

SOFTWARE PLATFORM FOR VIRTUAL LABORATORY OPERATION BY REMOTE CONTROL OF INDUSTRIAL ROBOTS

POPA, S[telian]; BRICEAG, C[laudia] V[iioleta]; NICOLESCU, A[drian] F[lorin] & DORIN, A[lexandru]*

Abstract: Virtual laboratory concept is increasingly used year by year. Overcoming the time and space barrier, a virtual laboratory solution like the one presented has a supplementary major advantage: safety operation. Present paper describes custom developed software used by students to improve their robotics programming skills. The software has a graphical user interface helping users to operate industrial robots in virtual environment and remote operation the real robots, located in the robotics laboratory prepared to share its facilities.

Key words: virtual laboratory, industrial robots, remote operation, off-line programming and simulation

1. INTRODUCTION

The Internet plays a very important role in robotic remote operation applications. Today the bandwidth limitations are no longer a problem, as was 10 years ago when the first remote operating robotic systems were launched: the Mercury Project (Goldberg et al, 2000a), Telegarden Project (Goldberg et al, 2000b), and the system developed by University of Western Australia (Taylor & Dalton 2000).

Since many other applications were developed at different universities around the world, most of them use browser embedded applets written in JAVA programming language. The disadvantage of this approach is that end-users are restricted to download the JAVA applets whenever they start using the virtual laboratory. The internet connection delay is another problem, because if lag appears result can be unexpected and accidents can occur.

2. SOFTWARE APPLICATION DEVELOPMENT

2.1 Selected platform based on Internet statistics

In the past three months (February - April 2010) we have made a study over 47,000 Romanian internet users and the conclusions are as follow: 96.8% of users are using a Windows family operating system and 0.3% of users are using a Linux family operating system; only 77.11% of users have JAVA support (***) (2010a) - meaning that Java Virtual Machine is not installed, and it have to be installed before the applets start.

It results that, if we are targeting maximum compatibility with potential users it is preferable to develop the application in Windows operating system. Based on these observations we have developed StudRob - a software application designed in C++ offering some major advantage versus JAVA similar applications: better integration of serial port programming, a more accurate and spectacular 3D animation used for the simulation as well as an accomplishment of 96,8% integration with existing users actual requirements.

2.2 Laboratory facilities

The client application of StudRob has a component designed to simulate one real robotics laboratory, situated in EMST Faculty of Politehnica University of Bucharest.

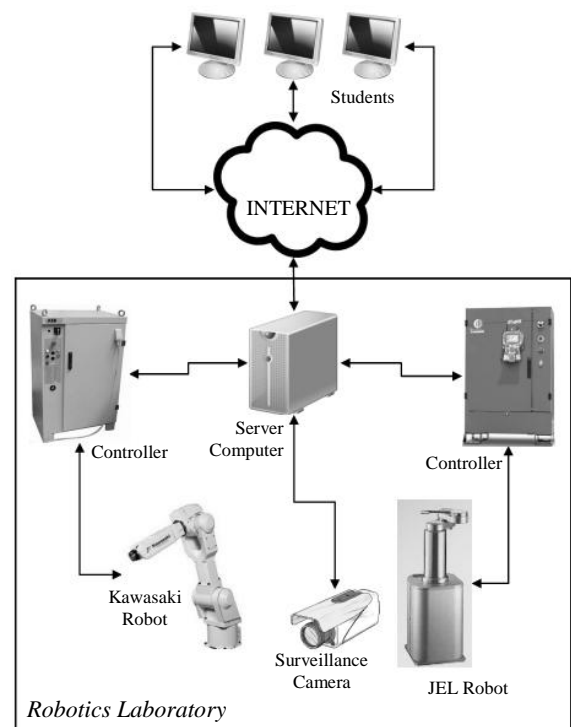


Fig. 1. Hardware schema of Robotics Laboratory

Hardware facilities of this lab includes: Jel SHR3000 (3-axes SCARA type robot, having a PLC controller), Kawasaki FS10 (6-axes articulated arm robot, having a D-Series controller), ABB IRB 1600 (6-axes articulated arm robot, having a S4controller, Fanuc ARC Mate 100iC (6-axes articulated arm robot, having a R-30iA controller), a PC as multirole server, and a video surveillance camera.

2.3 StudRob software description

PC server runs following main software packages: Debian - a free Linux-based operating system, CSF - a Linux Firewall, Apache - a HTTP server with PHP support, MySQL - a relational database management system, StudRob - the server component of our application.

StudRob is designed as a client – server applications and it has 2 components. First component is the client component running on student computer and the second one is the component running on laboratory server as a process on the Linux server located in the laboratory.

Every time the user application is started it connects via internet to the server. If this fails from different reasons the application can offline running only in simulation mode.

In order to offer a more intuitive user interface for StudRob a Ribbon interface was used where a set of toolbars are placed on tabs in a tab bar. The Ribbon (***) (2010b) is a contextual interface that offers functionality based on the context the user is working in. This Ribbon GUI provides the user interface of

the application with a large toolbar filled with graphical representations of control elements which are grouped by different functionality, as seen in Fig 2 (see also Ribbon using by are Microsoft Office Suite or RobotStudio).

