supported robots on http://wiki.ros.org/Robots
25/04/2017 : 27 Supported robots on http://wiki.ros.org/Robots
01/05/2014 : 2048 Supported packages on http://wiki.ros.org/Packages
25/04/2017 : 1400 Supported packages on http://wiki.ros.org/Packages

ROS: short history

2008	ROS started with Willow Garage
2010 - Jan	ROS 1.0
2010 - Mar	Box Turtle
2010 - Aug	C Turtle
2011 - Mar	Diamondback
2011 - Aug	Electric Emys
2012 - Mar	Fuerte Turtle
2012 - Dec	Groovy Galapagos
2013 - Feb	Open Source Robotics Fundation continues ROS
2013 - Aug	Willow Garage became Suitable Technologies
2013 - Aug	PR-2 Support is handled by Clearpath Robotics
2013 - Aug	Hydro Medusa
2014 - Jul	Indigo Igloo (EOL - Apr 2019)
2015 - May	Jade Turtle (EOL - May 2017)
2016 - May	Kinetic Kame (EOL - May 2021)
2017 - May	Lunar Loggerhead (EOL - May 2019)
2018 - May	Melodic Morenia (EOL - May 2023)

ROS: Release normalization

- Lunar Loggerheard (May 2017- May 2019)
  - Ubuntu Xenial (16.04), Yakkety (16.10)
  - Ubuntu Zesty (17.04 LTS)
  - C++11, Boost 1.62, Lisp SBCL 1.2.4, Python 2.7 (tests against Python 3.5 recommended) CMake 3.7.2
  - Ogre3D 1.9.x, Gazebo 7.5, PCL 1.8.0, OpenCV 3.x, Qt 5.7.1, PyQt5 5.7
- Melodic Morenia (May 2016 - May 2023)
  - Ubuntu Artful (17.10)
  - Ubuntu Bionic (18.04 LTS)
  - C++14, Boost 1.55, Lisp SBCL 1.3.14, Python 2.7 (tests against Python 3.5 recommended), CMake 3.10.2
  - Ogre3D 1.9.x, Gazebo 9, PCL 1.8.1, OpenCV 3.2, Qt 5.9.5, PyQt5 5.10.1
- Lien vers la ROS Enhancement Proposal (REPs) : `http://www.ros.org/reps/rep-0003.html`

Online exercices

What is a simulator ?

[Ivaldi,ICHR,2014]

System simulators

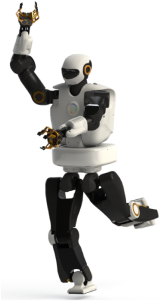
- Simulate sensors, actuators, environment
- Smoothly going from simulation to real robot
- Accurate representation of reality
- Stability

Control simulator

- Real-time
- Catch the robot main dynamics
- To be integrate inside the control loop

What is a simulator ?

[Ivaldi,ICHR,2014]



System simulators

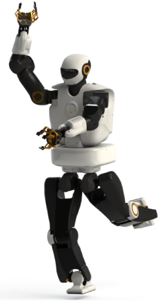
- Gazebo
- Stage
- Morse
- OpenHRP

Control simulator

- MuJoCo
- RBDL
- Pinocchio
- Robotran

What is a simulator ?

[Ivaldi,ICHR,2014]



System simulators

- Gazebo
- Stage
- Morse
- OpenHRP

Dynamics Engine: ODE

Control simulator

- MuJoCo
- RBDL
- Pinocchio
- Robotran

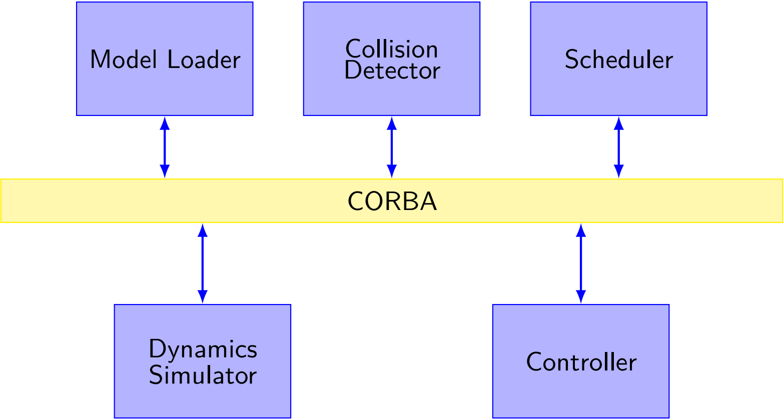
Problem 1: Reality

- Impact
- OpenHRP 2.x: 800 N
- Real robot: 1300 N
- Giving up speed
- Compliant material
- Dynamics
- Vision

