DISTRIBUTED PLATFORM FOR TRAINING IN MOBILE ROBOTICS THROUGH INTERNET

L. Payá, O. Reinoso, L.M. Jiménez, A. Gil, C. Fernández
Departamento de Ingeniería de Sistemas Industriales
Miguel Hernández University. Avda. de la Universidad s/n
Ed. Torreblanca. 03202 Elche (Alicante). Spain.
E-mail {lpaya, o.reinoso, luis.jimenez, arturo.gil, c.fernandez}@umh.es

Abstract

Nowadays, Internet-based techniques have become a powerful tool in all the fields of engineering education. This paper presents a distributed platform that allows the students to access the robots available in the laboratory through Internet. With this platform, a remote environment is created so that the students can carry out different experiments over several available equipments in the lab, with a flexible schedule. This way, the users can create algorithms of basic reactive control and test them on real robotic platforms. Currently, there are three kinds of robots available. The main objective of the work is create a common platform that allows the access to all of them using a common interface of communication in a transparent way from the point of view of the user.

Keywords
Mobile robots, Collaborative robotics, Distributed control, Remote laboratory, CORBA.

1. INTRODUCTION

In engineering education, the experimental formation of the students has always played a very important role. This way, the laboratories must provide all the necessary resources to put into practice the knowledge the student acquires in the theoretical study of the subjects. However, the traditional laboratory has a set of lacks that may make difficult the learning of the student, such as the scheduling, the limited number of equipments and the evaluation process. The students cannot experiment freely and cannot dispose of all the time they require to achieve the objectives of the practical lesson. Besides, the cost of the setting and maintenance of the laboratory is usually high, so the contents of the practical lessons will be strongly conditioned by the number of available equipments. This way, several students will have to share the same equipment and they can not experiment freely to improve their own knowledge. To finish, the traditional evaluation system is carried out through a report that is checked by the professor to verify if the student has reached the desired level of knowledge. However, the student does not know the result of the evaluation until the professor feedbacks to him the result and, in addition, in that moment the student can not modify the report delivered to improve the evaluation.

The use of Internet-based techniques to carry out practical lessons in engineering supposes several advantages. Students are highly motivated to make use of and benefit from the resources available in remote environments through Internet [1]. The remote access to the laboratory through Internet to perform experiments solves the problems previously exposed. The students can access the laboratories in a free timetable, from their own house, and disposing of all the time they require to accomplish the aims of the lessons. Also, the students can access individually to the systems, independently of the number of equipments available. At last, through an online evaluation system, the student can know the evaluation results in real time and the professor can take into account not only the final results but also the work the student has actually carried out to evaluate the practices. Even, several educative centres could share their equipments and so, the associated expenses [2].

This paper presents a platform that allows the students to access the mobile robots available in the laboratory through Internet. Several remote labs using robots in a remote environment have been developed. In [3], an educational virtual laboratory for training in robotics that allows either the simulation of a robot arm and the tele-operation of the equivalent of a real robot is presented. Also, in [4] a client/sever architecture is described for the remote control of a manipulator robot. [5] presents an architecture and an interface to control and teleoperate the irB21 robot. At last, [6] presents and discusses several applications using mobile robots through Internet at EPFL in Lausanne.

The distinguishing feature of the proposed platform consists on having a team of different mobile robots that the students can use to test the algorithms they develop. The main objective is to develop a common platform that allows this access from the same graphical interface independently of the kind of robot the user wants to monitor or control.

The paper is organized as follows. Section 2 describes the architecture of the system that is based in the CORBA standard. It allows the access to the different robots. In Section 3, the application to a specific Robot (WiFiBot) is described. Section 4 explains the use of the system, from the point of view of the final user. At last, section 5 summarizes the conclusions of this work.

