VOL. 15, NO. 11, JUNE 2020 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

DESIGN AND IMPLEMENTATION OF A VIRTUAL AND REMOTE LABORATORY TO CONTROL A SPEED PLANT

Faiber Robayo Betancourt, Jonatan Legro Pastrana and Camilo Andrés Muñoz
Department of Electronic Engineering, Faculty of Engineering, Surcolombiana University, Neiva, Huila, Colombia
E-Mail: faiber.robayo@usco.edu.co

ABSTRACT

This paper presents the design and implementation of a virtual and remote laboratory created for controlling a speed plant located in the control laboratory at Surcolombiana University. The aim purpose is that the student has the first contact with the plant (virtually) before working with the physical one, understands its operation, restrictions and prevents any damage to the same equipment. Students can handle the actual plant remotely by another application created for this purpose, enabling it to take data that later will serve for the identification, processing and implementation controllers designed to meet the desired specifications. Also, it has an IP camera to visualize and understand the effects of its running. The project is based on the Easy Java Simulations software (EJS), responsible for all the interactive and visual interface part, linked to the Matlab software. Performance validation of the generic controller was carried using Matlab tools, achieving great results, confirming the reliability of the applications created.

Keywords: easy java simulations, virtual, speed plant, remote laboratory.

1. INTRODUCTION

Universities and other higher education institutions around the world are facing the challenge of offering their services to a growing population of more socially and culturally diversified students. Nowadays education is increasingly active and participatory, adopting fewer traditional lessons, i.e. master classes. Current pedagogy should encourage the development of creative skills and abilities, the appropriate selection of information and the ability to formulate more pertinent questions and find more suitable responses. To the same extent, in our society, distance education is shown as the ideal solution for a group of groups (students, teachers, researchers, etc.) that demand to have more flexible, accessible and adaptable teaching systems (without spatial or temporary limitations) [1]. Therefore, the educational work of universities is subject to profound changes as far as the traditional methodology of knowledge construction is concerned.

The Internet and the growing speed of digital media allow the use of distributed software systems for remote access to virtual or physical laboratories to carry out experimental research activities remotely [2]. In the field of engineering education, distance learning has also evolved to such an extent that it is now common to find engineering specialties taught remotely; such is the case of Control Engineering. The teaching of this specialty requires that the theoretical concepts, which are taught through master classes, are complemented by practical activities in laboratories. Through these experiments students learnt some theoretical contents that generally become complicated to assimilate by means of traditional methods. Distance education is a drawback when incorporating practical experimentation, because with the traditional method students have to attend face-to-face classes in the laboratory to carry out this type of activities [3].

Information and Communication Technologies (ICT) have been applied substantially to improve paradigms to carry out practical learning and experimental research remotely and globally. These new paradigms are virtual and remote laboratory systems [4]. In this context, experiments can be adapted so that students can access them remotely through the Internet. In this way, teaching adjusts to the non-physical presence of students in the university to conduct their laboratory work. In turn, the laboratory experimentation can be complemented with a simulation of the process that the student will face during the process/practice with the real plant. These simulations allow the student to know the process and interact with the system in a virtual way, this being a good initial approach before facing the real equipment. This type of applications that work with a simulation of the process and allow access to real plants remotely are known in the world of engineering education as virtual and remote laboratories [3].

A virtual laboratory allows to simulate physical phenomena and models, abstract concepts, controlled simulation environments, among others. The mathematical model is transparent for the student and the simulated phenomenon is shown interactively. The previous ones have evolved transforming into remote laboratories, where a student uses and controls the real equipment through a local network or through the Internet thanks to the increasing complexity of the activities in a traditional laboratory and the development of tools designed for the Internet [5], [6]. The design, development, deployment of virtual laboratories and the evaluation of the role of online experiments to provide self-learning an innovative pedagogical practices for user communities has been executed [7].

In view of the above, it was necessary to design and implement a virtual and remote laboratory for a speed plant of the control laboratory at Surcolombiana University. This project becomes the foundations for this as well as other institutions of higher education to continue developing this kind of proposals, thus achieving better

working conditions for students and introducing distance learning in laboratory practices.

2. MATERIALS AND METHODS

2.1 Proposed Prototype

This project solves the lack of certain applications to make a difference in the traditional education, the former applied in the field of laboratory practices, more specifically, in a control laboratory of the electronic engineering area. The project was divided into two stages: the first is the application for simulation (virtual part) and the other is the application for remote control (remote part).

