DISTANCE TRAINING IN ROBOTIC APPLICATIONS THANKS TO A PROPOSAL OF REMOTE LABORATORY

Electrical Engineering

2008

Herminio Martínez[1], Juan Gámiz[1], Joan Domingo[1], Antoni Grau[2]

Abstract — *This paper describes the development of a remote laboratory in the area of automatic and robotic engineering. The laboratory is based on an educational platform around a robotic arm that allows emulating a practical laboratory in which engineering students can corroborate their theoretical results comparing them with the real paths traced by the robot. Thus, the research work presented in the article is directed towards the consolidation of concepts in the field of robotics and control thanks to the use of the aforementioned remote laboratory.*

Index Terms — *Distance training, remote laboratory, robotics, automatic systems.*

INTRODUCTION

A considerable number of applications based on distance–learning platforms have been developed during the last years [1]–[2]. They have demonstrated the incredible potential in order to improve the training in the fields of automatic and robotic engineering.

In the presented platform, the aim of this application is the design and programming of a remote control application for an industrial manipulator acceding through Internet. For this purpose, the authors have started from an educational application that emulates a practical laboratory in which engineering students can confirm the theoretical results comparing them with the real paths traced by the robotic arm.

By this way, the student connects, through his Internet browser, to the remote laboratory's Web Site hosted at the University's Apache server. The user is identified by filling in the login and password gaps and the theoretic results are sent to the server's MySQL database. This process makes the application enters in a functioning routine in order to analyze the information received and, in case of being valid, the information is sent to a I/O 8255 card which links the computer server with the control board that takes control of the robot's articulation motor.

On the one hand, students can see on–line at any time the result of the practical experience through the laboratory's IP camera and consulting the record files which are generated.

On the other hand, professors can accede to the remote laboratory's Web Site as an administrator which allows him to work with student's database, generated previously, in order to search the previously obtained results by a student, to insert a new student, to erase previous results, etc. A complementary application has been programmed to load all student's personal information from a plain tabulated text file (*.txt) exported from Microsoft Excel to MySQL database.

ANALYSIS OF THE ROBOTIC SYSTEM

It is important to know the capacities and limitations of the elements involved in the system in order to determine the best solutions. The application has been based in an industrial manipulator, in particular, the NND from Nakanippon Electric Ltd., model which consists in five rotational articulations (Figure 1).

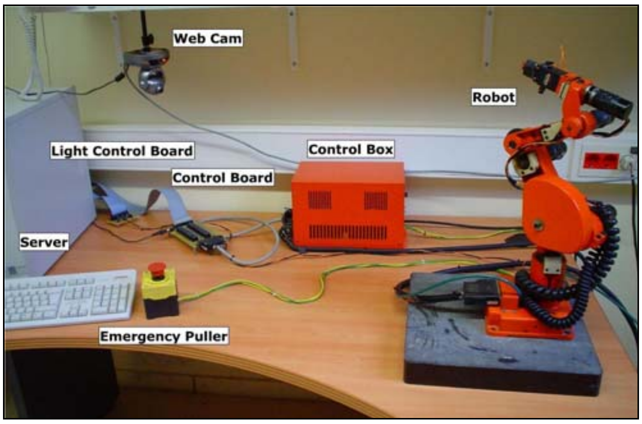


FIGURE. 1

GENERAL ASPECT OF THE APPLICATION ASSEMBLY. WE CAN SEE THE ROBOTIC ARM ON THE RIGHT

The control box allows taking control of the mechanical arm from a personal computer connected with a standard RS–232 cable using protocols or using the manual console. To communicate the computer with the servos and actuators without the manipulator's protocols, a board has been designed and implemented following the electrical scheme

[1] Herminio Martínez, Juan Gámiz y Joan Domingo, Escuela Universitaria de Ingeniería Técnica Industrial de Barcelona (EUETIB). Universidad Politécnica de Cataluña (UPC). C/ Comte d'Urgell, 187. 08036. Barcelona, Spain, {herminio.martinez, juan.gamiz, joan.domingo}@upc.edu

[2] Antoni Grau, Departamento de Ingeniería de Sistemas, Automática e Informática Industrial (ESAII). Universidad Politécnica de Cataluña (UPC). C/ Pau Gargallo, nº 5. Edifico U. 08028. Barcelona, Spain, antoni.grau@upc.edu

of the manual console replacing the pullers by electrical relays. In this way, we can alter the status of the relays from a personal computer through the I/O 8255 card which communicates with the manipulator's servos.

Controlling the manipulator's engines by relays involves working with time variables instead of the angular ones. Analyzing each articulation, direction and speed to deduce, using mathematical interpolated calculations, the equations, allows converting the angles introduced to time variables for the control of the relay status obtaining the angle value needed. In order to perform the analysis we have designed an optocoupler circuit with an encoder, designed with 15º between their holes (Figure 2) carefully.

