

A GLOBAL APPROACH FOR THE REMOTE PROCESS SIMULATION AND CONTROL

J.L. Díez, M. Vallés, A.Valera

*Dpto. Ingeniería de Sistemas y Automática
Universidad Politécnica de Valencia
Camino de Vera 14, 46022 Valencia (Spain)
E-mail: jldiez@isa.upv.es, mvalles@isa.upv.es, giuprog@isa.upv.es*

Abstract: Systems Engineering and Control teaching has a problem in common with other technical studies when distance learning is required: laboratory lectures. It is related to the studied concepts in the area and the requirement of using specific software tools (i.e. Matlab) and process.

This paper presents a global approach for the remote process simulation and real control using Matlab Web Server (MWS) tool. This work allows the execution of Matlab developed applications from any remote computer having a Web browser. The server obtains configuration parameters from a Web page and runs M-file Matlab programs. Once the simulation or the real process control has finished, MWS returns the output results to the remote computer through a new Web Page.

Keywords: Distance-learning, Virtual Laboratory, Remote Laboratory, Digital Computer Applications, Computed Control, Control Education, Data Acquisition.

1. INTRODUCTION

For the time being, trend in education (Poindexter, 1998) and, particularly, in control engineering education (Paprock, 1998), is to incorporate Information Technology in learning process. It is due to the increasing amount of computers at home connected to the Internet and decreasing effective teaching hours at University in new undergraduate programmes. They both make necessary an optimisation of educational process.

The use of Internet resources leads to an increase in time and space flexibility of the educational process. Clearly, distance education takes more advantage

from new technologies than standard education, but the last can make use of information technologies (when properly used) as an effective support (Paprock, 1998).

Based on Internet, two different options can be found for the laboratories developing: virtual laboratories and remote laboratories. A virtual laboratory allows, for example, continuous access to a simulated process in a computer. Under this structure, a remote interaction between the student and the virtual process is possible.

Halfway between traditional and virtual laboratories are the remote laboratories that offer a real

experiment to remote users. Incorporating a webcam allows watching real system evolution in addition to variables of interest. Most equipment needed for setting up a virtual/remote laboratory is available in traditional laboratories and only an interface between local application and Web server is the additional part to be developed.

MWS is a very powerful tool that helps developing virtual laboratories (Valera, et al., 2001), (Vallés, et al., 2001). However, direct process control is not possible if this software is used alone. This paper shows how to implement remote simulation and real process control using Internet platforms. In order to do this, *MWS* is complemented with some software functions. The hardware configuration is based on PC and industrial data acquisition cards. The software functions allow the low level access from the Matlab environment.

In the final part of this paper a real example of industrial process control (a DC motor) is presented in order to show the hardware and software configuration for the proposed remote laboratory.

2. MATLAB WEB SERVER CONFIGURATION FOR VIRTUAL AND REMOTE INDUSTRIAL PROCESS CONTROL

2.1 Software Configuration

Matlab Web Server is an integrated software suite that enables Matlab programmers to create applications that use the capabilities of the World Wide Web to send data to Matlab for computation and to display the results in a Web browser.

A set of programs enables remote Web access to Matlab applications (The MathWorks, 1999):

- *matlabserver*: Manages the communication between the Web application and Matlab. Matlabserver is a multithreaded server TCP/IP server that runs Matlab programs (M-file) specified in HTML document. The maximum number of simultaneous simulations and port number is specified here.
- *matweb.exe*: Located at HTML server is a TCP/IP client of matlabserver that is uses the Common Gateway Interface (CGI) to extract data from HTML documents and transfer it to matlabserver.
- *matweb.m*: Calls the M-file that Web application wants to run.
- *matweb.conf*: A configuration file that matweb needs for connecting to matlabserver.

For an extensive explanation of main features of *MWS* and commands used in creation of basic applications, (Vallés, et al., 2001) is suggested. In this paper, the particular configuration for authors' equipment is explained and some remote laboratories examples are presented.

The process of creating *MWS* applications involves usually a combination of Matlab M-files, HTML code and graphics, so that creating these applications consists of generating the following files:

- An HTML input document for data submission to Matlab.
- A Matlab M-file to process input data and compute results.
- An HTML output document for display of Matlab's computations.
- A test file to validate code before distributing the application over the Web.

However, *MWS* does not allow on-line access to the low level hardware. Therefore, remote laboratories and remote industrial process control is not possible under *MWS* software provided by Mathworks.

