

Remote Programming of Multirobot Systems within the UPC-UJI Telelaboratories: System Architecture and Agent-Based Multirobot Control

Raul WIRZ, Raul MARIN, Joseph FERNANDEZ, Pedro J. SANZ

Abstract—One of the areas that needs further improvement within E-Learning environments via Internet (A big effort is required in this area if progress is to be made) is allowing students to access and practice real experiments in a real laboratory, instead of using simulations [1]. Real laboratories allow students to acquire methods, skills and experience related to real equipment, in a manner that is very close to the way they are being used in industry. The purpose of the project is the study, development and implementation of an E-Learning environment to allow undergraduate students to practice subjects related to Robotics and Artificial Intelligence. The system, which is now at a preliminary stage, will allow the remote experimentation with real robotic devices (i.e. robots, cameras, etc.). It will enable the student to learn in a collaborative manner (remote participation with other students) where it will be possible to combine the on-site activities (performed “in-situ” within the real lab during the normal practical sessions), with the “on-line” one (performed remotely from home via the Internet). Moreover, the remote experiments within the E-Laboratory to control the real robots can be performed by both, students and even scientist. This project is under development and it is carried out jointly by two Universities (UPC and UJI). In this article we present the system architecture and the way students and researchers have been able to perform a Remote Programming of Multirobot Systems via web.

Index Terms— E-Learning, Robotics, Artificial Intelligence, Multirobot Systems

1. Introduction

The evolution of communication and information technologies has opened up new possibilities in educational methods. E-learning methodologies make autonomous learning easier, by offering course materials and tools. While there is a wide offer of Course Management Systems (CMS) [2] that seem to cover adequately the e-learning requirements of theoretical issues, e-learning on practical issues, working with real equipment, is still an unsolved matter.

The E-Laboratory will serve as a basis for

researchers to develop optimum algorithms that control robotic systems in an “intelligent” manner. Moreover, it will permit the students to improve their learning process as long as we have more access to real systems (24 hours a day, 7 days per week)[3][4]. At a first glance, the E-Laboratory effort will be in the design of techniques that improve the Remote Manipulation of a scenario via web (using a robot manipulator). It should be possible for the student or scientist to be situated at a long distance (i.e. thousands for miles away)[5]. The system will provide advanced tools to enhance the human-robot interaction, such as augmented reality, virtual reality and pattern recognition [6][7].

Firstly, a description of the e-learning environment will be described in section 2. Secondly, a description of the system architecture and of the proposed programming libraries will be analyzed in sections 3 and 4. Finally, an experiment over the UPC-UJI E-Laboratory using different locations is performed in section 5 to prove test system.

2. System Description

The main components of the e-learning environment are:

1) *Course Management System*. The basic tools of the learning process are multimedia materials as textbook, images, videos or interactive tutorials. These materials are structured in learning blocs. To manage and access materials, the Moodle [8] software package is used.

2) *Robotics Labs*. In this context, a Robotics lab is a set of equipment and resources having the entity of a complete, multi workspace, laboratory, which is available simultaneously for remote access and on site work. The laboratory arrangement and organization are oriented to mixed training, that is, to combine contact and remote learning activities. Usually, education institutions have different types of robots (robot manipulator, mobile robots), each one with a different programming languages and interfaces. In this work, a unified robot interface framework is designed in order to make easier and faster the use of robotic systems, such as educational manipulator robots, industrial

Manuscript received November 15, 2006; Revised May 12, 2007 and October 1, 2007.

Raul Wirz, Computer Science Department, Jaume I University, Castellon, Spain, email: wirz@icc.uji.es.

Raul Marin, Computer Science Department, Jaume I University, Castellon, Spain, email: rmarin@icc.uji.es.

Joseph Fernandez Ruzafa, Dpt. Enginyeria de Sistemes, Automatica i Informatica Industrial, Universitat Politecnica de Catalunya, email: josep.fernandez@upc.edu.