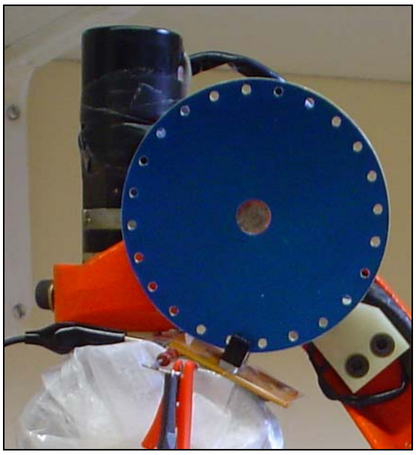


FIGURE. 2
DETAIL OF THE ENCODER

Using a digital oscilloscope connected to a personal computer, the values needed to determine the equations have been captured and the necessary values extracted. Using the toolbox named *polyfit* as others of the well–known software Matlab, the equations of the robotic arm have been obtained.

As an example, the graphics in Figure 3 show the values for the fourth articulation. This figure shows the time that the manipulator takes to cross the maximum distance in every direction (left and right) for two different speeds (high and low). By analyzing the results, we have determined to work only with 'high' speed, because is the most constant and there is not so many difference between the speeds in both directions.

On the other hand, is important to get the manipulator model by using the Denavit and Hartenber's method, analyzing the robotic arm structure to determine the cinematic and combined matrix of every articulation from which we can extract the orientation and position submatrix. The equations allow us to create an application in order to observe, using graphic simulations, the theoretical trajectories and determine the working manipulator volume (Figure 4).

The hardware elements that compose the application are, mainly, the robotic arm with its control box, the input/output 8255 card, a control board and another board designed in order to manage the operation status of the LED diodes incorporated in the system. As a safety measure, an emergency push–button, just for a local purpose, has been added. This element affects to the robot's power supply directly. Finally, the incorporation of an IP Webcam allows us to observe all the manipulator's trajectories.

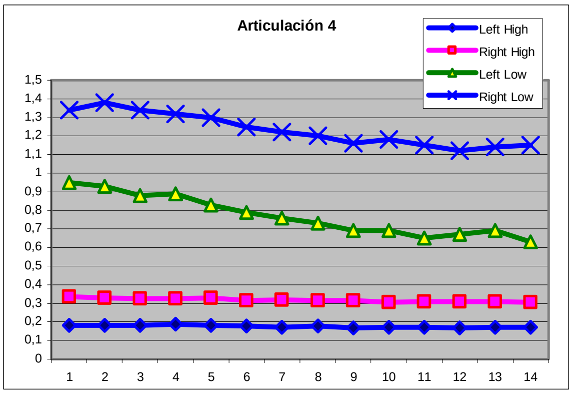


FIGURE. 3
TIME GRAPH OF ONE OF THE INDUSTRIAL MANIPULATOR'S ARTICULATION

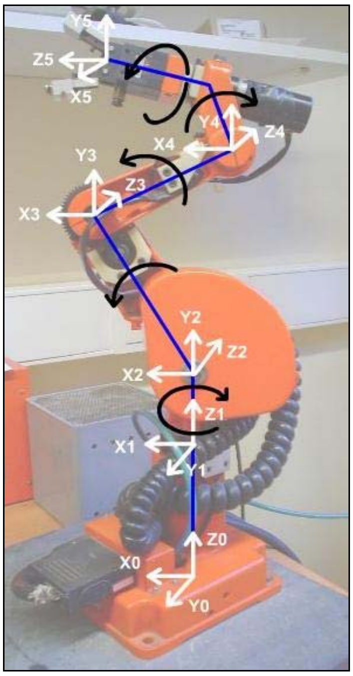


FIGURE. 4
STRUCTURE OF THE ROBOTIC ARM APPLYING THE DENNAVIT AND HARTENBER'S METHOD

As we said previously, authors have designed a control board that controls the operating status of the manipulator's servos by the activation of the relays, which are controlled by a personal computer. The board has two connectors: the first one for the servos, and the second one for the input/output 8255 card connected to the computer in order to send or acquire the signals from the controller board (Figure 5).

