the iCub project
an open humanoid platform

Sestri Levante
July 21th, 2011

Giorgio Metta

Cognitive Humanoids Laboratory
Robotics, Brain and Cognitive Sciences dept.
the Italian Institute of Technology
• rbcs (in short)
  – neuroscience
  – robotics

• robotics
  – iCub intelligence
  – iCub hardware
  – iCub software
  – iCub production
Cognitive Systems and Robotics in FP7 (2007-2012)

<table>
<thead>
<tr>
<th>Work Programme</th>
<th>Objective</th>
<th>Call (Evaluation)</th>
<th>Budget</th>
<th>Projects: ACS &amp; Robotics (total)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ICT Call 3 (2008)</td>
<td>97 M€</td>
<td>17 (23)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ICT Call 6 (2010)</td>
<td>80 M€</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ICT Call 9 (2012)</td>
<td>82 M€</td>
<td>??</td>
</tr>
</tbody>
</table>

*) No more interaction since 2009 - language-based interaction in a separate objective with its own budget
strategic research agenda for robotics in Europe

- RTD **strategy** document developed by EUROP members in the CARE project (FP6)
- **industry-driven**, based on extensive analysis of market development and future opportunities
- **commitment** of all European stakeholders
- **public release** in Brussels on July 7, 2009

http://www.robotics-platform.eu
flexible object manipulation, packaging industry, medical applications (e.g., prosthetic devices) (TOMSY)
Robot-arm/ML (ComPLACS)
Assembly (DARWIN/Intellact)

V-Charge: autonomous valet parking eCAR

SENSOR NETWORK/MONITORING/CONTROL INTELLIGENT BUILDING (iSense)

SURGICAL ROBOT (ACTIVE – I-SUR)

LOGISTICS (RobLog)

UNDERWATER (NOPTILUS / CoCoRo)
Search & Rescue/
Unmanned (EMICAB)
UAV – ML (ComPLACS)

ROBOT BUILDER (Goal-Leaders)

VR of Space manipulation (Intellact)

RUBICON: assisted living / transport of goods

CORBYS: MOBILE MANIPULATION / eSMCS / CORBYS REHAB
Xperience: HOUSEHOLD HELPER

NeuralDynamics: HR COOPERATION TABLETOP Robotics at

COMPLACS Assembly (DARWIN/Intellact)

GAME/ table-top

Game/ table-top

Game/ table-top

VR of Space manipulation (Intellact)

Flexible object manipulation, packaging industry, medical applications

(e.g., prosthetic devices) (TOMSY)

Robot-arm/ML (ComPLACS)

Assembly (DARWIN/Intellact)

V-Charge: autonomous valet parking eCAR

SENSOR NETWORK/MONITORING/CONTROL INTELLIGENT BUILDING (iSense)

SURGICAL ROBOT (ACTIVE – I-SUR)

LOGISTICS (RobLog)

UNDERWATER (NOPTILUS / CoCoRo)
Search & Rescue/
Unmanned (EMICAB)
UAV – ML (ComPLACS)

ROBOT BUILDER (Goal-Leaders)

VR of Space manipulation (Intellact)

RUBICON: assisted living / transport of goods

CORBYS: MOBILE MANIPULATION / eSMCS / CORBYS REHAB
Xperience: HOUSEHOLD HELPER

NeuralDynamics: HR COOPERATION TABLETOP Robotics at

COMPLACS Assembly (DARWIN/Intellact)
Call 7 outcome - SMEs

- Manipulation of deformable objects (folding/sorting clothes)
- Mobile manipulation in a restaurant
- Assistive robotics surgery
- Object hand-over
- Compliance & Modular design /assembly
- Safe Physical HRI
- Learning domestic tasks from web (cooking)
- Understanding and carrying out everyday activity (generalisation/learn action from spoken instruction/language-action link)
- Cognitive + Flexible robots for SMEs manufacturing
- Aerial cooperative manipulation
- Underwater (cooperative/advanced autonomy)
- Outdoor guide navigation in crowded environment + social HRI
- Intelligent prosthesis
- Safe Physical HRI
...a bit of (additional) history

- RobotCub (FP6): started 2004 – finished 2010, initial design
- ITALK: started 2008, extensions to language
- Poeticon: started 2008, supported the development of fingertips
- CHRIS: started 2008, supported the development of force control (for safety)
- RoboSKIN: started 2009, took over the development of a skin system
- Viactors: started 2009, study on intrinsic compliance and variable stiffness
- AMARSi: started 2010, compliance and learning, motor richness
- ImClever: learning and intrinsic motivations
- ROSSI: sensorimotor and social interaction
- Xperience: started 2011, cognitive architecture & affordances
- EFAA: started 2011, social interaction and learning from interaction
- Darwin: started 2011, manipulation and assembly
- Poeticon++: starting 2012, language and action
RobotCub goals

✓ design a **humanoid robot** platform, namely the iCub

✓ make it the **platform of choice** for researchers in artificial cognitive systems

✓ study **cognition** from a developmental perspective (neuroscience)
iCub community goals now

✓ maintain and improve the iCub to keep it alive

✓ make it the platform of choice for researchers in artificial cognitive systems

✓ study cognition from a multitude of points of view
iCub is an open source international endeavour initially funded by the EU project RobotCub

- a full humanoid robot
- is 104cm, weighs 22 kg
- has 53 degrees of freedom
- can crawl, sit and manipulate
- open design as LGPL/GPL/FDL
why is the iCub so special?

