Control of the HRP4 robot

JNRH-CAR

24 juin 2014

François Keith

Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier
LIRMM

UNIVERSITÉ MONTPELLIER 2
SCIENCES ET TECHNIQUES

cnrs

Summary

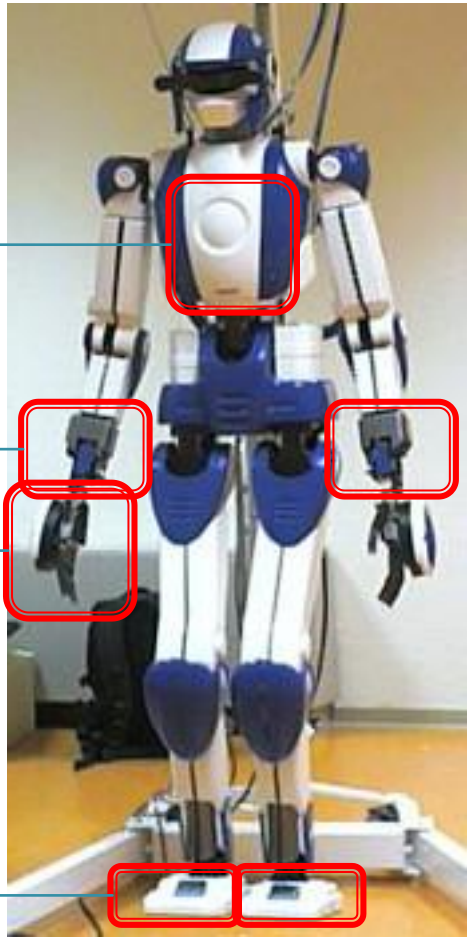
- HRP4 description
- Control softwares
  - Stack-of-tasks
  - Multi-contact controller
- Middleware: OpenRTM

Description of the HRP-4 robot

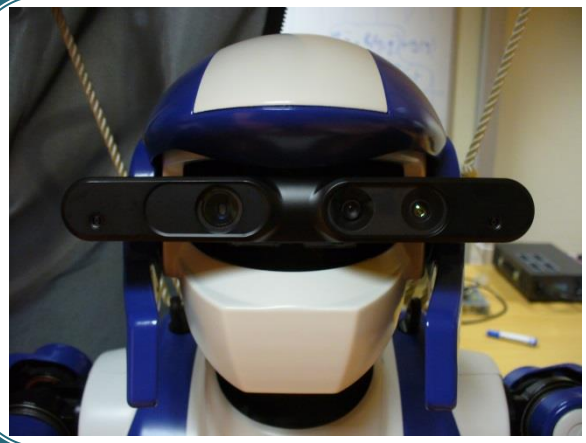
- 1.51m, 39kg
- 34 dofs
  - 6dofs/leg, 2/waist, 2/head, 6/arms, 2/hands
- Sensors
  - Accelerometer
  - Gyrometer
  - 6dofs sensor / foot
  - 6dofs sensor / hand

