REMOTE LAB FOR CONTROL APPLICATIONS USING MATLAB

Rafael Puerto Manchón
Oscar Reinoso García
Ramón P. Ñeco García
Nicolás García Aracil
Luis Miguel Jiménez García

email: r.puerto@umh.es
Department of Systems Engineering and Control
Miguel Hernández University
Av. Ferrocarril s/n (Ed. Torreblanca)
03202 – Elche (Alicante)
SPAIN

Abstract: In this document we present the general architecture of an application for the remote execution, via Internet and in real time, of physical process controller. This application has developed using the platform Matlab/Simulink. The motivation of this work is based on the little availability of real physical systems or laboratories to perform the experiments. We also present some examples that show the validity and applicability of the presented architecture.

Keywords: Controlled Systems, Real – Time, Remote Control, Internet

1. INTRODUCTION

The work presented in this paper pretends to provide a general architecture for remote execution in real time of physical processes via Matlab[1]. The main motivation is, on one hand, the lack of lab scale models or real physical systems, and for other, the little availability of schedules in the laboratories where the students of automatics and process control subjects develop the practical assignments corresponding with the theoretical contents. Therefore, the present application is focused on the real-time remote control of a physical system through Internet and Matlab as the development platform. This application will allow the students to simulate the operation of a controller for a certain physical process using Internet, and the control in real time, of the physical process in question, using the designed controller. In this way, it is allowed the student to develop this activities without being in the laboratory where the real system is installed. Also, the application returns the user all the information relating the performed execution, besides graphic and other data of interest.

We have chosen the platform Matlab / Simulink (with some additional toolboxes) for the development of this application for several reasons: first, Matlab, Simulink and the necessary toolboxes constitute a reliable, well-known platform and with a lot of technical support. Second, the time used to obtain the prototype and in the development is quite inferior to the time needed with other tools and platforms (direct programming in a programming language, etc.). Third, this platform porvides tools for remote execution of programs, and for real-time execution on a physical system, through a data system acquisition, using a specific control algorithm. The last reason, and not less important, is the great quantity of researchers that use this platform as development tool for simulations and real applications.

[1] Matlab and Simulink are trademarks of The Mathworks Company. The authors wish to thank Mathworks Spain for their help in the development of the present work.

2. GENERAL ARCHITECTURE SCHEME

The general scheme of the application architecture is shown in Figure 1.

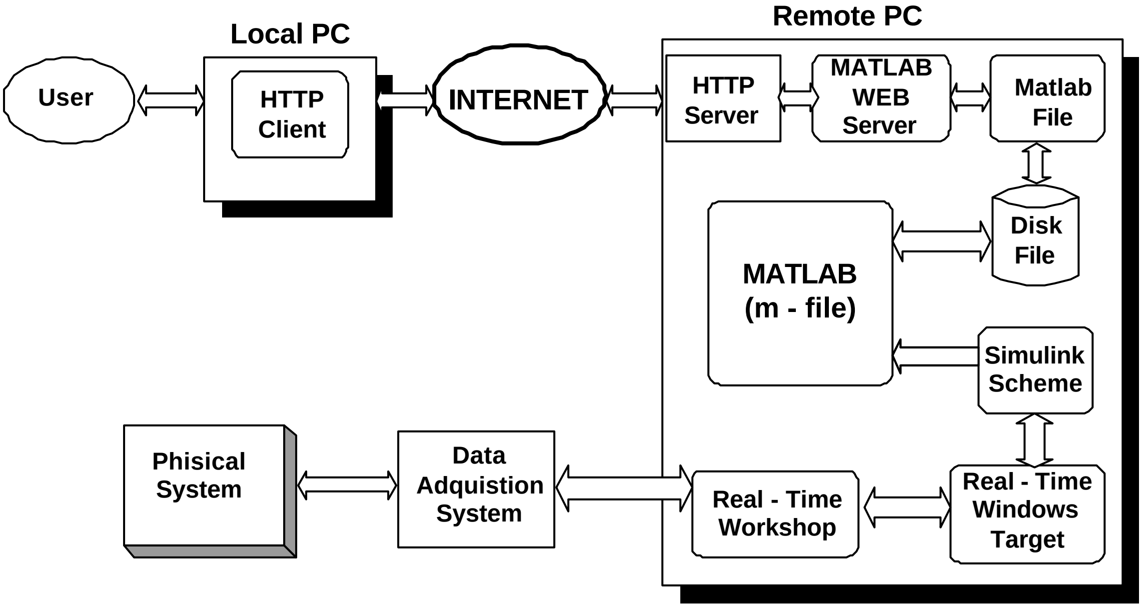


Figure 1.General architecture scheme

As can be seen in Figure 1, the hardware and software elements in the local and remote area are the following:

- Local area:
  - Computer
  - http client
  - Connection to Internet

- Remote area:
  - Computer
  - Data acquisition system
  - Physical system to control
  - Operating system that allows to stablish access security directives
  - Http Server
  - MATLAB R12
  - SIMULINK V. 4.1
  - Matlab Web Server V. 1.2.1
  - Real–Time Windows Target Toolbox V. 2.1
  - Real–Time Workshop Toolbox V. 4.1
  - Connection to Internet

The main problem that is presented to perform the real-time control using Matlab (with the Real-Time Workshop and Real-Time Windows Target toolboxes) and to present the results using Matlab Web Server is that an efficient communication between both mechanisms doesn't exist. Therefore, the data needed to carry out the real-time control (controller data, etc.) and the results obtained once the experiment has been carried out are read by means of files stored in disk. In the following section we comment how is performed this communication.

Next, the operations that are developed in the local and remote area are described, leaving for the following epigraph the description and operation of each one of the mentioned components.

OPERATIONS PERFORMED IN THE LOCAL AREA

1. **Connection by means of Internet and a http client to the remote machine.** When

this connection is established, the client presents an introductory web page where the system to control is presented in detail as well as the necessary guidelines for interacting with the system.

2. **Choice of the control scheme:** choice of the desired control configuration to carry out on the physical system (PID controller, state feedback, etc.).
3. **Controller data input:** the controller data input is carried out by means of a form that specifies the values of the coefficients of the polynomials that conform the numerator and denominator of its transfer function or the state feedback gain matrix.
4. **Execution request sending and results waiting.**

OPERATIONS PERFORMED IN THE REMOTE AREA

The operations in the remote area are performed by two different applications or entities: Matlab Web Server and Matlab and the toolboxes for the real-time execution.

a) **Operations performed using Matlab Web Server:**

1.- Reception of the data that appear in the form (transfer function of the controller, controller type, etc.). The http server receives the data of the controller and transfer them to Matlab Web Server. Matlab Web Server transfer this data to a Matlab program (executed by Matlab Web Server).

2.- Execution of a Matlab program by Matlab Web Server: this Matlab program will carry out the following tasks:

- To check the validity of the controller: the correct operation of the controller is verified. For this purpose it is analyzed the poles of the transfer function in closed-loop for this controller and an error message is generated in case these lead to a not desired behavior.
- To generate in disk a file (.mat) that contains all the necessary data to carry out the system control on the real physical process (controller type, controller data, etc.)
- To wait that the results files of the execution are available in disk, and once they are available, the program reads its content and deletes the file.
- To generate the web page with the results presentation and to send it to the user

b) **Operations performed by Matlab:** Matlab is continually executing a program that carries out the following tasks:

- To remain in an infinite loop awaiting that the file that contains the information to carry out the real-time execution is available in the disk
- Once the file is available, the program reads the data from this file and perform the control of the system in real time.
- Concluded the real-time execution, the program generates a file in disk with the result of the execution and other data of interest (duration, dates, hour, etc.)
- The program deletes the file generated by Matlab Web Server and starts again.

3. FUNCTIONAL DESCRIPTION OF THE ARCHITECTURE

In this section we describe the operation and function of each block that appears in Figure 1. Only those that appear in the remote area will be analyzed, because the blocks in the local have less interest.

1. http server: This server allows the communication of the computers using the http protocol. Also, it has been configured previously so that the data processing can be made by Matlab Web Server.
2. Matlab Web Server: this toolbox allows to use the mathematical and graphic capacities of Matlab and to present the results in a web page. Matlab Web Server generates the data input pages and the presentation of results. At the same time, it will execute the Matlab program that carries out the tasks mentioned in the previous section.
3. Matlab: executes the program that makes possible the real-time control of the system and the generation of the results in a file.
4. Simulink scheme: we chose to design a Simulink scheme (whose execution will be carried out in real time using the Real–Time Windows Target and Real–Time Workshop), for the real-time execution of the system controller. Real–Time Windows Target provides the interaction blocks with the data acquisition system, and the execution support in real time is provided by the Real time–Time Workshop.
5. Real–Time Windows Target: this toolbox allows to execute Simulink schemes in real time. For this purpose, it provides the necessary blocks for the interaction with the data acquisition system, at the same time that

it allows to use the Real–Time Workshop from Simulink.

