REMOTE CONTROL OF REAL EXPERIMENTS VIA INTERNET

Carine Campos Teixeira, carinect@ita.br
Jeeves Lopes dos Santos, jeeves@ita.br
Cairo Lúcio Nascimento Júnior, cairo@ita.br
Instituto Tecnológico de Aeronáutica, Divisão de Engenharia Eletrônica,
12228-900 – São José dos Campos – SP, Brazil

Luis Filipe Wiltgen Barbosa, wiltgen@univap.br
Universidade Vale do Paraíba - UNIVAP,
Faculdade de Engenharia, Arquitetura e Urbanismo - FEAU,
12244-000 – São José dos Campos – SP, Brazil

Abstract. *This article is concerned with the remote control of real experiments via internet with the aim of sharing expensive laboratory resources by making them available at any time to remote users. The following software architecture is proposed and implemented: a) the remote user sends commands to the real devices using a standard web browser running on his computer; b) these commands are received via internet by a local computer that runs a standard web server software and written to a command file; c) the same local computer runs another program (called "experiment control program") that: 1) reads the command file and accordingly controls the local real experiment using its physical interfaces, 2) updates a local status file with information about the current status of the real experiment; d) this status file is then read and properly formatted by the web server software such that the remote user receives feedback about the real experiment in his computer via web browser. Alternatively the web server software and the experiment control program can run in different computers which should be somehow interconnected (for instance, if the computers are in the same local area network they can simply use the file-sharing facilities provided by their respective operational systems). This article presents the implementation of this proposal for a student-built mobile robot called ROMEO III. An additional user feedback is provided in near real-time by video from a web camera which is displayed in the user web browser. By using the proposed architecture, any experiment could be used as a component of a real remote laboratory with minimal modifications, even if the experiment was not initially built to be remotely controlled.*

Keywords*: WebLab, real remote laboratory, web-based laboratory, distance learning.*

1. INTRODUCTION

The remote control of real experiments can be very useful in a laboratory environment, since expensive resources can be used more effectively (Bencomo, 2004). For instance, more users can have access to expensive equipment if they do not need to be physically present at the laboratory. However, an adequate hardware/software interface is needed by these remote users such that they can control and monitor the laboratory resources and if the remote equipment was not initially designed or built to be remotely controlled.

Nowadays the internet is ubiquitous, its protocols are used as the standard for network communications, and anyone knows how to use a web browser. Therefore it makes sense to use: 1) a web browser as the remote user interface, and 2) a web server and auxiliary programs to control and monitor the devices in the laboratory. The laboratory is then called a *weblab*.

In this article the following software architecture is proposed and implemented: a) the remote user sends commands to the real devices using a standard web browser running on his computer; b) these commands are received via internet by a local computer that runs a standard web server software and the commands are written to a local command file; c) the same local computer runs another program that: 1) reads the command file and accordingly controls the local real experiment using its physical interfaces, 2) updates a local status file with information about the current status of the real experiment; d) this status file is then read and properly formatted by the web server software such that the remote user receives feedback about the real experiment in his computer via web browser.

Several authors have proposed different approaches for sending the data from a web browser to a web server to control remotely experiments (Casini, Prattichizzo and Vicino, 2004; Chicullita and Frangu, 2002). Alternatively the web server software and the experiment control program can run in different computers which should be somehow interconnected. For instance, if the computers are in the same local area network they can simply use the file-sharing facilities provided by their respective operational systems. An additional and important user feedback can be provided in near real-time by video from a web camera which is displayed in the user web browser.

This paper is organized as follows. Section 2 presents the proposed weblab system architecture. Section 3 describes the remote experiment, the Romeo III mobile internet robot, and some results are shown. Concluding remarks and some directions for future researches are presented in the Section 4.

2. THE WEBLAB SYSTEM ARCHITECTURE

The weblab system architecture is composed of at least 2 computers in the same (local or wide area) network and the real experiments. One computer runs the web browser and the other computer runs the web server software and the software responsible for controlling and monitoring the real experiments. Fig. 1 shows how these systems are interconnected.