2 dofs/main

- Thumb
- Fingers

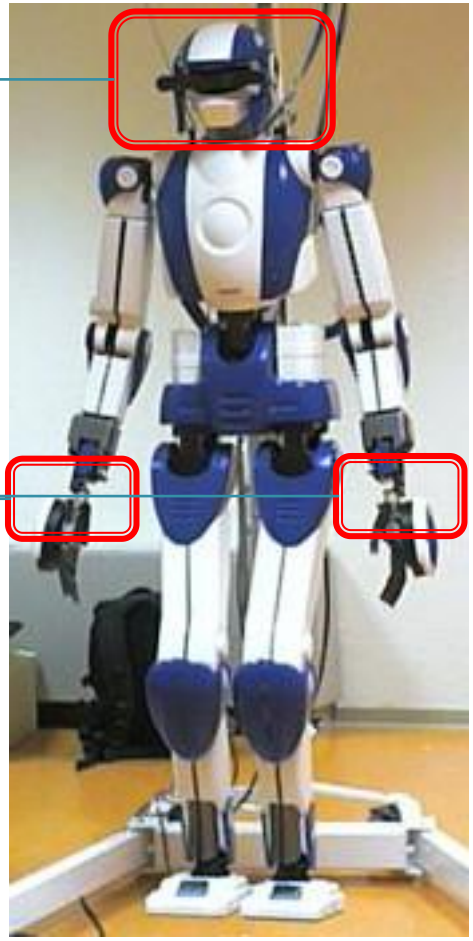


Description of the HRP-4 robot



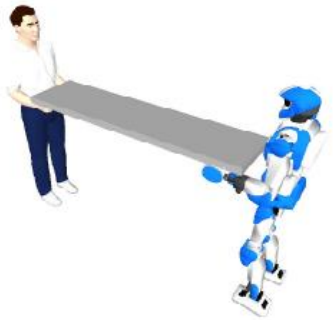
XTION PRO Live

- Schunk ATI Mini
  - More robust and precise

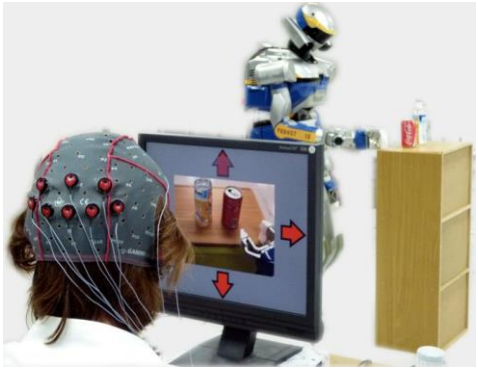


Research objectives

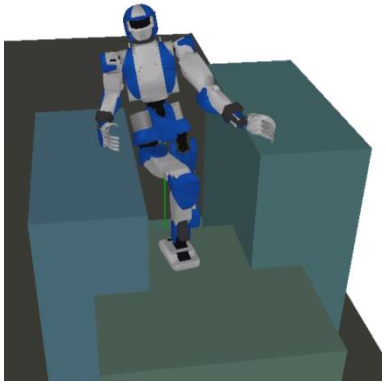
- Collaboration robot/humain
  - Collaborative transportation of a common object
  - Assistance to elderly people (provide support)



- Command and interaction human-robot using physiological signals



- Planification and control multi-contact



Stack-of-tasks

- Task-based hierarchical framework
- Handle inequality and equality tasks
- Implementation of a data flow

- Software architecture:
  - C++: definition of the entities and algorithms.
  - Python: building of the graph / scenario
  - Bridge with ROS: allows import/export of data and remote python interpreter.

STACK-OF-TASKS

TYPES OF CONTROL

Kinematic control

- Reference

$$J_i \dot{q} = \dot{e}_i^*$$

- Variables
  - Joint velocity $\dot{q}$
- Constraints
  - Joint limits

$$\underline{\dot{q}} < \dot{q} < \overline{\dot{q}}$$

$$\underline{q} < q < \overline{q}$$

Dynamic control

- Reference

$$J_i \ddot{q} + \dot{J}_i \dot{q} = \ddot{e}_i^*$$

- Variables
  - Joint torques τ
  - Joint acceleration $\ddot{q}$
  - External forces f_c
- Constraints
  - Dynamic equation