2.2 LISA

Intelligent Laboratory and Autodidactic Simulations (LISA), is the virtual and remote laboratory that has been developed for the control of a speed plant in a control laboratory. The application consists of a main panel, which is on the left of the interface, designed to control the simulation running and the interaction with the plant, as the input of a voltage step and the design of a PID controller that responds according to the "Speed Reference" parameter; a secondary panel on the right of the interface, which has three views or graphs: the first one shows the simulation of the plant response to a given step input, the second displays the plant Bode plot, and finally, the root locus of the plant. There is also a lower panel that shows the progress of the interaction, options to save the graphics, among others. Finally, it has a menu bar at the top of the interface, with tools that improve the interaction as shown in Figure-1.

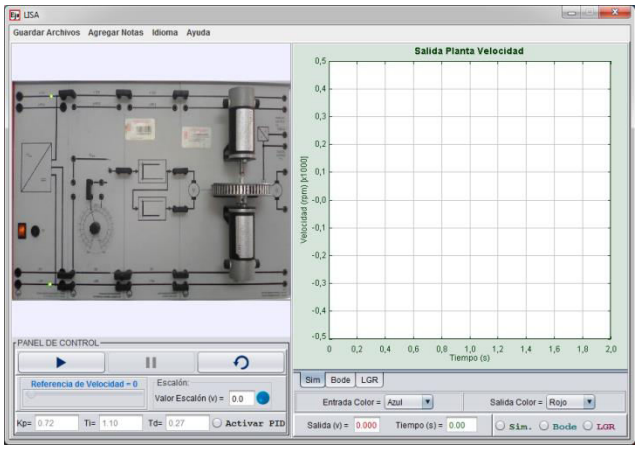

Figure-1. Main Window of Virtual LISA.

Through this application the student can see and analyze the diverse characteristics of the speed plant without being in person in a laboratory. One application is virtual while the other application will only allow remote control of the plant.

These applications have been designed through the Easy Java Simulations (EJS) software, an open source tool developed in Java, designed for the generation of dynamic interactive simulations [8]. This software was chosen due to its simple handling and interoperability with other tools such as Matlab and LabVIEW.

2.3 Implementation of the Virtual Laboratory to Control the Speed Plant

In the first place, the declaration of variables is defined in the EJS software, to then the differential equation in the sub-panel Evolution that is given by the transfer function of the plant is indicated, which was previously modeled as shown in Figure-2. Similarly, in that sub-panel the user can see a tab called Evolution_PID, where a code that allows the PID controller running and data storing are implemented.

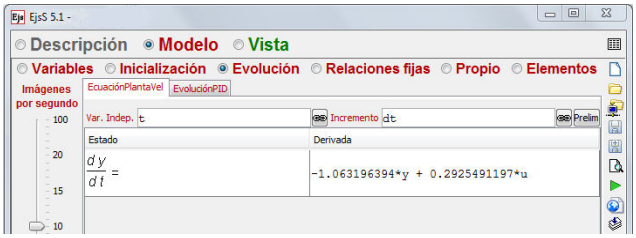

Figure-2. Differential equation of the speed plant entered in EJS.

Several tabs are included in the EJS variables sub-panel to keep an order in the creation of all the application elements. In addition, the programming is carried out so that the data of the plant response could be saved in a file (.m), where the variables included in the simulation are stored.

In the Initialization sub-panel of the Model panel, the code for creating the file (.m) is included in a tab. To adjust the numerical format of the data, the following tab allows the correct start-up and visualization of the program in Java versions. The third tab consists of initiating the values of the PID controller and the actions to be performed; finally, the last tab is in charge of making changes to the colors of the graphics.

In the Own sub-panel several lines of code are built in Java for the creation of different methods in order to store data in Matlab, assign the different colors for the simulation graphs, define a generic PID controller, modify the parameters of the controller and generate auxiliary information windows as presented in Figure-3.

```
public void mostrarpaginaAcerca()
{
  Acerca=false;
  _view.update();
  Acerca=true;
}

public void mostrarpaginaNotas1()
{
  Notas1=false;
  _view.update();
  Notas1=true;
}

public void mostrarpaginaNotas2()
{
  Notas2=false;
  _view.update();
  Notas2=true;
}

public void pasos () {
  _view.showDescription(false);
  _view.showDescription(true);
}
```

Figure-3. Equation Code of the sub-panel Own LISA in EJS.

