



HAL
open science

**Program / simulation tools / ROS / Gazebo /
OpenHRP**
Olivier Stasse

► **To cite this version:**

Olivier Stasse. Program / simulation tools / ROS / Gazebo / OpenHRP. Doctoral. GdR Robotics Winter School: Robotica Principia, Centre de recherche Inria Sophia Antipolis – Méditerranée, France. 2019. cel-02130166

HAL Id: cel-02130166

<https://inria.hal.science/cel-02130166v1>

Submitted on 15 May 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License



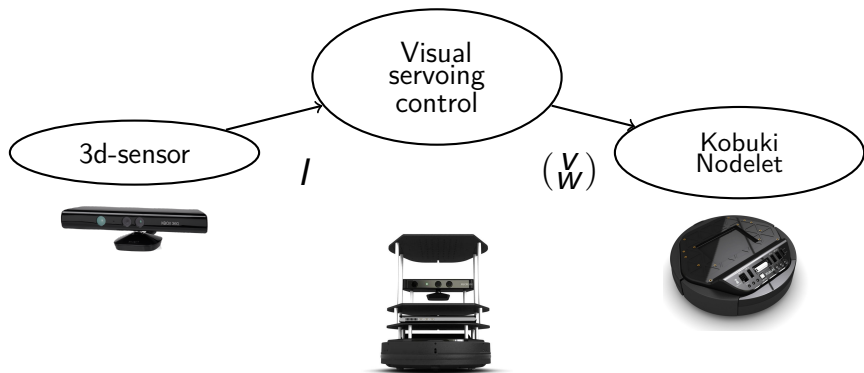
Program/simulation tools
ROS/Gazebo/OpenHRP

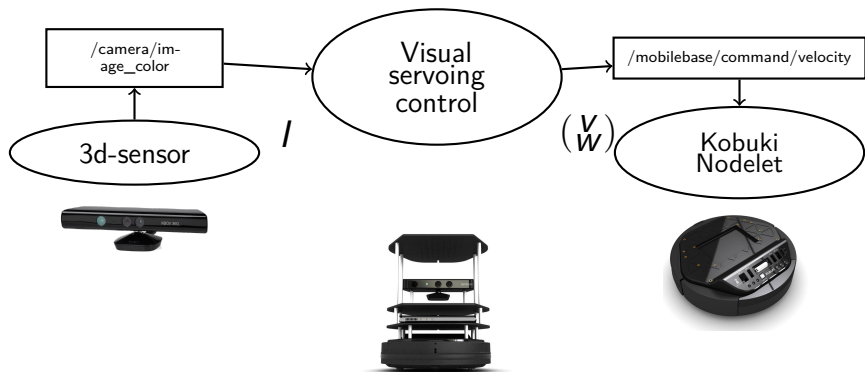
Robotics Principia

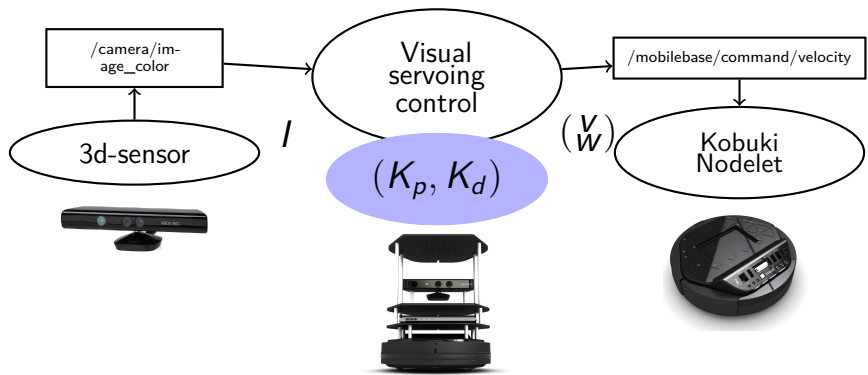
24th January 2019, Sophia-Antipolis, France