$$M\ddot{q} + b(q, \dot{q}) = \tau + J_c^T f_c$$

  - Contact constraint

$$J_c \ddot{q} + \dot{J}_c \dot{q} = 0 \qquad 0 \leq f_c^{\perp}$$

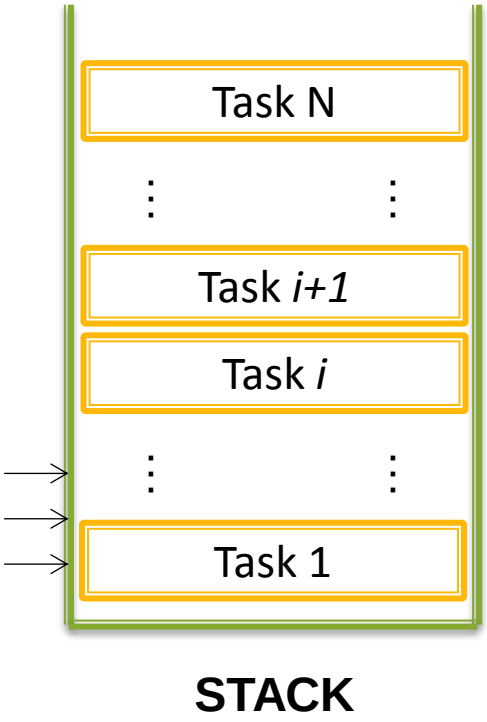
  - Joint limits

$$\underline{\tau} < \tau < \overline{\tau}$$

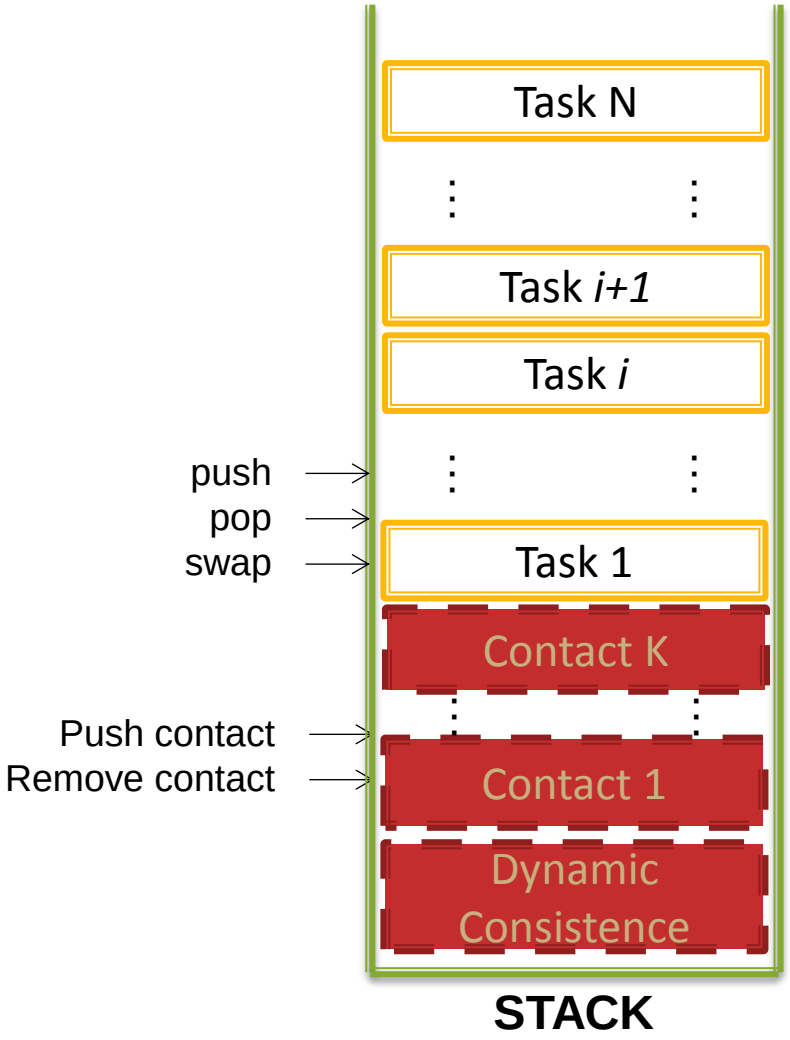
STACK-OF-TASKS

TYPES OF CONTROL

Kinematic control



Dynamic control

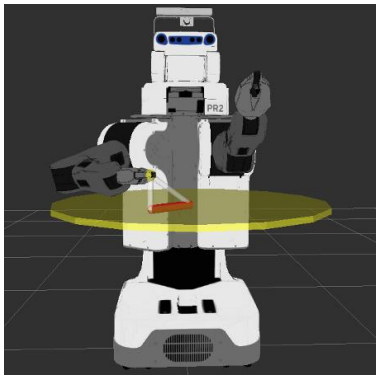


Stack-of-tasks

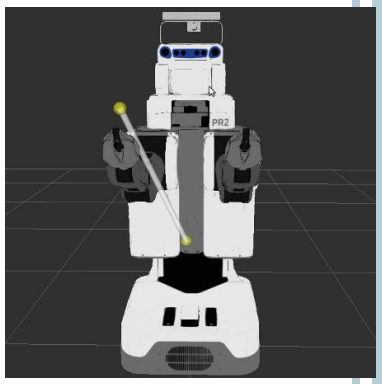
Bridging with expression-graph

- Basic tasks handled
  CoM, 6d position tasks, Visual task, joint limits…

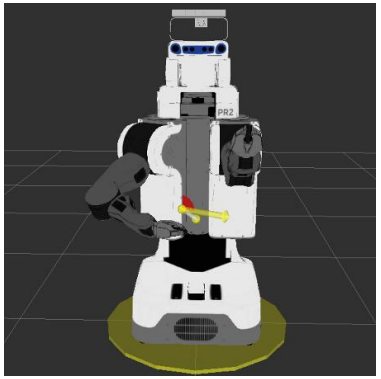
- Increase the set of tasks handled by combining with an expression-graph
- Allow to define constraints in non Cartesian spaces
- Definition of the tasks closer to the symbolic level