In order to make possible remote laboratories and remote control of real systems using *MWS*, an additional configuration with several data acquisition cards have been used and different routines have been programmed to access these cards from the Matlab environment and the web server. Above-mentioned file structure is shown in Figure 1.

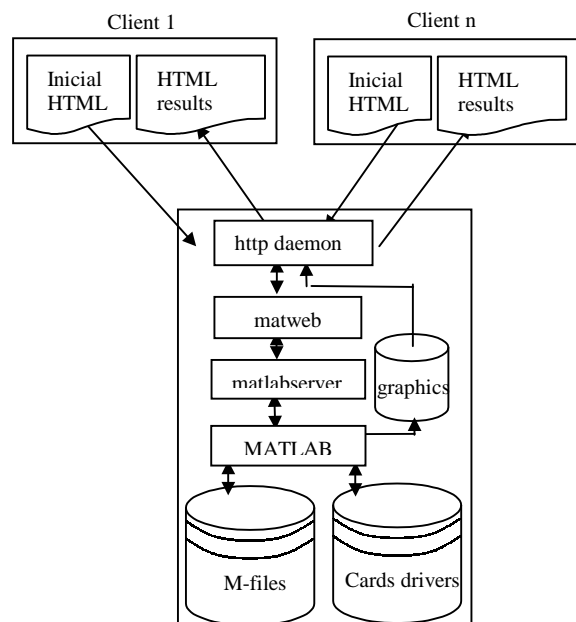


Fig. 1. Remote laboratory file structure.

The routines that have been programmed to access these cards from the Matlab environment allows the A/D conversion, the D/A conversion, Encoders

signals management, Digital inputs/outputs accessing and timer programming. All these functions have been programmed using C++ language in order to access to these cards directly in the “.m” files of the Matlab Web Server applications (available functions from Matlab toolboxes does not fulfil the application requirements). The Matlab executable functions are obtained as DLL files with the *mex* command, using Visual C++ v6.0 compiler.

```
>> mex read_812.cpp
```

2.2 Hardware Configuration

System configuration (Figure 2) consists of a Web browser running in client workstations, while Matlab, MWS, and the Web server daemon (httpd) are running in another machine. Additionally, several data acquisition cards (DACs) have been used. An *Advantech™ PCL-812* industrial card has been used in order to obtain the analog outputs; another one, *PCL-726*, has been used for supplying the control actions; and the *PCL-833*, have been used for reading encoders signals. Finally, a real process has been connected to DACs. DACs and real processes are not necessary if a virtual control is performed.

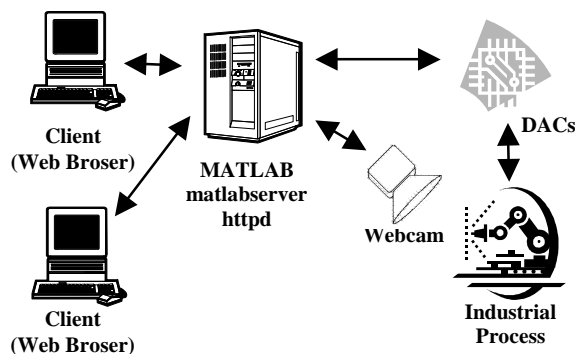


Fig. 2. Remote laboratory hardware configuration.

3. CREATION OF VIRTUAL APPLICATIONS

As stated in section 3.1, three files are needed for executing Matlab applications through the Internet. First, an HTML file for data submission to Matlab is needed. This file have to specify all M-file variables and a *submit* input for submitting data to the program. Text variable names used at web pages must coincide with Matlab file variables.

M-file for processing data input and calculate results must be implemented at computer server. The M-file is a normal Matlab function that accepts input from HTML input document and return results to a HTML output document. Different options exist for showing results and choice must depend on the kind of result.

- **Scalar variables display**

The simplest way to show output results is to display a scalar or character string. In order to do so, variable name (between \$ signs) have to be written in the HTML output file.

- **Matrix display**

If Matlab function output is more complex, results can be shown in a matrix. A table, based on information included in a Matlab variable, can be dynamically generated for that purpose by means of HTML command *table* where *autogenerate* option corresponds to Matlab variable to display.

- **Graphic display**

Finally, graphics can be generated when much information must be displayed. *MWS* generates *jpeg* graphics from Matlab. An graphic identifier (*mlid*) must be defined and used subsequently for file name creation.