Pedro J. Sanz, Computer Science Department, Jaume I University, Castellon, Spain. email: sanzp@icc.uji.es

manipulator robots, mobile robots or telerobotics systems [9]. This includes human-robot interface and, when possible, robot programming language.

3) *Security & Integrity Modules*. A main issue in remote systems is security and integrity. Two security levels are defined to avoid malicious or erroneous use of the system: User level, -user has to be authenticated, and depending on the assigned profile he/she can only make a restricted use of the labs-, and Scenario level – the system only authorizes the access/movement to safe positions-.

4) *Resource Center*: The robotics community, from amateurs to researchers, is very active. A huge volume of information on image, design or resources about robotics is available in the Internet. These formal or informal materials are very good for promoting and enhancing autonomous learning. The presented environment will include an organized repository of robotics resources on the Internet. The Resource Center will facilitate the distribution and publication of documents, videos, images, and other information relative to the teaching/learning of a specified subject. Likewise, the Resource Center would facilitate the integration of that material with the corresponding exercises within the real Tele-Laboratory.

5) *Study Center*: This tool would permit scientists and students to share, discuss and work in a collaborative way in a real problem within the tele-laboratories. These people could access the Study Center from any computer connected to the Internet.

6) *Tele-Stadium* [10]: As a novel contribution to the Robotics and Computer Networks community, the Project presents the possibility to perform robotic competitions via Internet. In a first step, a team of LEGO robots would be provided that could perform tasks designed remotely from any person connected to the tele-laboratory. The system would extract the results of these algorithms from the real robotic scenario in order to assess the user knowledge.

7). *Intelligent Multimedia Tutorials*: The Tele-laboratories would be enriched with the design of Intelligent Multimedia Tutorials, that explain in a simple way the required subjects and allows the student to practice the correspondent exercises with a real robot through the tele-laboratory. The system would include assessments in order to automatically extract the level of knowledge of the student, and then present the subjects in a way that is appropriate for that specific level.

3. System Architecture

The presented e-learning environment is built from the existing on-line laboratories at the UPC and UJI Universities. The work is directed to enhance and integrate existing laboratories, despite that they are

designed independently, together with the other components presented in Section II.

A. *Lab Equipment*: The system is composed of 4 tele-laboratories, two belonging to the UPC University and the other two to the UJI University. The UPC University has a laboratory for industrial applications having 5 manipulator robots and the second one with 10 LEGO robots. The UJI University has one laboratory that is located in the teaching building and is basically used by undergraduate students within the Robotics subject practices (see Figure 1). The research laboratory has several industrial robots that can be programmed remotely as well as from the Internet. The application of this research tele-laboratory is more related to Ph.D students and researchers too.

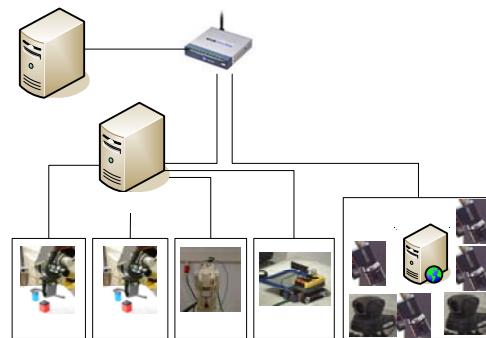


Figure 1. SNRP UJI-UPC Tele-Laboratories.

B. *Software Architecture*: The main software architecture offers a simple way of integrating and synchronizing the four tele-laboratories. As shown in Figure 2, the software architecture defines a northbound interface to any one of the tele-laboratories. It means a user can get access to any device of a telelaboratory as long as it has this common interface, which is common to every telelaboratory. This interface is service oriented and named as *Simple Network TELelab Protocol* (SNTELP), which is in fact one of the novel contributions of this paper. It uses the HTTPs protocol as a basis, and permits any external program to get access to every one of the robots belonging to the telelaboratory. Inside a SNTELP provided telelaboratory, every device (sensors and robots) implement a SNRP interface (Simple Network Robot Protocol), which permits controlling them in a very simple manner by just using a standard URL. This is very motivating for students, because it is very easy to get the robot move by just clicking at a Web link. This lets students focus on the application instead of spending several hours in specific and technical aspects of a given robot.