In our case the ROMEO III mobile robot was the remote device. A webcamera was added such that the remote user can watch the experiment in near real-time in his computer. The webcam provides an internet video link from the experiment to the remote user and it runs independently from the laboratory web server (the webcam has a built-in stand-alone web server).

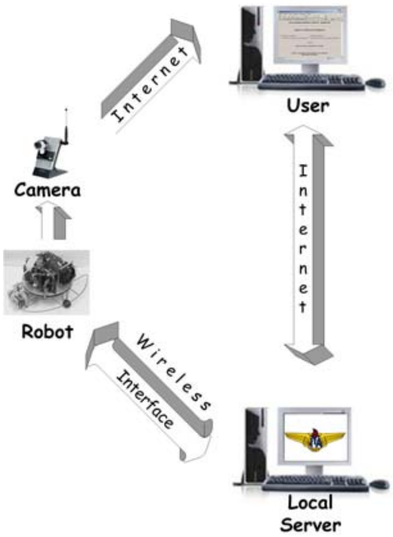


Figure 1. The weblab system architecture

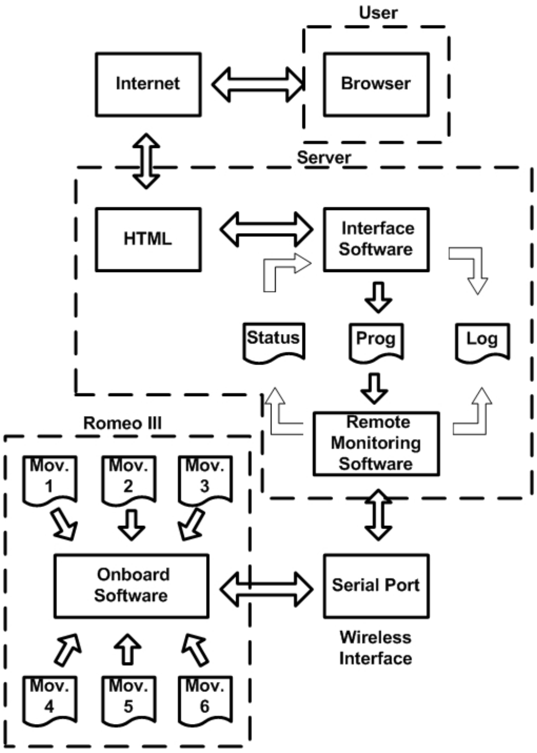


Figure 2. Flow of information between the remote user computer, the laboratory server and the remote equipment.

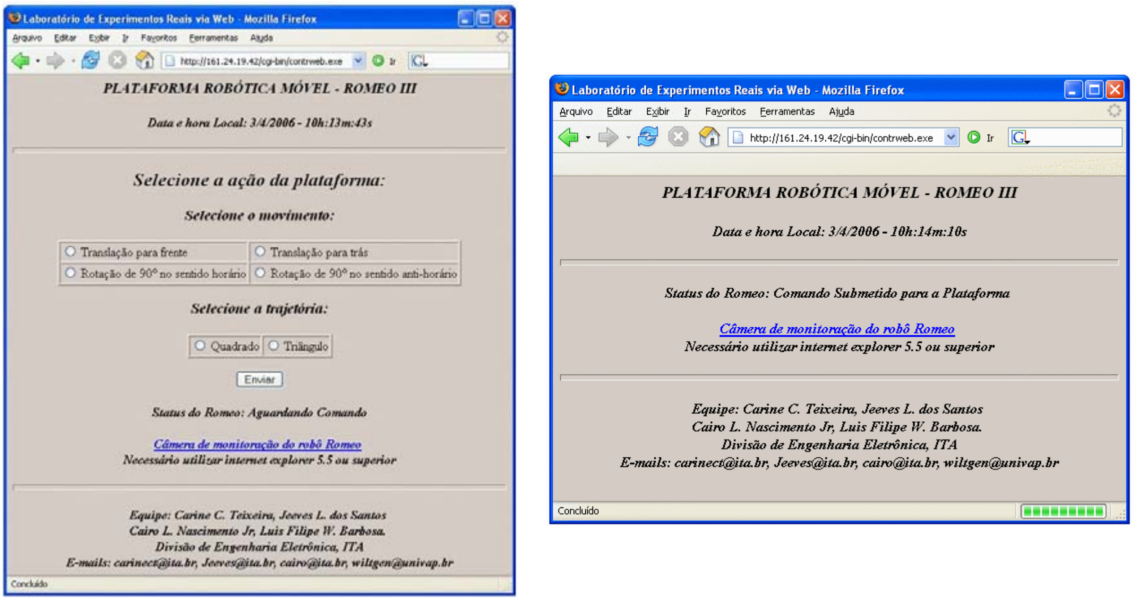
The flow of information between the computer running the web browser, the computer running the web server and the remote equipment is as follows (Fig. 2):

