Control and Perception Architecture for the Tele-Operation of Humanoid Robot COMAN

E. Mingo Hoffman, A. Rocchi, A. Settimi, M. Ferrati,
C. Pavan, A. Ajoudani, J. Lee, Z. Li, L. Muratore, A. Cardellino, N. G. Tsagarakis
and all the WALK-MAN Team

IN SUPREMÆ DIGNITATIS
1343

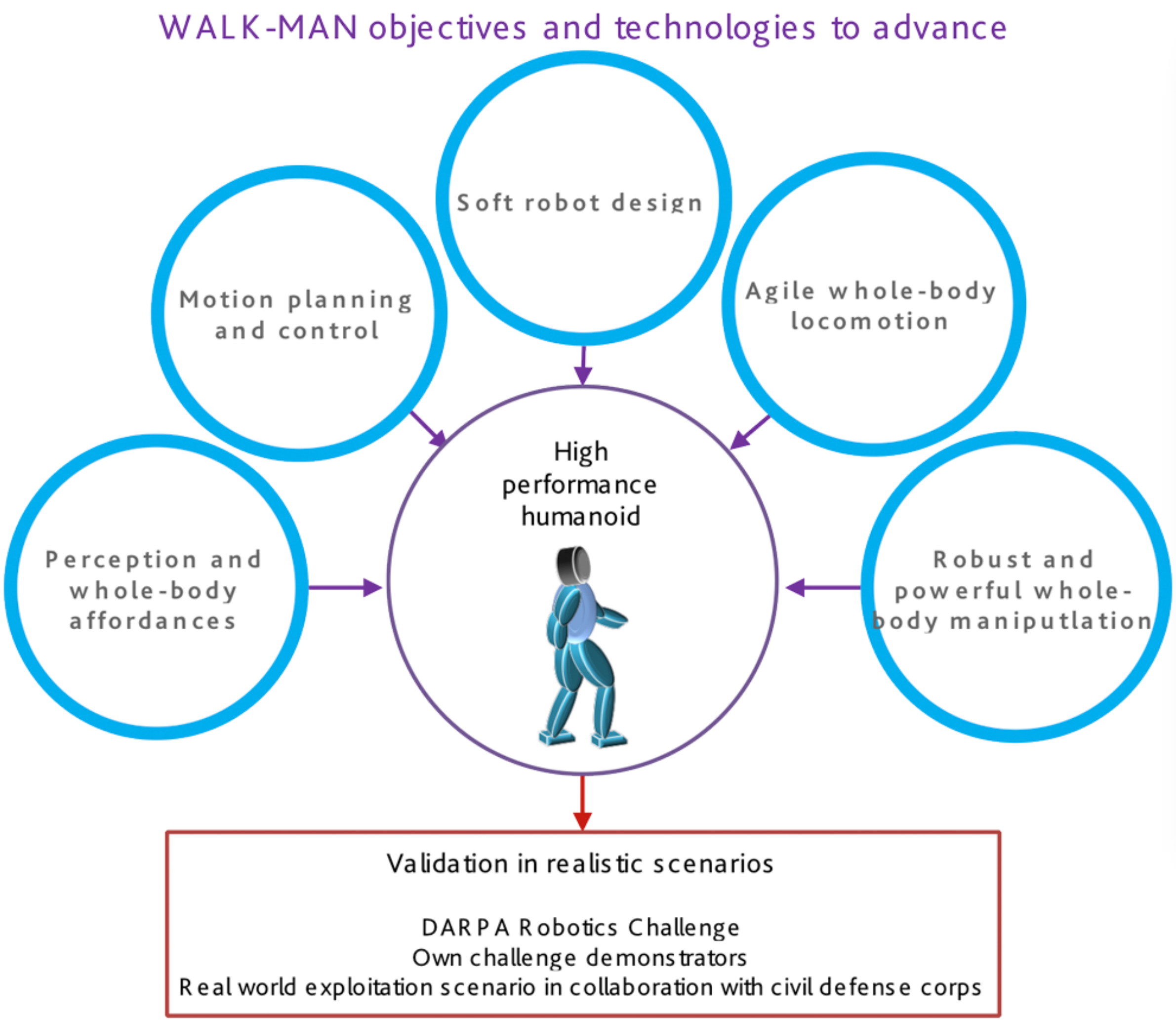
ROSCon

12-13 September 2014,

Chicago

ISTITUTO ITALIANO
DI TECNOLOGIA

COMAN & The WALK-MAN Project



WALK-MAN objectives and technologies to advance

Soft robot design

Motion planning and control

Agile whole-body locomotion

Perception and whole-body affordances