2. ARCHITECTURE OF THE SYSTEM.

In our robotics laboratory, three different kinds of mobile robots are available at the moment. The first one, the B21r is a 4-wheels synchronous kinematics robot used mainly for research. It carries two on-board computers with Red Hat Linux and the Mobility Software (libraries based in the Corba 2.0 standard used to communicate with the robot). The communication with the robot is made through a wireless Ethernet link and it carries a wide variety of sensors: sonar, infrared, laser and encoders. Besides, it carries on its top a pan-tilt unit with two Sony XC999 parallel cameras. The second robot is the WiFiBot. It is a low-cost non-holonomic robot. 2 models are available at the moment, the SC and the 4G. The first one includes a BECK controller (SC12) with a real time operating system (RTOS). The second has an ‘Access Cube’ from 4G-Systems (MIPS AMD Processor, 400 MHz, 64 Mb RAM and 32 MB Flash Memory). Both models carry an Ethernet colour camera, 2 infrared sensors in the front side and 4 optical encoders, and the communication is made through WiFi. Also, a PC-104+ board with Linux Debian operative system has been added in both models. To finish, the EyeBot robot is managed by a Motorola 68332 Microcontroller and it is a robot with very limited sensorial capabilities, with colour camera, infrared sensors and encoders. These robots are shown on fig. 1.

Fig. 1. B21R, WiFiBot and EyeBot robots used to develop the common communication platform.

Then, having a heterogeneous team of robots, that have different hardware architecture and sensorial capabilities, the objective is to create a common communication layer that allows the access to each element of the system in an easy and transparent way to the user.

2.1 Communication Architecture for distributed applications CORBA.

CORBA (Control Object Request Broker Architecture) [7] provides an object-oriented methodology defined for the implementation of distributed applications. It will constitute the reference model used in the general design of the platform. Two fundamental features of CORBA are:

(a) Separation between interface and implementation. The interfaces of the objects are specified in a language specially defined with this goal, IDL (Interface Definition Language), which is part of the CORBA standard. IDL isolates the interface of the object from its implementation, offering more portability and interoperability this way.

(b) Independence of the location. An object offers services through its operations and attributes, which are visible through its interface. The clients can require the accomplishment of certain operations over an object with independence of its location. The element that allows this transparency in the location and access to the objects is ORB (Object Request Broker).

Then, a CORBA architecture allows that a client application requires to make operations over a CORBA object, that resides in a server. The implementations of the objects that are in the server side define the methods that implement the IDL interfaces. These objects can be implemented in different programming languages. This standard has been used in similar tasks of communication, resulting efficient to interchange information in a team of heterogeneous robots in RoboCup F2000 league. [8].

2.2 Developed Model.

The CORBA-based architecture developed is shown on this figure.

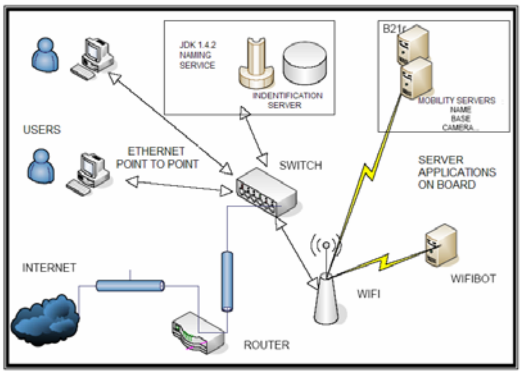

Fig. 2. Architecture developed to monitor and control the robots.

The components of this architecture are:

- Identification server. For the correct functioning of the system, it is necessary to know which elements are connected in each moment and which services provide each of them. This function is carried out by the identification server. It provides updated information abut the active servers (robots) in the system. It has been created using Java JDK 1.4.2.
- Robots servers. These servers manage the services that each robot offers. This way, their objective is to solve the high level requirements that arrive and run the code that implements each service (e.g. reading odometer values, capturing camera images, etc.). This server has been created using C++ with ORBacus 4.3.0 software.
- Supervision Application: It is a graphical interface that has been designed to maintain a list of the members of the team that are active and to show the most relevant information of each element that is connected to the network. This application is based on Java JDK 1.4.2. Its goal is to solicit, periodically, the services of the robots that are active in the system. It has been taken profit of the interoperability that exists between ORBacus and Java JDK to implement these three applications. C/C++ is required to program the services in the mobile robots, and JAVA to develop the supervision tool.

3. SERVICES IMPLEMENTED ON THE WIFIBOT ROBOT.

This section exposes how the previous concepts have been applied to one of the available robots in the laboratory, the WiFiBot. Four main applications have been developed to integrate this robot in the platform:

- Server. The robot is provided with a server program for the SC12 controller and a Windows client application that acts as an interface to control and monitor the robot. This server provides methods to control the linear and steering speeds of the robot and to read the current values of the sensors. Methods to calculate kinematics and odometer, a timer to update the odometer and a function to linearize the infrared sensor lecture have been added. Also, the protocol has been changed to text mode in order to be used by any telnet client to manage the robot.