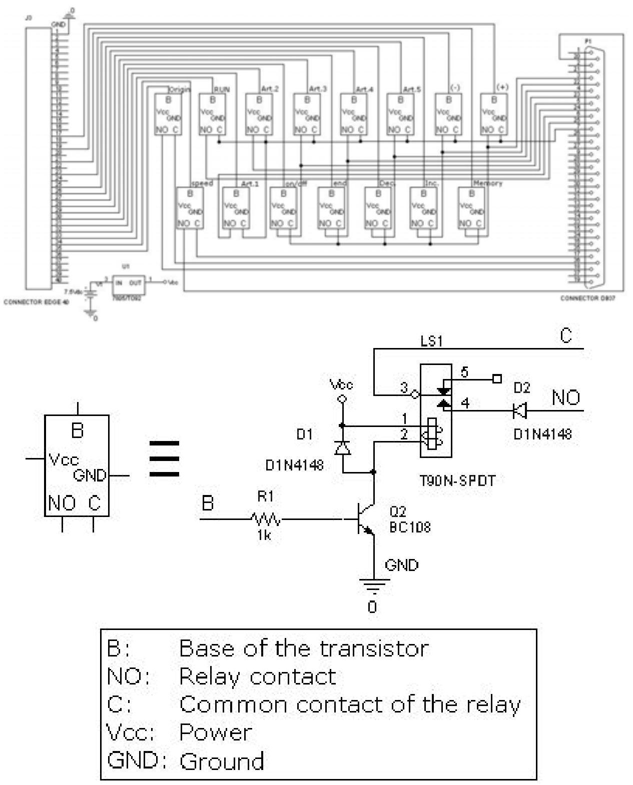


FIGURE. 5

CONTROLLER BOARD CONNECTED TO THE CONTROL BOX AND THE 8255 CARD CIRCUIT

The light control board has been designed to make two functions. On the one hand, two LED diodes (green and red) indicate to the Webcam's observer whether the robotic arm is available or whether it is in the middle of a process. On the other, the control of another light point located in the manipulator free end, makes possible to obtain easily the real trajectory by a video capturer and, in this way, to determinate the real position in the space. This trajectory can be contrasted with the theoretical data introduced by the student previously.

SOFTWARE STRUCTURE

The software that takes part in this application can be classified in two sections according to their operating scope: remote applications and local applications.

At the same time, the remote applications can be classified in two sections: the ones that work as a server (Apache to host the Web Site, and MySQL for database) and the ones that allow interacting with this server (Web navigators).

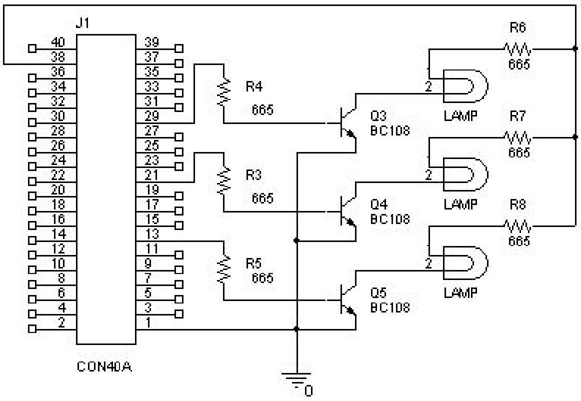


FIGURE. 6

LED CONTROL BOARD CONNECTED TO THE CONTROL BOX AND THE 8255 CARD

Apache Web Server

A Web server is an application which implements the HTTP (HyperText Transfer Protocol) protocol in a computer, typically a personal computer. Using a Web navigator the client makes an HTTP request to the server. This one answers with the solicited content.

In the server–side application, the server executes the application and converts it in HTML code in order to send it to the client who does not need to have any additional capacity installed in his Web navigator. In this way, any client who has a basic Web navigator can use this sort of application.

The HTTP Apache server has been chosen because it is free–license software with open source, which allows us to adapt widely the application to our application, adding the PHP (Hypertext Pre-Processor) module to allow working with MySQL database. To run with Apache server in this application, we have created a Service called Apache 2.

PHP 5.0 Module

The PHP 5.0 module is designed to extend the Apache functionality in order to execute applications in PHP code. We have chosen PHP programming language because it is a free language that allows the creation of dynamic content and emphasize its connectivity with MySQL.

MySQL

Apart from MySQL server, there are three other applications which help in the MySQL database's management:

- **MySQL Server:** Contain the MySQL Server and a client application in console mode.
- **MySQL Administrator:** Allows supervising the Server process, to change the configuration options, to access to register information, to manage user's privileges and manage security copies.
- **MySQL Query Browser:** Visual client for MySQL.

Creation of User Registers in Database

We need to create two kinds of users, because each one has his/her own permissions and has to be configured according to their necessities and restrictions. These two profiles are *professor* and *student*.

Database Creation

Basically, we can distinguish between *schemes* (or database) and *table* (or data tables).

On the one hand, the *table* is where we can create the fields we want to work with. On the other, *schemes* are the groups where one or more tables are stored. Notice that, once a kind of user has been defined in the MySQL Administrator, the permissions that are given to this user are attached to the schemes (and not to the tables). In this way, if we want to create two kind of users (for instance, as it is the considered case, *student* and *professor*) and restrict their permissions in an effective and safe way, we must create each table in a different scheme. Next, Table I shows the 'schemes' and 'tables' we have created in this application.

TABLE I
ACCESS TO MYSQL ADMINISTRATOR

Permissions	***DB Homework explanation***	***DB answer***	***DB professor***
Student	All, except *Insert, Delete, Create*	All, except *Delete*	None
Professor	All	All	All

In order to avoid conflict with PHP code, we should not use numbers as a name of a field. The numbers are not interpreted as a part of the name when MySQL variables are called from PHP.