Following the simulation development, the Bode diagram of the speed plant and its Geometrical Place of the roots were drawn. This is done using Matlab, taking out data from the "bode" and "rlocus" functions.

In the View panel, the creation of different panels is carried out, where the start, pause and restart buttons of the simulation are placed. The step input box that allows to visualize the response of the speed plant before any input value (0 - 20 volts) is located in another panel. This action can be done by entering the desired value in volts in the numerical field or just by turning the blue knob to the right. There is a slider to choose the desired speed value for the plant (this works when the PID controller is active); the boxes to enter the Kp, Ti and Td parameters of the PID controller designed are located in a different panel. It also has a button to activate the controller.

Boxes to show the simulation length are inserted in a different panel, that is, the numerical value of the output of the plant. Other three boxes were designed to save the simulation graphs, bode diagram and LGR. In another panel of that section, two tabs were included to change colors of graphics in the simulation.

Four tabs are located in a Menu bar, each one with different options: Menu_Save was programmed to save the simulation data in a file (.m) with an option; the next one is to save the desired graphic so that when having one of the boxes selected that curve can be saved as an image (.png); Menu_Notes, where two blank spaces are located for the student to write some notes when required. Menu_language gives the option to set up all the simulation either in Spanish or English. Finally, Menu_Help, was programmed with two options: First Steps and About LISA. The first one contains a description of each tab and LISA functions, the second one includes information of the designers and the version.

2.4 Implementation of the Remote Laboratory to Control the Speed Plant

For this application, variables of the Model panel are defined in the sub-panel, everything related to the variables that are associated with the block diagram implemented in Simulink and EJS, for the correct association with the real plant. In the Initialization panel of the Model panel, a couple of code lines are defined that allow the connection between EJS and Matlab Simulink as shown in Figure-4.

```
simulink.connect();
simulink.open("TesisLisaRemota.mdl");
```

Figure-4. Sub-panel Remote LISA Initialization in EJS.

In the other tabs of the Initialization option the user will find the necessary code to allow the same functions mentioned in LISA Virtual.

In the Evolution sub-panel of the Model panel, code lines are created to obtain data from Simulink execution, in order to deliver data to Simulink execution and perform a number of steps. In addition, extra lines to execute the controller correctly and a line to store data are written as presented in Figure-5.

```
simulink.set();
simulink.step(1);
simulink.get();

Ref_real=Referencia;

if(ControladorPlanta)
{
  u=controladorPID (pidControlVel,y,RefYVel,u);
}

recordData();
```

Figure-5. Remote LISA evolution sub-panel in EJS.

In the Own sub-panel of the Model panel, five tabs containing codes that allow the functions mentioned in LISA Virtual (saving data, first steps, updating parameters, among others) are located. In the Elements sub-panel of the Model panel, an external element was added to the EJS program so that it could be linked to the Simulink block diagram and be able to run the two programs simultaneously.

Figure-6 shows the Simulink Element editor, which allows the user to link two computers to work an application remotely. The server public IP address is placed in the Server address option. The port where the information will be sent and received can be found in the

Port box, and in the Transport option, the "http" option must be selected.

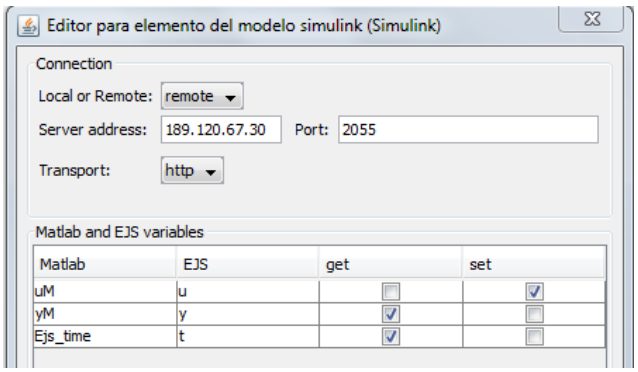

Figure-6. Editor for element of the Simulink model.

In the box Server address the word “localhost” must be written when it is required that a single computer works as a server and the client at the same time. The additional boxes shown are those that allow the association of the variables between the Simulink block diagram and those of the model in EJS.

The model implemented in Simulink, is supported by Matlab / Simulink thanks to the Arduino library. It was possible to add blocks that allow analogous reading and writing. In addition, there are gain blocks and constants that enable the correct conversion of data for reading them in the plant and writing to it. There are also some variables that are associated with EJS for the appropriate running of both programs as shown in Figure-7.

