RH2-Y – Toward A Cooperating Robot for Home Applications

Jean-Daniel Dessimoz, Pierre-François Gauthey

Robocup-at-Home League, Proceedings Robocup World Competition 2007,
Georgia Tech, Atlanta, USA, 30 June-10 July 2007

RH2-Y – Toward A Cooperating Robot for Home Applications

Jean-Daniel Dessimoz, Pierre-François Gauthey

HESSO.HEIG-VD West Switzerland University of Applied Sciences, CP,
1400 Yverdon-les-Bains, Switzerland
{Jean-Daniel.Dessimoz@heig-vd.ch, Pierre-Francois.Gauthey@heig-vd.ch); http://www.iai.heig-vd.ch/)

Abstract. RH2-Y is an autonomous robot for home applications, with flexible architecture, distributed processors, multi-agent and real-time programming environment, robust communication and implementation, cognitic properties of high level. Hardware components include camera with onboard processor for vision; microphone; PLC for robust signal acquisition and generation, as well as for distributed, fast reflex programming; servoboard for individual joint control; planar laser; arm and hand; and small-sized PC; mostly interconnected with TCP/IP Ethernet protocol. Among software components, about 20 agents switch their individual tasks at high rate. Thorough visualization and simulation capabilities have been developed for man-machine interface, mostly for development and configuration phases. The programming environment includes real-time, hierarchical trajectory definition, compound location support and motion primitives with industrial robotics flavor. User interaction can be done with a variety of channels, and programmer's interaction is in particular possible in four steps of increasing complexity, depending on needs and operator expertise.

1 Introduction

The subject of the paper, RH2-Y team description, is at the frontier of new developments, where automation reaches cognitive tasks, i.e. cognitics [e.g. 1], and practical solutions touch autonomous operations in the physical world, including robotics at home [2].

Overlap between cognitics and mobile robotics is obvious. Competitions in mobile robotics have progressively appeared and developed. They are useful in order to provide benchmarks, to foster exchanges and to force a systemic, holistic view on what an autonomous system must be and do.

RH2-Y is the current direct update of RH1-Y, which is a specific instance of our versatile platform for autonomous robots, ARY. ARY robot family has been progressively designed since 1998 (e.g. [1, 3, 4]). The core platform includes a number of interesting components and aspects, in terms of architecture, distributed processors, multi-agent and real-time programming environment, robust communication and implementation techniques, as well as in terms of cognitic properties of high level or of classical robotic approaches.

2 Major hardware components

The architecture of RH2-Y distributes functions on several hardware components each controlled by their own processors. Communication is handled differently depending on the proximity to physical world. Locally, lower level procedures and standards are ad hoc, adapting in the best possible way to specific constraints. At higher level, Ethernet TCP-IP communication paths are provided, interconnected with a hub.

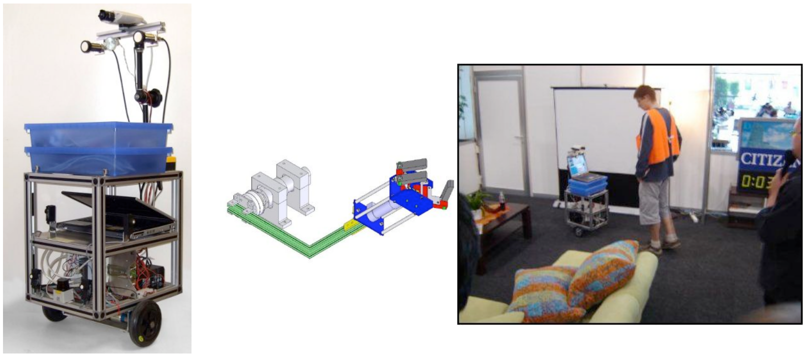


Fig.1 Early view of RH2-Y (left), with arm and hand (center) and a view of RH1-Y in Bremen Robocup competition 2006, Follow-a-Human task (right)

Major hardware components include a camera with onboard processor for vision ; a PLC for robust signal acquisition and generation, as well as for distributed, fast reflex programming ; servoboards for individual joints (wheels, arm and hand); and a small-sized PC .

The camera has wide angle lens, is configured in low resolution RGB (80 to 320 columns). Embedded processors continuously compensate for lighting conditions.