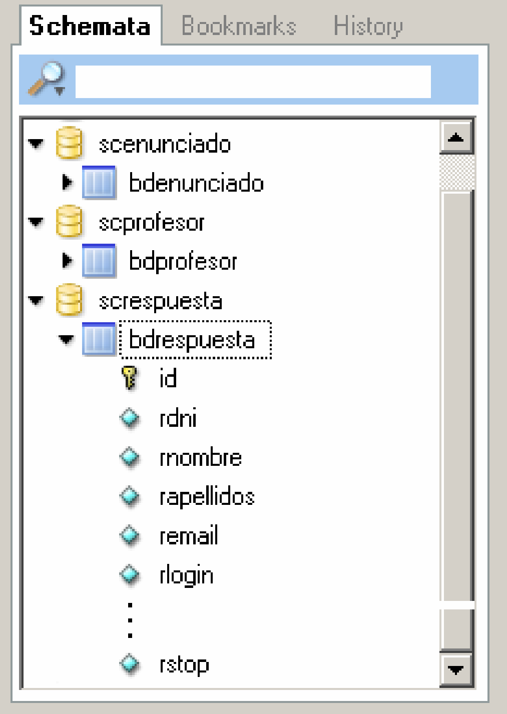


FIGURE. 7
CREATED DATABASE

WEB SITE

The Home page allows the users of the implemented application to connect as a *professor* or as a *student*, asking them in each case for the corresponding 'login' and 'password' and maintaining these authentication variables during the whole session. In 'Home' the user can link to the 'Help' page, which gives general information about the current application and its designers.

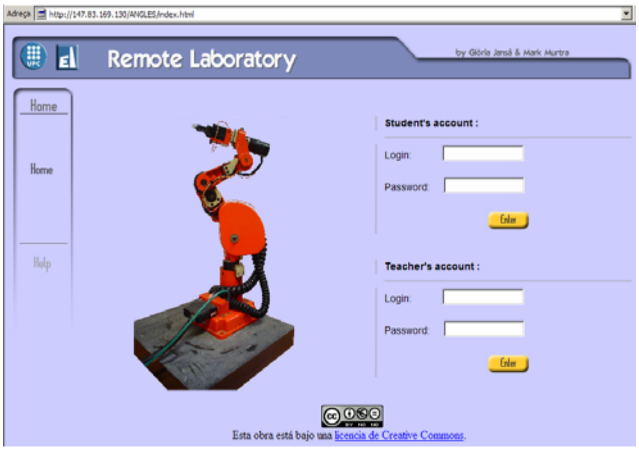

Adreça
http://147.83.169.130/ANGLES/index.html
Remote Laboratory
by Glòria Jansà & Mark Murtra
Home
Home
Help
Student's account :
Login:
Password:
Enter
Teacher's account :
Login:
Password:
Enter
Esta obra está bajo una licencia de Creative Commons.

FIGURE. 8
HOME PAGE OF THE APPLICATION

Connection as a 'Professor' User

When a user is connected as a *professor*, the system verifies that the 'login' and 'password' are correct, contrasting the information in the MySQL database connecting to it as a 'professor' user.

The main page shows whether the professor has been identified correctly. From this point, the user can link to the pages: Search, Insert, Delete, Help and Close (figure 9).

The '*Search*' page contains a field to write the student's identification number that wants to be found. Internally the page connects with MySQL database connecting as a 'professor' user in order to show the results in the Web page.

To insert student's data manually, the user have to link the '*Insert*' page, filling in the 'text box' with the student's information and, when pressing the button 'Insert', the site internally connects to MySQL database as a *professor* user and creates the new register automatically.

The '*Delete*' section allows identifying the student by his personal identification number, to delete his data from the database. When pressing 'Delete' the page calls an SQL action that deletes the register from MySQL database.

The '*Help*' page contains useful information for the professor about the Web Site. Pressing the '*Close*' option, the site redirects the user to the Home page of the application.

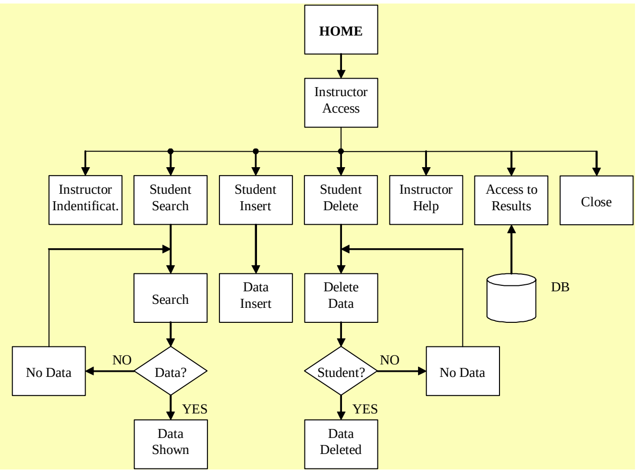


FIGURE. 9
FLOW DIAGRAM OF THE 'PROFESSOR' WEB SITE

Connection as a 'Student' User

Connected as a *student* user, the system verifies that the 'login' and 'password' are also correct, contrasting the information in MySQL database connecting as a '*student*' user. From this point, the user can link to the pages: Data, Results, Robot, Help and Close (figure 10).