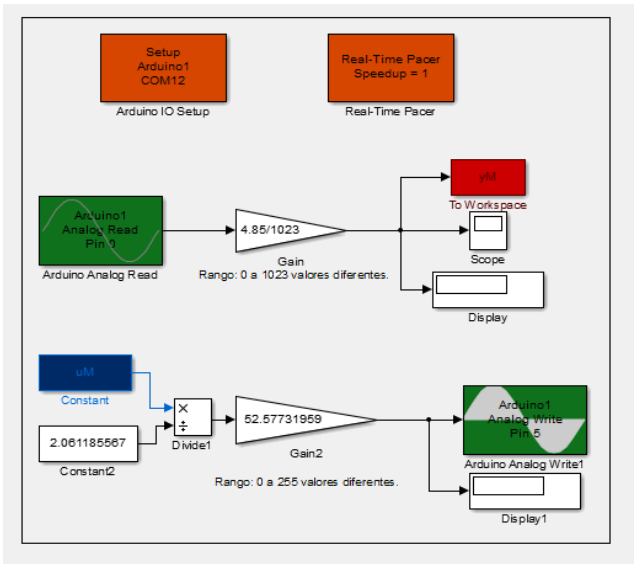

Figure-7. Sub-panel Editor for element of the Simulink model

In the computer that has the server role, the command presented in Figure-8 must be run at the Windows command prompt (cmd) or in the Linux terminal. The JIM Server is a file that was created by the EJS developers to allow remote connection between computers linked through the Matlab and EJS programs. It is a free tool; servers have also been developed that work with other programs such as LabVIEW.

Figure-8. Execution of the JIM server.

Following the description of the Remote LISA application, a control panel is proposed in which the user applies a voltage that does not exceed an established limit of 10 volts, this is for the safe running of the speed plant. There is also a button to activate the PID on this panel and it has some boxes to enter the designed values of the PID controller. While the Activate PID button is activated, the 'Step Value' bar cannot be moved because the controller that is entered will look for the appropriate input voltage for the plant to respond as entered in the 'Speed Reference' bar. In addition, as a novelty to the LISA Virtual application, two additional buttons were created to allow the streaming video of the speed plant to be connected and disconnected. It should be noted that the Play button must be press first and then the Connect button in order to have a correct visualization of the video otherwise only one photo of the plant will be displayed.

3. RESULTS AND DISCUSSIONS

3.1 LISA Virtual

When executing the LISAVirtual.jar file, two windows will appear; one of them is named "First Steps" where a quick guide is shown for the proper handling of the application, the other one is the main window of the application where everything pertinent to the speed plant could be simulated as presented in Figure-1.

When using a 5-volt input in the simulation and then a 7-volt input, the plant responds as shown in Figure-9. The input and output curves of the simulation have the red and blue colors by default, but it was programmed so that the user can choose between 8 colors to see the answer.

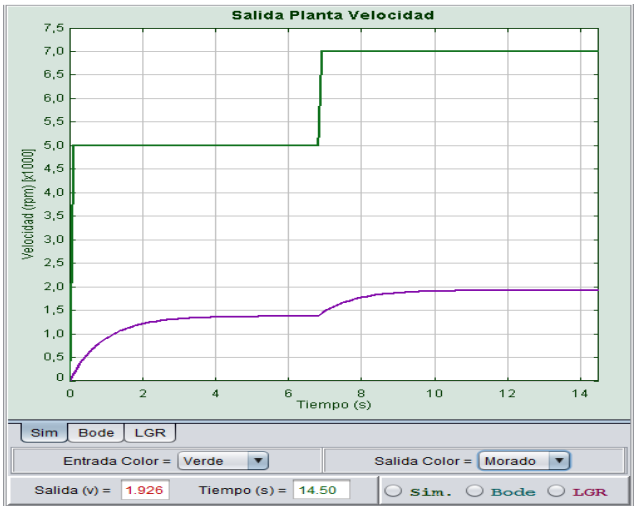

Figure-9. Virtual LISA response.

When changing from the 'Sim' to the 'Bode' option in the bottom right-hand of the application, the Bode diagram (magnitude and phase) of the plant is displayed. In addition, the LGR option can be changed, where the locus of the speed plant will be displayed in Figure-10.