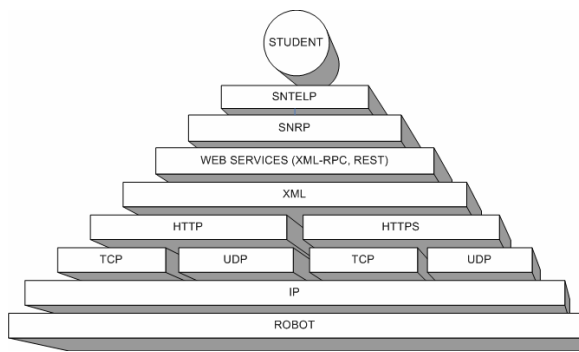


Figure 2. Software Architecture.

C. *Remote Programming*: In order to have the same interface for the existing on-line laboratories, three programming methods are proposed (Figure 3):

- 1) *Native Robot Programming*. The native robot programming language is used (RAPID; VAL; ACL; NQC, LEJOS; ..). The user writes robot programs, and after validation, they are uploaded and executed on the robot controller. In this way, all robots features are accessible in an efficient way, but the user has to learn different robot languages if he/she wants to use robots from different vendors.
- 2) *Standard Robot Programming*. As a part of the presented work, a standard robot

programming is defined. This standard covers the basic features required to control a robot. The user writes robot programs using the standard language, and after validation, they are uploaded in the lab server that translates it to the native robot language. Finally, the lab server transmits the translated program to the robot controller for execution. Lab users therefore have to learn only the standard language, independently of the robots used. This feature also allows us to reuse the same programs for a new robot, so that only a suitable translator is required.

- 3) *Command based Programming*. The user writes robot application programs using a computer programming language (C++; Java, Visual Basic ...). Programs that are executed on the client-side have to store and process the data coming from the robot. Using a robot library the client interacts with robot, acquiring robot and world information, and sending robot commands. Using a Robot Identifier in each exchanged message, a multi-robot system is build up. This programming method makes use of the power of computer programming language, and allows it to implement a multi-robot system.

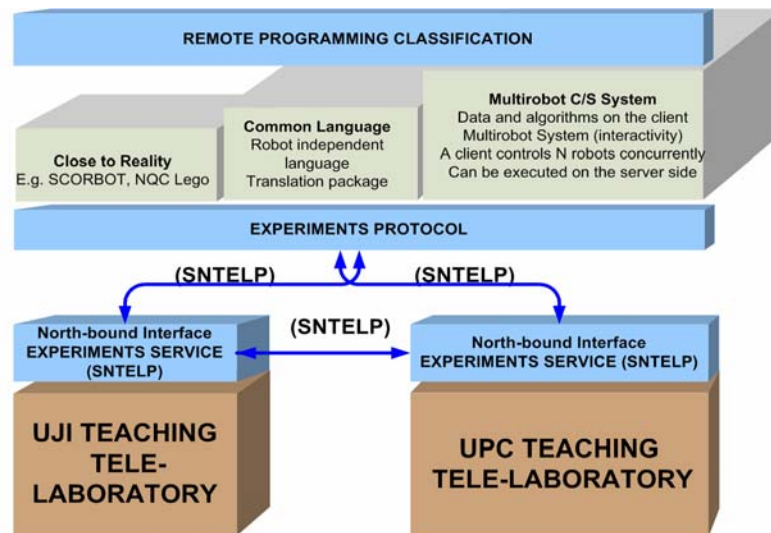


Figure 3 Remote Programming Classification at the UPC-UJI E-Laboratory

4. Remote Programming

In order to program the Tele-Laboratory remotely, a TCP/IP interface is provided and can be easily managed through an already implemented Java Library

called “Experiments”. This library already includes templates that are examples of simple experiments that manage the remote robots and the cameras. (Figure 4)

To facilitate implementation of the experiments even more, an “Experiment_template” is provided that already inherits from the “Experiment” class and

presents the structure of a typical Remote Programming experiment, which is: (1) extending the “Experiment” class, (2) creating an instance of the experiment, (3) executing the corresponding actions on

the Telelaboratory using SNRP commands, and (4) closing the connection.