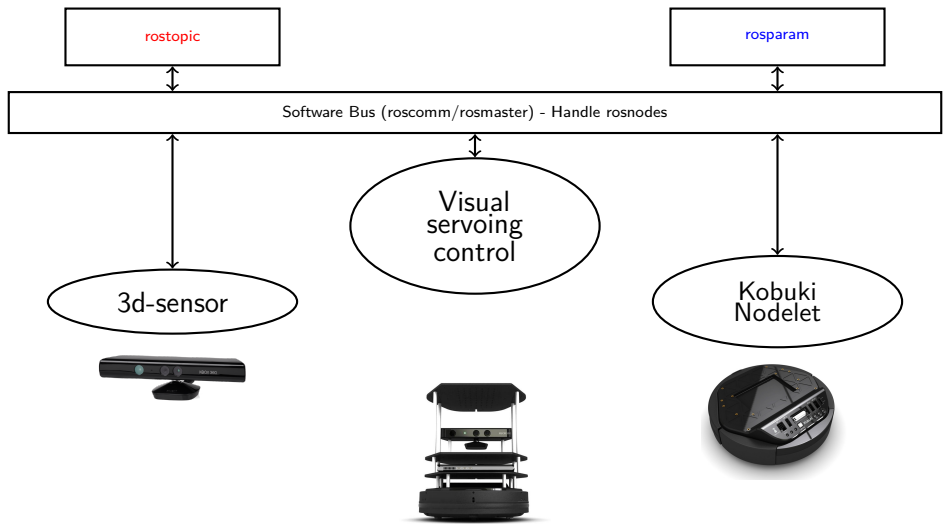
O. Stasse, Gepetto,
LAAS-CNRS

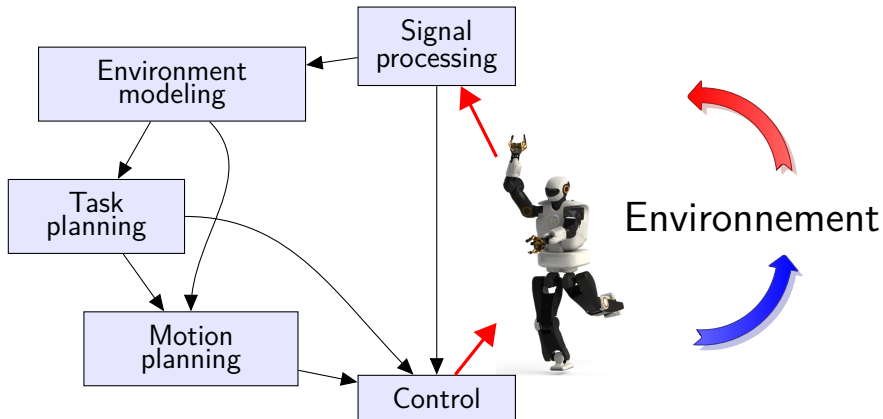
- 1** ROS
- 2** Simulators
- 3** Simulator architecture
- 4** Dynamics engine
- 5** Motivations - Scientific approach

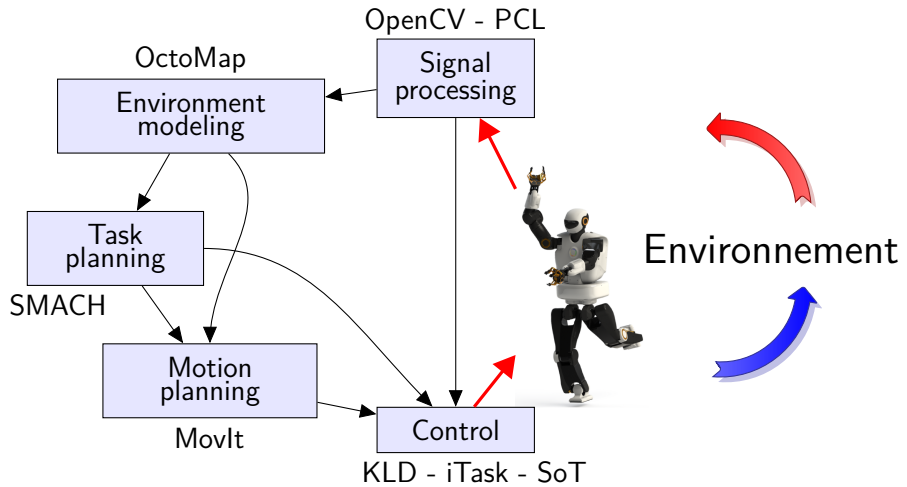


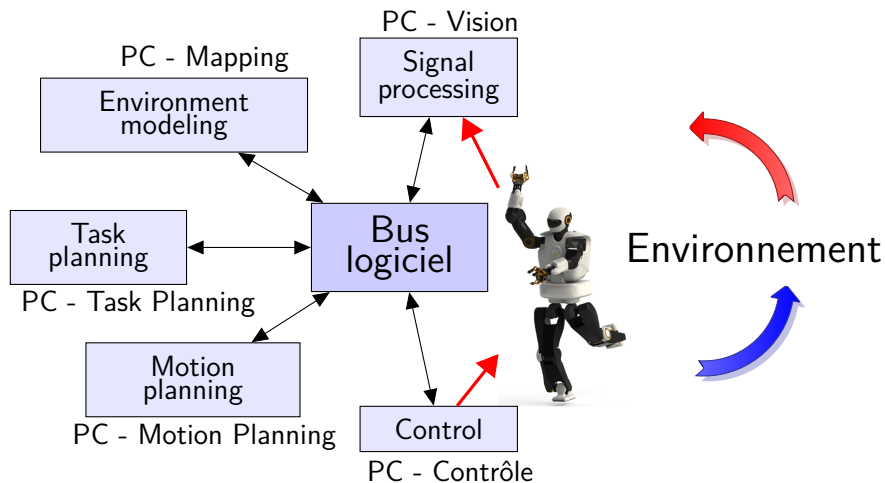












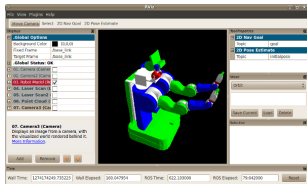
The robotics community must *collaborate* to create robotics systems of the same quality than in the computer science and physics community.
ROS is the tool that allowed to scale-up.