1. The remote user enters the internet address of the laboratory web server in his web browser.
2. The user web browser contacts the laboratory web server and receives from it an HTML page with the current status of the remote equipment (in this case if the robot is ready to receive tasks or not) and possible tasks (in this case movements) for the equipment in the laboratory.
3. The remote user selects one of the possible tasks/movements and the web browser sends this information to the laboratory web server.
4. The laboratory web server resends this information to the robot *Remote Monitoring Software* that may run on the same computer.
5. The remote monitoring software updates the log file (the selected task is registered) and the status file (with the information that a task has been submitted to the robot or that the robot is executing a task) and sends the selected task/movement to the *Robot Onboard Software* using a wireless serial interface.
6. The robot onboard software then activates the robot onboard systems to execute the selected task/movement. When the robot completes the task/movement, the onboard software sends a confirmation to the remote monitoring software.
7. The remote monitoring software then updates the status file with the information that the task was completed and the equipment/robot is now ready to receive and execute a new task.

After the web server sends the selected task to the remote monitoring software, it will continue to serve an HTML page with the information that the remote equipment is executing a task whenever it is contacted by any computer in its network.

3. THE ROMEO III INTERNET ROBOT EXPERIMENT

Figure 3 shows respectively the web pages that the remote user will see in his web browser: a) when he firstly enters the internet address of the laboratory web server in his web browser, and b) when a task has been submitted to the robot. These pages were generated using the standard HTML language such that the remote user can use any standard web browser (MS Internet Explorer, Mozilla Firefox, etc.).


Laboratório de Experimentos Reais via Web - Mozilla Firefox
Arquivo Editar Exibir Ir Favoritos Ferramentas Ajuda
http://161.24.19.42/cgi-bin/contrweb.exe
Ir
Enviar
Concluído

PLATAFORMA ROBÓTICA MÓVEL - ROMEO III

Data e hora Local: 3/4/2006 - 10h:13m:43s

Selecione a ação da plataforma:

Selecione o movimento:

○ Translação para frente	○ Translação para trás
○ Rotação de 90° no sentido horário	○ Rotação de 90° no sentido anti-horário

Selecione a trajetória:

○ Quadrado ○ Triângulo

Status do Romeo: Aguardando Comando

Câmera de monitoração do robô Romeo
Necessário utilizar internet explorer 5.5 ou superior

Equipe: Carine C. Teixeira, Jeeves L. dos Santos
Cairo L. Nascimento Jr, Luis Filipe W. Barbosa.
Divisão de Engenharia Eletrônica, ITA
E-mails: carinect@ita.br, Jeeves@ita.br, cairo@ita.br, wiltgen@univap.br

a) Initial web page.

PLATAFORMA ROBÓTICA MÓVEL - ROMEO III

Data e hora Local: 3/4/2006 - 10h:14m:10s

Status do Romeo: Comando Submetido para a Plataforma

Câmera de monitoração do robô Romeo
Necessário utilizar internet explorer 5.5 ou superior

Equipe: Carine C. Teixeira, Jeeves L. dos Santos
Cairo L. Nascimento Jr, Luis Filipe W. Barbosa.
Divisão de Engenharia Eletrônica, ITA
E-mails: carinect@ita.br, Jeeves@ita.br, cairo@ita.br, wiltgen@univap.br

b) Web page showing that a task has been submitted to the remote equipment.