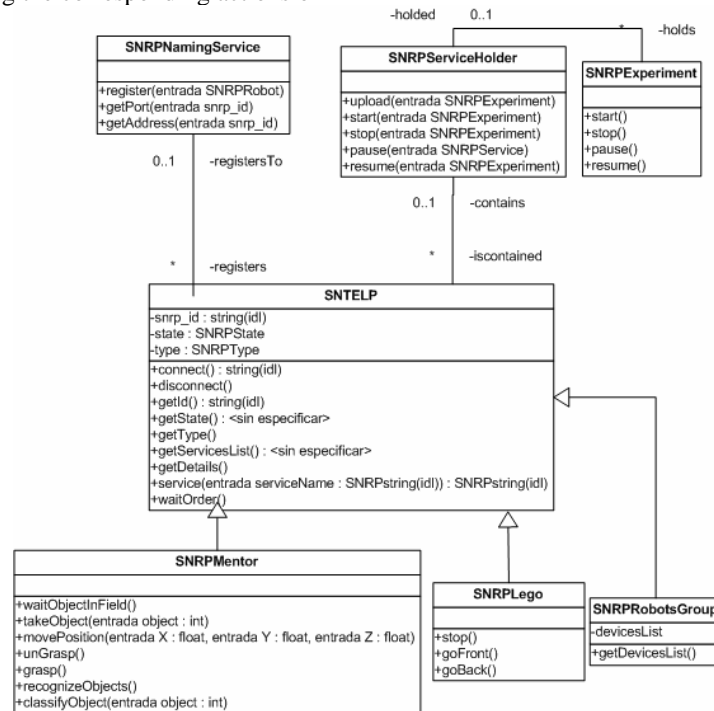


Figure 4. Remote Programming Template

To make heterogeneous robot programming easier, a new protocol was developed to allow that a single program get access to any robot capabilities in a very simple way (i.e. using XML)

4.1. SNTTELP & SNRP Protocol

Different approaches of multi-robot systems have been proposed [11][12]. However, our solution is a new novel application-layer protocol was developed to establish and maintain a communication between different tele-laboratories (SNTTELP) and different robots (SNRP). The protocol can be used from any programming language and several Java libraries have been provided to the students for letting them program every specific kind of robot in a very simple way.

SNRP is a high level language program that gives us the opportunity of program a type of robot without learn the low level language of the robot. In other words, different robots that share similar features are programmed with the same commands. For example, manipulator robots have a grasp that it can be opened or it can be closed. A Mentor robot will use the command: “*moveToSteps 128 128 128 128 128 128*” to close the grasp; A Barret hand will use the command: “*gc*”. Using SNRP, the student needs only

to know that the high-level command “Grasp” will close the grasp in any robot that would have a grasp.

SNTTELP is a high-level language program that gives us the opportunity of program together different robots that can be situated in the same tele-laboratory or in different tele-laboratories. It is a protocol that helps us to synchronize and establish communication between different robots.

4.2. Example Libraries.

SNRP and SNTTELP libraries are provided in Java to the students to perform experiments. Examples of these libraries, which are used in the experiment of the next section, are following explained:

4.2.1. UjiSNTTELP Library

This Java abstract class has two different parts: A part that contains the commands and a part that it is a thread. The following command can be found in the library:

Connect: The user connects to the robot in a tele-laboratory and reserves this resource until the process is concluded.

Disconnect: User disconnects to the robot.

WaitOrder: The specified Robot waits until it receives an order.

RegisterToEvent: Robot waits until a specific event has been activated in other robot.