Middleware

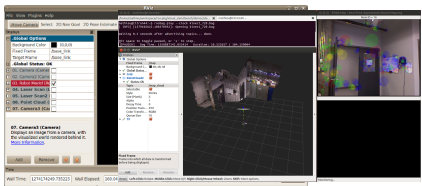
- Publication/subscription transmission on anonymous messages (Message Passing)
- Record and playback messages
- Requests/answers using remote procedure calls
- Distributed parameter server

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

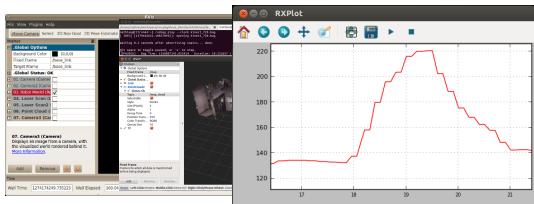


Motivations - Tools

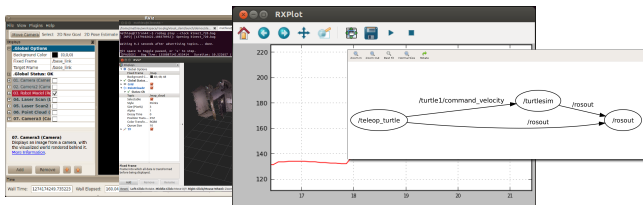
- rviz 3D graphical interface
- rosviz Record and data visualizing
- rxplot Online viewing quantities
- rxgraph Graph application structure display
- rqt Incremental Graphical User Interface
- catkin Compiling and packages management system



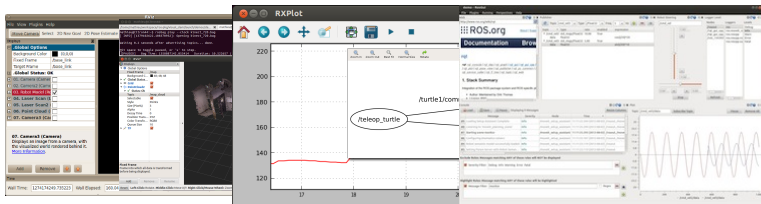
- rviz 3D graphical interface
- roscap Record and data visualizing
- rxplot Online viewing quantities
- rxgraph Graph application structure display
- rqt Incremental Graphical User Interface
- catkin Compiling and packages management system



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



- 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




- 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




Operating Systems

Robots



Packages

Support :  Ubuntu

Experimental :

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

 OpenEmbedded/Yocto  Debian  Arch Linux

 Windows  Ångström  UDOO

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 Foundation 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)

- Lunar Loggerhead (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/repos/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 ■

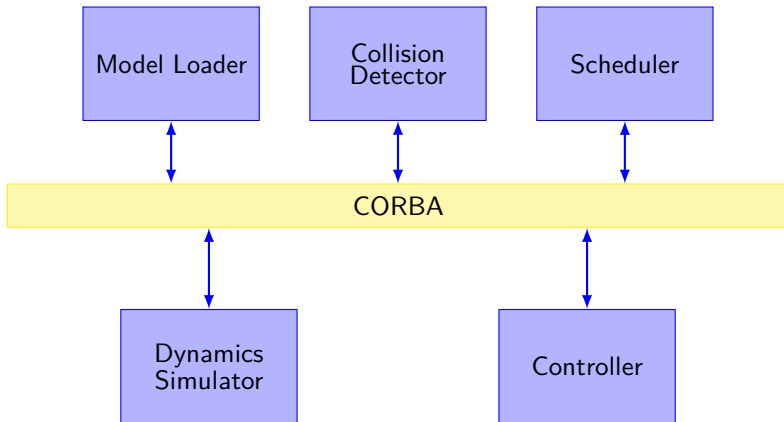


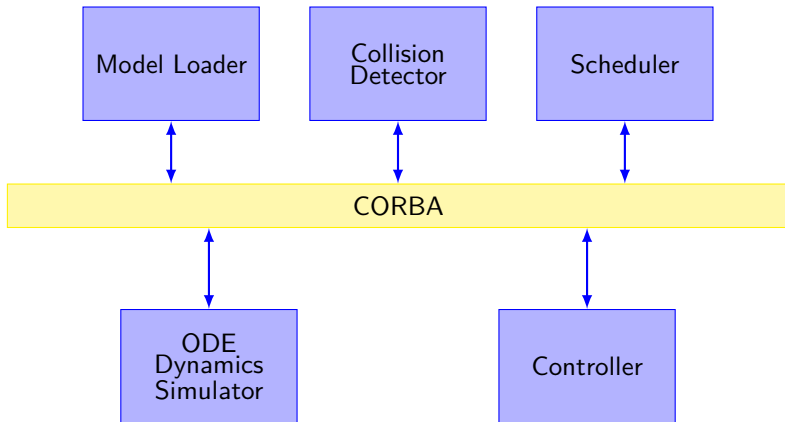
Problem 1: Reality

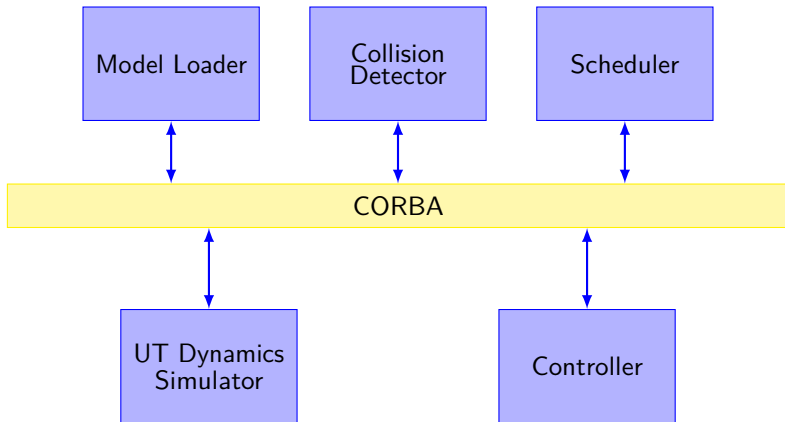
- Impact
- OpenHRP 2.x: 800 N
- Real robot: 1300 N
- Giving up speed
- Compliant material
- Dynamics
- Vision

