“Give me the Purple Ball” – he said to HRP-2 N.14

E. Yoshida*, A. Mallet†, F. Lamiraux†, O. Kanoun†, O. Stasse*, M. Poirier†,
P-F. Dominey‡, J-P. Laumond†, K. Yokoi*
AST/ISRI-CNRS/STIC Joint French-Japanese Robotics Laboratory (JRL)

* Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technology (AIST)
Central 2, Umezono 1-1-1, Tsukuba, 305-8568 Japan. firstname.lastname@aist.go.jp
† LAAS-CNRS, University of Toulouse
7, av. du Colonel Roche, 31077 Toulouse, France. firstname.lastname@laas.fr
‡ ISC-CNRS
67, bld. Pinel, 69675 Bron cedex, France. dominey@isc.cnrs.fr

***Abstract*— This paper reports current experiments conducted on HRP-2 based research on robot autonomy. The contribution of the paper is not focused on a specific area but its objective is to highlight the critical issues that had to be solved to allow the humanoid robot HRP-2 to understand and execute the order “give me the purple ball” in an autonomous way. Such an experiment requires: simple object recognition and localization, motion planning and control, natural spoken language supervision, simple action supervisor and control architecture.**

I. Introduction and related work

Thanks to the recent successful developments in mechatronics, humanoid robots are certainly today the most challenging platforms we may expect to support fundamental research on robot autonomy. They are challenging because of their anthropomorphic mechanical structures: humanoid robots may perform a lot of tasks as complex as a human being does. Such platforms are also challenging for robot algorithms due to the complexity of the underlying dynamical systems: the number of degrees of freedom is high, higher than in other existing robot platforms, and the system stability becomes a critical issue, much more critical than for wheeled robots.

This paper reports on the current level of robot autonomy we reach today in the research performed at JRL-France on the humanoid robot HRP-2. The paper is not focused on a specific technical topic. Its objective is to pinpoint the critical issues that had to be solved to allow the humanoid robot HRP-2 understanding and executing as autonomously as possible the high level order *“Give me the purple ball”*. The degree of autonomy depends on the a priori knowledge given to the robot on the world it is acting to. In the considered context we provide the robot with a geometric map of its environment. The map contains the fixed obstacles. The location of the balls the robot has to grasp is unknown: the balls can be everywhere in the environment, on a support table whose height is unknown. When the robot cannot succeed in filling a current (sub)goal, it reports to the operator via spoken language. For instance, due to the limitation of its vision system, the robot cannot see small objects far from it. When the robot does not see the ball, it may ask the operator for complementary information (e.g. what is the color of the support table?). Even limited, such experiments require robot capabilities in:

- speech recognition and synthesis, and natural language interaction,
- simple object (ball, support table) recognition and localization,
- navigation planning including obstacle avoidance,
- grasping, and
- action supervision.

Such capabilities should be effective and organized in a coherent architecture. The paper gives an account of both the main functional modules (perception, motion planning and control, natural language interaction) and their global integration into a single system.

In addition to humanoid robotics research focused on special topics as mechatronics, motion control, motion imitation, etc., several research projects currently emphasize on robot autonomy at large. Among them, let us mention Johnnie [18]: here the focus is done on the capacity to navigate on uneven terrains in an autonomous way. The perspectives of future applications of humanoid robotics in the daily life stimulate researches focusing on domestic environments. The humanoid robot H7 [22] is defined as a research platform for autonomous behavior, combining vision, motion planning and control capabilities [19] in such contexts. The HRP-2 robot family is made of 15 clones. Several of them are the support of research in robot autonomy mainly at University of Tokyo [8], [23], [24] and AIST [29], [28], [21]. Other android research projects are steered by robot autonomy objectives. ARMAR [3] is a wheeled humanoid robot: the wheeled locomotion simplifies the motion control; it allows researchers to focus on perception and reasoning, including challenging issues in task learning [25]. Anthropomorphic systems for autonomous manipulation are also investigated from platforms as Justin at DLR [4] or Robonaut at NASA [1].