High performance humanoid

Robust and powerful whole-body maniputlation

Validation in realistic scenarios

DARPA Robotics Challenge

Own challenge demonstrators

Real world exploitation scenario in collaboration with civil defense corps

Scenario

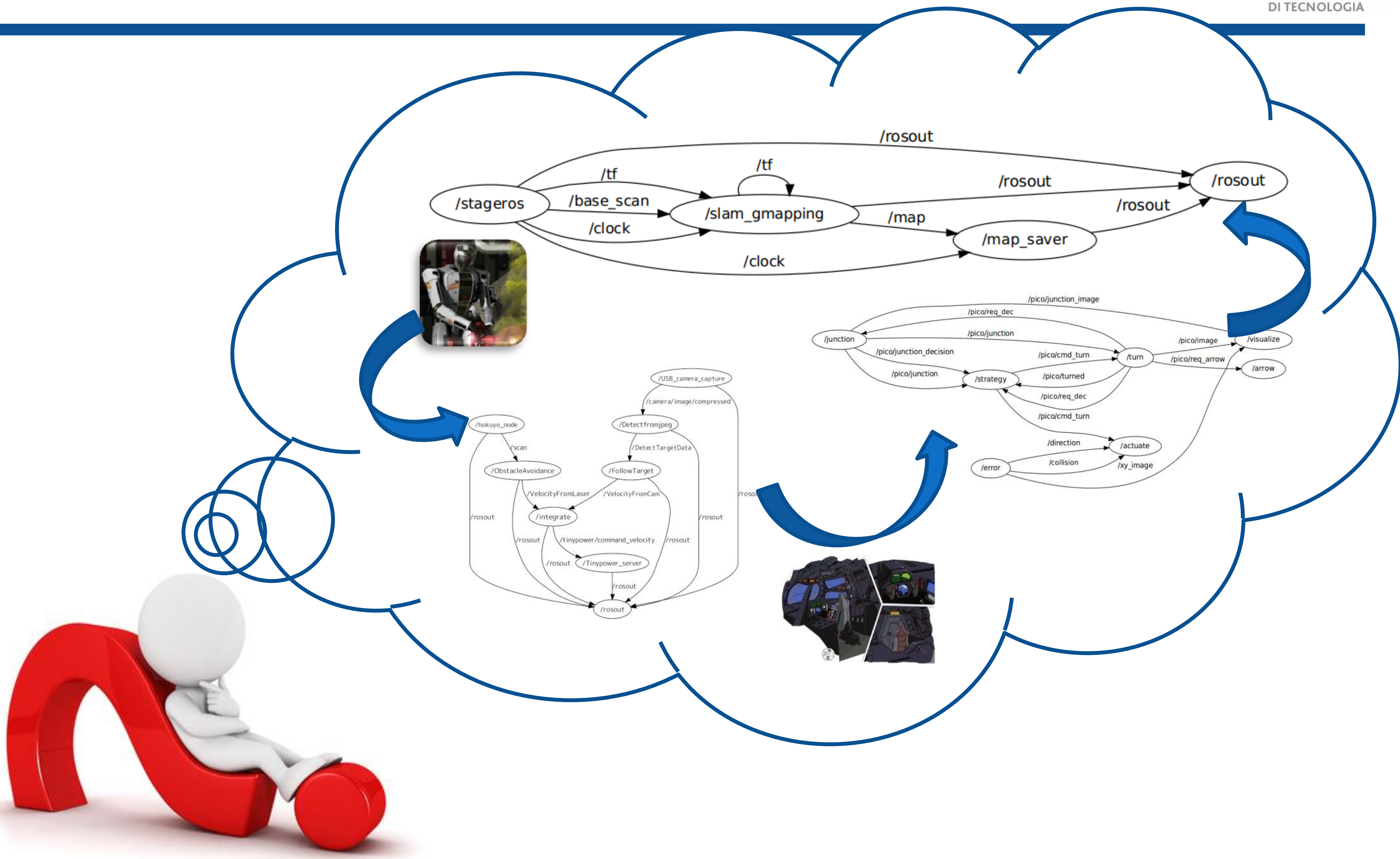
Command & Control Station

- monitoring & diagnostic
- different levels of control
- high computational power

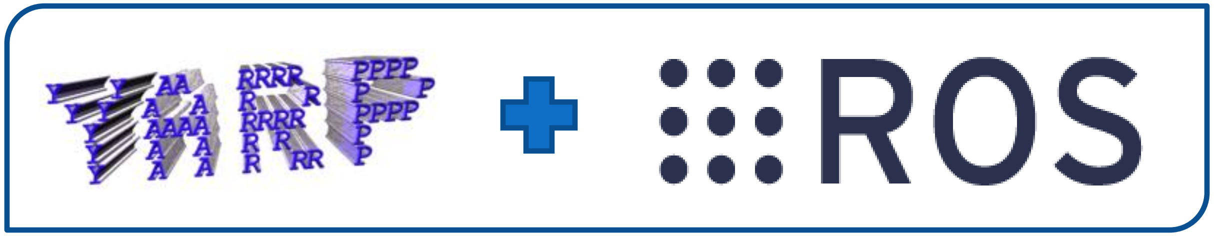
Humanoid Rescue Robot

- semi – autonomous
- operational space tasks
- robust
- basic environment recognition