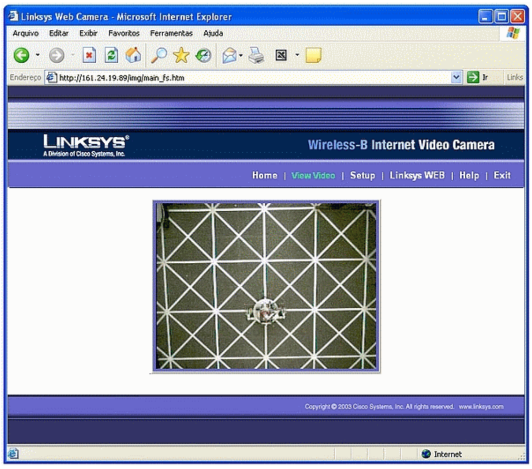
Figure 3. Examples of web pages seen by the remote users.

Figure 4 shows a camera view for the robot in the laboratory while the robot is executing a movement. In the experiment the webcam was a wireless (802.11b) internet video camera model WVC11B from *Linksys* (a division of Cisco Systems). The webcam provides an internet video link from the experiment to the remote user and it runs independently from the laboratory web server (the webcam has a built-in stand-alone web server). However, the video stream from this camera can only be viewed using Microsoft Internet Explorer$^{®}$ version 5.5 or above as the remote user web browser.

The Romeo III mobile robot, shown in Fig. 5, is a robot built at ITA (Gabriel, 2005) as an experimental base for research in robotics. Its dimensions are 35 cm x 20 cm (diameter x height) and about 3.5 kg of weight. This robot is capable of autonomous navigation in a structured, static and previously known environment. It uses 5 infrared light sensors to follow white lines in a black floor (shown in Fig. 4) and a wireless Bluetooth serial link to communicate with a base computer. It is important to note that the ROMEO III robot was not originally designed or built to be controlled remotely.

The autonomous mobile robot ROMEO III is comprised of 5 systems as follows:

1) External Communication System: a wireless Bluetooth serial link.
2) Onboard Computing System: a 386Ex DOS-based SBC (single board computer) and two 8051 microcontrollers.
3) Sensory System: 1 contact detection sensor, 2 odometry sensors, 5 white line detection sensor, 1 low-battery detection sensor.
4) Energy System: a 12V 3.2 Ah battery.
5) Propulsion System: a power driver electronic circuit and 2 DC permanent magnetic motors.


Linksys Web Camera - Microsoft Internet Explorer

Endereço http://161.24.19.89/img/main_fs.htm

Wireless-B Internet Video Camera

Copyright © 2003 Cisco Systems, Inc. All rights reserved. www.linksys.com

Arquivo
Editar
Exibir
Favoritos
Ferramentas
Ajuda
Ir
Links
LINKSYS
A Division of Cisco Systems, Inc.
Home | View Video | Setup | Linksys WEB | Help | Exit
Internet

Figure 4. Camera view for the robot in the laboratory while the robot is executing a movement.

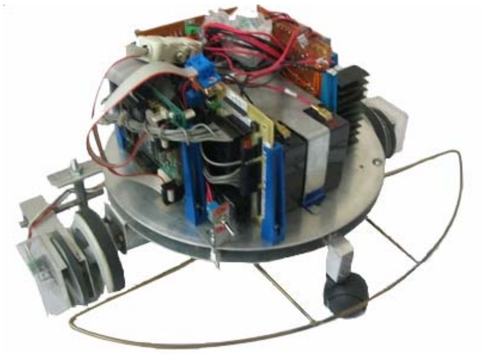


Figure 5. Autonomous mobile robot ROMEO III.