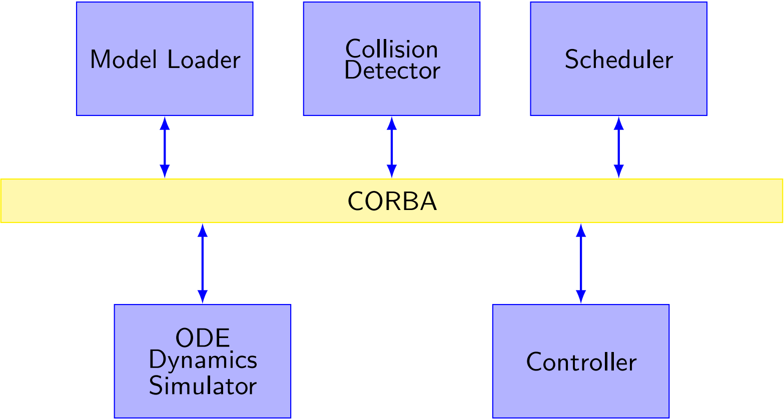
Problem 2: Software flexibility

- From simulation to real robot
- Being able to change parts of the simulator
- Strong middleware
- Sensor simulation

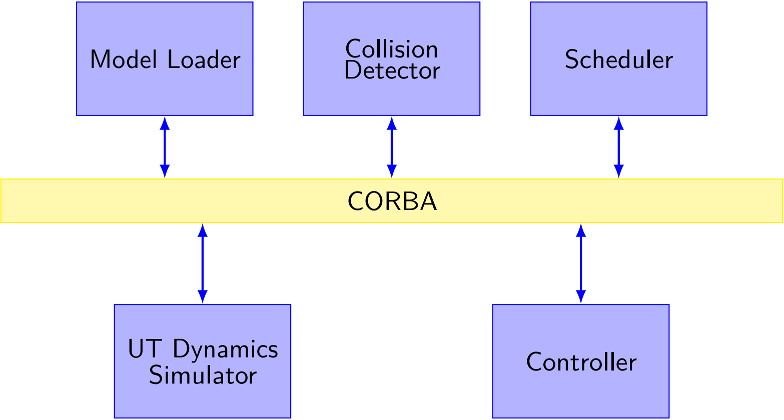
OpenHRP architecture



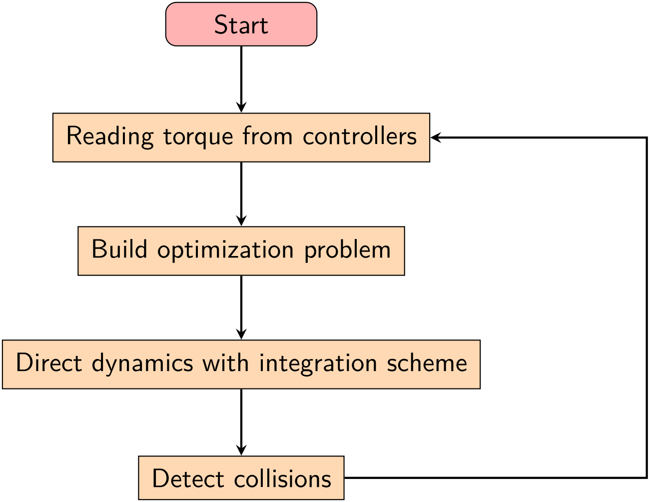
OpenHRP architecture



OpenHRP architecture



Global sequence - scheduler

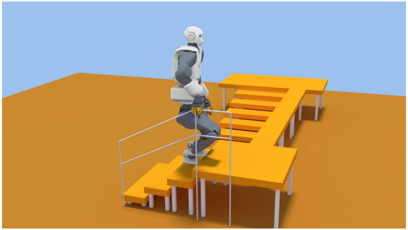


OpenHRP AIST Dynamics Simulator

- Uses a mixture of *generalized* and *maximal* coordinates formulation
- The *generalized* coordinates are used to compute the robot state.
- This assumes a perfect rigid body dynamics (true for a high PID control)
- The *maximal* coordinates are used to compute the rest of the world state.
- Specific compliant joint are introduced in the direct dynamics of the robot.

