

Telemanipulation using a PLC for web based education

DORIN POPESCU, CLAUDIU CHICULITA,
DAN SELISTEANU and LIVIA C. POPESCU

Department of Automation and Mechatronics

University of Craiova

107 Decebal Street, Craiova, 1100

ROMANIA

<http://www.robotics.ucv.ro>

Abstract: - Current technology enables the remote access of laboratory equipment via Internet. This work is about remote programming of a Programmable Logic Controller (PLC) that controls a manipulator, for didactic purpose (tele-education) or industrial purposes. The aim of the work was to develop the hardware and software structures to be used in a teaching context (remote learning by Internet). It was developed an external board, which link a PLC with a computer and a web server application. The web server application permits access to HTML pages with JavaScript routines. The application runs on the local computer (connected to the PLC) whereas the client need of an Internet-linked computer.

Key-Words: - Telemanipulation, Programmable Logic Controller, Internet

1 Introduction

Considering the importance of practices in Engineering education and training, the goal of the work was to implement a remote system useful in programming of a PLC in order to control a manipulator

The high cost necessary to carry out experiments (for didactical purposes) with manipulators and robots in various environments led to the development of remote facilities where the physical system, sensors and operating environment can be at a great distance

Some researchers achieved an analysis of the current challenges in Internet tele-programming and tried to find the possible solutions under the current environment, where exist some unsolved common problems associated with this new technology such as limited bandwidth and unreliable signaling [1], [2], [3].

Web-based tele-robotic systems have only become available on the Internet in the last decade. In the beginning they utilized a CGI interface to access the robots [4], [5], [6], [7], [8], [9]. With the introduction of Java and its integration into web browsers, developers could create tele-operated systems that sustained an interactive link to the robot during its execution. Many new tele-robotic systems were created that gave the user much more control and provided functionality, which under CGI could have never been possible [10], [11], [12].

The Internet lab technology offers the students the opportunity to work with sophisticated equipment,

of the kind they are more likely to find in an industrial setting, and which may be too expensive for most faculties to purchase..

This paper gives a solution for tele-programming of a certain Programmable Logic Controller (PLC). It tries to show an aspect of higher education and training based on Information Technology. The final purpose of the remote laboratory is to allow the student to take full control of the equipment, in order to fulfill the task required by the teacher.

2 Telemanipulation system

Remote laboratories are actual laboratory experiments that are run remotely via a web interface. Usually, the student can set some parameters on the web, then a software interface converts those parameters to a form that is accepted by the local computer running the experiment or he can tele-operate an equipment [1], [2], [3], [13], [14]. We proposed more, namely the student can achieve a program, which can run on the PLC. The student has thorough freedom to choose what experiment wants to do.

Our goal is remote programming of a PLC, which controls a manipulator for use in tele-education. The manipulator is controlled through PLC and Internet connection.

Our system consists of an AP-MICRO Programmable Logic Controller, a manipulator and a computer (Fig. 1).



Fig. 1. Telemanipulation system

2.1 Programmable Logic Controller (PLC)

The Romanian AP-MICRO PLC is a simple micro-PLC with 64 digital inputs, 48 digital outputs and 8 timers. This PLC has a large programmer console with a keyboard like a computer but with a small monitor [15]. We abandoned this programmer and an interface between a PC and the PLC was designed. It was achieved the software for editing PLC programs with its own language, too [16]. Our interface board is connected to the computer by a parallel port [17].

The achieved editing software for AP-MICRO PLC permits the realization of programs in statement list. Our application has a help menu about instructions of AP-MICRO PLC and about the way to realize programs.

2.2 Manipulator

The manipulator is made with pneumatic components. This manipulator is a pick-and-place manipulator. The positions are determined by discrete signals - "on" causes the manipulator to move to one extreme position and "off" moves it to the other extreme position. The manipulator has five powered pneumatic solenoids. If all solenoids are off, no air is applied to the manipulator's actuators.

Traditionally, a manipulator is connected to a PLC via digital I/O or specific hardware interfaces. In this case, the manipulator and PLC programs can only communicate at a very low level [18].

The control of the manipulator is made with the AP-MICRO PLC. Our application illustrates how a PLC may be used to control a manipulator from a great distance.

3 Methods of implementation

The use of Internet technology for remote programming application offers the advantage of

low-cost deployment. Almost every computer connected to the Internet can be used to control a tele-operable device. The downside is the limitation of varying bandwidth and unpredictable time delays. These Internet features should be defined and considered before designing an efficient remote programming system. Besides that, several functional requirements should also be defined before designing any tele-operable system.