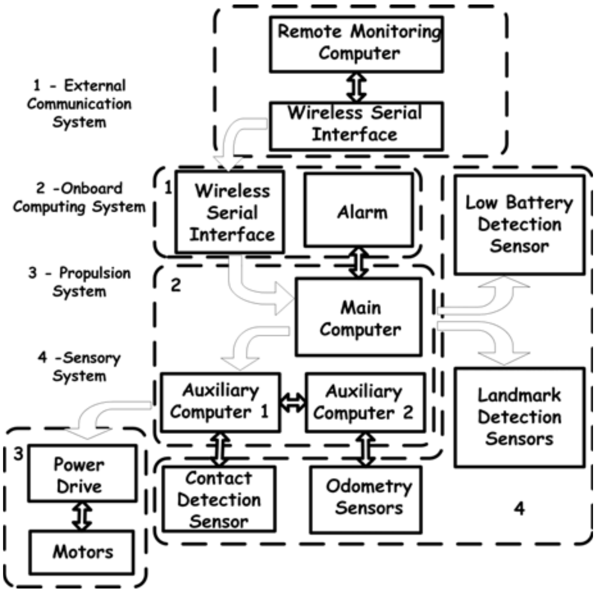

1 - External Communication System
2 -Onboard Computing System
3 - Propulsion System
4 -Sensory System
Remote Monitoring Computer
Wireless Serial Interface
1
Wireless Serial Interface
Alarm
Low Battery Detection Sensor
2
Main Computer
Auxiliary Computer 1
Auxiliary Computer 2
Landmark Detection Sensors
3
Power Drive
Motors
Contact Detection Sensor
Odometry Sensors
4

Figure 6. Block diagram of the ROMEO II robot system architecture.

Initially, the ROMEO III robot receives from the remote monitoring computer, via its wireless serial interface, a file with the description and parameters for the possible trajectories or movements that it will be later available for the remote user. In our experiment the possible movements that a remote user can select are: a) forward translation, b) backward translation, c) 90° left rotation, d) 90° right rotation, e) square trajectory, f) triangular trajectory.

When the robot is ready to accept commands, the remote monitoring computer just need to send a code that identifies the selected trajectory. The robot sends back 3 possible codes that identify in which phase it is: 1) the command with the selected trajectory was accepted, 2) the trajectory is being executed, and 3) the trajectory has been completed and the robot is ready to accept new commands.

The computer that runs the web server and the remote monitoring software is a standard PC (ATTHLON XP 1500+ processor with 1.35 GHz, 450 MB RAM) running MS-Windows XP as the operational system. The web server software is Apache version 1.3.34. The web server receives the trajectory that was selected by the remote user as HTML data in a form. This form is interpreted by a CGI (*Common Gateway Interface*) program written in C that communicates with the remote monitoring software (written in MATLAB® version 6.5.1) via files.

5. CONCLUSIONS AND FUTURE RESEARCH

This article proposes a simple software architecture for a remote control laboratory using Internet technology (web browser, web server, internet protocols). The proposal was tested using an autonomous mobile robot which was not designed or built to be remotely controlled. The tests have shown that the proposed software architecture works well for the autonomous mobile internet robot experiment.

Possible directions for future research are: 1) testing of the proposed software architecture with other remote control experiments, such as the magnetic levitation experiment (Barbosa *et al.*, 2004), 2) testing of the proposed architecture using just one web server to control multiple local experiments, 3) improvement of the remote user interface using Java programming, 4) investigation of remote user interfaces are not web browser based.

6. ACKNOWLEDGEMENTS

The authors wish to acknowledge the support for this research provided by CAPES and the Division of Electronics Engineering at the Aeronautical Institute of Technology (ITA). The authors wish to thank Gabriela W. Gabriel and Eduardo H. Yagyu for designing and building the ROMEO III robot and for their help and suggestions.

7. REFERENCES

Barbosa, L. F. W., Costa, F. E. D.F., Ludwig, G. O. and Nascimento Jr., C. L., 2004, "Controle Analógico de um Levitador Magnético (MagLev) de Simples Construção e Operação", XV Brazilian Conference on Automatics, Gramado, Brazil.

Bencomo, S. D., 2004, "Control Learning: Present and Future", Annual Reviews in Control, Vol.28, pp.115-136.

Casini, M., Prattichizzo, D. and Vicino, A., 2004, "The Automatic Control Telelab – A Web-Based Technology for Distance Learning", IEEE Control Systems Magazine, Vol.24, No. 3, pp. 36-44.

Chiculita, C., Frangu, A., 2002, "A Web Based Remote Control Laboratory ", Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics, Orlando, USA.

Gabriel, G. W., 2005, "Projeto e Construção de uma Plataforma Robótica Móvel em Ambiente Estruturado", MSc. Thesis, Aeronautics Institute of Technology (ITA), São José dos Campos, Brazil. (in portuguese).

8. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.