4. EXAMPLES OF VIRTUAL LABORATORIES

Some examples have been developed with *MWS* and a set of appropriately designed web pages, in order to test tool capabilities and the easy access of the whole simulation framework via the Internet. Nowadays, available virtual laboratories are related to “IIR Digital Filter Design” and “Controllers Design”, both of them quite common in basic control courses [2]. In this way, any student following a subject in which concepts related to developed web pages are taught, can perform proposed virtual laboratory work. The only one requirement is to have a PC connected to the Internet and a Web browser, and no Matlab version is required on the student computer.

Developed laboratory structure is as follows: an “*index.html*” web main page has hyperlinks to two new main pages, each one corresponding to one of the previously mentioned examples: controller and filter design. Main page for controller design is “*pract1.html*” and “*pract2.html*” is the same for filter design.

4.1 IIR Filter Design Laboratory.

Once IIR filter design option is chosen, different possibilities exist:

- signal generation
- filter design, and
- filter behaviour test

Signal generation option is in “*gen_sen1.html*” file and it allows, by submitting to Matlab sinusoidal signals parameters (amplitude, frequency, phase), generation of a complex signal by addition of

sinusoidal components. Indirectly, the user has access to *sen* command.

After parameters are introduced at text entry boxes, *Enviar* (submit) button must be pressed in order to make data arrive at Matlab applications server, where a ".m" file will process data and results will be sent to client computer. In this example, a time and frequency analysis of generated signal is done (by means of *plot* and *fft* commands) and a plot including both is returned. Figures 3 and 4 show input data web page and obtained result, respectively.

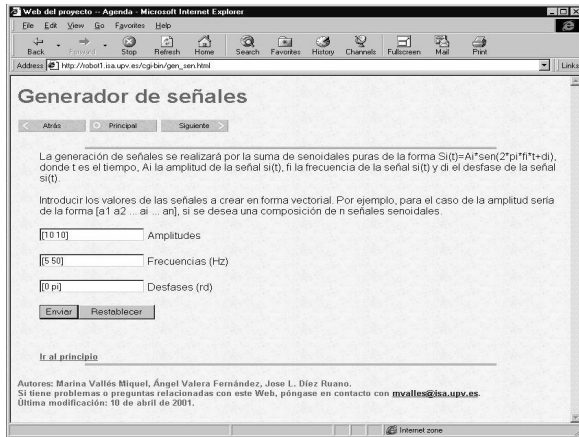


Fig. 3: Input data web page.

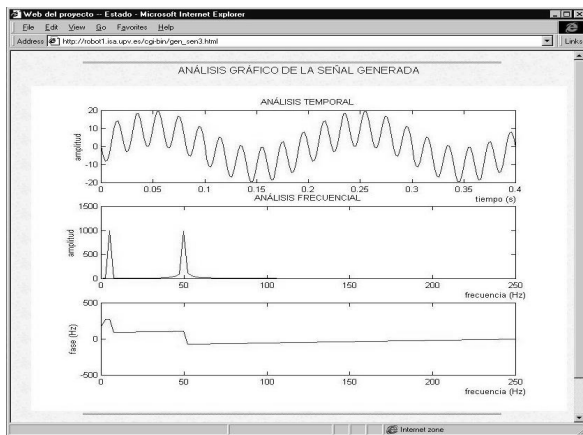


Fig. 4: An example of analysis of generated signal.

Filter design option is managed by "*dis_fill.html*" file, who submits Matlab bandwidth of the signal that filter will process (in order to choose an appropriate sampling frequency) and design filter parameters: frequency passband edge ripple, frequency stopband edge ripple, passband ripple, stopband ripple, filter type (Butterworth, Chebyshev I, Chebyshev II or Elliptic) and main filter characteristic (lowpass, highpass, bandpass, bandstop).

Once parameters are introduced at the menus and text entry boxes, submit (*Enviar*) button must be pressed and a plot of frequency analysis of designed filter (answer of *freqz* Matlab command) is shown.

Third (and last) option referring filter design is in "*apl_fill.html*" file and can be used for testing designed filters. In this case, all parameters explained in the two previous options (signal generation and filter design) must be submitted to Matlab.

Server output is a plot of time and frequency analysis of original (unfiltered) and filtered signals, where behaviour of designed filter can be easily checked.