3. WORKING PROCEDURE:

In order to download StudRob from <http://virtual.robotica-upb.ro> the users have to be first (registered) members of the website. After downloading and launching the application the user is asked to enter its unique username and password in a login form. These credentials are given by a system administrator. After application launching, the main screen (fig.2) shows a basic representation of the real robotics laboratory facilities, available components being shown in one row.

In this step the user decides what robot is going to operate and click on its icon. To point out the selected object selected robot's background becomes blurry and appropriate second screen is activated (Fig.3) accordingly robot type selection.

The second screen activates the menu and buttons from the ribbon. In order to operate selected robot, following actions are available: New session - creates a new working session and saves it on the server. Any actions of the users will be recorded and a history will be saved further; Open session - download the list of older sessions and allows users to open one. It acts like a browser history and allow student to continue a previously developed work; Save session - save the current



Fig. 2. StudRob main screen: robot selection

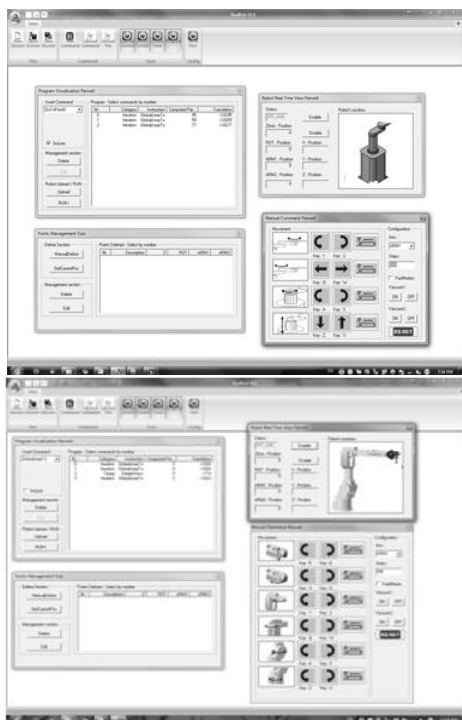


Fig. 3. StudRob second screen: active windows

working scheme and upload it on the server; Run command - Send the 'Run' command to the selected robot; Upload command - Upload the written program into the memory of the robot controller, Upload file - Upload the written program on the server and saves in into a directory.

Manual control for virtually teach-in the robot - Activates a new window (Fig. 3) including a virtual teaches pendant that allows virtual control of the robot. The results of virtual operation can be followed by user in the simulation window. Performing virtual robot's teach-in procedure the user may become familiarly with robot programming without any risk regarding real robotic system damaging by inappropriate operation.

Robot programming - Using same intuitive interface with buttons and drop-down menus the user can insert commands, in order to accomplish a full real robot operation working program (Fig. 3). When the work is finished the user can upload its program on the laboratory server and may request a time slot to view his program running on the real robot, in real time operation mode. After program checking and validation, the system administrator decides about scheduling of user's request and fixes an appointment for user remote access on real robot operation visualization. Before viewing the result in real mode, the user may review once again the virtual robot operation. For this job DirectX framework is used. In order to follow up real time robot operation the user need to skip from virtual operation window to the real robot operation window. The real time robot operation window contains is supplied by live streaming from the laboratory surveillance camera following up real robot operation accordingly robot programming made by user. As **Safety precaution**, to prevent real robotic systems damaging the operating speed of real robots is limited to 20% of the maximum speed and the working program can be loaded into robot controller only if administrator is in the laboratory. While the program is running the video captured by the surveillance camera is sent to StudRob and the student sees the action in real time.

4. CONCLUSIONS

The paper presented an original software package StudRob developed by authors in order for founding a virtual robotics laboratory allowing distance learning processes, robot's virtual programming and simulation as well as remote operation (via Internet) of real scale industrial robots. The software may be used by students to improve their robotics programming skills, as well as for performing personnel training procedures in commercial companies' by remote operating robotics facilities existing in company's training centres.

5. REFERENCES

- Goldberg, K.; Genter, S.; Sutter, C. & Wiegley J. (2000a). *The Mercury Project: a feasibility study for Internet robots*, IEEE Robotics & Automation Magazine, vol. 7, pp. 35-40.
- Goldberg, K.; Kusahara, M.; Dreyfus, H.; Goldman, A.; Grau, O.; Gržinic, M.; Hannaford, B.; Idinopulos, M.; Jay, M. & Kac, E. (2000b). *The robot in the garden and telepistemology in the age of the Internet*. MIT Press, Cambridge
- Taylor, K. & Dalton, B. (2000). *Internet robots: a new robotics niche*. IEEE Robotics & Automation Magazine, vol. 7, pp. 27-34.
- *** 2010a, Java (programming language), http://en.wikipedia.org/wiki/Java_%28programming_language%29C++, [http://en.wikipedia.org/wiki/C%2B%2B_\(02-06-2010\)](http://en.wikipedia.org/wiki/C%2B%2B_(02-06-2010))
- *** 2010b, Ribbon (computing), [http://wikipedia.org/wiki/Ribbon_\(computing\)_\(17-05-2010\)](http://wikipedia.org/wiki/Ribbon_(computing)_(17-05-2010))