6. Real–Time Workshop: this toolbox generates the C code that, once compiled, will be executed in real time. For space reasons, we don’t describe the details of the operation of the Real-Time Windows Target, which is much more complex than the short description presented in this paper.

As can be appreciated, the critical point in the presented procedure is the communication and synchronization among Matlab Web Server (that reads the necessary information for the execution and presentation of the results) and Matlab (that carries out the real-time execution and generates the results). The mechanism of operation of both programs (the one executed by Matlab Web Server and and the one executed by Matlab) is presented in the following programs written in pseudocode:

Program executed by Matlab Web Server

```
Read data from user
Testing the controller
If (Controller OK) Then
   Generate data file
   While not exist results file
      %Loop
   end
   Load results file
   Delete results file
   Generate results from user

Else
   Generate error message from user
```

Program executed by Matlab

```
While True % Infinite Loop
  While not exist data file
      %Loop
  End
  Load data file
  %Real time execution
  Connect to Real – time Kernel
  Start Simulink Model execution
     %Real Time Execution
  Stop Simulink Model execution
  Disconnect from Real – Time Kernel
  Save results file
  Delete data file
end
```

The arrows show the flow in the execution of the programs. As can be observed, the execution in real time of the simulink model is performed by means of command line functions.

4. EXAMPLE

The example that we present in this paper allows to execute a controller of a pneumatic system. This system consist of a pneumatic double-effect cylinder. This pneumatic cylinder is commanded by a proportional valve of 3 byas, which is fed to a pressure of 6 bars. As the position sensor an incremental encoder is used. In Figure 2, the designed platform of a pneumatic cylinder is presented. As you can be seen, the system is made up of a pneumatic cylinder, a position sensor (encoder), and a support system in its top that allows to modify the load supported by the cylinder. An scheme that allows to detail the operation of the system is presented in Figure 3.

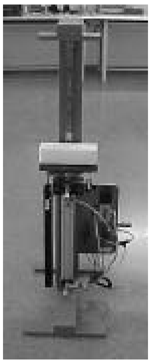


Figure 2: Servo Pneumatic Platform.

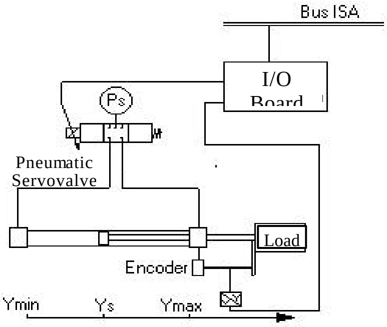


Figure 3: Pneumatic system scheme .

The used components that integrate the system appear in Table 1.

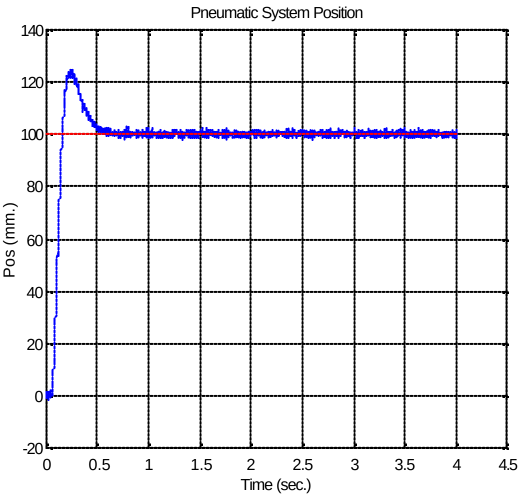
System	*Values*	*Manufacturer*
Servo-pneumatic Cylinder	32 mm ∅ chamber 12 mm ∅ Rod 800 mm Stroke 10bar Max. Pressure	Boshc 015
Pneumatic Servovalve	24v 0-10v Command 10bar Max.	Festo MPYE-5-1/8-HF
Position transducer	5v 9.75 pulses/mm	Unimeasure LX-EP-40