4.2 Controller Design Laboratory

From "*index.html*" global main page, second laboratory main page ("*pract1.html*") can be easily accessed, where different buttons for three different exercises are shown. Available exercises are in this case:

- open loop system analysis,
- closed loop system analysis, and
- controller design

Open loop system analysis is managed by "*bobert1.html*" file. This web page submits to Matlab "*bobert.m*" file (located at remote server) numerator and denominator coefficients written by the user at text entry boxes in the same way as explained in previous laboratory example. In this case, remote user has access to *dstep* Matlab command.

Closed loop system analysis exercise works in the same way, being its main page "*btanecat1.html*" file. In this case, the web page submits numerator and denominator coefficients of the system to be analysed to Matlab "*btanecat.m*" remote file. In this case, remote user has access to *rlocus* Matlab command. The exercise output (remote server output) is a root locus plot, which will help user in controller design tasks.

Finally, controller design exercise is in "*control.html*" file. This web page submits to Matlab "*control.m*" file (located at remote server) numerator and denominator coefficients of the system and the controller (the user have to design this controller) to be tested in a unit feedback configuration. In this way, the user has remote access to *conv* and *dstep* Matlab commands. Server output is a plot where closed loop system step answer is shown. Input to the system (step reference) is also plotted, in order to allow easy comparison between signals and control system performance.

5. REMOTE INDUSTRIAL PROCESS CONTROL BASED ON MATLAB WEB SERVER

Two different operating options are available in remote labs: batch and on-line. On-line operation

considers control algorithm at remote computer, being control actions and sensors information transmitted through the Internet. Reference and parameters can be changed while experiment is being carried out, but variable delays appear as a consequence of Internet traffic. Batch operation avoids Internet delays because reference and controller parameters are sent to server before experiment starts and, once finished, process output is sent to remote computer.

A remote laboratory in batch operating mode has been done as an example and it has been developed using *MWS* with additional features explained in section 2 and a set of appropriately designed web pages. In this way, any student having a PC connected to the Internet and a Web browser can control an industrial process, and no Matlab version is required on the student computer. Additionally, a webcam in order to allow watching real system evolution has been included using Windows Media Technologies 7 free software (Microsoft Corp, 2000).

A hyperlink to broadcasting has been added to experiment main page (see figure 7) and, when visualisation of experiment is chosen, Windows Media Player is open (figure 5) and real time (with some delay) images are visualised by users.



Fig. 5. Remote Process visualisation using Webcam.

5.1 Example: Remote DC-Motor Control

In this example, a DC motor that has been used as can be seen at the Figure 7, but any other physical process can be used if the signals (control actions and process outputs) are available.

Figure 6 shows the laboratory setup used for this experiment.

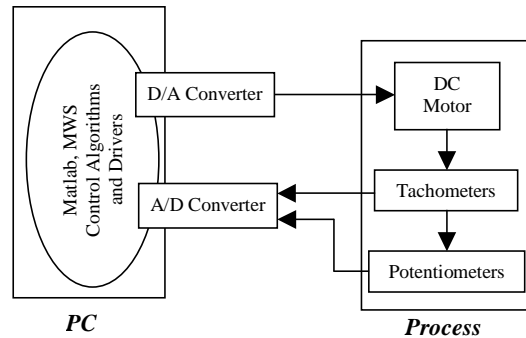


Fig. 6: Remote Laboratory Setup

Technically this control architecture works as follows: The potentiometers and the tachometers give the DC motor position and the velocity. These signals are introduced in the computer server using the A/D converters. Then, the computer server calculates and applies (using the D/A converters) the control action. In order to control more complex systems, other card drivers have been programmed, like the encoders signals and the digital inputs/outputs. All these cards and drivers are used to control an industrial gantry robot.

For the implementation of this remote laboratory, an HTML file for data submission to Matlab is needed. This file have to specify all M-file variables and a *submit* input for submitting data to the program. In this case, initial Web page is “*control.html*” file and it submits to Matlab all experiment parameters (reference values, controller parameters and experiment time). Figure 7 shows the input Web page.