• **hands:** we started the design from the hands
  – 5 fingers, 9 degrees of freedom, 19 joints

• **sensors:** human-like, e.g. no lasers
  – cameras, microphones, gyros, encoders, force, tactile…

• **electronics:** flexibility for research
  – custom electronics, small, programmable (DSP)

• **reproducible platform:** community designed
  – reproducible & maintainable yet evolvable platform
why humanoids?

• scientific reasons
  – e.g. elephants don’t play chess

• natural human-robot interaction

• challenging mechatronics

• fun!
why open platforms?

• repeatable experiments
• benchmarking
• quality

this resonates with industry-grade R&D in robotics
development tools (in the case of the iPhone)
development tools (in our case)
Yet Another Robot Platform

- YARP is an open-source (LGPL) middleware for humanoid robotics

- history
  - an MIT / Univ. of Genoa collaboration
  - born on Kismet, grew on COG, under QNX
  - with a major overhaul, now used by the iCub project

- C++ source code (some 400K lines)
- IPC & hardware interface
- portable across OSs and development platforms
exploit diversity: portability

• operating system portability:
  – Adaptive Communication Environment, C++ OS wrapper: e.g. threads, semaphores, sockets

• development environment portability:
  – CMake

• language portability:
  – via Swig: Java (Matlab), Perl, Python, C#
The iCub online manual:
Software documentation:
http://icub.org ➔ iCub software
Software

A container for all software in $ICUB_ROOT/src. More...

Modules

- **iCub modules**
  This is the list of iCub modules, i.e.

- **Graphic User Interfaces**
  This is a list of Graphic User Interfaces.

- **Libraries**
  This is a list of static libraries.

- **iCub mod library**
  Objects that can be conditionally compiled in the main build.

- **Tools**
  This is a list of iCub basic tools.

Detailed Description

A container for all software in $ICUB_ROOT/... For clarity this is further divided in groups.

X Find: documenta next previous highlight all match case

Done

autoAssociativeMemory

The auto-associative memory (AAM) module - autoAssociativeMemory - effects the following functionality: More...

The auto-associative memory (AAM) module - autoAssociativeMemory - effects the following functionality:

- when an image is presented to the memory, it attempts to recall that image;
- if a previously-stored image matches the presented image sufficiently well, the stored image is recalled;
- if no previously-stored image matches sufficiently well, the presented image is stored;

Since images are streamed continuously, processing an image (i.e. reading it from the input port and attempting to recall it from the memory) is triggered by the presence of the threshold value on the input port. The next image to be read and stored/recalled is only processed when a new threshold is written to that port.

The AAM has the following inputs:

- an input image
- a tuple containing an image id, number and a match value r, 0 ≤ r ≤ 1. These are type integer and double respectively.

**Dependencies**

YARP.

**Parameters**

Command Line Parameters

The following key-value pairs can be specified as command-line parameters by prefixing -- to the key (e.g. --from file.ini). The value part can be changed to suit your needs; the default values are shown below.

- from autoAssociativeMemory.ini specifies the configuration file
- context autoAssociativeMemory/conf specifies the sub-path from $ICUB_ROOT/icub/app to the configuration file
- name aam specifies the name of the module (used to form the stem of module port names)
Applications

$ICUB_ROOT/app/ stores directories which group sets of configuration files and scripts to run a certain group of modules together. More...

Modules

armCartesianController
An application that enables the user to select objects lying on a table, in order to steer the robot gaze to them and ultimately command a grasp.

attentionDistributed
This is the application used to run the attention system, distributed on all ICUB slave nodes.

cartesianSolver
The Cartesian Solvers Launcher.

crawlingDemo
A set of modules for crawling and reaching on all fours.

crossPowerSpectrumVergence
Starts the crossPowerSpectrumVergence module.

default
Place here default configuration files and scripts.

demoAAM

A set of modules for crawling and reaching on all fours. More...

A set of modules for crawling and reaching on all fours.

Description

The Crawling application consists of a set of modules that are used for crawling, steering and reaching on all fours (please refer to Deliverable 3.8, for more information). The architecture can be extended to visually guided crawling.