The thread, that is continuously running, is the part of the program that controls the security and the experiment supervision. For example, if a robot doesn't listen correctly or the robot dies, the thread will stop the experiment.

4.2.2. MentorSNRP Library.

This Java class has been developed to manage the Mentor robots. The provided instructions are the following:

WaitObjectInField: Robot waits until an object is dropped on the work-area.

TakeObject: Robot picks up the object that we want.

MovePosition: Robot is moved to a specific position.

Ungrasp: Robot drops the object that has been grasped.

Grasp: Robot closes the grasp.

RecogniseObjects: Robot recognizes object in field using statistical methods and an already existing database.

ClassifyObject: Robot drops the object in the same area as other similar object.

4.2.3. LegoSNRP Library.

This Java class has been developed to manage the Lego robots. In this specific case, our Lego robot has been built to function as a conveyor belt, so the Lego Robot merely waits for orders from the Mentor robots to move in a specific direction. Then, in this experiment, we will use the following instructions:

GoFront: Robot is moved from start position located in Mentor 1 work area to end position located in Mentor 2 work area.

GoBack: Robot is moved from the end position located in Mentor 2 work area to the start position located in Mentor 1 work area.

Stop: Robot lego stop any operation.

5. Remote Multirobot Experiment

To test the proposed architecture and network protocols (i.e. SNTLP & SNRP), a multi-robot experiment has been decided to carry out using three robots, two manipulator robots and one mobile Lego robot. This experiment consists of performing a common task in such a way that all the robots work together and a cooperative manner using the same programming language in all elements that participate

in the experiment. So the final task is to classify an object.

Each robot works in the following manner:

Mentor 1 Robot: This robot waits until an object is dropped on the work-area. Once it happens the robot takes the object and puts it on the Lego robot.

Lego Robot: The Lego robot performs the same way as a conveyor belt. It transports objects from Mentor 1 robot to Mentor 2 robot.

Mentor 2 Robot: When the Lego robot arrives at the Mentor 2 robot work-area, the robot recognizes the object and classifies it. In other words, the robot takes the object and drops the object on a specific area depending on the object name. The experiment can be observed using three cameras: two cameras situated on the mentor's work area and a camera pantil in a side.

To carry out this experiment, it has been programmed three independent tasks, one per robot. The code has been developed in JAVA using Threads, one thread per robot (see Figure 5). And each thread uses the specific SNRP robot interface that controls its particular robot.

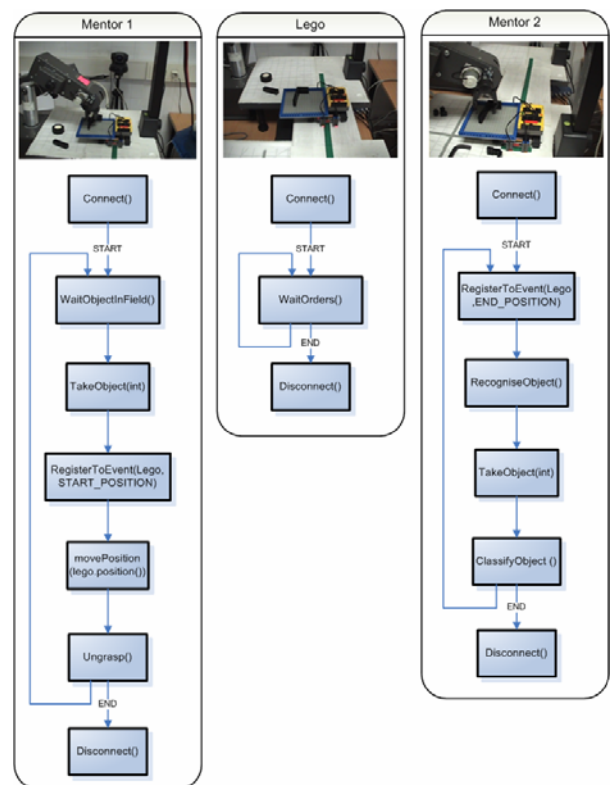


Figure 5. Multi-robot Concurrent Tasks. Each robot runs its own code.