Architecture



Our choice: Frameworks

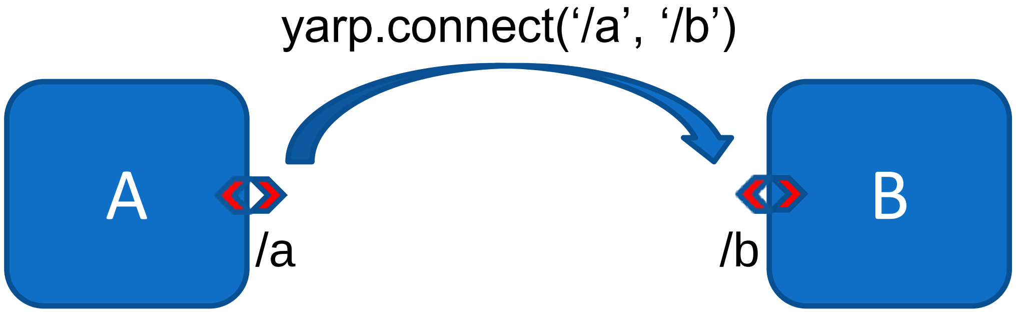


Control modules: YARP based + ROS facilities

Perception modules: ROS based

Visualization & Diagnostic: ROS based

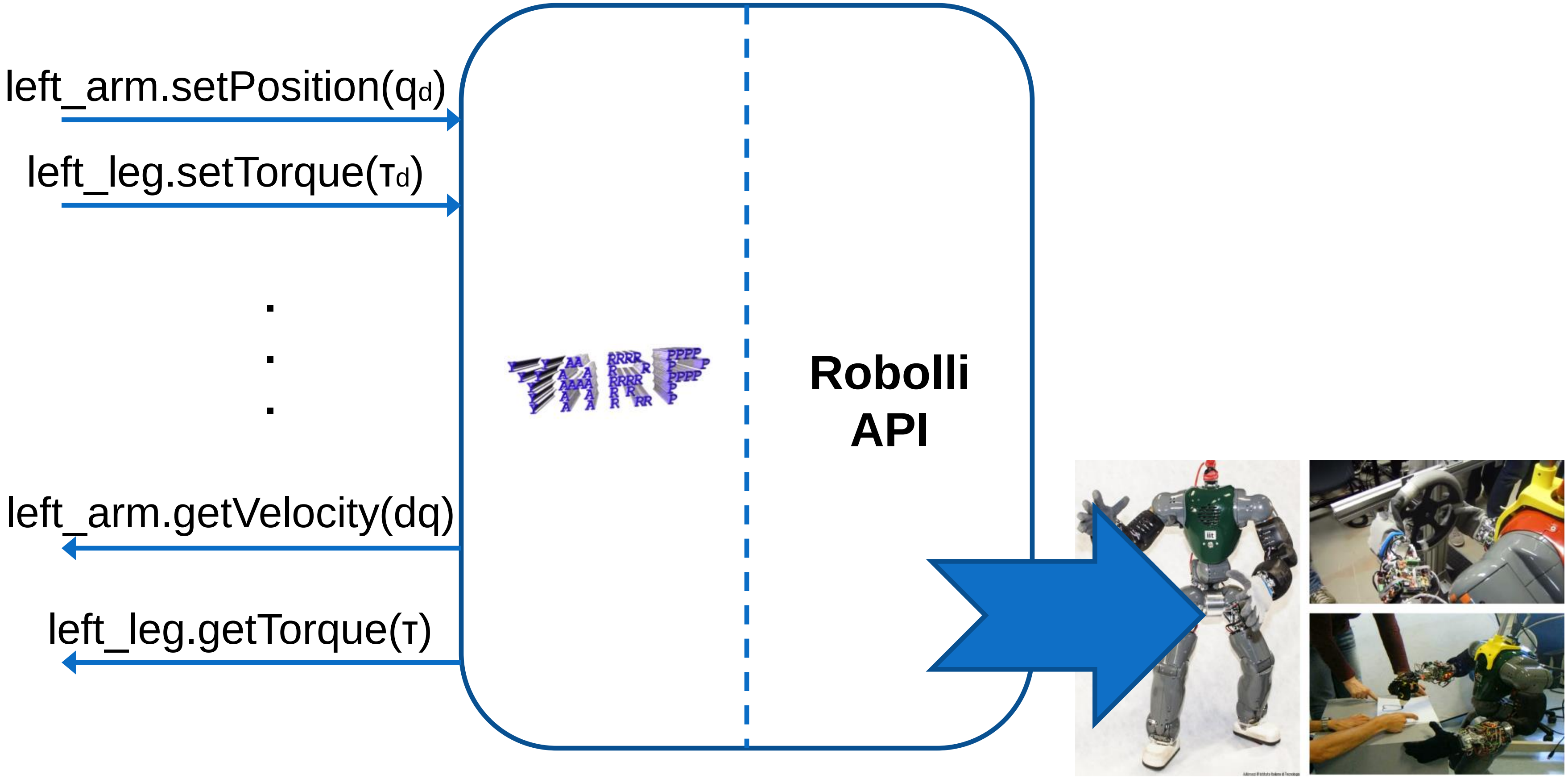
YARP



- Ports, RPCs, Interfaces
- TCP/IP, UDP, ShMem, …
- Typed/Non-Typed ports
- Successfully used in iCub and many other robots

http://wiki.icub.org/yarp/

COMAN Interface



left_arm.setPosition(q_d)

left_leg.setTorque(τ_d)

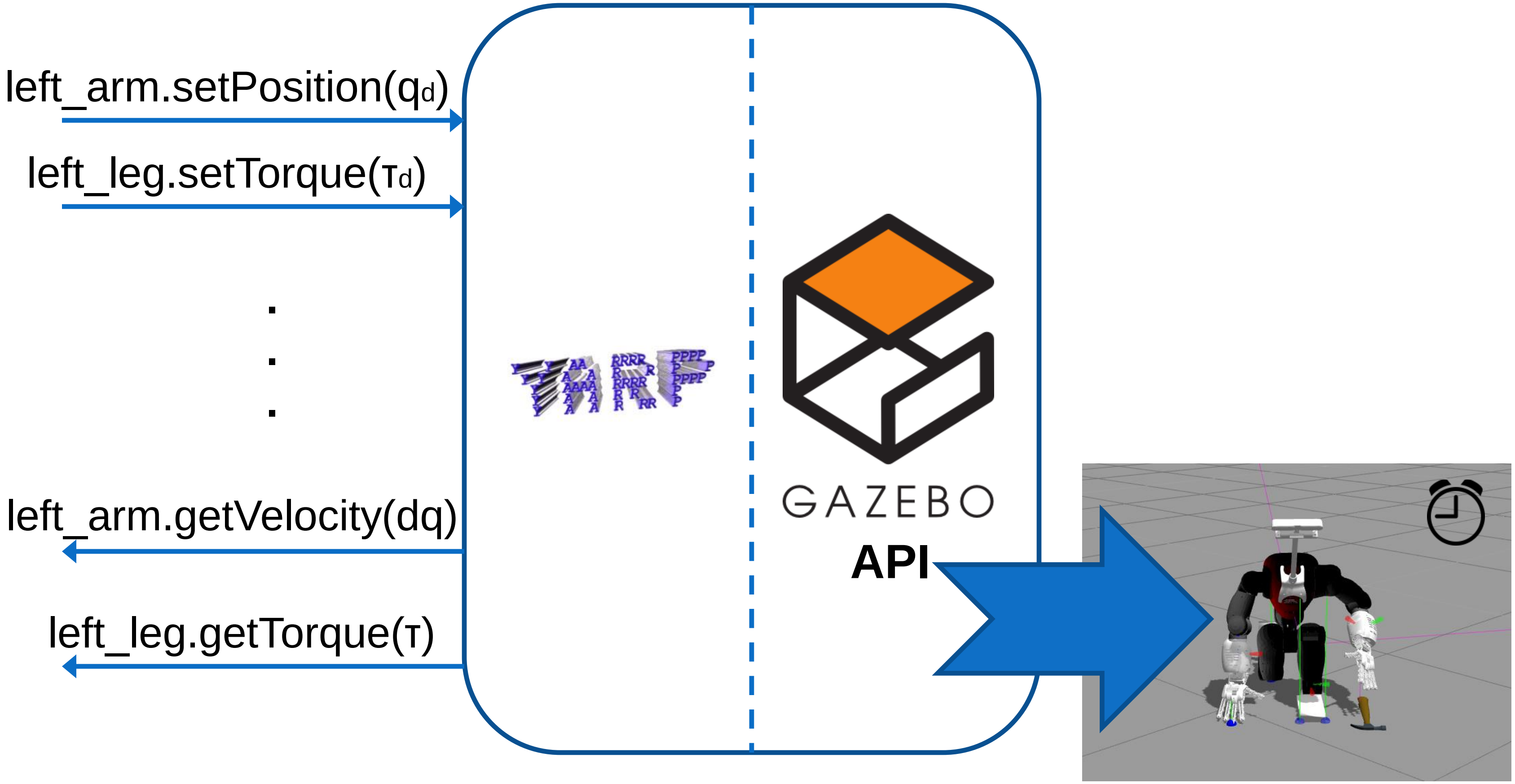
.
.
.

left_arm.getVelocity(dq)

left_leg.getTorque(τ)

Robolli API

gazebo_yarp_plugins



left_arm.setPosition(q_d)

left_leg.setTorque(τ_d)