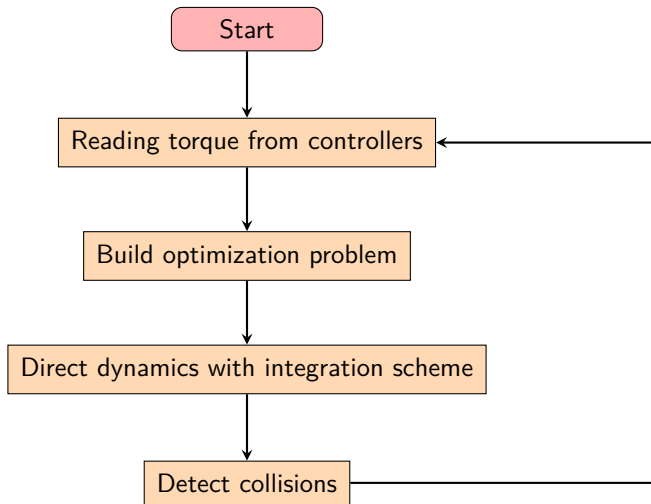
Problem 2: Software flexibility

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









- 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.

Definitions

[Hsu, Simpar, 2014]

i -th body
- position,
quaternion

$$x, q$$

velocity

$$\dot{v}_i = [\dot{x}^T, \omega^T]^T \in \mathbb{R}^{dim(i)}$$

force

$$F_i \in \mathbb{R}^{dim(i)}$$

acceleration

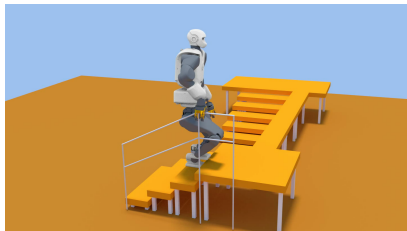
$$\ddot{v}_i$$

then

$$M_i \dot{v}_i = F_i$$

$$M \dot{v} = F$$

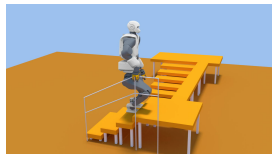
$$M_i \dot{v}_i = \begin{bmatrix} m_i \delta & 0 \\ 0 & I_i \end{bmatrix} \begin{bmatrix} \ddot{x}_i \\ \dot{\omega}_i \end{bmatrix} = \begin{bmatrix} f \\ \tau - \omega \times I \omega \end{bmatrix} = F_i$$



[Hsu, Simpar, 2014]

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

Let us linearize constraints



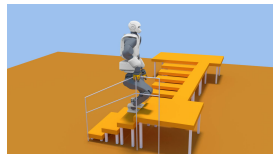
[Hsu, Simpar, 2014]

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

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

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

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



[Hsu, Simpar, 2014]

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

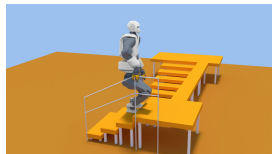
speed $J_{i1}\nu_1 + \dots + J_{ik}\nu_k + \dots + J_{in}\nu_n + c_i = 0$

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

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

with $F^c = J^T \lambda$

then $M\dot{\nu} = J^T \lambda + F$



[Hsu, Simpar, 2014]

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

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

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

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

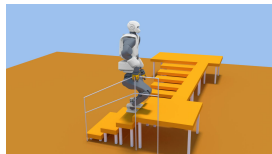
with $F^c = J^T \lambda$

then $M\dot{\nu} = J^T \lambda + F$

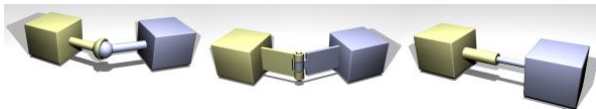
then with (1) $J(M^{-1}J^T \lambda + M^{-1}F) + c = 0$

then $A\lambda = b$

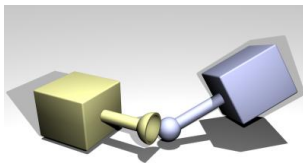
with $A = JM^{-1}J^T$ et $b = -(JM^{-1}F^{\text{ext}} + c)$