5.1. Experiment Description

In Figure 6, three Concurrent Agents are running in the experiments, one for every robot in the scenario. These agents are located in the computers that control the robots.

First of all, the Mentor 1 robot waits until an object is dropped on the work-area (WaitObjectInField) and, once it detects it, the Mentor 1 robot picks it up (TakeObject). After that, the Mentor 1 robot waits until the Lego robot is situated in the start position (RegisterToEvent start_position). Once it happens, the Mentor1 robots is moved on top of the Lego Robot (MovePosition(lego.position())), and drops the object (Ungrasp). After that, it tells the Lego robot to move to end position at the Mentor2 work area (GoFront).

When the Lego robot arrives to the end position, the event “end_position” is activated. As the Mentor 2 robot is waiting for this event, it is able to proceed accordingly.

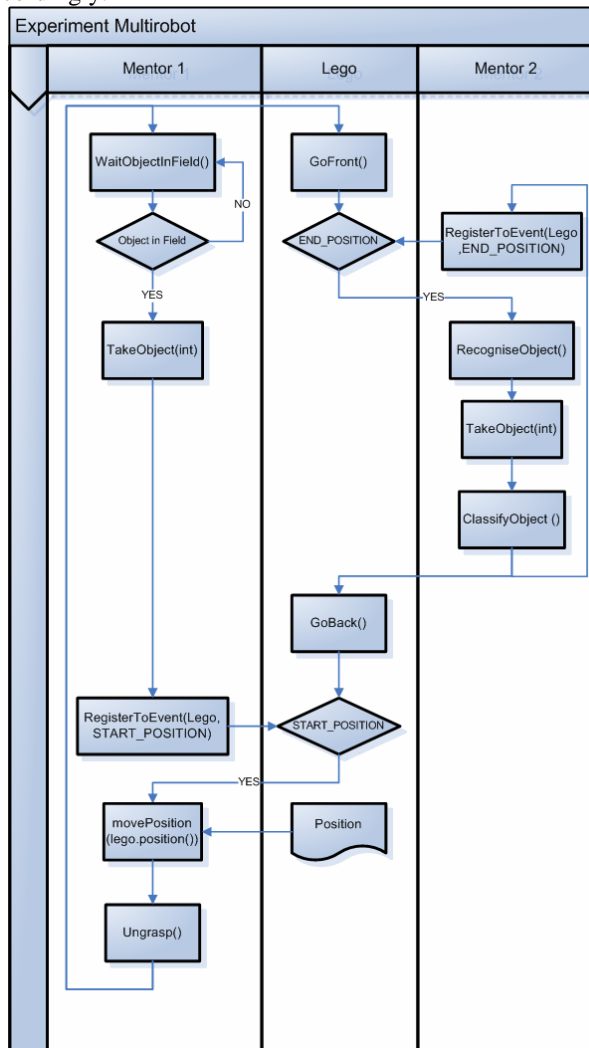


Figure 6. Process diagram.

When the Mentor 2 robot detects the event, it first recognizes the object (RecogniseObject), picks it up (TakeObject), classifies it (ClassifyObject) and orders the Lego robot to return to its start position (GoBack). Finally, Mentor 2 robot goes to its home position and waits until a new end_position event gets activated (RegisterToEvent end_position).

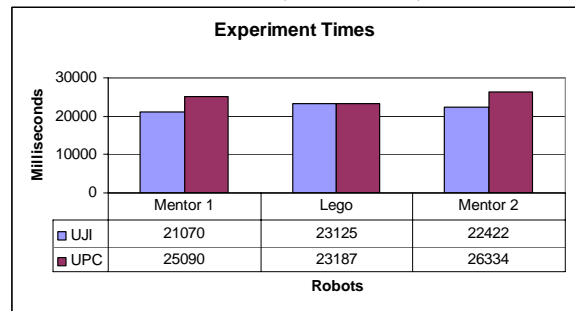
When the Lego Robot arrives at the start position, the start position event (start_position) is activated, so the Mentor 1 Robot starts the same loop again.