This paper gives an account of the recent experiments conducted on the platform HRP-2 N.14. The paper is organized along four main sections that constitute the key issues we haven been dealing with. Each of them does not constitute an original contribution by itself. Some of them have been already published by the au-

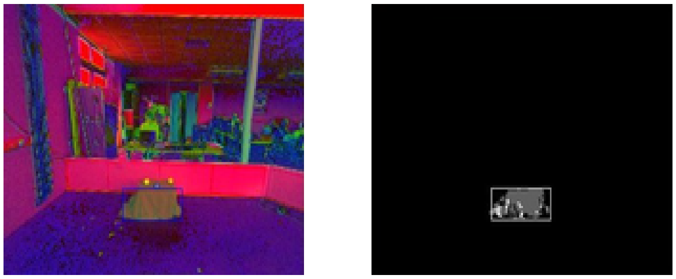


Fig. 1. Table detection. Left image shows the HSV image and right image is the back projection of the table color model in the source image. Rectangle is the result of the execution of the CamShift algorithm on the back projection image.

thors. Other ones are based on state of the art technology. The first section deals with the presentation of the 3D vision system we have developed to recognize and locate very simple objects. Section III summarizes our approach for obstacle avoidance and whole body motion planning. Section IV deals with human interaction for action supervision. Section V gives an overview of the technical choice we made in terms of software development strategy. The resulting experiments are presented in the last section as well as in the videos available at `http://www.laas.fr/~mallet/purple`. The added value of such experiments with respect to the related work above lies mainly on the integration of whole body motion for grasping, the human based supervision and the quality of software integration.

II. Simple Object Recognition and Localization

The HRP-2 robot is equipped with two pairs of firewire digital color cameras, configured as two independents stereo-vision benches. Different lenses on the two benches allow the selection of the appropriate bench to deal with either narrow images and close objects or global scenes.

The perception software is made up of standard, state of the art components: a image acquisition module, performing image resampling, distortion correction and rectification and which feeds a stereo-vision component in charge of computing dense 3D images. Finally, an object detector is able to find simple objects using video color images.

A. "Hue Blob" detection

The algorithm used to localize objects is based on the *Continuously Adaptive Mean SHIFT (CAMSHIFT)* algorithm [5]. Objects to be detected are previously *learned* by taking a sample image, manually cropped around the region of interest. Given this image, a two dimensional histogram in the $\{Hue, Saturation\}$ color space is computed and represents the *model* of the object to be recognized.

The object detection is performed by *back projecting* the object histogram onto a video image. The back projection image is obtained by replacing each $\{H, S, V\}$ pixel value by the corresponding value in the histogram, leading to a probability image where each pixel value is the probability of that pixel to belong to the object model. The *CamShift* algorithm then locates the object center and orientation in the back projection image.

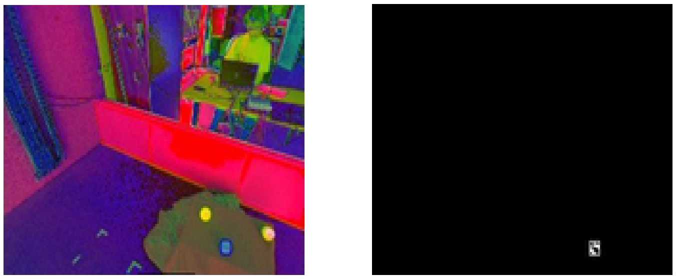


Fig. 2. Purple ball detection. Images represent the same steps of the CamShift algorithm as in Figure 1.

B. Object localization

A stereo-vision algorithm by pixel correlation is applied on the stereo image pairs, and produces a dense three dimensional image of the current scene. Even though pixel correlation is known to give poor results in indoor environments, the objects to localize are sufficiently textured so that precise enough 3D points can be obtained in the vicinity of the objects.

Object localization is performed by averaging the 3D coordinates of all the 3D points that belong to the object (*i.e.* inside the region returned by the CamShift algorithm), weighted by the probability of each pixel to belong to the object (*i.e.* the pixel value in the back projection image). The advantage of this simple method is that we don't require a 3D model of the object, and thus we can localize any object. Of course, this approach would not work if we were to grasp more complex object than simple balls.

III. Motion Planning and Control

After the goal is localized by the vision system, the necessary motion are generated by the "motion planning and control" system. We present here its three components: navigation planning, whole-body motion to take the ball and real-time dynamic motion controller.

A. Navigation planning

We apply a two-stage motion planner, composed of a collision-free path planner using non-holonomic vehicle model and a dynamic walking pattern generator that transforms the planned path into locomotion motion. We assume that the position of obstacles is known and robot moves on a plane.

In the first stage, we need to plan a smooth path towards the goal position. It is desirable for the robot to move forward rather than sideways in order to look the ball and to take it. This is a nonholonomic constraint well known in wheeled mobile robotics [15]. Motion planning techniques for such robots are well-developed and they are useful to generate smooth paths that can be easily followed by biped robots.

A sampling-based method consists in searching a collision-free path in a road-map generated as a graph whose nodes are randomly sampled in the free space $\mathcal{C}_{free}$ in configuration space $\mathcal{C}$. Typical sampling-based methods are known as RRT (Rapidly-exploring random tree) [17] or PRM (Probabilistic RoadMap) [13].