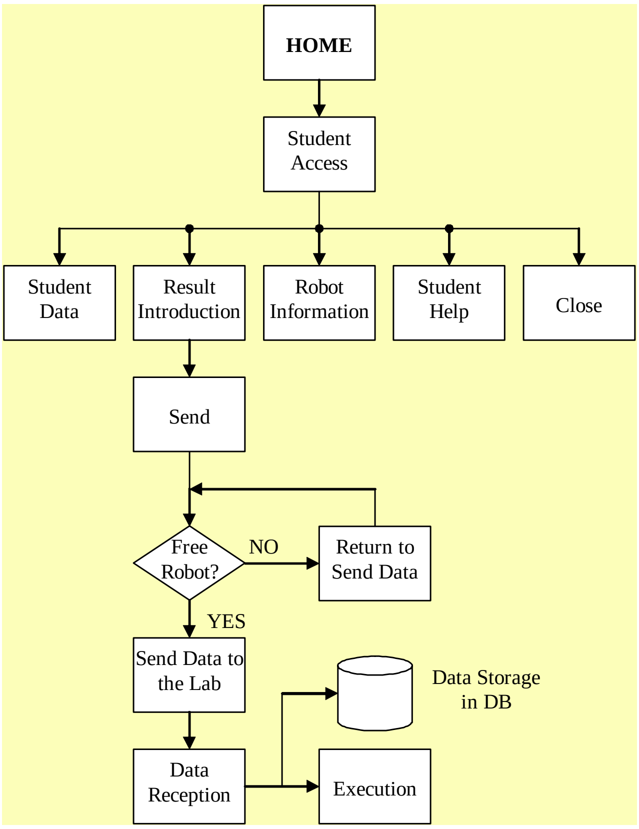
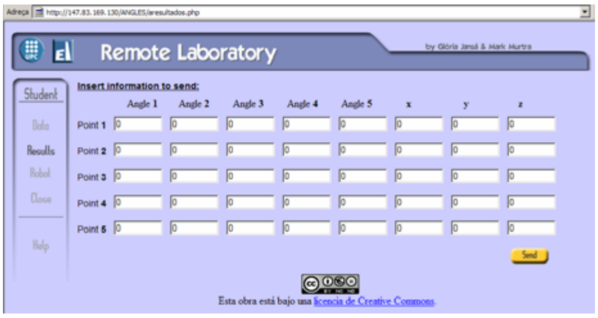


FIGURE. 10
FLOW DIAGRAM OF THE 'STUDENT' WEB SITE

'*Data*' option shows information about the student and his exercise or homework. In addition, the '*Results*' option allows introducing data which has to be sent to the remote laboratory (figure 11). When the send–data command is activated, the site checks if another user is working with the robotic arm. Thus, if the robot is busy, the site would redirect the user to the '*resending*' page where data would remain intact, waiting the robotic arm to be available. On the other hand, if the robot is available, data will be sent to MySQL database of the server connecting with the database as a 'student' user, and a special flag is set to '1' indicating that the robotic arm is busy. When the trajectory of the robot is finished, the Visual Basic application resets this flag to '0' automatically, informing that the robotic arm is available).


Adreça http://147.83.169.130/ANGLES/aresultados.php
Remote Laboratory
by Glòria Jansà & Mark Murtra
Student
Data
Results
Robot
Close
Help
Send

Insert information to send:

	Angle 1	Angle 2	Angle 3	Angle 4	Angle 5	x	y	z
Point 1	0	0	0	0	0	0	0	0
Point 2	0	0	0	0	0	0	0	0
Point 3	0	0	0	0	0	0	0	0
Point 4	0	0	0	0	0	0	0	0
Point 5	0	0	0	0	0	0	0	0

Esta obra está bajo una licencia de Creative Commons.

FIGURE. 11
'STUDENT RESULTS' WEB SITE

The '*Robot*' option contains the robotic arm's structure and the data necessary to make the theoretical calculus as well as the maximum and minimum articulation limits. The '*Help*' section contains information for the student about the Web Site. Finally, pressing the '*Close*' option the site redirects the user to the Home page.

The local project applications are made with the Visual Basic software. In particular, there are three applications: the main application, an executable to close the first one (the main application), and finally an application to fill the database with student's information from a plain text (*.txt) document.