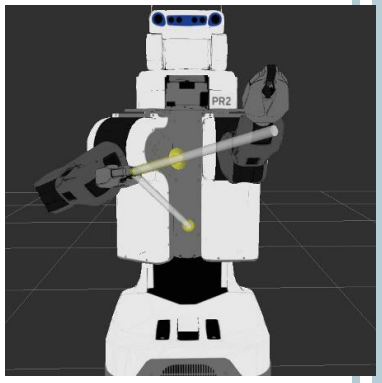
Distance 1 axe



Distance



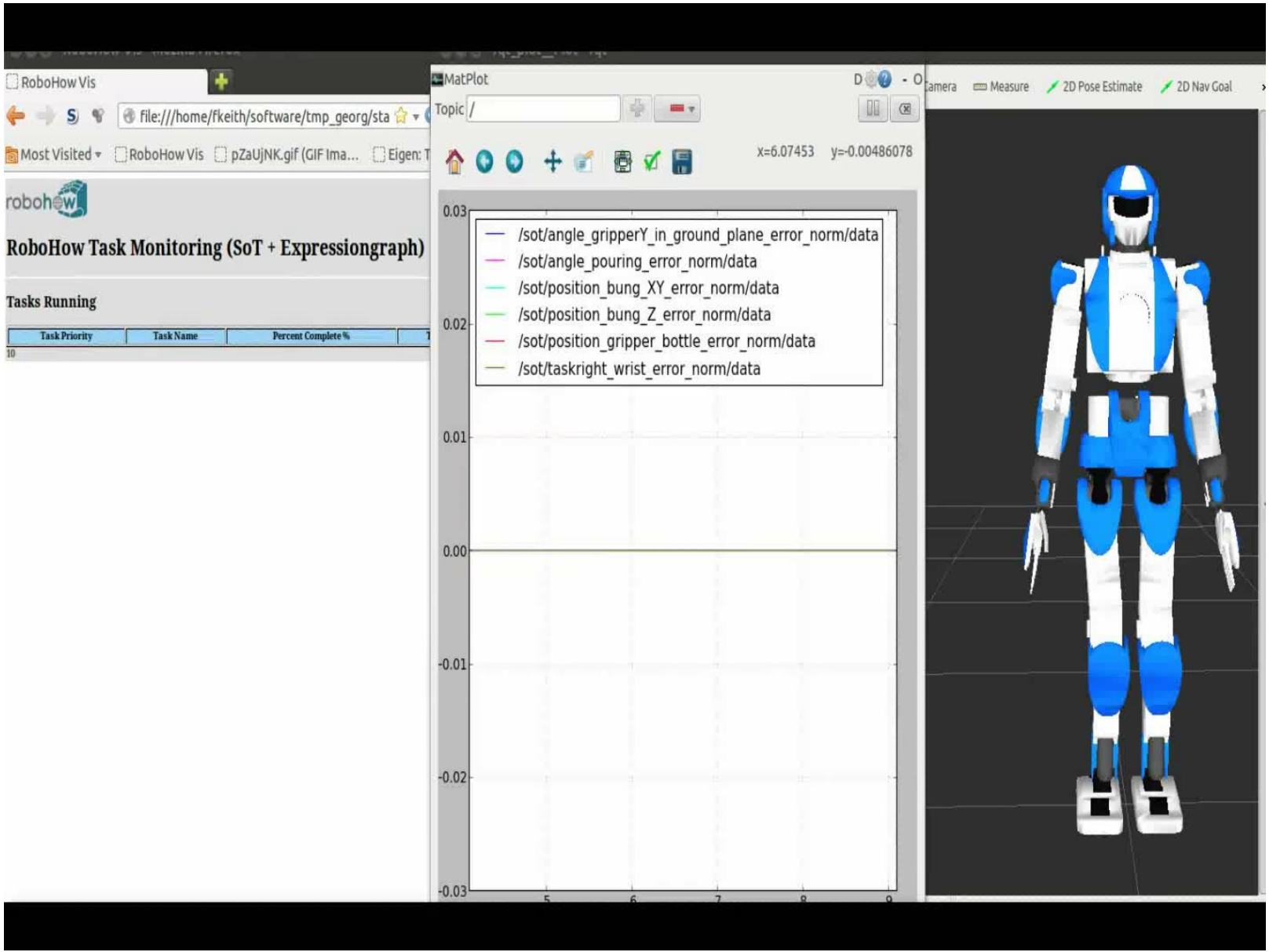
Angle



« Pointing at »

Stack-of-Tasks

Bridging with expression-graph

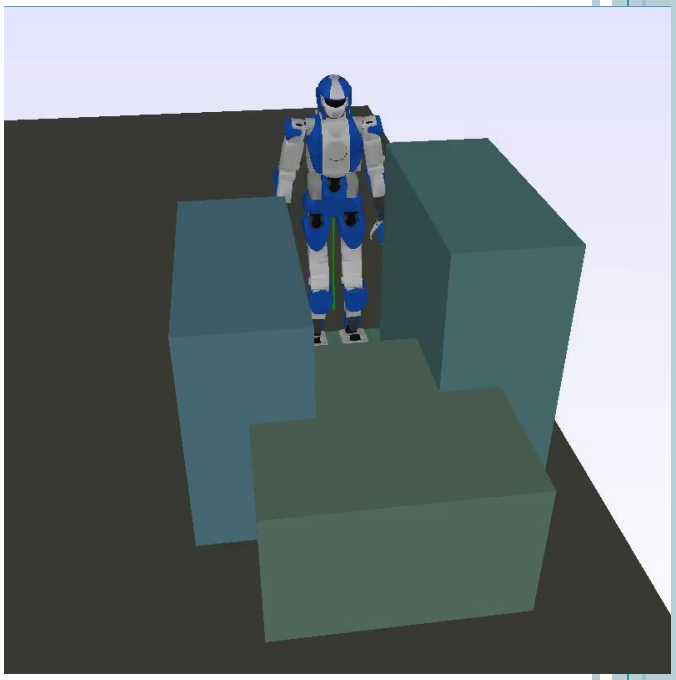


Stack-of-tasks

Brain control interface example

MULTI-CONTACT CONTROL

- Goal
  - Use the whole body to navigate in the environment.
- Approach:
  - Generation of stable postures with several contacts
  - Find a sequence of tasks linking the initial and final states
    - Contact addition/removal or COM target change
  - Generate a motion to link those postures
- Software architecture:
  - ROS dependent architecture
  - Solver wrapped into a ROS node
  - Provides the result of the control law via ros message.