We apply this planning method to the bounding box of the humanoid robot. It is essential to define how to connect the sampled configurations for this method. We here call "steering method" a method that computes an admissible path from a starting configuration to a goal. Taking account of the above nonholonomic constraint, we introduce a steering method for smooth motion for car-like vehicle [14]. The idea is to connect the two planar configurations smoothly with the same curvature. The considered configuration space is then of dimension 4. The method uses the flatness property of the vehicle mobile robot and builds paths through planar curves with given position, orientation and curvature at both ends. According to the flatness property, any admissible trajectory of the vehicle can be represented by the curve followed by the center of the vehicle. The orientation of the vehicle is the direction of the tangent vector to the curve. This method is applied to connect collision-free configurations to derive a smooth path from initial and goal position of the bounding box as shown in Fig. 3.

The planned collision-free path is transformed into dynamic humanoid locomotion in the second stage of navigation planning. After the path is converted into footsteps, a walking pattern generator is applied to generate a dynamically stable walking motion using a method proposed by Kajita et al. [10] based on a preview controller for zero moment point (ZMP). In this method, the ZMP reference trajectory is derived from the footsteps and the motion of center of mass (CoM) is finally computed by using preview control for inverse pendulum. The planned motion is sent to the real-time controller presented later in III-C to be executed in the robot.

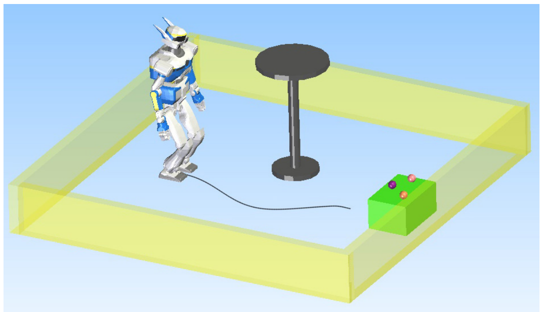
We have implemented these algorithms on a motion planning software kit KineoWorks^{TM} [16]. Fig. 3 shows a planned path for the robot to achieve the goal position.

B. Whole-body motion generation

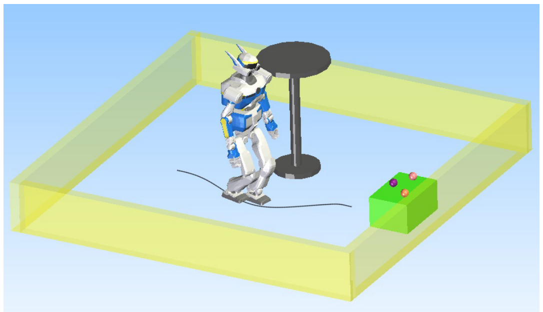
For the motion to reach and get the object, we adopt a general framework of whole-body motion generation [30] including support polygon reshaping. Based on a generalized inverse kinematic (IK) method (e.g., [20], [26]), such tasks as stepping, hand motion, and gaze control are treated with priorities.

Fig. 4 shows an overview of the method. The task is specified in the workspace with priority from which the generalized IK solver computes the whole-body motion as joint angles of the robot. Meanwhile, several criteria such as manipulability, stability or joint limits are monitored if they do not impediment the desired whole-body motion.

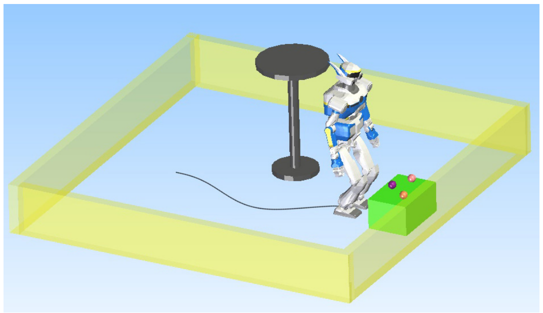
As long as the criteria are satisfied, the computation of whole-body motion continues until the target of the task is achieved. If the task cannot be achieved due to unsatisfied criteria, support polygon is reshaped to extend reachable space. A geometric module determines the direction and position of the deformation of support polygon so that the incomplete task is fulfilled. The position of a foot is then derived to generate the motion of CoM by using the same dynamic pattern generator introduced for locomotion in previous section.



(a) Initial position with the planned collision-free path.



(b) The corresponding walking motion.



(c) Final position.

Fig. 3. Collision-free smooth path planning and motion generation for locomotion.