- Client. A C++ client class has been developed so that the communication is transparent for the user, who has just to develop the algorithm to control the robot independently of the model of WiFiBot he wants to control. Through this client class, the user can program any of the functions that the implemented server offers. Also, a client class for the camera has been developed. The user has to create a socket with the IP and PORT required and to send a request to the camera, using the HTTP protocol. To do all this, a client class has been implemented for the camera, including an independent thread that is in charge of the image reception and other methods to obtain the last image in jpg format.

- Multi-thread bridge server. This server runs on the PC board mounted on the robot, and allows several programs to access simultaneously the services that the SC12 provides. The main advantage of this server is that several clients can access the low-level services that the robot offers. For example, a program to monitor the robot can be running simultaneously with the program that controls it. However, when a client is controlling the robot, the rest of clients who want to connect could just monitor it. This server establishes a link with the SC12 using the client class and a socket remains opened waiting for possible clients. If a new client accesses the system, an independent thread is created to attend this client. The different threads run in parallel, sharing the same space in memory. To avoid errors of communication with the SC12 produced by simultaneous requests of several clients, a mechanism of synchronization must be used. In this case, a mechanism of mutual exclusion (mutex) has been implemented, blocking the fragments of code where the requests to the SC12 are carried out, so, only a thread con be running this fragment of code at the same time. The rest of threads are blocked until the first thread finishes the request process.

4. SUPERVISION APPLICATION TO CONTROL THE ROBOTS.

A Java GUI application has been developed to provide the remote access to the robots. This application acts as a CORBA client, so it is capable to demand services that act over the actuators and sensors of the robots. When the client application begins, it shows a hierarchy tree that contains the available robots and their interfaces. All this information is provided by the Naming Service. Fig. 3 shows the global appearance of the application. There is a panel for each robot that is connected at the moment. The left one is the WiFiBot panel. At its top, there are some controls to move the robot manually. With the arrow buttons, linear and steering speeds can be increased or decreased, and the central button stops the robot. At last, the current linear and steering speeds are shown.

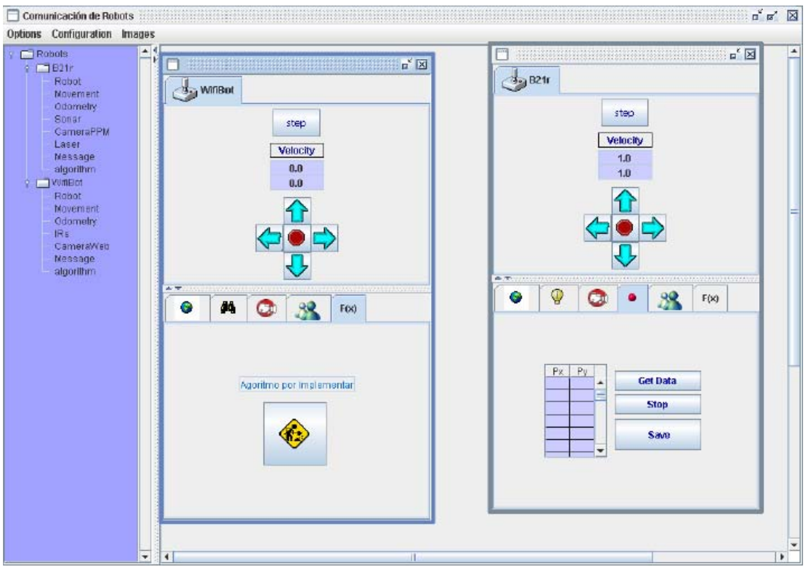

Fig. 3. Interface developed. There is a sub-window for which robot that is connected at the moment. Through this sub-window, the student is able to control and monitor the status of the robot.

Fig. 4. Panels for the monitoring of the WiFiBot (a).