ODE: The problem

Definitions [Hsu,Simpar,2014]



i-th body - position, quaternion	x, q
velocity	$\dot{\nu}_i = [\dot{x}^T, \omega^T]^T \in \mathbb{R}^{dim(i)}$
force	$F_i \in \mathbb{R}^{dim(i)}$
acceleration	$\dot{\nu}_i$
then	$M_i \dot{\nu}_i = F_i$
	$M\dot{\nu} = F$

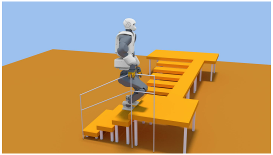
$$M_i \dot{\nu}_i = \begin{bmatrix} m_i \delta & 0 \\ 0 & I_i \end{bmatrix} \begin{bmatrix} \ddot{x}_i \\ \dot{\omega}_i \end{bmatrix} = \begin{bmatrix} f \\ \tau - \omega \times I\omega \end{bmatrix} = F_i$$

ODE: The problem

[Hsu,Simpar,2014]

i-th constraint $h_e(x,q) = 0,\ h_i(x,q) \geq 0,$

Let us linearize constraints



ODE: The problem

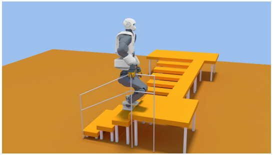
[Hsu,Simpar,2014]

i-th constraint $h_e(x,q) = 0,\ h_i(x,q) \geq 0,$

speed $J_{i1}\nu_1 + \cdots + J_{ik}\nu_k + \cdots + J_{in}\nu_n + c_i = 0$

all together $J\dot{\nu} + c = 0$ (1)

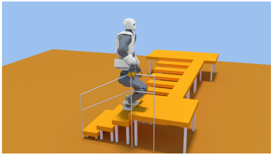
$M\dot{\nu} = F^c + F$



ODE: The problem

[Hsu,Simpar,2014]

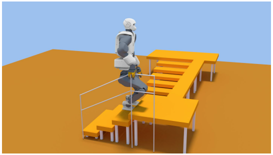
i-th constraint	$h_e(x, q) = 0,\ h_i(x, q) \geq 0,$
speed	$J_{i1}\nu_1 + \cdots + J_{ik}\nu_k + \cdots + J_{in}\nu_n + c_i = 0$
all together	$J\dot{\nu} + c = 0$ (1)
	$M\dot{\nu} = F^c + F$
with	$F^c = J^T\lambda$
then	$M\dot{\nu} = J^T\lambda + F$



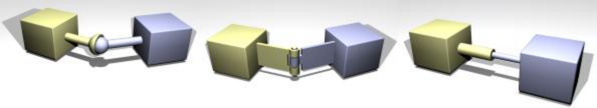
ODE: The problem

[Hsu,Simpar,2014]

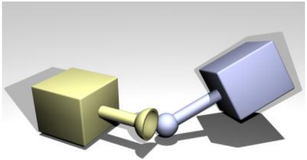
i-th constraint	$h_e(x,q) = 0,\ h_i(x,q) \geq 0,$
speed	$J_{i1}\nu_1 + \cdots + J_{ik}\nu_k + \cdots + J_{in}\nu_n + c_i = 0$
all together	$J\dot{\nu} + c = 0$ (1)
	$M\dot{\nu} = F^c + F$
with	$F^c = J^T\lambda$
then	$M\dot{\nu} = J^T\lambda + F$
then with (1)	$J(M^{-1}J^T\lambda + M^{-1}F) + c = 0$
then	$A\lambda = b$
with	$A = JM^{-1}J^T$ et $b = -(JM^{-1}F^{ext} + c)$



ODE: The problem



Joints when everythings goes well

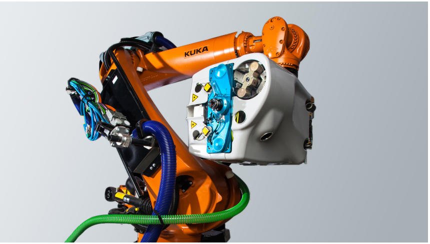


Error Reduction Parameter

Tutorials

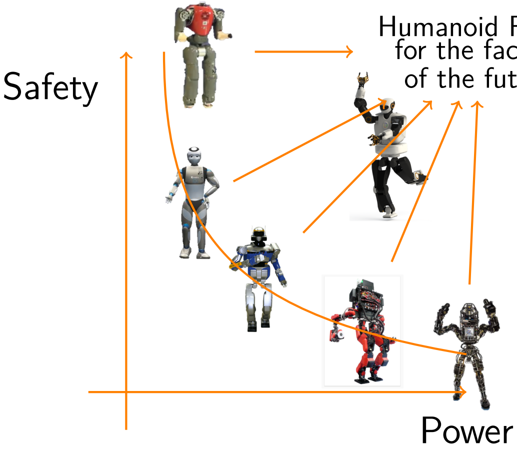
Online Exercices

The problem



Costly, heavy,
not user friendly

The problem - The vision



125 millions of holes drilled by year

75 % are done *manually*