The general implementation for the basic crawling architecture is pictured below:

The four limbs as well as the head and the torso are controlled. For each part, the trajectories of all the dofs are produced by a CrawlGenerator module (referred to as CPG on the figure), some of them being coupled together to ensure the gait (green arrows). On top of these, the CrawlManager module is responsible for sending the control (i.e., high-level) parameters corresponding to the different behaviors.

The behavior of the robot can be modulated by the user using the keyboard, more precisely:
the entire project is under LGPL/GPL/FDL
iCub 2.0 sneak preview

new mechanics

compliant actuators

force/torque measurements
skin

principle

lots of sensing points

structure of the skin
fingertips

• capacitive pressure sensor with 12 sensitive zones
• 14.5 mm long and 13 mm wide, sized for iCub
• embedded electronics: twelve 16 bit measurements of capacitance
  – either all 12 taxels independently at 50 Hz or an average of the 12 taxels at about 500 Hz
joint torque sensors
learning new actions
YARP progress (past year)

- Major rewrite of the **build system** using newer CMake features
- Official support for **MacOS**
- Bug fix, support
- **New test system** using CTest: http://dashboard.icub.org
- RPC client servers classes
- Persistent nameserver
- **Compatibility with other system** (e.g. ROS protocol)
- Change of licensing from GPL to **LGPL**
- **Port authentication** mechanism using HMAC with SHA
- **Binaries**, yes, eventually!
iCub progress (past year)

- Better **coexistence with external projects**
  - Now possible to use the iCub software as a CMake package
  - Separation between “main” and “contributed” software
- Support for new hardware (torque sensors, skin, new canbus device)
- **New modules** (force control, machine learning, inverse kinematics, logpolar attention system)
- Bug fixes and support
- Constant improvement of the manual and documentation
- Make **installation easier**: binaries of dependencies for windows
- New test system: http://dashboard.icub.org
- Work in progress:
  - better usability, gui, new simulator
  - binaries
New installation procedure

Scalability: coexistence with other projects

iCub applications & modules

iCub repository

main

contrib

other projects/repositories

EU project repositories using svn externals to connect to iCub main/contrib
...why is this important?
Publish or perish

• Successful researchers learn to maximize the number of published papers
• **Loss of work and knowledge**
  – code is lost
  – “implementation details” get lost

...but maybe the “implementation details” are what made the algorithm work!
Some questions

• How much time does it take to implement an algorithm from a paper? Does it always work?

• How difficult it is to choose the algorithms that work best?
Serious problems
Serious problems

• Lack of reward for producing reusable code
Serious problems

- Lack of reward for producing reusable code
- No mechanism for promoting successful software components
Serious problems

- Lack of reward for producing reusable code
- No mechanism for promoting successful software components
- Difficulty to compare experiments
Serious problems

- Lack of reward for producing reusable code
- No mechanism for promoting successful software components
- Difficulty to compare experiments
- No incremental progress
Serious problems

- Lack of reward for producing reusable code
- No mechanism for promoting successful software components
- Difficulty to compare experiments
- No incremental progress
- Our students tend to get stuck in “simple” problems
iCub community (snapshot at vvv10)
• 14 developers contributed code to YARP in the past 12 months
• 48 developers contributed code to iCub repository in the past 12 months
A new paradigm?

• Within the iCub community we have a great opportunity
A new paradigm?

- Within the iCub community we have a great opportunity
- See science as “what works”, peer usage as opposed to peer review
A new paradigm?

• Within the iCub community we have a great opportunity
• See science as “what works”, peer usage as opposed to peer review
• Don’t stop at experiments, think seriously about code development
A new paradigm?

• Within the iCub community we have a great opportunity

• See science as “what works”, peer usage as opposed to peer review

• Don’t stop at experiments, think seriously about code development

• **Publish** the code you write, make it **reusable**, write **documentation**
A new paradigm?

COMMIT or PERISH!!
Code Darwinism, the survival of the fittest code
...one of the largest open source teams in the world, it ranks in the top 2% of all projects teams registered on Ohloh.
more challenges for the future

new actuation?

new sensors?

new materials?
maybe robotics is next
sponsors

- EU Commission projects:
  - RobotCub, grant FP6-004370, http://www.robotcub.org
  - CHRIS, grant FP7-215805, http://www.chrisfp7.eu
  - ITALK, grant FP7-214668, http://italkproject.org
  - Poeticon, grant FP7-215843 http://www.poeticon.eu
  - Robotdoc, grant FP7-ITN-235065 http://www.robotdoc.org
  - Roboskin, grant FP7-231500 http://www.roboskin.eu
  - Xperience, grant FP7-270273 http://www.xperience.org
  - EFAA, grant FP7-270490 http://efaa.upf.edu/

- More information: http://www.iCub.org