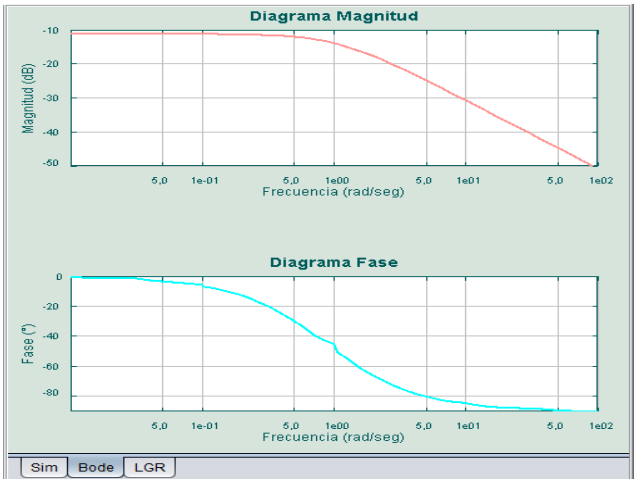

Figure-10. Bode of the speed plant represented in LISA Virtual.

To use the PID controller in the simulation, the button in the main window of LISA Virtual must be activated (Activate PID). This allows to see the action of the three boxes that make up the controller, Kp, Ti and Td, which can be modified to achieve that the output of the plant reaches a desired Speed Reference, in RPM. Figure-11 shows the response of the speed plant to a controller with a settling time of 10.5 seconds. Additionally, the simulation data can be saved in a file (.m) and save the graphics (.png) of the response of the plant, the Bode diagram and the LGR.

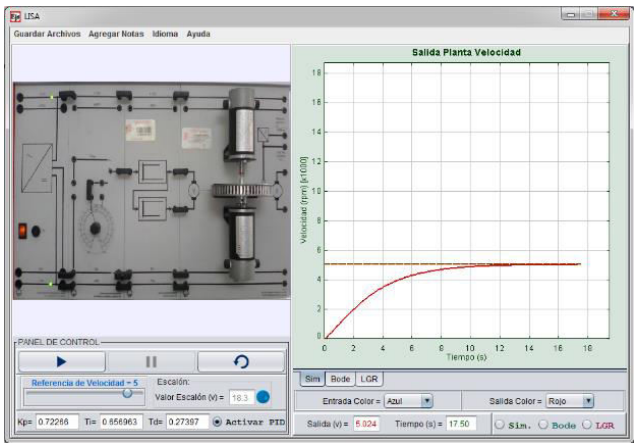

Figure-11. Response of the speed plant with a PID controller in LISA Virtual.

3.2 LISA Remote

Application with which the speed plant can be remotely controlled and real-time streaming can be visualized by means of an IP camera so that the user can monitor the response of the real plant and its output in RPM in another window of the application. When executed, two windows are displayed: one of them is the description of LISA and the other window is the main window of the application. When clicking on the Play button or when changing the "Step Value" slider, the remote control of the speed plant starts. When the value of the slider is changed, a voltage of that numerical value is applied to the plant and its response is observed in the window called "Exit Plant Speed".

In Figure-12 it can be seen that an input value of 5 volts was entered. The picture of the real plant is shown. This image is seen in this way because the IP camera is located on the module created by the designers to help the remote control of the speed plant.

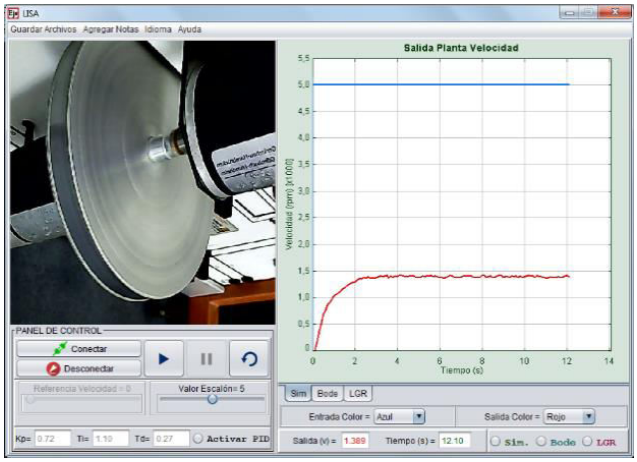

Figure-12. Remote LISA running with a 5v input.

It should be mentioned that Remote LISA has the same options as LISA Virtual, such as: saving data of the simulation or remote control for this case in a file (.m), saving figures as images (.png), adding notes, changing

the language, changing colors, and helping with a description and functions of LISA.