ODE: The problem

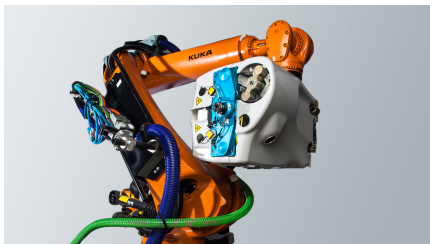


Joints when everything goes well



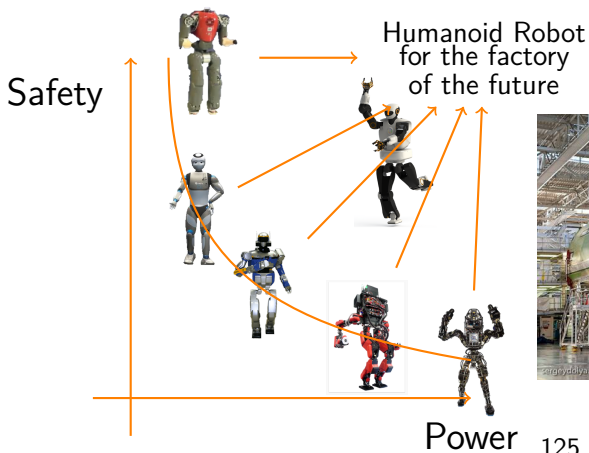
Error Reduction Parameter

Online Exercices



Costly, heavy,
not user friendly



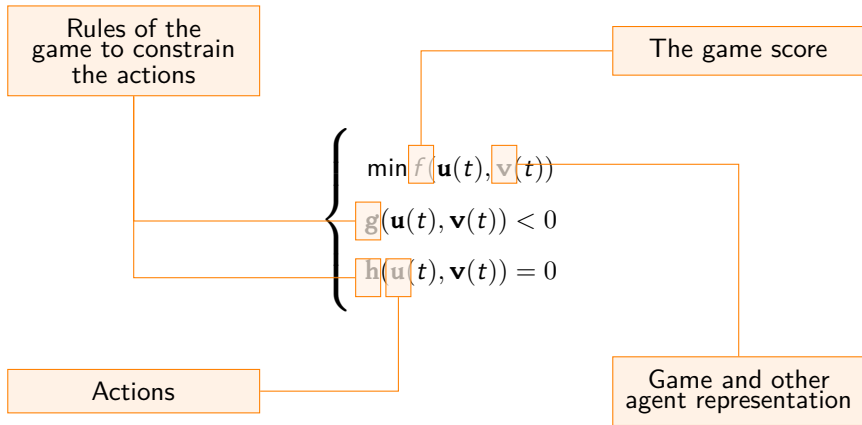


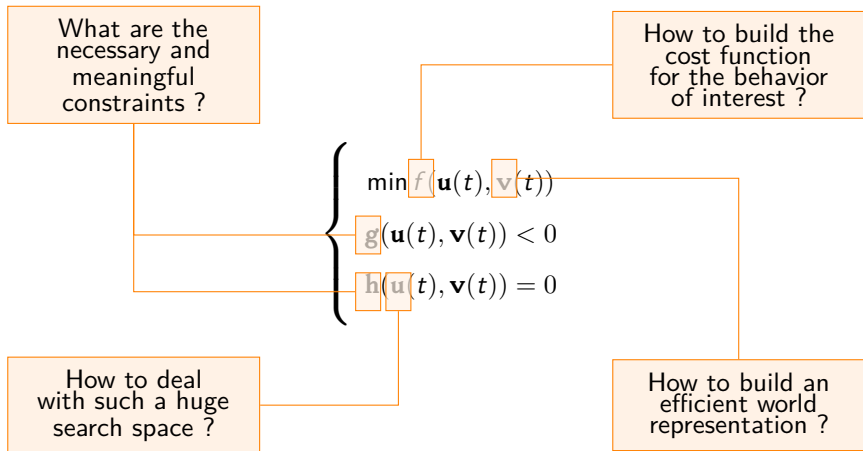
125 millions of holes drilled by year

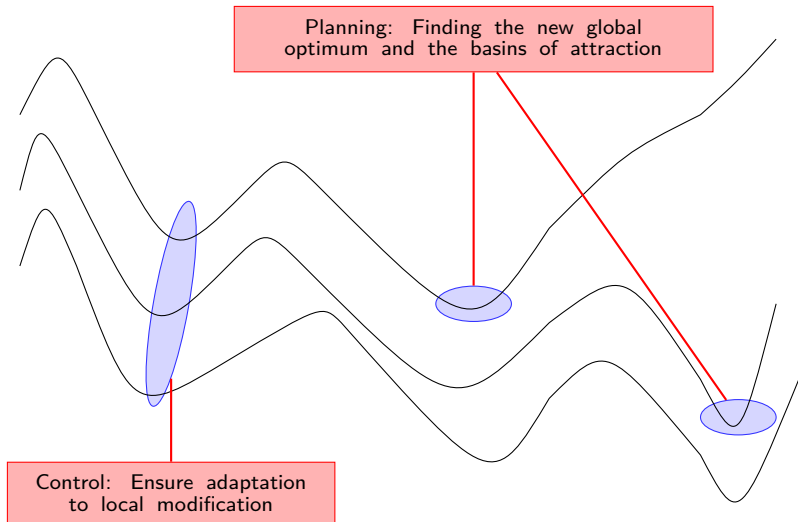
75 % are done *manually*

- Problem 1:** Variability according to the customers
- Problem 2:** Variability in the realization (humans)
- Problem 3:** Variability in the delivery
- Problem 4:** Collaboration with humans
- Problem 5:** Certification









$$\left\{ \begin{array}{l} \min f(\mathbf{u}(t), \mathbf{v}(t)) \\ \mathbf{g}(\mathbf{u}(t), \mathbf{v}(t)) < 0 \\ \mathbf{h}(\mathbf{u}(t), \mathbf{v}(t)) = 0 \end{array} \right.$$