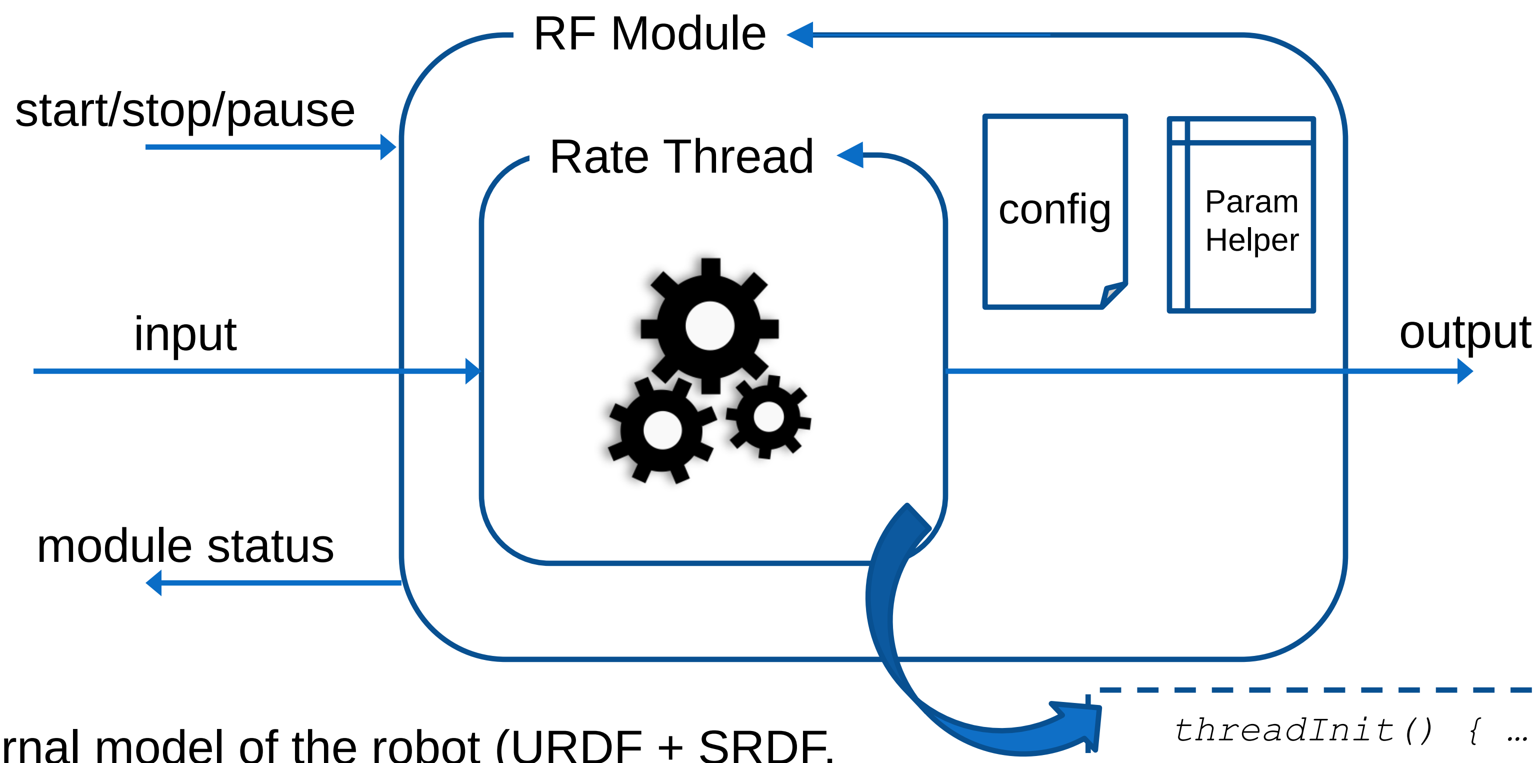
.
.
.

left_arm.getVelocity(dq)

left_leg.getTorque(τ)

https://github.com/robotology/gazebo-yarp-plugins

Control Modules (YARP)

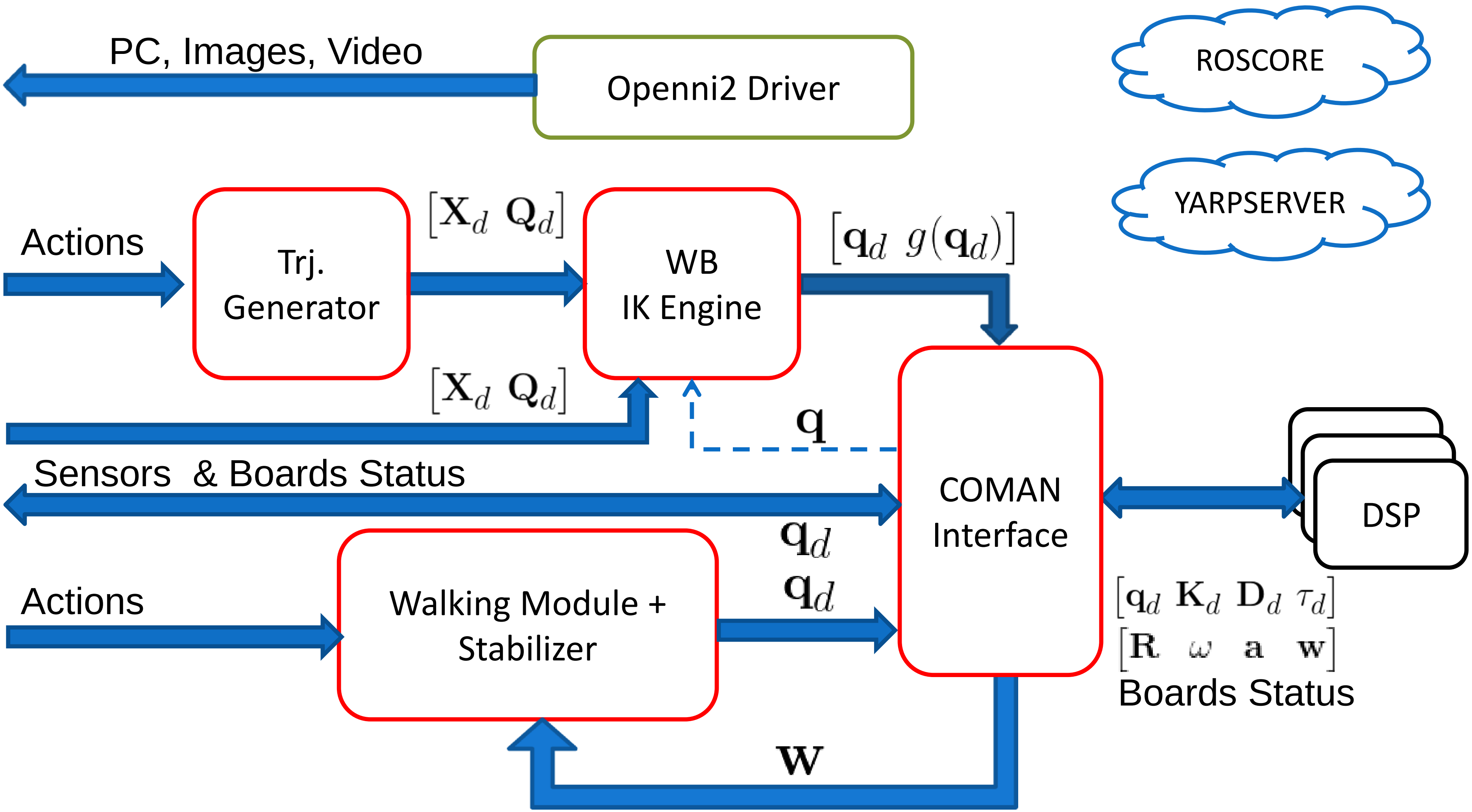


```
threadInit()  { … }

run()  { getFeedback()
         updateModel()
         ctrlLaw()
         move()}

close()  { … }
```