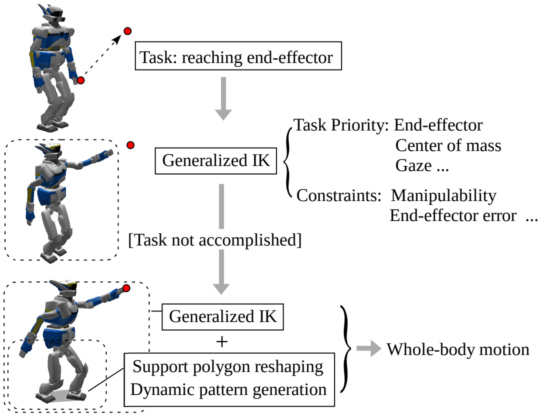


Fig. 4. A general framework for task-driven whole-body motion including support polygon reshaping

Using this CoM motion, the original task is then redefined as the whole-body motion including stepping that is recalculated using the same generalized IK solver. Finally we obtain a blended motion including reaching, stepping and gaze direction to be executed by the robot.

C. Dynamic motion control

On board computing for dynamic motion control is an important issue for autonomous robots. The presented motion planners are installed in the one of the two on board computers of the humanoid robot HRP-2 [12]. The generated motions are sampled by the control cycle time of 5 [ms] as joint angle trajectory and reference ZMP and sent to the other computer that takes charge of real-time motion control for the humanoid.

The controller and stabilizer implemented on OpenHRP humanoid controller [11] finally execute the planned motions. The detailed description of the control architecture of the software modules is given in V.

IV. NATURAL SPOKEN LANGUAGE SUPERVISION

We have been developing high-level programming interface for humanoids using spoken language [6]. Taking advantage of this high-level programming scheme, we have carried out an experiment of teleoperation through internet.

Dialog management and spoken language processing (voice recognition, and synthesis) is provided by the CSLU Rapid Application Development (RAD) Toolkit (http://cslu.cse.ogi.edu/toolkit/). RAD provides a state-based dialog system capability, in which the passage from one state to another occurs as a function of recognition of spoken words or phrases; or evaluation of Boolean expressions. In the mixed initiative dialog system we developed, the system prompts the user and waits for the user to respond with one of the commands (Table 5) and these are immediately executed.

The spoken language interface technology provided by the CSLU RAD system was running on a PC Pentium III Windows machine located at the University of Lyon. This machine communicated with the OpenHRP at the LAAS in Toulouse via the internet using an ssh secure connection. In this manner, spoken language commands evoked in Lyon were used to control the HRP-2 several hundred kilometers away, in Toulouse and in real time.

The behavioral result of a spoken action command that is issued either directly or remotely is the execution of the corresponding action on the robot. Each of these actions, specified in Figure 5, is achieved by the execution of a simple tcl script on the robot. The hueblob script that is associated with the “find” action takes a color argument, and returns the 3D coordinates of that colored object. The “go there” action uses these coordinates to produce a walking behavior that takes the robot to those coordinates. The “grasp it” action uses these coordinates to generate a full body motion grasp. Script execution for all of the actions specified in Table 1 is triggered remotely by the CSLU toolkit, which communicates directly with the low-level OpenHRP framework (Fig. 6).

Motor Command	**Resulting Actions**
find the [orange, purple] ball	Execute hueblob with color argument
find box	Execute hueblob with green argument
go there	Locomote to corrdinates returned by hueblob
grasp it	Use whole-body body motion to grasp object at coordinates returned by hueblob
look [left, right, up, down]	Execute look
turn around	180° walking turn
walk forward	Walk 1.5 meters
give	Use whole-body body motion to give object at predefined coordinates
try again	Repeat the previous high-level action.

Fig. 5. HRP-2 Specific Action Commands.

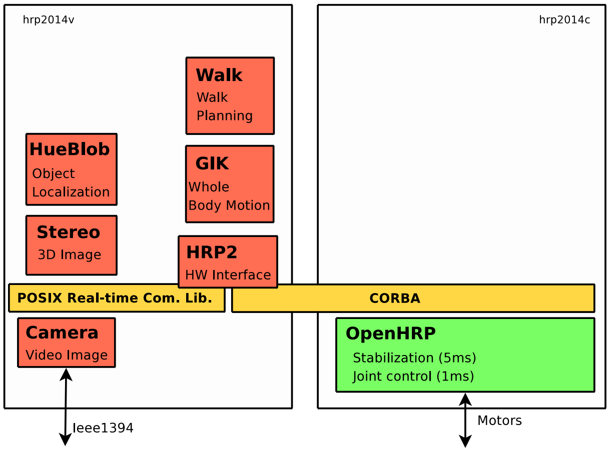


Fig. 6. Software components running onboard the robot.