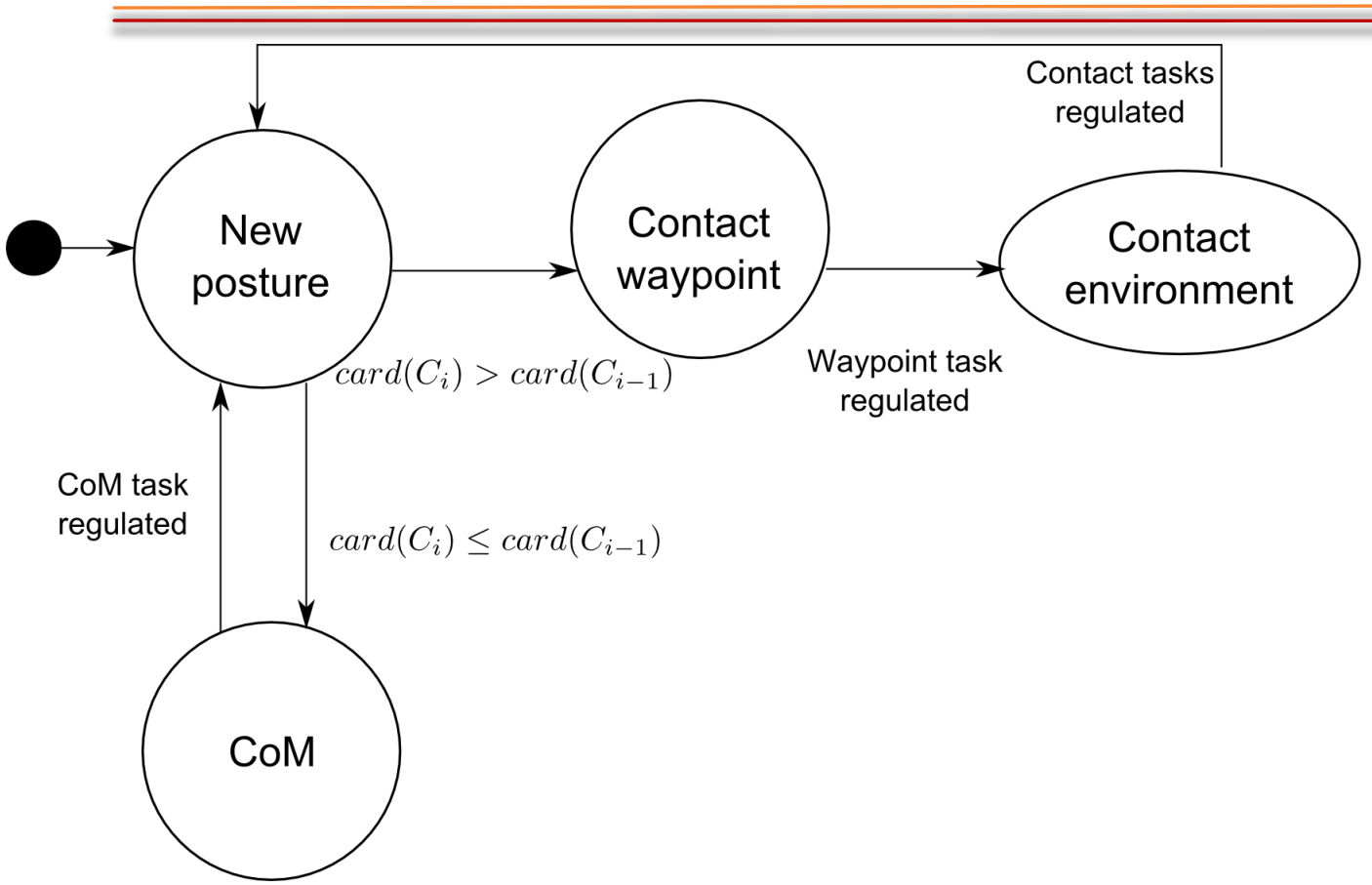
MULTI-CONTACT CONTROL

QP DEFINITION

- Weighting between the tasks
- Dynamic control
- Variables
  - Joint acceleration, force
- Constraints:
  - Dynamic equation
  - Contact constraints
  - Joint limits
  - Collision avoidance