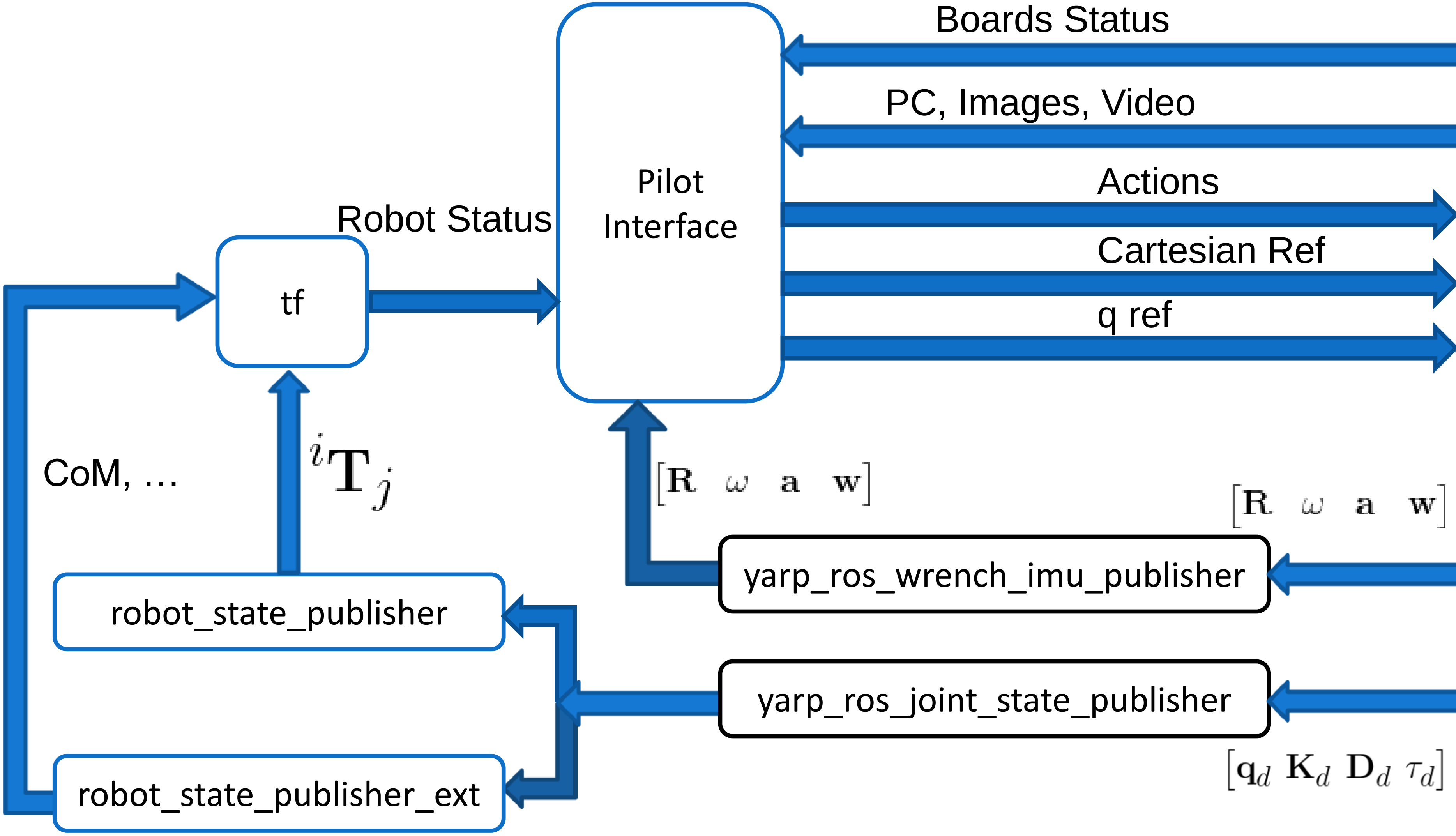
- Internal model of the robot (URDF + SRDF, REP 120)
- Decentralized IK/FK + ID (iDynTree)
- All kinematic chains controllable

https://github.com/robotology-playground/idyntree

Robot Modules


PC, Images, Video
Openni2 Driver
ROSCORE
YARPSERVER
Actions
Trj. Generator
$[\mathbf{X}_d \ \mathbf{Q}_d]$
WB IK Engine
$[\mathbf{q}_d \ g(\mathbf{q}_d)]$
$[\mathbf{X}_d \ \mathbf{Q}_d]$
$\mathbf{q}$
Sensors & Boards Status
COMAN Interface
DSP
$\mathbf{q}_d$
$\mathbf{q}_d$
Actions
Walking Module + Stabilizer
$[\mathbf{q}_d \ \mathbf{K}_d \ \mathbf{D}_d \ \tau_d]$
$[\mathbf{R} \ \ \omega \ \ \mathbf{a} \ \ \mathbf{w}]$
Boards Status
$\mathbf{w}$