5.2. Results

Since the objective is letting students from any University have access the Tele-Laboratory resources, the multi-robot remote experiment has been tested being launched from both, the UJI University of Castellon and UPC (University Technique of Catalonia) using the SNRP libraries. The bandwidth at the UJI is 100 Mbps and the time-delay around 3 ms. The UPC connection has a 18 ms time-delay till the Tele-laboratory servers.

In the following table we can observe the time required for each robot in both cases:

Table I Time Results (milliseconds)



The time-delay obtained for the whole experiment is around 150- 200 ms for the UPC localization and 50-100 ms for the UJI.

Conclusions

In this paper we have presented a software and a hardware architecture that lets students and scientists program remotely intelligent tasks in a tele-laboratory via the Internet without learn the programming language of each element of the tele-laboratory. Besides this, it is possible to perform the remote programming made by a single robot or even using a more sophisticated configuration by multiple heterogeneous robots.

One special feature of the architecture presented using the SNTLP and SNRP protocols, is that it is very easy for students to develop remote multi-robot programming over the Internet. This is due to the fact that most of the intelligence is located on the robot side, and the commands required to program the robot are high-level ones (e.g. takeObject 1).

Finally, the paper has presented a real multirobot remote programming experiment that demonstrates that this technique is very appropriate for education,

research and even industrial applications. As shown in the results, by using the system in the same campus, the time delay may be considered reasonable for their proper performance of multi-robot experiments.

Acknowledgements

This work has been partially funded by the Spanish Ministry (MEC) under Grants TIC2003-08496, DPI2004-01920, TSI2004-05165-C02-01, by the Fundació Caixa Castelló under Grants P1-1B2002-07, P1-1B2003-15, and P1-1A2003-10, and by the Generalitat Valenciana under grant GV04A-698.

References

- [1] R. Marín, P.J. Sanz., A.P. del Pobil, "The UJI Online Robot: An Education and Training Experience." *Autonomous Robots*, pp. 283-297. vol 15, Number 3, 2003.
- [2] Josep Fernández and Alicia Casals. "The Open Laboratory paradigm for engineering education via e-learning" Special Issue on Immersive Telelaboratories for engineering designer education. *IEEE Learning Technology Magn.* Pp 13-15 vol. 7, Issue 3. July 2005.
- [3] R. Marín and P. Sanz. "Grasping Determination Experiments within the UJI Robotics Telelab", *Journal of Robotic Systems, Internet and Online Robots for Telemanipulation Special Issue (Part 2)*, pp 203-206 vol 22, Number 4, 2005.
- [4] R. Wirz, R. Marín, E. S. Quintana-Orti. "Distributed System for Remote Programming of Multiple Network Robots: System Performance & Parallization Issues", *CEDI 2005 1st Spanish Conference on Computer Science CEDI 2005, Workshop on Parallelization (JP 2005)*, Granada, pp 431-442 September, 2005.
- [5] D. Lee and M. W. Spong. "Bilateral Teleoperation of Multiple Cooperative Robots over Delayed Communication Networks: Theory" In *Proc. of the IEEE Int. Conf. On Robotics and Automation (ICRA)*. Pp 360-365 Barcelona, Spain, April 2005.
- [6] D. Kragic. "Visual Servoing for Manipulation: Robustness and Integration Issues." PhD thesis, CVAP, NADA, KTH, Stockholm, Sweden, 2001.
- [7] B. K. Kim et al., "Web Services Based Robot Control Platform for Ubiquitous Functions" In *Proc. of the IEEE Int. Conf. On Robotics and Automation (ICRA)*. Pp 691-696 Barcelona, Spain, April 2005.
- [8] Moodle Website. <http://www.moodle.org>.
- [9] G T McKee, D I Baker, P S Schenker : "Robot Spaces, Module Networks and Distributed Robot Architectures", *Proceedings of the IROS 2004 Workshop on Networked Robotics:issues, architectures and applications*, Sendai, Japan (2004).
- [10] J. Fernandez, R. Marin, R. Wirz, "Online Competitions: An Open Space to Improve the Learning Process," *IEEE Trans. on Industrial Electronics*, vol. 54, no. 6, pp. 3086-3093, Dec. 2007.
- [11] Elhajj, A. GORADIA, N. XI, C. KIT, Y. LIU, T. FUKUDA, "Design and Analysis of Internet-Based Tele-Coordinated Multi-Robot Systems", *Autonomous Robots* 15, 237-254, 2003.
- [12] Alaa Khamis, Ayman Abdel-Rahman, Mohamed Kamel, "A Distributed Architecture for Mobile Multirobot Remote Interaction," *IEEE 2006 Workshop on Distributed Intelligent Systems*, Prague, Czech Republic, June 15 - 16, 2006.