3.1 Client-server method

First version of our implementation used an application client/server that it consisted of two programs: the server runs on the local computer (connected to the PLC; Fig. 2) whereas the client runs on any Internet-linked computer [17].

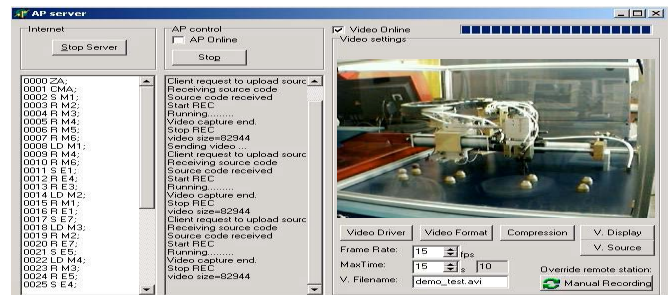


Fig. 2. Server application

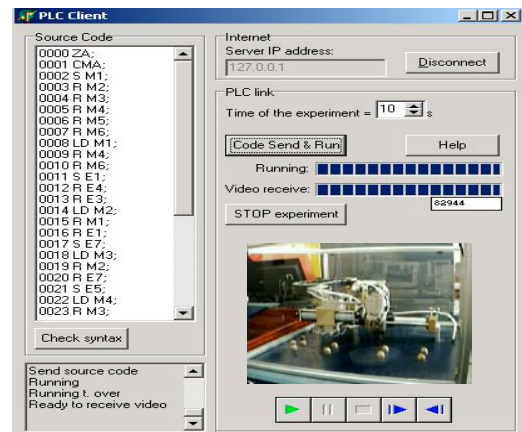


Fig. 3. Client application

A user (student or operator) might find himself in any location, with a computer and Internet. The user could do it in the following way: he could do a user program written in the original language of the AP-MICRO PLC. He could connect to the server and after acceptance he could send his user program. The user program for PLC run and the user could observe the movements and operation of the manipulator by means of a web-cam (Fig. 3). If there were delays the user could ask to record a movie (avi file) with the movements of the

manipulator and after a cycle of movements could download the movie and see it.

For the video feedback there were considered two options. The first was to send a live video feed (streaming) concurrently with the experiment. This test done by the authors proved that the existing connection did not permit a reliable good-quality live video feed.

The second solution, chosen by the authors was that of recording the video to the server and sending the resulting file to the client when the experiment was over. In this way the received video quality did not depend on the variations of the Internet connection speed, and the client could analyze (by reviewing) the received video.

The user opened the client application and wrote the source code of the program in the assembly language of the AP-MICRO PLC. On the client application it was also implemented the compiler so the user could check the syntax of the program at any time. When he was ready he connected to the server and his connection was accepted (if there were no other users active). He could choose some parameters regarding the video frame rate and the duration of the experiment based on how fast the process was and how much the program had to run. Then he sent his program to the server.

On the server application, once the code was received, it was compiled into the AP-MICRO PLC's machine code. If the compiler encountered any errors, an error message was sent to the client specifying the line number containing the error. If the code was compiled successfully then the execution was started.

Also the video camera started recording using the frame rate specified by the client and the video compressor selected by the server supervisor (operator) depending on the available ones. Once the time allocated (specified) for the experiment was over, the PLC and the video recording are stopped and the video was sent to the client.

3.1 Web base method

We wanted to have only one application and we developed the web server application with Delphi environment [19]. The web server application, using a web-cam, generates and manages the video image transmission to user. We changed the method of update of images when user interface is a classical web browser.

A user with a computer and Internet has to work in the following way after he navigate through our website: he has to ask permission to work with our PLC; he will receive an ID and a password; he will connect to our server and can read or download our

documentation about system (PLC-manipulator); he has to answer to some questions and then he can write a user program with the original language of the AP-MICRO PLC; the application points out the syntactic errors; he can send his user program. The user program for PLC will be run and the user can observe the movements and operation of the manipulator by means of a web-cam (Fig. 4).

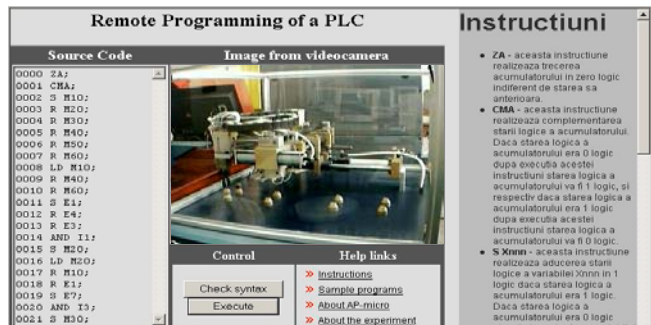


Fig. 4. User interface

The main learning goals for students are: learning of the structure of AP-MICRO PLC; learning of AP-MICRO PLC statements; learning of the way to implement a flowchart and a ladder diagram by statement list; learning to achieve a program for AP-MICRO PLC in order to control a manipulator.