V. CONTROL ARCHITECTURE

The whole software presented in the previous sections is running on board the robot. In order to build the necessary software components, we used the standard LAAS control architectures tools. In particular, we used the GenoM [7] tool that is able to generate robotics components. GenoM components can encapsulate C or C++ source code into an executable component that provides requests that can be invoked through simple TCL scripts or through more complex supervision software. The components can also be dynamically inter-connected together at run-time, providing a modular and programmable control architecture.

A. Software Components

Figure 6 shows a subset of important components that have been defined for the experiment. All the components but the real-time control (OpenHRP [11]) runs on a Linux 1.8 GHz Pentium-M processor. The real-time part is operated by Art-Linux [2] on a similar hardware.

Request	**Description**
`camera::OneShot`	Acquire stereo images pair.
`stereo::Compute`	Compute 3D image given the current video images.
`hueblob::Find` *x*	Localize object *x* in current video and 3D images.
`walk::Goto` *x,y,theta*	Plan a navigation path to the *world* coordinates *x, y, theta*.
`walk::Execute`	Execute last planned path.
`gik::Grasp` *hand,x,y,z*	Plan a grasping whole body motion to grasp a small object located at *x,y,z* in *waist* coordinates with the *hand* (left or right).
`gik::Look` *x,y,z*	Plan a motion so that cameras look at the *x, y, z waist* coordinates.
`gik::Reach` *hand,x,y,z*	Plan a motion so that the *hand* reach the position *x,y,z waist* coordinates.
`gik::Execute`	Execute last planned motion.
`hrp2::Track` *feed*	Connects to the *feed* (`gik` or `walk` in this experiment), read configurations to be executed from that feed and send them to OpenHRP for execution.
`hrp2::SetJointAngle` *joint,angle*	Move the *joint* to the *angle* value.

Fig. 7. Main requests available to supervision.

As presented in Section II, the vision processing chain is made up of three components: image acquisition (`camera`), stereo-vision by pixel correlation (`stereo`) and object detection and localization (`hueblob`). The motion planning software is split into two components: `walk` that generates navigation trajectories along which the robot can walk (Section III-A) and `gik` that handles the whole body motion generation (Section III-B). Finally, an interface component (`hrp2`) make the connection with the OpenHRP software and bridges the CORBA communication bus of OpenHRP to the GenoM communication bus (*Posix Real-Time communication library* on Figure 6).

All these components define requests that can be invoked by a human operator, by supervision software or by the natural language processing system (Section IV). Main requests are described on Figure 7.

B. Data Flows

Regular data, like images or object localization information flows through the components by using the standard mechanism of *posters* [7] defined by GenoM. Posters are basically shared data structures that can be written only by one producer (the owner) and read asynchronously by several readers without requiring code execution in the owner process context. The semantics of posters is such that only the latest produced data is available for reading, and overwrites any older data.

This semantics is not well suited for stream data flows, like the ones generated by the execution of a planned path or the whole body motions where each configuration must be sent for execution every 5ms. To handle this type of data efficiently, we have chosen to extend GenoM and define a new "FIFO" data type[1]. This new FIFO object allows us to send the configuration to be executed from non-real-time processes (`gik` and `walk`) to the real-time execution part in OpenHRP. Safety was ensured thanks to the dimensioning of a configuration buffer sufficiently large to store 1s of motion and tolerate slights variations in the period of non-real-time tasks.

```
hrp2::Track gik-target
gik::Reach $hand $x $y $z
gik::Execute
gik::HandClench $hand 0.0
gik::Execute
gik::Halfsitting
gik::Execute
hrp2::TrackStop
```

Fig. 8. Sample high-level control script that is invoked through natural language recognition. This example action gives an object at coordinates `x,y,z` with the left or right `hand` and returns to the rest position called *Half-Sitting*.

C. Simple Action Supervision

Action supervision was kept as simple as possible. We used the TCL interface of GenoM module to define high-level elementary tasks to be executed either directly by a human operator or through internet and the natural language processing software. Available actions are described in Figure 5 and Figure 8 shows the details of a (simplified) example script that sequences several requests to the `gik` and `hrp2` software components.

Each action can report failures (*e.g.* failure in grasping an object). It is thus possible to implement error recovery strategies by analyzing the reports of the actions.

VI. EXPERIMENTAL RESULTS

A. First Experiment

The first experiment was remote controlled through internet. The movie is available for download at the address mentioned in the introduction of this paper. Figure 9 shows some sample images of the execution.