[Saidi, IROS 2007]

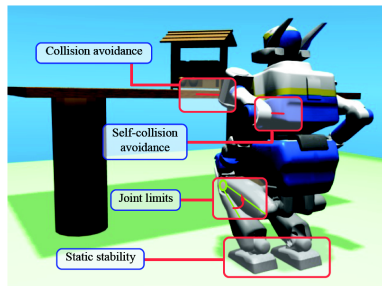
$f(t)$: The cost function

$\mathbf{u}(t)$: The control vector

$\mathbf{g}(t)$: The inequality constraints

$\mathbf{h}(t)$: The equality constraints

$\mathbf{v}(t)$: The environment representation



$$\left\{ \begin{array}{l} \min_{\mathbf{q}(\cdot), \mathbf{u}(\cdot)} \int_0^T L(t, \mathbf{q}(t), \mathbf{u}(t)) dt \\ \dot{\mathbf{q}}(t) = f(\mathbf{q}(t), \mathbf{u}(t)), \quad t \in [0, T] \\ \mathbf{q}(0) = \mathbf{q}_0, \quad T > 0 \\ 0 \leq \mathbf{gh}(\mathbf{q}(t), \mathbf{u}(t)), \quad t \in [0, T] \end{array} \right.$$

[Saidi, IROS 2007]

f: The system dynamics

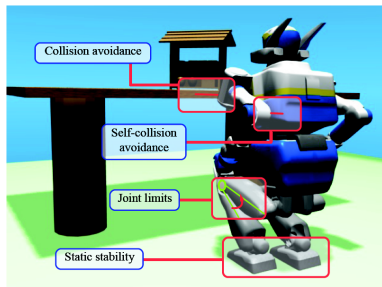
u: The control vector

q: The system state

gh: The (in)equality constraints

v: The environment representation

L: The cost

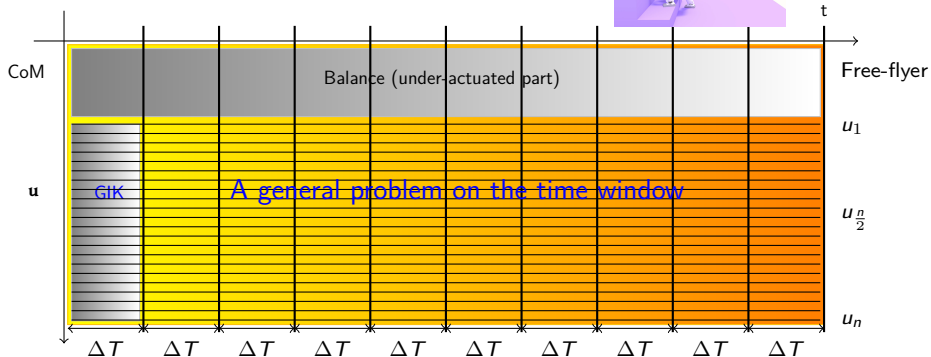
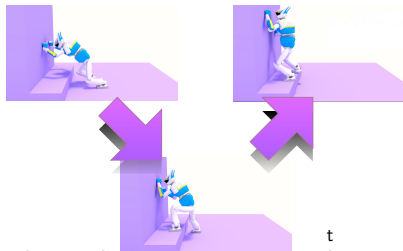


- Size of the problem

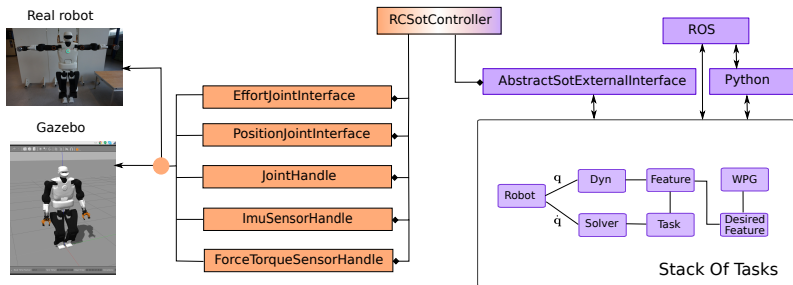
$$1.6 \times 200 \times 30 = 9600 \text{ variables}$$