VISUAL BASIC APPLICATION

Connection between Visual Basic and MySQL

All the project applications need a connection to MySQL database. Thus, it is necessary to have the 'Chain' General module in the Visual Basic that contains the path to the DLL (Dynamic Linking Library) and create and insert the ODBC (Open Data Base Connectivity) connector.

The ODBC connector is an administrative Data Origin tool that is used as a footbridge between MySQL database and the Visual Basic application. This connection allows to the application the interaction with the registers of one or more tables. To use this connector, it is necessary to have installed the ODBC and, later, create a DNS (Domain Name System) of the system.

The Main Application

When the application is executed, the subroutine 'Connect' is called to create a connection with the database in order to get the student information.

The whole functionally application is in the 'Timer' routine, the application stays in a polling iteration to database. While the 'flag' value is '0' (informing that there is not any student connected to the remote laboratory), the application will be still polling to database. If the 'flag' changes its value (through the Web Site) to '1' (thus informing that there is a student connected to the remote laboratory), the application will leave the polling iteration, and checks the 'STOP' field in the database, closing the program if this field value is equal to '1'.

Once the student is connected to the remote laboratory, the first process is defining the I/O 8255 card port status. The system sets all the points the user sends from the Web Site to variables, as well as the student's identification data.

To check the possibility that the trajectories are out of the security limits (figure 12), the system call to a subroutine where the angles obtained from the students are compared with the angle limits for every robotic arm's articulation. In case of an angle exceeds the security margin, the application will return to the initial waiting iteration and sends an error message that would be saved in an activity register's document located in the server. This document can be revised by the student checking which particular articulation is exactly out of the specified margins.

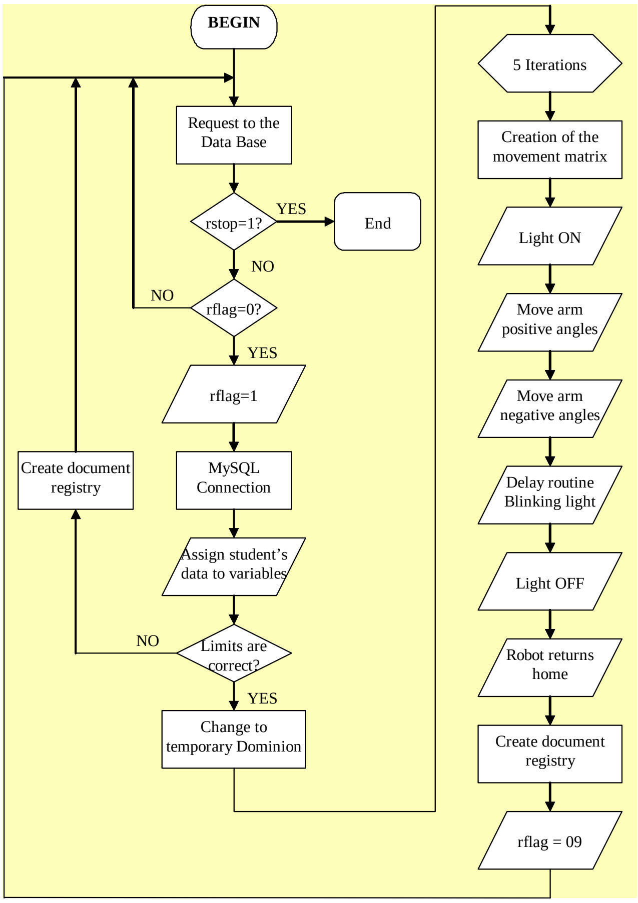


FIGURE. 12
FLUX DIAGRAM OF THE MAIN APPLICATION IN VISUAL BASIC

If all the angles are in the security margins, the system converts from the angle domain to a temporal domain using specific equations for each articulation and direction. Once the system has obtained the data inserted by the student in the time domain, it returns to the main routine and starts an iteration in which the mechanical arm follow the trajectory. In this step, the light indicator located in the manipulator free end is switched on.

To make the trajectory's movement, the application will work on the I/O 8255 card, switching its ports to '0' or '1' so the control board would be able to control the robotic arm's servos.

The application will create a matrix (articulation/time) that will be completed with logical '0' and '1' values. These values represent the activation of the corresponding pins of the I/O 8255 card. In this way, the system will save in every row of the matrix the status of the robotic arm's articulations in a time instant. Therefore, the robotic arm will get the effect of moving making a sweep of the aforementioned matrix. This matrix will be reloaded for every point; first, the system makes the movement of the positive angles and later the negative ones.