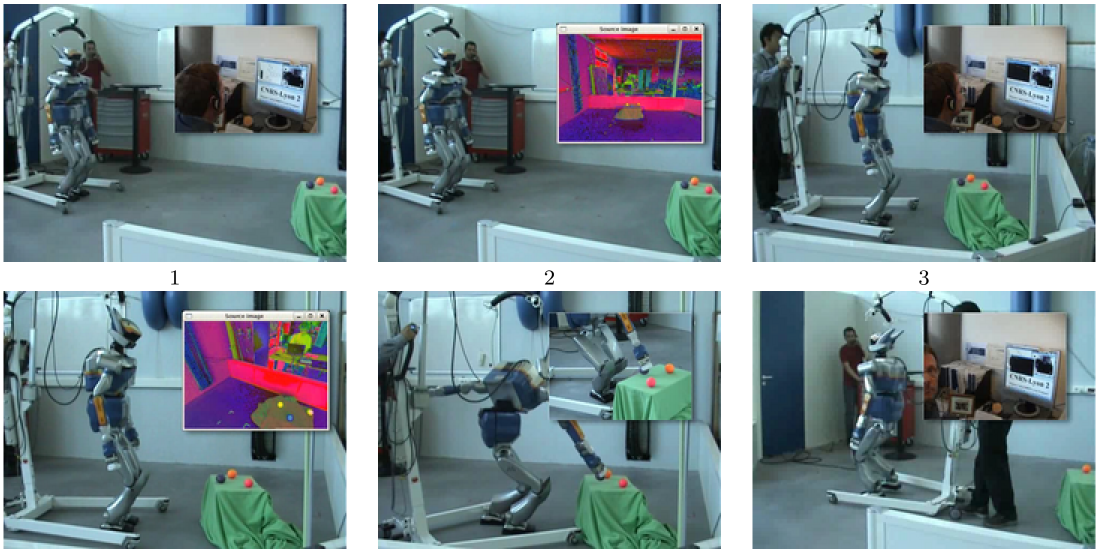
Here we provide the literal transcription of the human-robot dialog, annotated with explanation about the corresponding human and robot behavior and key system functions in [] brackets. Robot language will be indicated in *italics*, and human language in plain text, with main commands in **bold**. Robot confirmation of understood commands is removed for better readability.

Dialog: [Robot initiates the interaction] *I am ready.* **Find the purple ball.** *I cannot see the purple ball. I am ready.* **Find the box.** *Looking for the box.* [System executes vision processing algorithm]. *I am ready.* **Go to the box.** *Going.* [Robot walks to the box]. **Look down.** Looking down. [Robot looks down] **Find the purple ball.** *Looking for the ball. Describe an action.* **Grasp it.** *Grasping.* [Robot executes full body motion to grasp the ball]. **Turn around.** *Turning* [Robot turns around]. **Walk forward.** *Walking forward.* [Robot walks forward] Turn around. *Turning.* [Robot turns around]. **OK – Finish learning**. *You said OK, so I will store the learned program.*

The performances of the robot are quite satisfactory, except for the time required to compute the whole body motion. On the robot hardware, computing such a motion

---

[1] FIFO are, of course, very common objects available in many control architecture frameworks.



1 2 3 4 5 6

Fig. 9. First experiment with remote control of the robot through natural language interaction. The robot is asked to go to the table, grasp the ball there and come back to the initial position.

requires almost twice the time of the motion duration itself (50 seconds of computing for a motion that lasts 30 seconds in this experiment). However, on recent hardware we are able to reduce this time to less than the duration of the motion.

B. *Second Experiment*

The second experiment was controlled by a human operator and the same high-level commands as those used in the tele-operated experiment were available. You can see images of the execution on Figure 10.

The mission was to bring the purple ball to a person sitting on a chair. As an improvement to the previous experiment, a static environment model was loaded into the navigation planning module to enable the obstacle avoidance feature. The environment contained a big table and the wall delimiting the environment.

The sequence of commands was the following: **Find the box**, **Go to the box**, **Look down**, **Find the purple ball**, **Grasp it** (which failed, see Figure 10), **Grasp it** (which worked), **Turn around**, **Go to home**, **Give the ball**.

When going to the box, the robot was able to avoid a big table that was impeding it to reach the box directly.

As an interesting fact, the first attempt to grasp the ball failed. This is due to the relative simplicity of the perception system and some uncertainty on the coordinates of the detected object. Such failures are expected to happen, but can be easily detected by *e.g.* using the force sensors in the robot hands. Furthermore, the environment was modified during the motion by accidentally moving some ball. However, recovering from the failure was possible by invoking the algorithm to detect the ball again and recompute a new grasping movement with the new computed ball position.