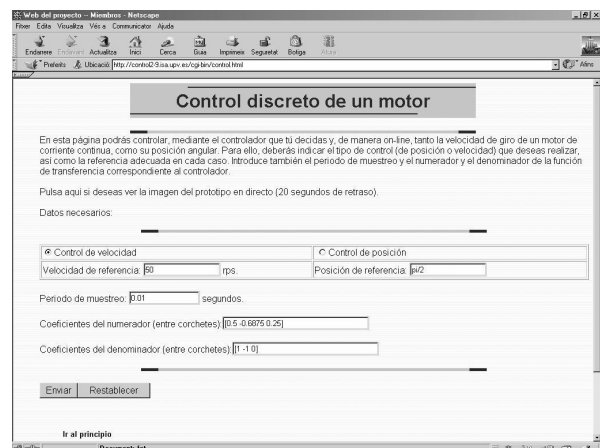


Fig. 7: Input data page for “DC-Motor Velocity Control”.

After *Enviar* (submit) button is pressed and the data arrives to the server, Matlab starts with the execution of the M-file associated function (*control.m* for the motor velocity control). This function is a normal M-file Matlab function that accepts input data from HTML input document, starts the control experiment and return calculated results to a HTML output

document. With this file, results will be sent to client computer and shown in "control2.html" file. In this example, a time response of DC-motor to chosen reference is returned in a graphic, as is shown in Figure 8.

Another remote control implemented with this DC motor is the position control. In a similar way, the user must specify the reference position and the control parameters and the results are showed in a graphic also.

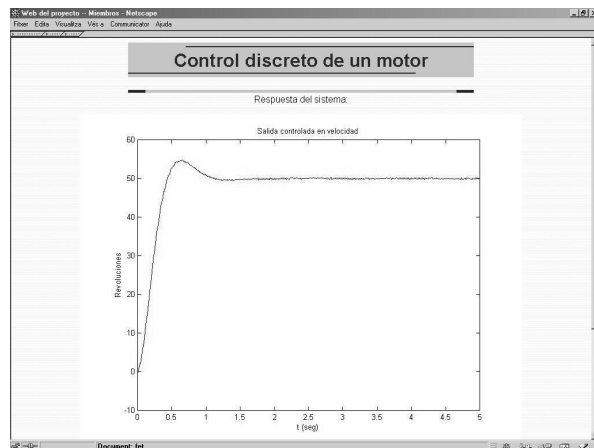


Fig. 8: Output of "DC-Motor Velocity Control".

6. CONCLUSIONS

Matlab Web Server is an interesting tool for making accessible Matlab applications from any remote computer with a Web browser. In order to do that, the following files must be generated: a HTML input document for data submission to Matlab, a Matlab M-file to process input data and compute results and a HTML output document for display (text or graphics) Matlab's computations.

If remote industrial process control is required, an additional configuration with several data acquisition cards and different routines to access these cards from Matlab environment are needed. A practical example for a simple industrial process control has been presented and the tool behaviour was simple and fast.

Finally, as usual in a remote laboratory, a webcam has been incorporated in order to allow watching real system evolution in addition to the graphical information returned by the web application.

Acknowledgements

The authors wish to acknowledge the support provided by "Vicerrectorado de Investigación, Desarrollo e Innovación" from the Universidad Politécnica de Valencia for partial funding of this

work, under research project PPI-6-00-20010031 (Proyecto de I+D para Grupos Emergentes).

REFERENCES

- Johansson M., Gäfvert M., Aström K. J. (1998) "Interactive Tools for Education in Automatic Control", *IEEE Control Systems Magazine*, Vol. 18, pp. 33-40.
- Overstreet J. W., Tzes A. (1999). "An Internet-Based Real-Time Control Engineering Laboratory", *IEEE Control Systems Magazine*, Vol. 19, pp. 19-34.
- Paproch, Kenneth., (1998). *Distance Learning: The Ultimate Guide* London. Sage Publications.
- Poindexter S.E., B.S. Heck, (1999) "Using the Web in your Courses: What can you do? What should you do?", *IEEE Control Systems Magazine*, Vol. 19.
- Microsoft Corp. (2000). "Microsoft Windows Media™ JumpStart CD".
- The MathWorks Inc. (1999). "Matlab Web Server user's guide".
- Valera, Á., Vallés, M., Tornero, J., (2001) "LabConRob: Virtual Laboratory of Real-Time Robot Control", *Robotica*. **43**. ISBN 152701/00. SIN 0874-9019. Pp 40-45
- Vallés, M., Valera, Á., Díez J.L., (2001). "Setting up a Virtual Matlab Control Laboratory.", *Education in Automatic Control EDCOM 2001*.