At the bottom of the WiFiBot sub-window, there is a panel with several lapels that allows the access to the sensor data and the control of the robot. These lapels, shown on Fig. 4, are:

- Odometer panel: The user can reset the odometer of the robot or set it to a pre-determined value. Also, the odometer data can be continuously read and saved into a file.
- Infrared panel: Current distances are shown and it can be configured to be continuously written in a file.
- CameraWEB panel: The user can visualize the images captured by the Web Camera. When 'stop' button is pressed, the camera gives up capturing and the last image can be saved in a jpg file.

- Algorithm panel: The user can download through this lapel the program he has implemented to control the robot so he can test different control algorithms. First of all, the user has to look for the C++ source file to be sent. Once it is localized, when pressing the button 'Compile', the file is transferred and compiled in the remote server (the user has not got the necessary libraries to compile it in his computer) and a window appears to the client, showing him the results of the compilation. In there is no error, when pressing the button 'Run', the uploaded algorithm will begin running. At any moment, the user can stop execution using the button 'Stop'.

Currently, this system is at the disposal of the students so that they can connect to it during a pre-established period to train with the robots. The student needs a validation for accessing the system. This way, the students can build algorithms of basic reactive control. Reading the measures of the sensors and the camera, and taking into account the kinematics of each robot, the student must program the behaviour of the robot so that it accomplishes a determined task (e.g. following a pre-recorded route). During the navigation of the robot, all the data can be saved in several text files so that the student is able to get the graphical evolution of the necessary variables, such as trajectory, speed, etc. Also, taking into account that the platform developed allows accessing each robot in a transparent way, it will also allow communication between robots. This way, the student can program algorithms to carry out tasks that suppose collaboration between the robots.

5. CONCLUSIONS.

A distributed platform for the communication with and between the members of a team of heterogeneous robots has been presented. With this platform, the students of robotics are able to develop their own algorithms and test how they work in a real robotic platform, in a remote way, through Internet, and managing with all robots using the same graphical interface. Besides, this platform allows, not only testing algorithms of basic reactive control but also carrying out tasks that need collaboration between robots, what implies sharing information between the members of the team. This system is currently being used for the accomplishment of practices in mobile robotics by engineering students, taking profit of the advantages the education through Internet has. This platform has been used for a collaborative task (multi-robot formations) using several WiFiBots with good results [9].

Acknowledgements

This work has been supported by Ministerio de Educación y Ciencia through project DPI2004-07433-C02-01 'Herramientas de Teleoperación Colaborativa. Aplicación al Control Cooperativo de Robots' and the Project PCT-G54016977-2005 'Robots Cooperativos para la vigilancia e inspección de edificios e instalaciones industriales'.

References

[1] Stafford, R. F., "Understanding motivations for Internet use in distance education", *IEEE Transactions on Education,* 49(2), 1995, pp. 301-306

[2] Berger, K. A., Topol, M. T., "Technology to enhance learning. Use of a web site platform in traditional classes and distance learning", *Marketing education review*, 11, 2001, pp. 15-26

[3] Candelas, F. A., Puente, S. R., Torres, F., Ortiz, F. Gil, P., Pomares, J., "A virtual laboratory for teaching robotics". *International Journal of Engineering Education,* 19(3), 2003, pp. 363-370

[4] Thamma, R., Huang, L. H., Lou, S. J., Diez, C. R., "Controlling robot through Internet using Java", *Journal of Industrial Technology,* 20(3), 2004

[5] Khamis, A., Rivero, D. M., Rodríguez, F., Salichs, M., "Pattern-based architecture for building mobile robotics remote laboratories", *Proceedings of the IEEE International Conference on Robotics and Automation,* 3, 2003, pp. 3284-3289

[6] Siegwart, R. and Saucy, P., "Interacting mobile robots on the Web", *Workshop Proceedings of the IEEE International Conference on Robotics and Automation*, 1999

[7] OMG, Object Management Group, "Common Object Request Broker: Architecture and Specification. Revision 2.0", 1995

[8] Utz, H., Stulp, F., Moeld, A., "Sharing belief in teams of heterogeneous robot", *Proceedings of RoboCup-2004 symposium,* 2004, pp. 508-515

[9] Payá, L., Gil, A., Reinoso, O., Ballesta, M., Ñeco, R., "Behaviour-based multi-robot formations using computer vision". To appear in *Proc. of the 6th IASTED International Conference on Visualization, Imaging and Image Processing*