VII. Conclusions

The experiments above can be repeated on demand. Such a feature supposes an important effort in terms of software development coordination. The targeted robustness has been reached thanks to the use of experienced methods integrated in a carefully defined software architecture. Future works include the development and the integration of new modules, e.g., self-localization, incremental 3D map building, face recognition, automatic action supervision, sensor- motor feedback control, etc. In parallel to this incremental development strategy, specific challenging routes in robot autonomy studies are open by humanoid robotics. The richness and the scope of possible actions benefiting from anthropomorphic mechanical structures, as well as their similitude with the human body, impose the development of original models of the action in synergy with live sciences. Such lines of research are emerging (e.g. [27], [9]).

References

[1] R. O. Ambrose, H. Aldridge, R. S. Askew, R. Burridge, W. Bluethman, M. A. Diftler, C. Lovchik, D. Magruder, and F. Rehnmark F. Robonaut: Nasa's space humanoid. *IEEE Intelligent Systems Journal*, 15(4), August 2000.

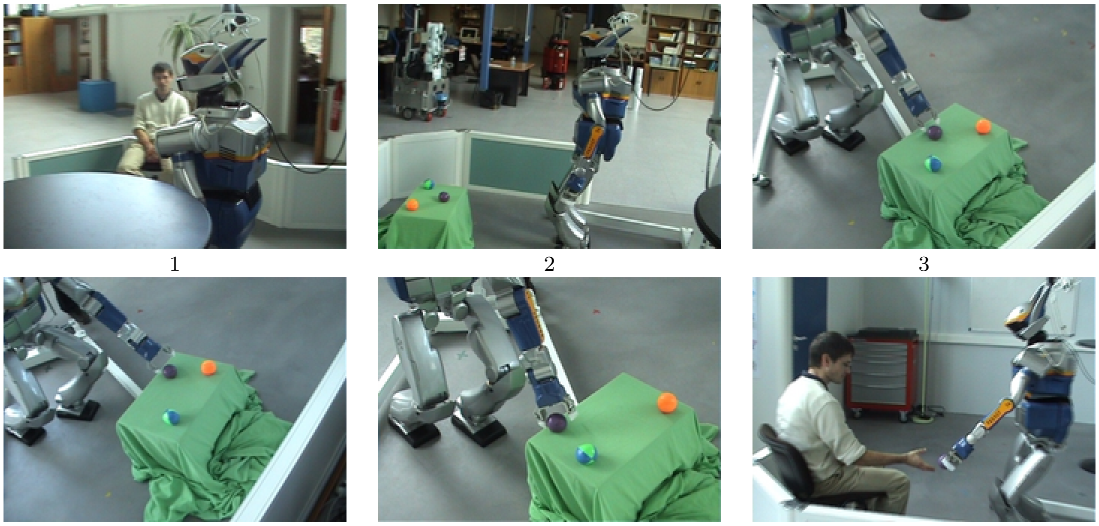
[2] ART Linux. `http://sourceforge.net/projects/art-linux`.

[3] T. Asfour, K. Regenstein, P. Azad, J. Schröder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann. Armar-iii: An integrated humanoid plattfrom for sensory-motor control. In *IEEE-RAS International Conference on Humanoid Robots*, Genova (Italy), December 2006.

[4] C. Borst, C. Ott, T. Wimböck, B. Brunner, F. Zacharias, B. Bäuml, U. Hillenbrand, S. Haddadin, A. Albu-Schäffer, and G. Hirzinger. A humanoid upper body system for two-handed manipulation. In *IEEE International Conference on Robotics and Automation*, Roma (Italy), April 2007. IEEE.

[5] G. R. Bradski. Computer video face tracking for use in a perceptual user interface. *Intel Technology Journal*, Q2, 1998.

[6] P-F. Dominey, A. Mallet, and E. Yoshida. Progress in programming the hrp-2 humanoid using spoken language. In *IEEE International Conference on Robotics and Automation*, Roma (Italy), April 2007. IEEE.



1 2 3

4 5 6

Fig. 10. Second experiment. The robot is asked to give the purple ball to the person sitting on a chair. It walks toward the table, avoiding known obstacles in the environment. It first fails to grasp the ball (image 3 and 4), but with a simple error recovery strategy it finally succeeds (image 5) and accomplishes the mission.

[7] S. Fleury, M. Herrb, and R. Chatila. Genom: A tool for the specification and the implementation of operating modules in a distributed robot architecture. In *International Conference on Intelligent Robots and Systems*, Grenoble (France), September 1997. IEEE.

[8] T. Inamura, K. Okada, M. Inaba, and H. Inoue. Hrp-2w: A humanoid platform for research on support behavior in daily life environments. In *9th International Conference on Intelligent Autonomous Systems*, pages 732–739, 2006.

[9] T. Inamura, I. Toshima, H. Tanie, and Y. Nakamura. Embodied symbol emergence based on mimesis theory. *International Journal of Robotics Research*, 23(4), April 2004.