To perform an experiment with our system the student has to read the documentation about system and PLC, because before the remote experiment he has to answer some questions. Only if the student answers correctly he can access the user interface, writes a program and run it.

Application structure:

1. A server application written in Delphi that:
 - a. generates pages to the client like an ordinary web server,
 - b. verifies the answers given by the student to the test,
 - c. assure the interface with the PLC,
 - d. generates and controls the stream of images sent to the client;
2. A client interface, available as a web page. At the client side only a simple web browser is required, no other plugins are necessary.

3.2 Transmission of images

The web server application generates and manages the video image transmission to user. In first implementation the image transmission was not so good (in real time) and the students was no satisfied from this point of view. We tried to improve this by video recording and then downloading of avi file. The image was better but it was no in real time. From this reason we changed the method of update of images.

The presented implementation is focused on availability to the user. Although there are a number of protocols that allow for video streaming and languages like Java that facilitate writing applets with a rich user interface and have support for different image transmission techniques, they have a number of disadvantages like resource usage and the need to have installed plugins in order to make them work. This implementation was intended to be available also on the simplest browsers and to run on the lowest hardware platforms.

The simplest method of displaying images from a video camera, when the client interface is a simple web browser is presented in Fig. 5. Because the request-response nature of the HTTP protocol, a new image is sent from the server to the client only when the client opens a connection and makes a request (through the web browser). The following steps take place:

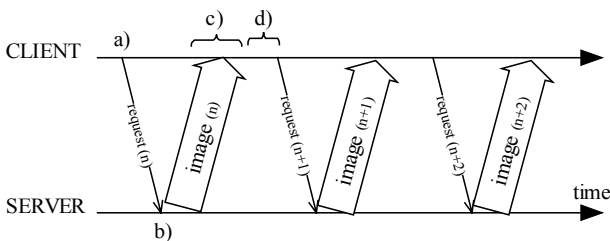


Fig. 5

- a) the browser requests an image from the server;
- b) the server receives the request and sends back the image immediately;
- c) the image loads progressively into the browser (the time taken to be loaded completely depends on the speed of the connection);
- d) after the image has been completely loaded, a short interval is waited for (in order to let the human operator perceives the whole image);
- e) the browser requests a new image, the old one on the screen being cleared;

The solution implemented by us consists of the following:

- a) the browser requests several images from the server;
- b) the web server will not serve the images immediately, it will acquire a image from the video camera and will serve only the first request, the other requests being maintained on hold, in order to be served at regular intervals according to the algorithm in the server;
- c) the browser loads the image in background; as soon as the image finishes loading it is displayed on the interface, replacing the old one. With every image completely loaded in browser a

new request is sent to the server; this way, the browser maintains a fixed number of connections opened to the server.

Each new request for an image contains a sequence number that is used by the server as an acknowledge signal, that a previous image was fully loaded into the browser, allowing the server to calculate the time it took the image to be transmitted from the server machine to the client. Knowing this time allows the server to calculate the moment when a new image should be sent to the client. The server can send a new image to the client on any moment because it has some requests kept on hold and can serve a new image on one of these connections. The algorithm of the server tries to send as much images as possible to the server, using the bandwidth at its maximum. Also, because the trip time of each image is computed, the server can adapt to the changes in network traffic. Thus, if the network becomes congested each image sent by the server will take a longer time to reach destination, and the server will decrease the rate of sending the image, avoiding further congestion of the network; if the network traffic comes to normal or in the case of a high bandwidth connection the web server will increase the rate of sending the images until reaches the full capacity of the link.

The advantages of using this protocol for sending images are:

- loading the images in background and displaying them only when are fully loaded eliminates the progressive loading effect and assures that on the screen there is always displayed a fully loaded image;
- the number of images per second received by the client is significantly increased (Fig. 6);
- adaptation to the speed of the connection between server and client and also to the unexpected variations in network traffic.