MULTI-CONTACT CONTROL

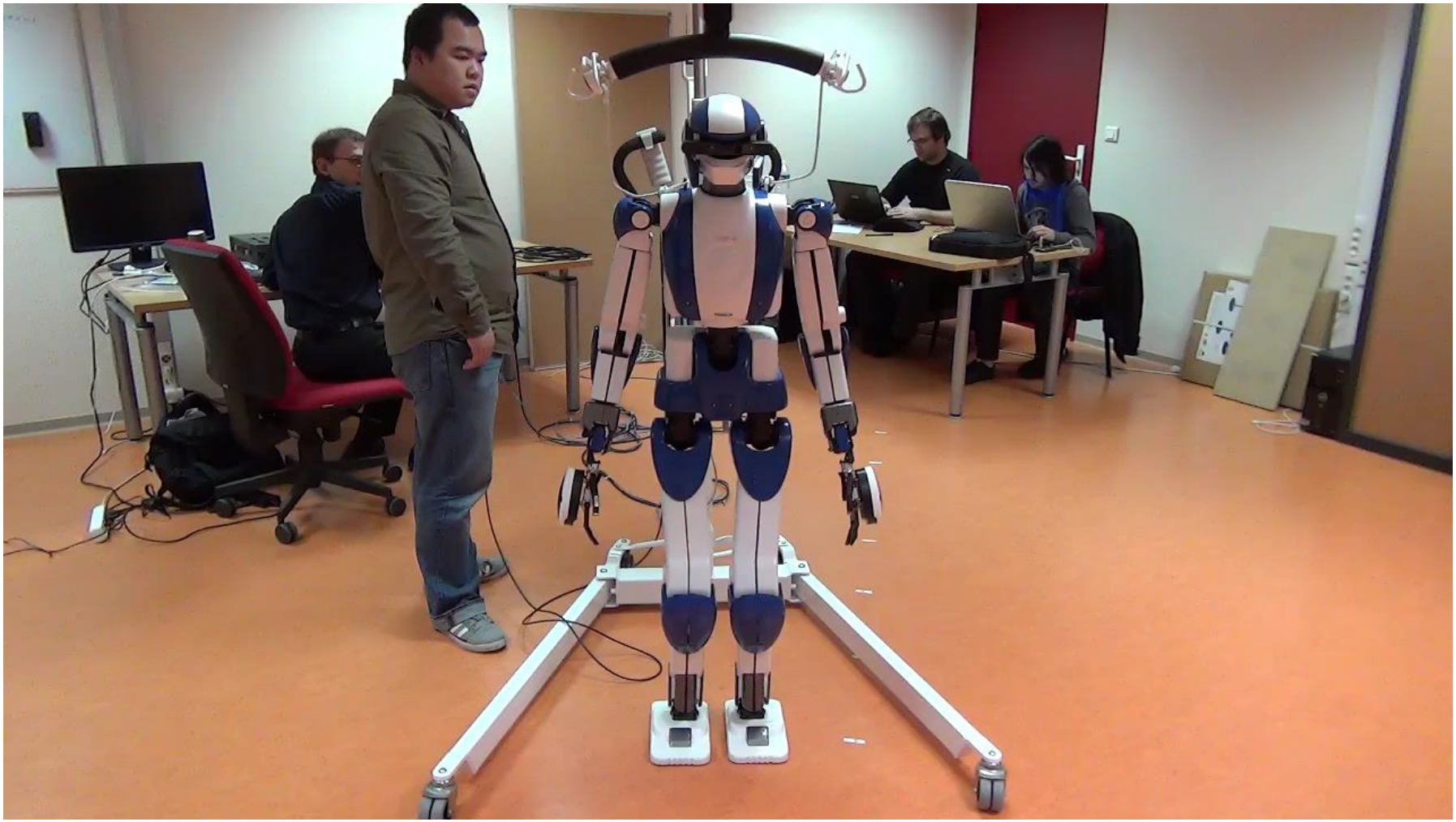
FSM



- Pro: Faster computation allowing to fit in 5 ms.
  - Whole control loop takes 1ms on a 2.6 GHz
  - 100 variables, 24 eq., 70+ ineq constraints
- Cons: Manual gain parameterization