[10] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *International Conference on Robotics and Automation*, pages 1620–1626. IEEE, 2003.

[11] F. Kanehiro, H. Hirukawa, and S. Kajita. Openhrp: Open architecture humanoid robotics platform. *International Journal of Robotics Research*, 23(2):155–165, 2004.

[12] K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawa, T. Kawasaki, M. Hirata, K. Akachi, and T. Isozumi. The humanoid robot hrp-2. In *International Conference on Intelligent Robotics and Systems*, pages 1083–1090. IEEE, 2004.

[13] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.

[14] F. Lamiraux and J-P. Laumond. Smooth motion planning for car-like vehicles. *IEEE Transactions on Robotics and Automation*, 17(4):498–501, 2001.

[15] J-P. Laumond, editor. *Robot Motion Planning and Control*, volume 229 of *Lectures Notes in Control and Information Sciences*. Springer, 1998.

[16] J-P. Laumond. Kineo cam: a success story of motion planning algorithms. *IEEE Robotics and Automation Magazine*, 13(2):90–93, 2006.

[17] S. LaValle and J. Kuffner. *Rapidly-Exploring Random Trees: Progress and Prospects*, pages 293–308. Algorithmic and Computational Robotics: New Directions. A K Peters, 2001.

[18] K. Lóffler, M. Gienger, and F. Pfeiffer. Sensors and control concept of walking 'johnnie'. *International Journal of Robotics Research*, 22(3), 2003.

[19] P. Michel, J. Chestnutt, J.J. Kuffner, and T. Kanade. Vision-guided humanoid footstep planning for dynamic environments. In *IEEE-RAS International Conference on Humanoid Robots*, Tsukuba (Japan), December 2005.

[20] Y. Nakamura. *Advanced Robotics: Redundancy and Optimization*. Addison-Wesley Longman Publishing, 1991.

[21] E. S. Neo, K. Maruyama, T. Sakaguchi, Y. Kawai, and Kazuhito Yokoi. A behavior level operation system for humanoid robots. In *IEEE-RAS International Conference on Humanoid Robots*, Genova (Italy), December 2006.

[22] K. Nishiwaki, J.J. Kuffner, S. Kagami, M. Inaba, and H. Inoue. The experimental humanoid robot h7 : A research platform for autonomous behavior. *Phil. Trans. of the Royal Society (special issue "Walking Machines")*, pages 1–28, 2006.

[23] K. Okada, T. Ogura, A. Haneda, J. Fujimoto, F. Gravot, and M. Inaba. Humanoid motion generation system on hrp2-jsk for daily life environment. In *IEEE International Conference on Mechatronics and Automation*, volume 4, 2005.

[24] K. Okada, T. Ogura, A. Haneda, D. Kousaka, H. Nakai, M. Inaba, and H. Inoue. Integrated system software for hrp2 humanoid. In *International Conference on Robotics and Automation*. IEEE, 2004.

[25] M. Pardowitz, S. Knoop, R. Dillmann, and R. D. Zöllner. Incremental learning of tasks from user demonstrations, past experiences, and vocal comments. *IEEE Transactions on Systems, Man, and Cybernetics*, 37(2), April 2007.

[26] B. Siciliano and J-J. E. Slotine. A general framework for managing multiple tasks in highly redundant robotic systems. In *5th International Conference on Advanced Robotics*, pages 1211–1216, Pisa (Italy), June 1991.

[27] P. Souères, C. Jouffrais, S. Celebrini, and Y. Trotter. *Robotics Insights for the Modeling of Visually Guided Hand Movements in Primates*. Biology and Control Theory: Current Challenges, Lecture Notes in Control and Information Sciences. Springer Verlag, 2007.

[28] K. Yokoi, E. S. Neo, T. Sakaguchi, H. Arisumi, E. Yoshida, O. Stasse, Y. Kawai, K. Maruyama, T. Yoshimi, and S. Kajita. Humanoid robot hrp-2 no.10 with human supervision. In *International Symposium on Experimental Robotics*, Tokyo (Japan), December 2005.

[29] K. Yokoi, E. S. Neo, T. Sakaguchi, O. Stasse, Y. Kawai, and K. Maruyama. Humanoid robot hrp-2 with human supervision. In *International Symposium on Experimental Robotics*, Rio de Janeiro (Brazil), July 2006.

[30] E. Yoshida, O. Kanoun, J-P. Laumond, and C. Esteves. Task-driven support polygon reshaping for humanoids. In *IEEE-RAS International Conference on Humanoid Robots*, pages 208–213, Genova (Italy), December 2006.