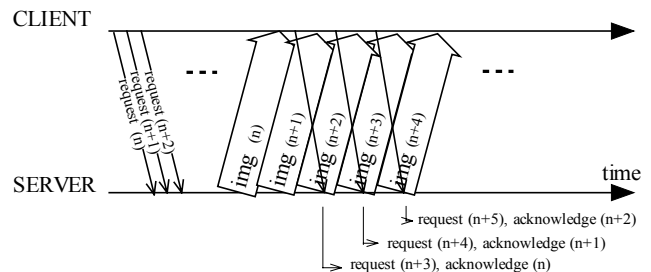


Fig. 6

4 Conclusion

This project is currently being used at University of Craiova (in the PLC class). Such remote laboratory experiment method enables student to use laboratory

equipment, which is not usually available to the students.

The application has been tested at University of Galati where a computer has been connected to Internet and students have used to program the PLC in order to control our manipulator which it was in the University of Craiova, a hundreds kilometers of Galati. Students can see the manipulator by our software and chatting with the PLC supervisor.

For the majority of remote experiments via Internet in robotics the student can tele-operate a robot or he can set some parameters on the web and then he can see the robot movements.

We performed more, namely the student can write a program, which can run on the PLC in order to move the robot how he wants.

The project touches on the important problems for didactic implementations. The application is provided with friendly GUI. The exercises help students to understand and compare different control methods of a manipulator using a PLC. Students are satisfied and their results are very good.

References:

- [1] S. Amin, M. Zakaria, N. Majid, L. Siong, L. Horvath, J. Tar, "Internet-based telerobotics: UTM's experince and future direction", *The 10th International Conference on Advanced Robotics*, Budapest, 2001.
- [2] G. Belmonte Bermudez, M.A. Perez Sanchez, "Robots Tele-programming", *Ist Workshop on Robotics Education and Training*, RET 2001, Weingarten, Germany, 2001.
- [3] R. Marin, P. Sanz, "The UJI Telerobotic Training System", *Ist Workshop on Robotics Education and Training*, Weingarten, 2001.
- [4] M. Cox and J. Baruch, "Robotic telescopes: An interactive exhibit on the world-wide web", *Second International WWW Conference*, 1994
- [5] R. Simmons, "Where in the world is Xavier, the robot? Robotics and Machine Perception", Special Issue: *Networked Robotics*, 5, 1996.
- [6] K. Goldberg, M. Mascha, S. Gentner, J. Rossman, N. Rothenberg, C. Sutter and J. Wiegley, "Beyond the web: Excavating the real world via mosaic", *Second International WWW Conference*, 1994.
- [7] K. Schilling, H. Roth and R. Lieb, "Teleoperations of rovers – from Mars to education", *Proceedings IEEE International Symposium on Industrial Electronics*, 1, 1997, 257-262.
- [8] N. Stedman and S. Ditlea, "Thriving community is seeded by tele-gardening on the web", *New York Times* article, 2 Aug. 1996.
- [9] K. Taylor and B. Dalton, "Issues in Internet telerobotics", *FSR 97 International Conference on Field and Service Robotics*, 1997.
- [10] M. Stein, "Painting on the world wide web: The PumaPaint project", *IEEE/RSJ Int. Conference on Intelligent Robotic Systems*, 1998.
- [11] S. Grange, T. Fong and C. Baur, "Effective vehicle teleoperation on the world wide web", *IEEE International Conference on Robotics & Automation*, 2007-2012, 2000.
- [12] P. Backes, K. Tso, J. Norris, G. Tharp, J. Slostad, R. Bonitz and K. All, "Internet-based operations for the Mars polar lander mission", *IEEE International Conference on Robotics & Automation*, 2025-2032, 2000.
- [13] D. Gillette, C. Salesman., R. Longchamp, D. Bonvin, "Telepresence: An Opportunity to Develop Practical Experimentation in Automatic Control Education", *European Control Conference*, Brussels, 1997.
- [14] D. Popescu and K. Schilling, "TOM – A Remote Laboratory for Mobile Robots Education", *7-th International Conference on Intelligent Engineering Systems*, Assiut, Egipt, pp. 447-453, 2003.
- [15] D. Popescu, *Programmable logic controllers*, Ed. Sitech, Craiova, 2001
- [16] ***, *AP-MICRO PLC Documentation*.
- [17] D. Popescu, C. Chiculita, "PLC Tele-Programming", *The 3rd International Conference on Information Technology Based Higher Education and Training*, ITHET 2002, Budapest, Hungary, 2002.
- [18] M. Ivanescu, *Industrial robots*, Ed. Universitaria, Craiova, 1994.
- [19] B. Patrut, *Aplications in Delphi*, Ed. Teora, Bucharest, 1999.