Command & Control Station Modules



Boards Status

PC, Images, Video

Actions

Cartesian Ref

q ref

Pilot Interface

Robot Status

tf

CoM, …

${}^{i}\mathbf{T}_{j}$

$[\mathbf{R} \quad \omega \quad \mathbf{a} \quad \mathbf{w}]$

$[\mathbf{R} \quad \omega \quad \mathbf{a} \quad \mathbf{w}]$

yarp_ros_wrench_imu_publisher

robot_state_publisher

yarp_ros_joint_state_publisher

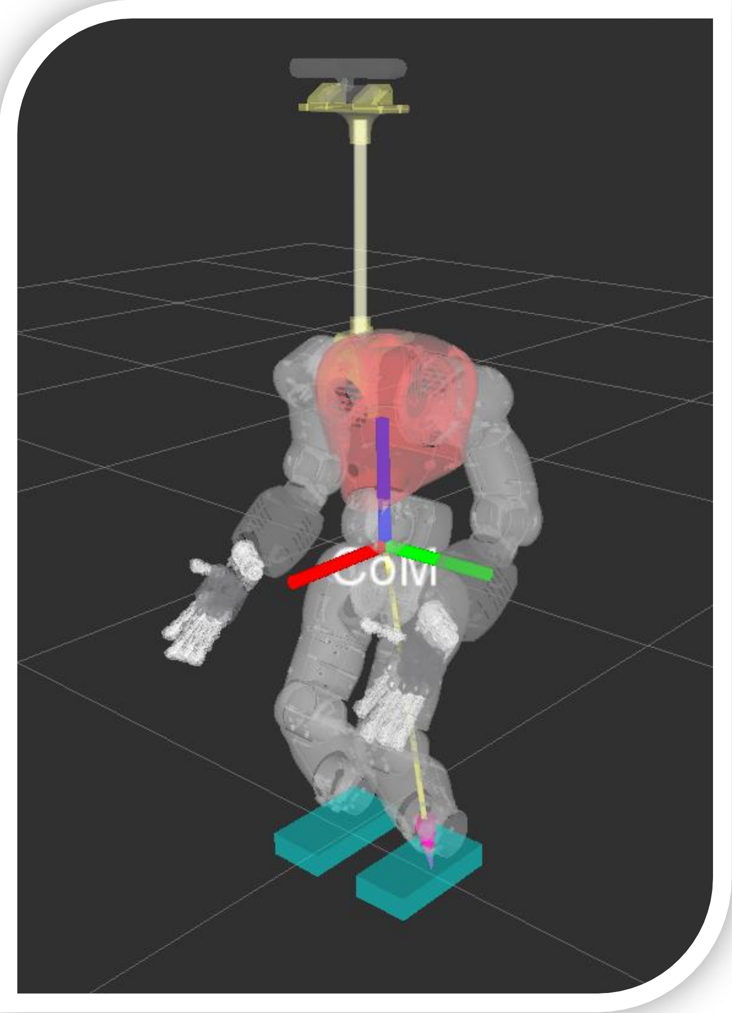
$[\mathbf{q}_d \; \mathbf{K}_d \; \mathbf{D}_d \; \tau_d]$

robot_state_publisher_ext

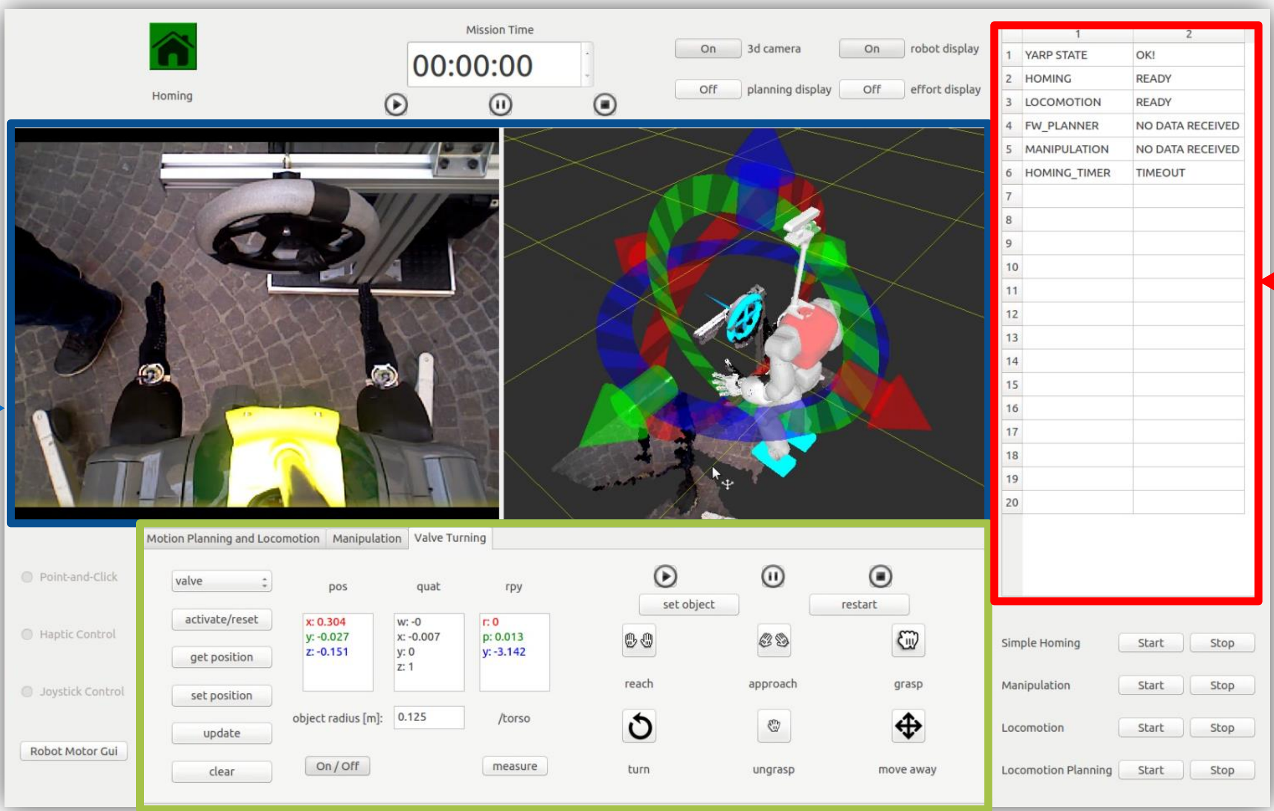
robot_state_publisher_ext

ROS Node similar to robot_state_publisher with more informations related to Humanoid Robots:

- CoM Position
- CoM Projection
- Convex Hull
- CoP
- …



Pilot Interface



Homing

Mission Time

00:00:00

On 3d camera

On robot display

Off planning display

Off effort display

	1	2
1	YARP STATE	OK!
2	HOMING	READY
3	LOCOMOTION	READY
4	FW_PLANNER	NO DATA RECEIVED
5	MANIPULATION	NO DATA RECEIVED
6	HOMING_TIMER	TIMEOUT
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		

○ Point-and-Click

○ Haptic Control

○ Joystick Control

Robot Motor Gui

Motion Planning and Locomotion | Manipulation | Valve Turning

valve

activate/reset

get position

set position

update

clear

pos: x: 0.304, y: -0.027, z: -0.151

quat: w: -0, x: -0.007, y: 0, z: 1

rpy: r: 0, p: 0.013, y: -3.142

object radius [m]: 0.125

/torso

On / Off

measure

set object

restart

reach

approach

grasp

turn

ungrasp

move away

Simple Homing Start Stop

Manipulation Start Stop

Locomotion Start Stop

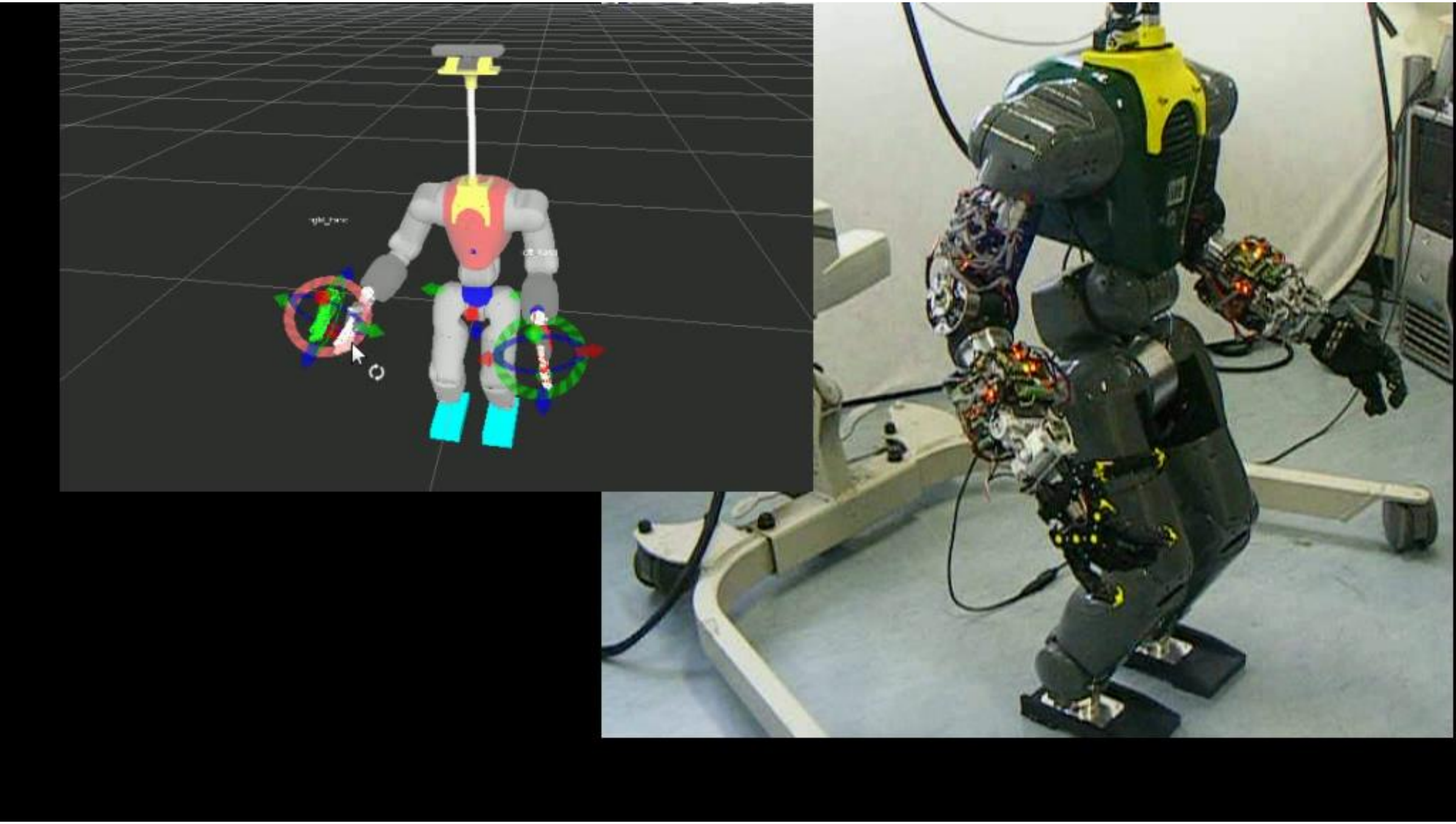
Locomotion Planning Start Stop

Module status

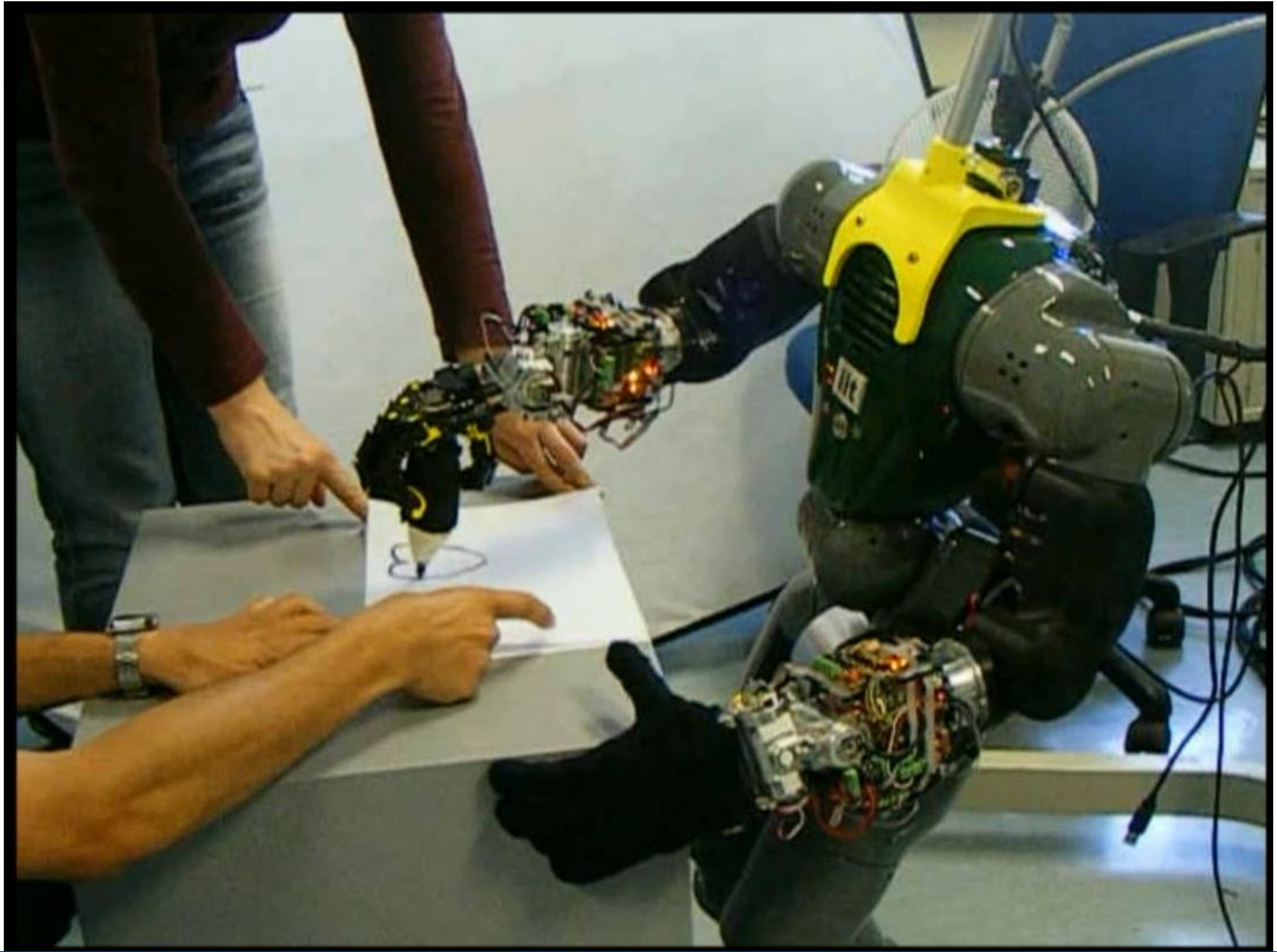
Cameras, PC,
Cartesian Control, Status, …

Module’s related commands

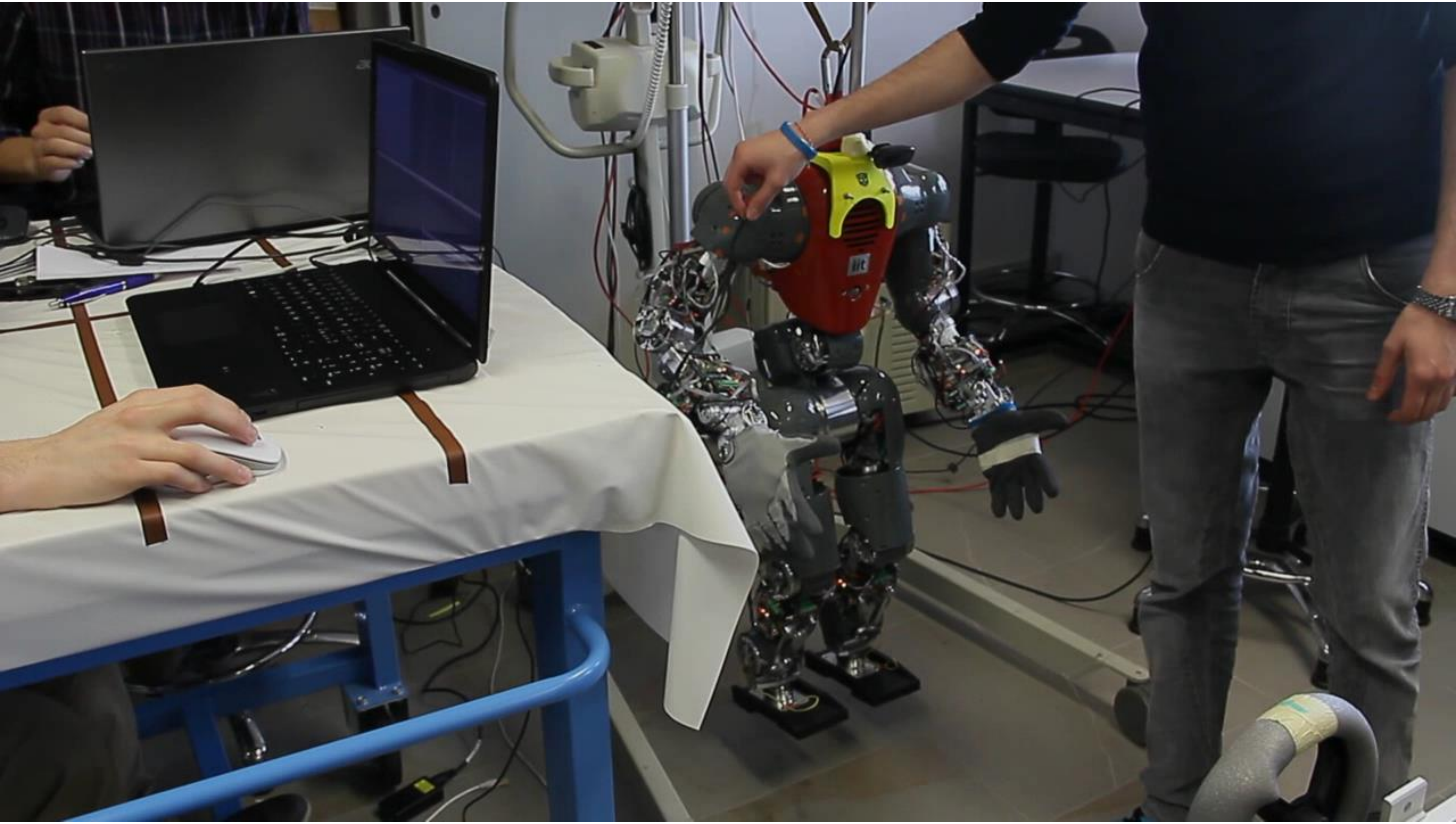
+ sot_VelKinCon



+ sot_VelKinCon



+ Valve



Conclusion & Future Work

- **Final Remarks**
  - Dedicated build system (SuperBuild)
  - Automatic generation of URDF/SRDF
- **Future Work**
  - Network management
  - Advanced perception
  - More autonomy in task execution
  - Automatic generation of modules