The motion controller guarantees position of 2 motors, drives 2 active wheels. It can detect the direction of motion and integrate wheel rotation increments, control the individual wheel motion, with predefined maximum speed and acceleration features during motion. Arm and hand will be individually controlled with Fiveco units.

In principle 2 passive wheels are provided for balance and stability. One of them is mounted on a flexible suspension with damper, in order to reduce the risk of hyperstatics, i.e. to keep a minimal load on ground through the motorized wheels, while keeping as high as possible the eigenfrequency of the platform , for accuracy purpose.

The PLC drives digital output signals, and optically reads isolated Boolean input signals, and get analog signals as well, typically transmitted by sensors. Local computing is possible on specific, local CPU.

The supervisory computer is a small sized computer, with all the usual features (hard drive, memory extensions, battery packs, Ethernet ports, etc.).

For external communication, a wireless port is another important hardware component which is sometimes used. Additional resources for vocal dialogue are considered.

3 Major software components

RH2-Y features various types of software components, depending on the processor considered. Low-level components have embedded processors, mostly with their specific environment: linux-based for camera processor; proprietary internal and user-level environments for motion controller; for PLC, IEC-61'131 standard "programs" can be defined and uploaded. The most interesting software components however are to be found on the supervisory computer.

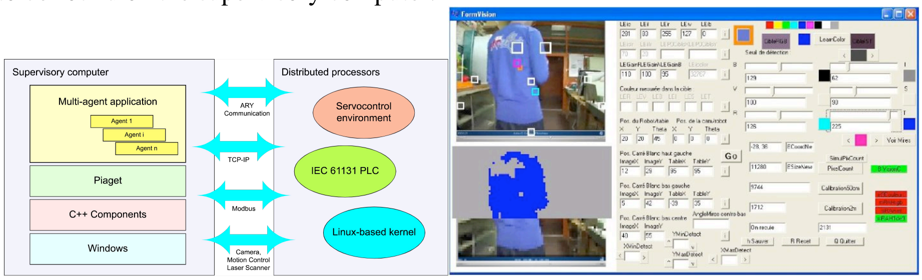


Fig. 2). Overview of control resources and architecture of RH2-Y (left). Configuration board for visual tracking (right)

On supervisory computer, our original Piaget environment is being used. In terms of software components, the platform includes about 20 agents, switching their tasks by time slices of about one hundred nanosecond duration.

Several agents are responsible for motion control. The software system is capable to control a good variety of motorized solutions: typically linear and circular interpolation, ARY redundant, multi-phase control strategy for non-holonomic platform (as well as low-level stepper motor and DC-motor management in some former cases of parallel port interfacing).

Other agents deal with vision, I/O transfer and processing, command interpretation for communication with other robots and remote systems, as well as human-machine interfacing (see point 5), and strategy deployment.

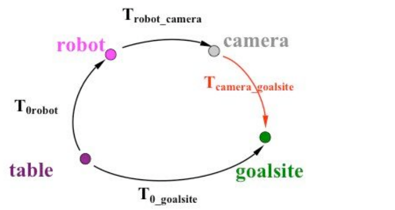
While the initial version of Piaget was implemented in Pascal and could run under Dos environment, more recent versions have been written in C++, and are compatible with Windows (version XP-Pro and other ones). Other implementations include a subset, "Piaget Light", written in C, running under RTOS on a PC-on-a-Chip.

4 Definition of locations and motion trajectories

A special feature of Piaget programming environment is to include real-time hierarchical trajectory definition, compound location support and motion primitives in a way similar to good practices in industrial robotics (e. g. VAL).

Piaget trajectory control includes primitives and procedures for Cartesian space coordinates, direct and inverse kinematics solutions, joint coordination as well as joint-level servocontrol. Implementation is hierarchical and relies on a distributed architecture.