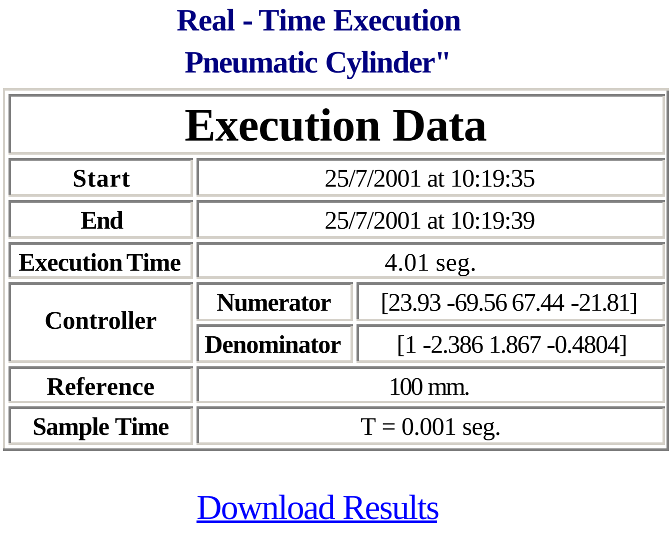
Table 1

In order to allow the user to carry out previous simulations before choosing the appropriate controller and scheme control, we performed a system identification. Although the system is strongly not lineal due to its inherent characteristics, a model that represents under some appropriate margins the behavior of the system is:

$$G(s) = \frac{3.583s^2 - 730.4s + 5.07e04}{s^3 + 98.04s^2 + 1873s - 1587}$$

It is necessary to note that the system presents a high dynamic that it is very different between the extreme of the cylinder and the centre of the camera. If the gains of the controller are not adjusted according with a table that takes into account the different behaviour of the system instability effects are originated. A part of the results obtained for the discrete controller whose transfer function if given by the fields “Numerator” and “Denominator” of the table “Execution Data” is presented in figure 4.





Real - Time Execution Pneumatic Cylinder"

Execution Data		
Start	25/7/2001 at 10:19:35	
End	25/7/2001 at 10:19:39	
Execution Time	4.01 seg.	
Controller	**Numerator**	[23.93 -69.56 67.44 -21.81]
	Denominator	[1 -2.386 1.867 -0.4804]
Reference	100 mm.	
Sample Time	T = 0.001 seg.	

Download Results

Figure 4. Results example

5. CONCLUSIONS

In this article we present an architecture that is able to perform real-time executions and to return the results to the user through Internet. For this purpose we use the Matlab platform with the necessary toolboxes. The use of this platform is justified, on one hand, for the fact of being broadly used in control engineering, and for other, for the great simplicity, speed and reliability with which it is allowed to develop applications. Lastly a real example has been presented that shows the validity of the presented architecture.

6. REFERENCES

Brun X., Belgharbi M.,Sesmat S.,Thomasset D., Scavarda S., (1999): ”Control of an electropneumatic actuator, comparison between some linear and nonlinear control laws", Journal of Systems and Control Engineering.

Burrows C.R., (1972): "Fluid Power Servomechanisms", Van Nostrand Reinhold Co., London, England, pp 132-155.

Ming-Chang Shih & Shy-I Tseng, (1995): "Identification and Position Control of a Servo Penumatic Cylinder", Control Engineering Practice Vol. 3, No. 9, pp 1285-1290.

Rubio A.E., Reinoso O., Saltarén R., Pérez C., Jiménez L.M., (2001): "Identificación experimental de un cilindro neumático", Simposio Internacional de Ingeniería Eléctrica SIE'2001 Santa Clara. Cuba

Schmid, Chr (1999): “Virtual Laboratory for Engineering Education”. Proc. 19th World Conference on Open Learning and Distance Education, ICDE'99, Wien, Paper u2a1.

Schmid, Chr.(2000): “Virtual Control Laboratories and Remote Experimentation in Control Engineering”. Proc. 11th EAEEIE Annual Conference on Innovations in Education for Electrical and Information Engineering, University of Ulm, Ulm, S.213-218

The Mathworks Inc. (2000): Matlab R12, Simulink, Real – Time Workshop, Real – Time Windows Target and Matlab Web Server. Online Manuals.

Wang J., Pu J. Moore P., (1999): "A practical control strategy for servo-pneumatic actuator systems", Control Engineering Practice 7 pp. 1483-1488.