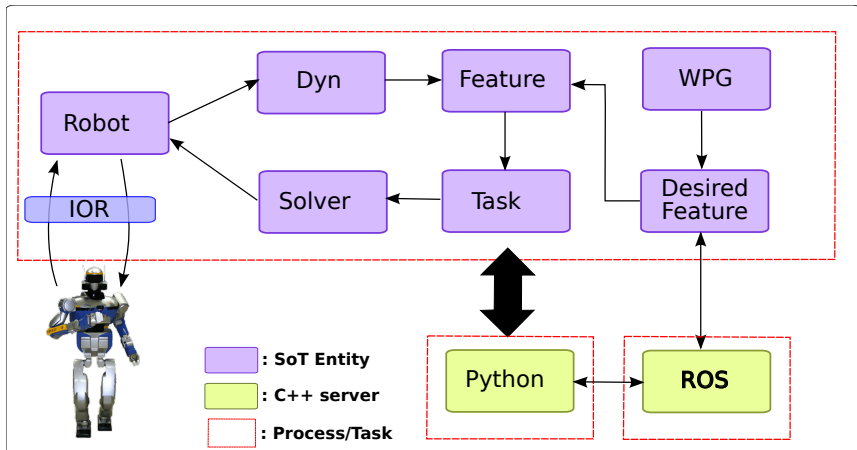
- Non linear constraints

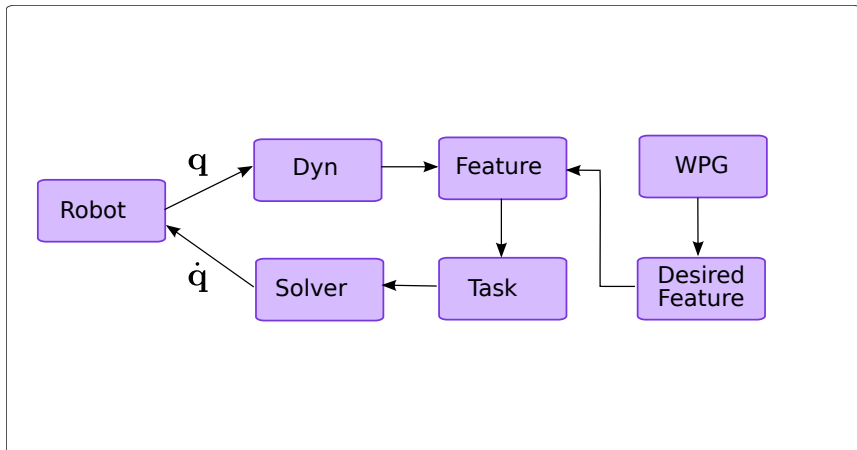
- Discrete nature due to contacts

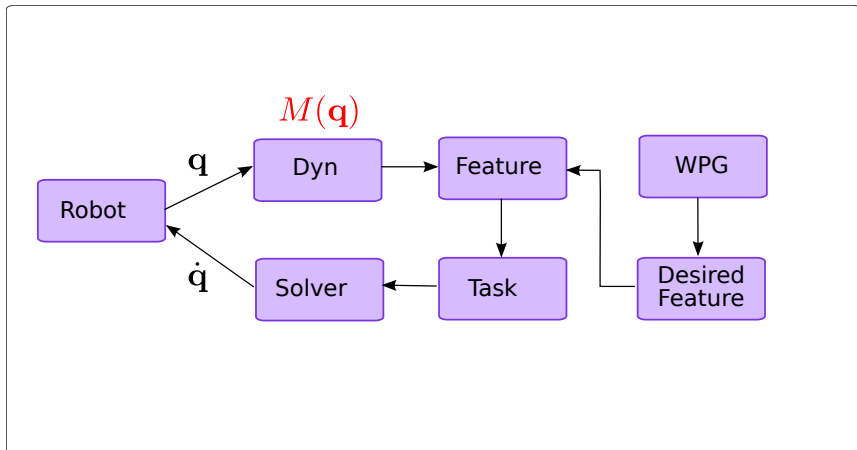


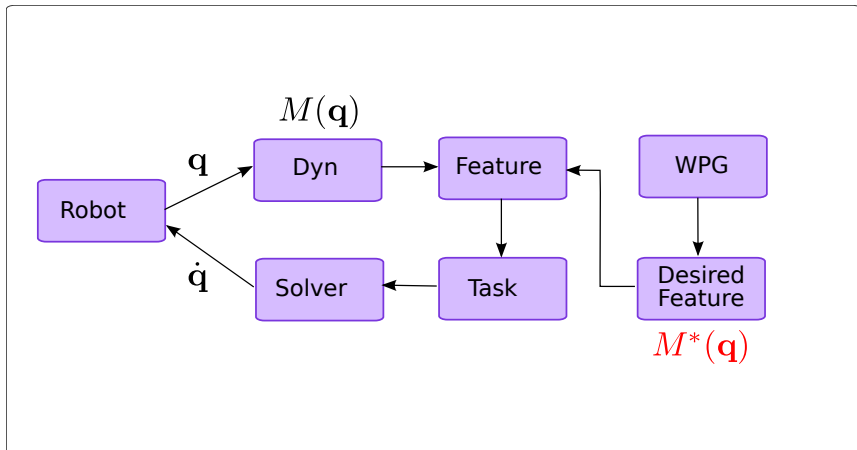
<https://www.youtube.com/watch?v=WbsQBPzQakc>