Pedro J Sanz is Associate Professor at the Department of Computer Science & Engineering of University of Jaume-I (Spain). Dr. Sanz has been active since 1990 in R&D within several European and National research projects on Advanced Robotics. He is author or co-author of a broad range of research publications, IEEE member and other international scientific societies such as IAPR or ECCAI. Dr. Sanz is working presently as associate Editor in *IEEE Trans SMC(C)* and others some ISI International journals, and as Program Committee member of several international outstanding conferences, including IEEE ICRA and IROS. His current research interests are Multisensory based dexterous Manipulation, Assistive Robotics, Telerobotics and Human-Robot Interaction.



Josep Fernández (M'04) received the MSc degree in Computer Science from the School of Computer Science of Barcelona (Spain) in 1989, followed by the Ph. D. in Computer Science from the Technical University of Catalonia, Barcelona, Spain, in 1998.

He is currently an Associate Professor in the Automatic Control and Computer Engineering Department at the Technical University of Catalonia, giving lectures on Digital systems design and Robotics. His research areas are in the field of Computer Vision and Robotics, mainly oriented to sensor based navigation in robotics applications, telerobotics and telelaboratories. He is currently working in projects on telerobotics, online laboratories and robotics systems for disable people.

Dr. Fernández is author of many contributions in international conferences and book chapters in the field of computer visions and robotics and he was member of the organisation committee in various international conferences. He is member of the EURON Education and Training group.



Raúl Marín received a B.Sc. degree in computer science engineering (1996) and a Ph.D. in Engineering (2002) by the Jaume I University of Castellon (Spain). In 1996 he worked in the Multimedia department of BYG Systems Ltd, Nottingham (UK), as a software developer. In 1997, he joined Lucent Technologies (Bell Labs Innovations) and worked as researcher, software developer and software architect at the Switching and Access Division. In 1999 he began to teach and research as professor at the Jaume I University. Now, he is working as researcher at the Department of Computer Science of the Jaume-I University (Spain) and lectures Computer Networking and Distributed Systems in the same university. His research interests lie mainly in the field of Multirobot Distributed Systems, High-Performance FPGA-based Vision, Internet Telerobotics, Network Protocols, Human-Computer Interfaces, and Tele-Education. He is author or co-author of multiple research publications on these subjects.

Nowadays, Dr. Marín is conducting a reasearch project in collaboration with the UPC University in the area of Tele-Laboratories and Network Protocols for Multirobot Architectures (E-RIADS-MEC).



Raúl Wirz received the M.S. degree in computer science from the University of Jaume I (UJI), Castellon, Spain, in 2004, and is currently working toward the Ph.D. degree in computer science and telerobotics. His Ph. D. thesis is about Networked Robots.

In 2003, he joined the Robotic Intelligence Laboratory, UJI, and he worked on research projects related to Internet telerobotics, Internet transport protocols and human-robots interfaces. His current research interest includes teleoperation, visual servoing, 3-D virtual environments and IP protocols.