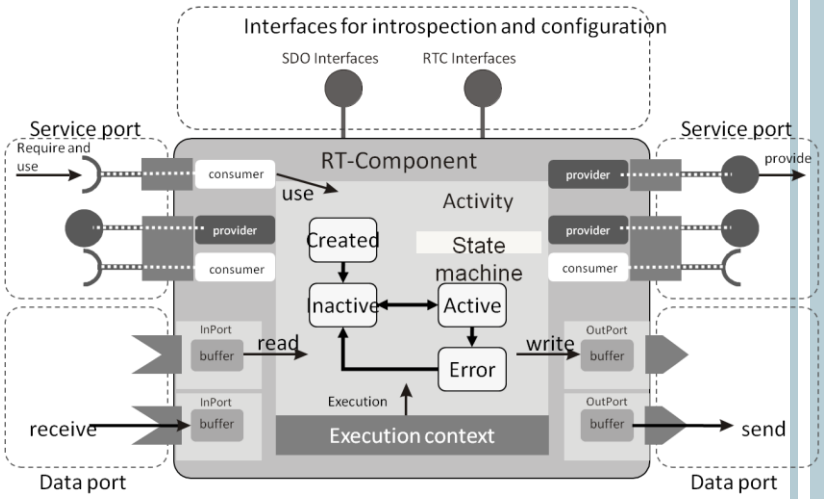
MULTI-CONTACT CONTROL

VALIDATION OF THE CONTROL ON THE ROBOT



Middleware: OpenRTM-aist

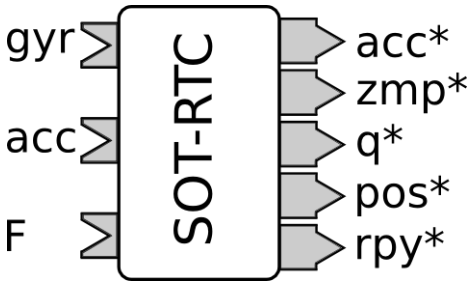
- Middleware embedded on HRP4
- Component-based framework
- Corba based
  - Service mechanism (Request/reply)
  - Point to point transport
- Allows real time control



Representation of a RT-component

Embedding the controller in OpenRTM

- Realize the computation directly in the real time loop
- Stack-of tasks
  - ✓ easy to design and implement
  - ✖ Computation not realized in 5ms
    - Computer not powerful enough
    - Use of non-optimized dynamic library.



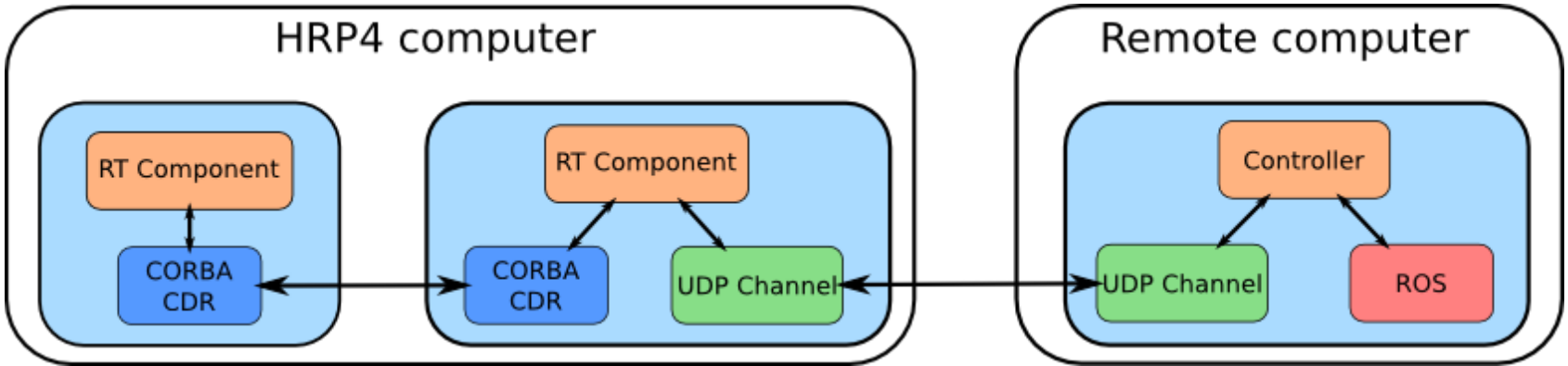
- Multi-contact solver
  - ✖ Impossible to embed the controller (computation in more than 5ms)
  - ✖ Impossible to use ros messages to communicate (ros is not real time).

OpenRTM-aist with UDP

- Computation realized on a remote computer
- Two servers are wrapped in the RTM component in the robot. Test of the following transfert protocols
  - TCP: Discontinuities (step- like shape of the control)
  - UDP(+): better results.

✓ Allow to overcome the limitations of the internal computer of the robot (software and hardware)

✖ Delay of a tick between the control and the corresponding sensors values



Conclusion and discussion

- Robot (finally) complete
  - Now able to do proper force sensing
  - Impedance control
  - Real multi-contact motions
- Two working controllers, but that cannot be embedded.
- Remote control (but presents some flaws)
- Potential solutions:
  - Enhance hrp4 hardware.
  - Use the openrtm-aist / ros bridge
    - Allow to directly interact with start via ros
    - Heavy
  - Use the ros_control interface?

Merci pour votre attention