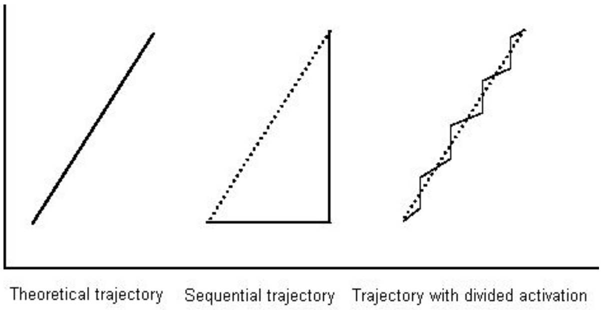


FIGURE. 13
POSSIBLE THEORETICAL TRAJECTORIES

To make the end of the mechanical arm follows a trajectory, instead of activating the servos in a sequential mode (in case of doing a diagonal trajectory it would made just one big step), we have designed a chopped movement (figure 13). Thus, all the articulations are activated at the same time, the one who have the longer time will be turned on during the trajectory, and the other articulations will spread their activation time, stopping at regular intervals. The whole sum of the total active times will be the same as if it was activated in a sequential mode. In this way, we get a stepped diagonal more representative of the real trajectory and all the articulations will finish their movement nearly at the same time (figure 14). Depending on the robot sensitivity, we can adjust the step in a minor or major value.

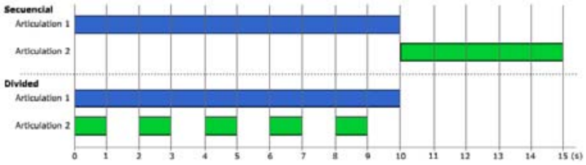


FIGURE. 14
MATRIX OPERATING

Once the movement of every point is made, all variables are set to zero, as while as the matrix, to let the system ready to the next point. The LED indicator placed at the end of the mechanical arm start blinking, showing the change of point. When the five points are done (figure 15), the indicator turns off. The robotic arm will stay stopped in the last point during a short time to finish going back to the beginning 'Home' position.

When the trajectory is finished, the application creates a text file as a register about the identification and data introduced by the user. In addition, the file informs if any error has happened during the process execution. To finish, the system changes the status of the flag indicator to '0' to inform that the robotic arm is now available, closes the connection with database and the application remains waiting in the polling initial iteration.

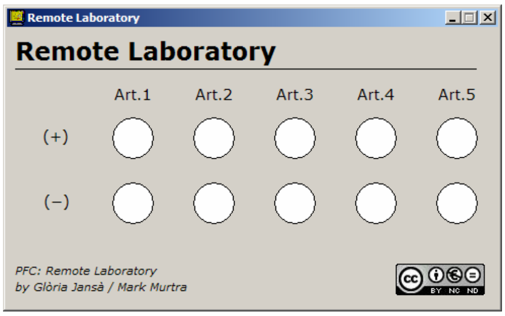


FIGURE. 15
VISUAL INTERFACE OF THE MAIN APPLICATION

'Close' Application

The main application needs another external application to close the first one (figure 16). The reason is that the main application is always in a waiting iteration. Pressing the button 'Close' call to an internal function which changes the value of the 'STOP' database's field to '1', this change makes both applications to be closed.

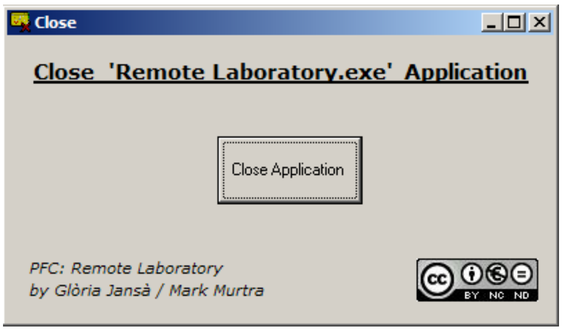

Close
Close 'Remote Laboratory.exe' Application
Close Application
PFC: Remote Laboratory
by Glòria Jansà / Mark Murtra
BY NC ND

FIGURE. 16
VISUAL INTERFACE OF THE 'CLOSE' APPLICATION

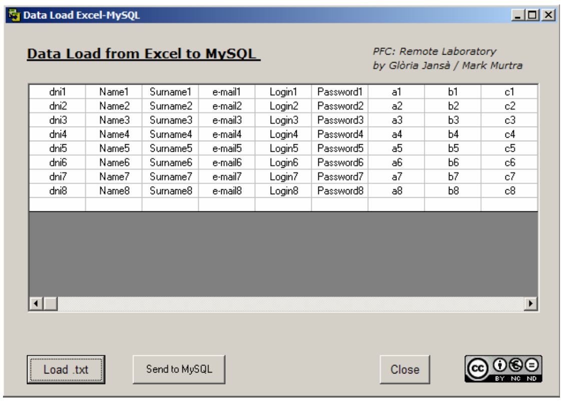
Excel–MySQL Application

Starting from an Excel document with just the data which have to be sent to database, it’s necessary to create a plain text with ‘.txt’ extension delimiting the fields thanks to tabs.

We execute the ‘data_load.exe’ (figure 17). Pressing on the ‘Load’ button will open a Common Dialog to select the ‘.txt’ file to load and the data will appear in the form.