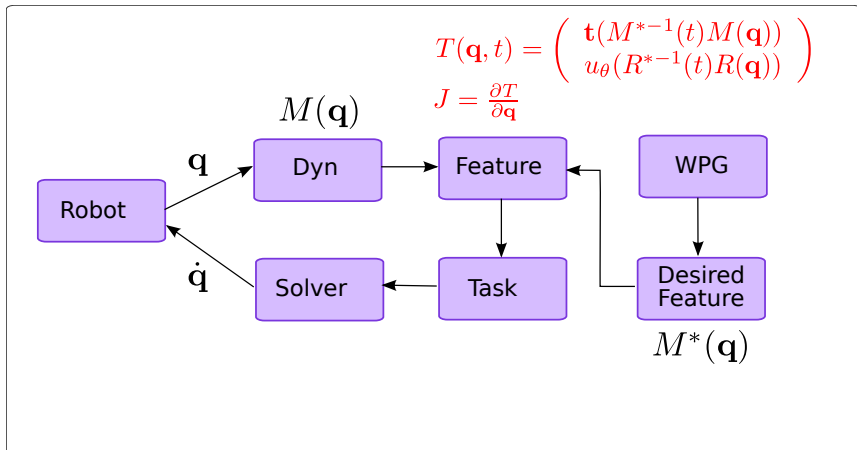


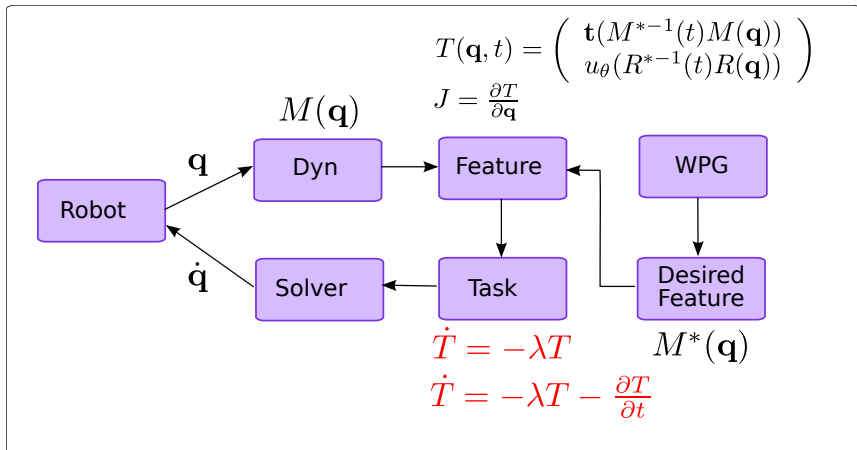


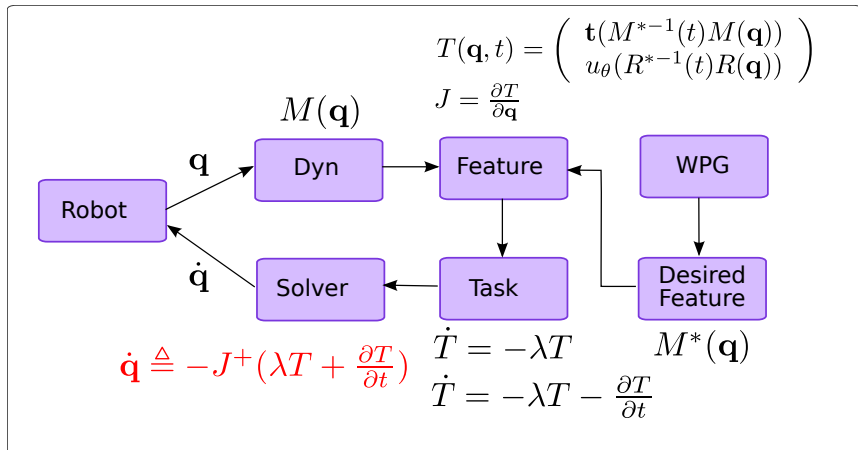


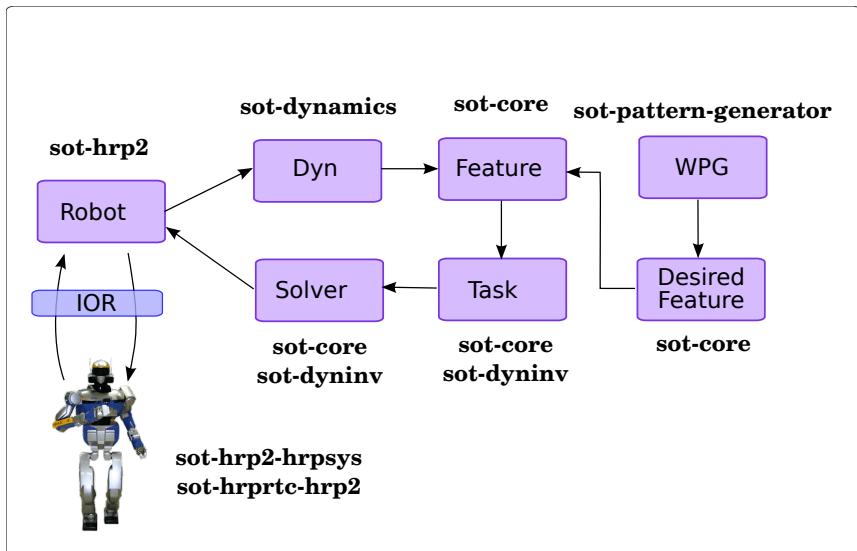














Inverse Dynamics

Weighted Pseudo Inverse

- Faster (!?) computation
- Easier to formulate
- Do not guarantee convergence
- Difficulty to tune the weights
- Do not handle properly inequalities

Hierarchical Quadratic Program

- Slower (!?) computation time
- Warranty on priority
- Handle easily inequalities
- Difficult to formulate (here hidden in the solver)
- Known problems with cycles and singularities management

Pros

- Generic to put instantaneous controller together
- Allow code reusability
- Real-time performance
- Adapted to complex applications
- Binary packages support
- Eigen support

Cons

- Research code
- The learning curve seems to be yet steep