Typically, two main categories of motions are provided. Mostly, use is made of our proprietary, 3-phase procedure (ARY). In this case, an intermediary step between joint coordinates (motorized wheel rotations, θ_r, θ_l) and application-oriented, Cartesian coordinates (x, y, α) is made, via 3 intermediate variables (θ_1, d_2, θ_3). The latter can be viewed as 3 virtual joints.



$$T_{0\_goalsite} = T_{0\_robot} \cdot T_{robot\_camera} \cdot T_{camera\_goalsite}$$

```
12: if(!SignalIn(NSIStart))
      GoState(6);
        else
      GoSTate(20);        break;case

20: ApproAGN(Table,30);  break;case
//Switch light on
21: SignalOutAGN(NSOLamp,true)
                          break;case
22: SleepAGN(0.05);      break:case
//Visual analysis of a row in scene
23: WatchRowAGN(R,CStart,CStop);
```

Fig. 3 Example of Piaget support for transformation calculus, and for computing compound locations: graphic representation of relative pose and matrix products and equation. Similarly, inverse matrices computation and implicit definition of approach and depart locations are integrated in the system (left). Example of Piaget langage, incl. approach motion primitive (right)

By this token, even though the system is non holonomic, motion is ensured redundantly in 3 subsequent phases, each of them characterized by 2 joints cooperating to effectively and successively implement one the virtual joint transformation (translation or rotation). This mode naturally yields linear motions in the plane. The second category of motions is of circular type.

5 User interaction

Home users can interact with RH2-Y addressing sensors: ultrasonic, laser-based, visual or phonic; and perceiving expressions: sound, display and motions. Only about 10 words are recognized at word level (re. MS SAPI 5.1), but with excellent synthesis capability, our multi-state dialogue manager makes those vocal primitives versatile.

Programmers can interact with the platform in several modes of increasing complexity, depending on needs and operator expertise. The modes range from step 1. Interactive configuration and parameterization, to 2. Strategy programming, in the application-oriented proprietary language, 3. Definition of new agents or low-level vision procedures, or 4. Changes in kernel functions, and all other items, typically defined in C++. Other interaction possibilities include input/sensor activation or relies on real-time, remote communication possibilities, in wireless mode.

A thorough visualization and simulation capability has been developed for man-machine interface, in order to make development and configuration phases more productive. A graphic display shows in particular the real-time position and orientation of ARY in its world. Specialized screens are provided for vision and communication. A very extensive proportion of input configurations can also be easily handled in real-time for simulation purpose.

Mode 1 is the most basic and commonly used way to "program" ARY. Compilation is not required as changes either have immediate effect and/or are eventually stored in a configuration file. In this mode, configurations can be flexibly changed in particular in terms of enabling or presetting input signals, setting maximum speeds and accelerations, choosing options and strategies, calibrating time and space increments, in very fine resolution increments (on the order of one microsecond for time and better than one part in 10'000 for linear and angular displacements).

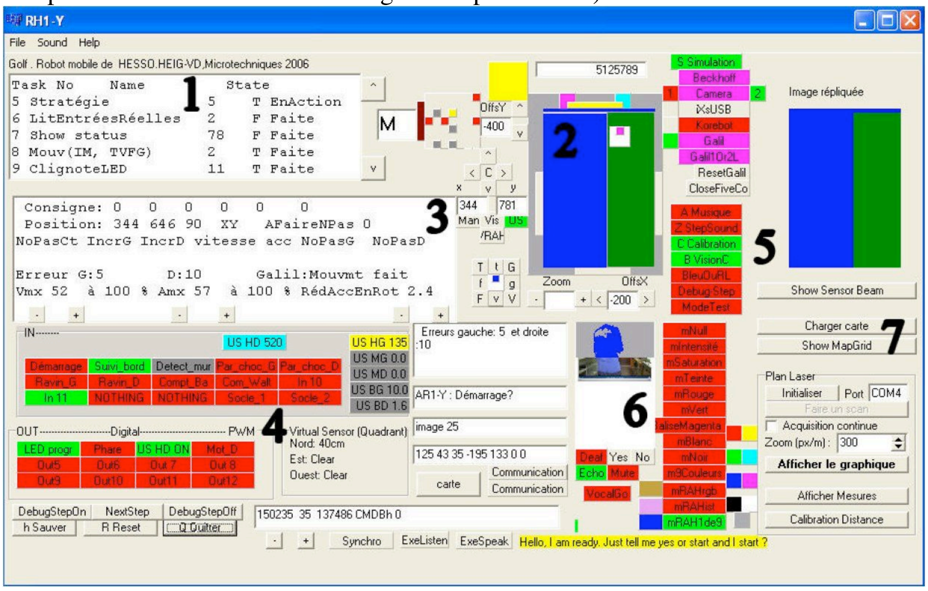


Fig. 4 RH2-Y main screen including agent status, I/O states, and simulation rendering