Internally a matrix with all the ‘.txt’ data has been created. Pressing ‘Send to MySQL’ executes the ‘Chain’ module which allows to create the database connection and connect to a table sending all the information saved in the matrix. A confirmation message pops up.

We have designed some representative Icons for each application (figure 18).



Data Load Excel-MySQL

Data Load from Excel to MySQL

PFC: Remote Laboratory
by Glòria Jansà / Mark Murtra

dni1	Name1	Surname1	e-mail1	Login1	Password1	a1	b1	c1
dni2	Name2	Surname2	e-mail2	Login2	Password2	a2	b2	c2
dni3	Name3	Surname3	e-mail3	Login3	Password3	a3	b3	c3
dni4	Name4	Surname4	e-mail4	Login4	Password4	a4	b4	c4
dni5	Name5	Surname5	e-mail5	Login5	Password5	a5	b5	c5
dni6	Name6	Surname6	e-mail6	Login6	Password6	a6	b6	c6
dni7	Name7	Surname7	e-mail7	Login7	Password7	a7	b7	c7
dni8	Name8	Surname8	e-mail8	Login8	Password8	a8	b8	c8

FIGURE. 17
VISUAL INTERFACE OF THE ‘DATA LOAD’ APPLICATION

FIGURE. 18
APPLICATION’S ICONS

CONCLUSIONS AND FUTURE WORK

The article has presented the implementation of a remote laboratory in the field of robotic and automatic engineering. The main objective of this laboratory is to use it in didactical applications for electrical and electronics engineers. The platform offers to the students the opportunity to learn how to program a real robotic arm without having one at proximity, in a way that closely resembles the real robotic arm programming.

An important point of this application is the interaction in the software level (various applications between them) and in hardware–software level. Extrapolating this application in a greater scale, it is possible to design and manage an industrial processors plant by means of this remote laboratory.

The applications are ready to continue working on it, increasing and extending its functions. In fact, the presented application is just the first step, as a functional base, of a more extensive and multidisciplinar application.

Therefore, in relation to the software, it is possible to add more modules to improve the robotic arm feedback to the student. For instance, the recording of the robotic arm trajectories, the access of the student to the Webcam through the same Web page of the application, the inclusion of an angular indicator for the articulations (for example, triangulating the images obtained from two or more cameras), etc.

Related to the hardware, we can obtain the sensor signals of the robot in order to get information of their status directly to an I/O 8255 card’s port connected to the computer. To improve the movements and the manipulator’s sensitivity, it is hardly recommended to control the articulation motor speed, for example using a PWM control. In this way, it is possible to extend the angular articulation movement.

Finally, it is important to highlight that, in order to add the virtual laboratory concept to the remote laboratory, it is necessary to use a simulation software which allows, through the Web Site, to superpose the theoretical trajectory with the real one that the robotic arm makes [4]–[6]. Another possible improvement is to depurate and filtering the pictures obtained from the camera to have a trajectory trace.

REFERENCES

[1] B. A. Foss, K. E. Malving, and T. I. Eikaas. “Remote Experimentation–New Content in Distance Learning”. *International Conference on Engineering Education*, Oslo, Norway, Aug. 6 – 10, 2001, pp. 8D1-1–8D1-6, Session 8D1.

[2] J. Sanchez, S. Dormido, R. Pastor, and F. Morilla. “A Java/Matlab–Based Environment for Remote Control System Laboratories: Illustrated with an Inverted Pendulum”. *IEEE Transactions on Education*, vol. 47 (nº 3): pp. 321–329, Aug. 2004.

[3] C. S. Tzafestas, M. Alifragis, N. Palaiologou, S. C. A. Thomopoulos, M. Brahman, and A.–E. Exarchou. “Development and Experimental Evaluation of a Remote Laboratory Platform for Teaching Robot Manipulator Programming”. *Proceedings of the International Conference on Engineering Education (ICEE’2004)*, Florida, Oct. 16–21, 2004.

[4] F. A. Candelas, S. T. Puente, F. Torres, F. G. Ortiz, P. Gil, and J. Pomares. “A Virtual Laboratory for Teaching Robotics”. *International Journal on Engineering Education (Special Issue on Distance Controlled Laboratories and Learning Systems)*, vol. 19 (nº 3): pp. 363–370, 2003.

[5] E. Guimaraes, A. Maffeis, J. Pereira, B. Russo, E. Cardozo, M. Bergerman, and M. F. Magalhaes. “REAL: A Virtual Laboratory for Mobile Robot Experiments”. *IEEE Transactions on Education*, vol. 46 (nº 1): pp. 37–42, Feb. 2003.

[6] C. S. Tzafestas, N. Palaiologou, and M. Alifragis. “Virtual and Remote Robotic Laboratory: Comparative Experimental Evaluation”. *IEEE Transactions on Education*, vol. 49 (nº 3): pp. 360–369, Aug. 2006.