In mode 2, users define the strategies robots must follow, in the powerful and simple to use, application-oriented, multi-agent Piaget language. In mode 3, users can define new parallel agents or various objects, such as for vision algorithms, or reflex capabilities. Mode 4 refers to the design and adaptation of the environment, e.g. for Piaget primitive implementation or system architecture. This requires a deep knowledge of the system. Remote communication is yet another user interaction mode possibly provided, with chat paradigm on user side and command interpretation on ARY. Other techniques to improve man-machine dialogue on ARY include directly displaying with numerous LED's the status of some sensors, of most of the I/O signals, of power circuits and in particular of the overall running status of the robot. Similarly various direct inputs can be given through switches and specific robot input signals.

6 Cognitive framework

Excellent performances can be achieved in cognitive terms. Some examples are given here for abstraction, concretization, expertise and other cognitive entities, as defined in MCS theory.

Abstraction is the ratio of input information quantity to output information quantity. E.g. the detection of robot location (x, y, α, with 1‰ accuracy, i.e. 30 bit) on the basis of an acquired image (320 columns by 240 lines, with RGB encoded, 3x8 bit pixels) is a process featuring an abstraction index of about 61’440.

Concretization is the ratio of output information quantity to input information quantity. For example the generation of a trajectory (e.g. max. of 3 m displacement, and 2x 180 degree reorientation, with 1cm and 1 degree resolution, i.e. (300+180+180)*(3*10) bit) on the basis of goal site Cartesian coordinates (3*10 bit) yields a concretization index of about 660.

Expertise is the ability of a cognitive system to take decisions fast and right. It is the product of knowledge, K and fluency, f. Considering the specific example above for position estimation, $K=\log_2(30*2**1'895'040))$, i.e. about 2 million lin. Considering that image acquisition and vision processing takes about 0.1 s, f has a value of 10. Therefore the expertise quantity of ARY in this domain is of about 2*10**7 lin/s.

7 Conclusion

The paper describes a versatile platform for the control of autonomous robots, which has been progressively designed in a few years, sometimes through several design iterations. As much as possible, components of the market are used, preferably those developed for standard industrial applications.

Cognitic properties of the platform are of high level, not only for what regards abstraction, concretization, and expertise, but also for other cognitive properties.

Many significant steps however are left to be considered, for the long term future, in the context of real-world applications of the kind aimed at by Robocup-at-home initiative; to name a few: capabilities of locomotion on uneven grounds, of dextrous handling of objects, and of accessing detailed, shared world descriptions.

Team members (Michel Brisseaux, Arvind S. Gautam, Pierre Boucher, Samuel Jabès, Varuj Sukhsevi, Parul Dubey, Ajit P. Singh and authors) wish to thank especially Aurélien Bour, Carl Kjeldsen, Pasd Putthapipat, Sébastien Villoz, Veerapot Kitcharoen, as well as technical departments at HEIG-VD for contributions to RH-Y.

References

1 Jean-Daniel Dessimoz. et al. « Ontology for Cognitics, Closed-Loop Agility Constraint, and Case Study – a Mobile Robot with Industrial-Grade Components », *Proc. 5nd IEEE Intern. Conf. on Industrial Informatics INDIN’06*, August 2006, Singapore

2. Tijn van der Zant and Thomas Wisspeintner, « RoboCup X: A proposal for a new league where RoboCup goes real world » In: Itsuki Noda et al.(Eds.) RoboCup-2005: Robot Soccer World Cup IX, Lecture Notes in Artificial Intelligence, Springer, 2006 ,7pp

3. Sébastien Villoz, « Réalisation d’un robot coopératif pour la compétition internationale Robocup@Home 2007 à Atlanta », Diploma Thesis, Dec. 2006, HESSO.HEIG-VD, pp. 42

4. More information on ARY and RH-Y robots: descriptions and examples, pictures and videos; http://ary1.populus.ch, and http://rahe.populus.ch