By pressing the 'Activate PID' button and considering the parameters of a controller that has an establishment time of 20.5 seconds, the operation of the remote LISA controller can be observed. By moving the "Speed Reference" slider and locating it at a specific RPM value, the response of the plant is obtained taking into account the parameters of the controller. It is also visualized how the rotation speed of the plant is increasing as well as the response in the window that has been arranged for that purpose as shown in Figure-13.

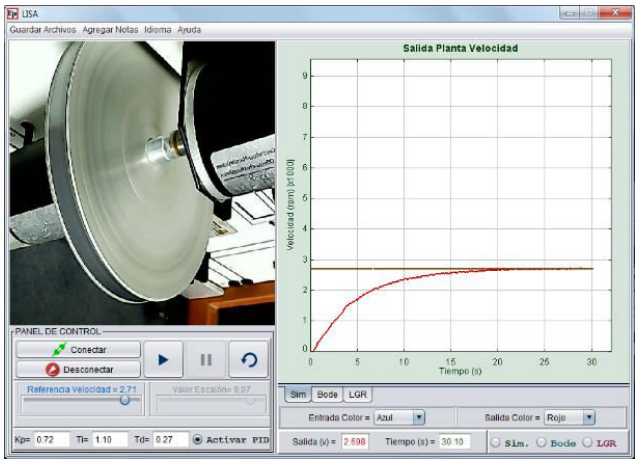

Figure-13. Remote LISA running with the PID controller activated.

Figure-14 shows the module created by the designers, which has a data acquisition card and an amplifier circuit, together with the speed plant, its power module and its power supply. This is the appropriate complement for Remote LISA to work. In this module an IP camera is integrated, which manages to visualize the streaming video of the plant fully operating in real time in the above mentioned application.

Figure-14. Final hardware product for remote LISA.

3.3 Validation of the PID Controller

The design of a controller developed with the "pidtool" tool of Matlab is presented. This controller is set up in such a way that the plant has a slow response as presented in Figure-15. Then the Kp, Ti and Td values are put in the boxes of LISA application as shown in Figure-16. It is observed that, in the simulation, the same slow response of the design is shown with a settling time of 10.4 seconds as shown in Figure-17.

Controller parameters

	Tuned
Kp	0.66267
Ti	0.52695
Td	0.13174
N	

Performance and robustness

	Tuned
Rise time	6.05 seconds
Settling time	10.4 seconds
Overshoot	0 %
Peak	0.999
Gain margin	Inf dB @ NaN rad/s
Phase margin	83 deg @ 0.314 rad/s
Closed-loop stability	Stable

Figure-15. Parameters of the slow controller obtained by "pidtool".

Figure-16. LISA control panel with the parameters of the slow controller.

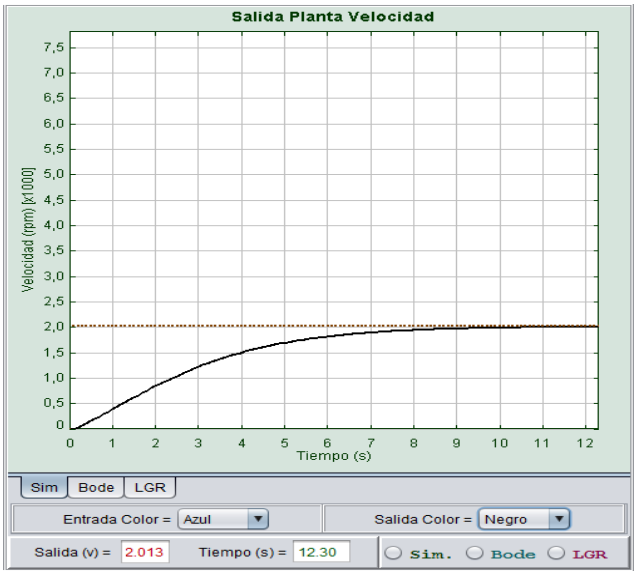

Figure-17. Output of the slow controller plant in LISA.

4. CONCLUSIONS

A virtual and remote laboratory to control a speed plant was designed and implemented, located in a control laboratory of Surcolombiana University. It is composed of two executable applications in Java: one simulates the behavior of the plant and with the other application the remote control of the real plant is achieved. Both have a great interactivity with the user and enable functions such as saving graphics and data generated during the simulation or execution to be processed later, among other functions useful for the student.

The Easy Java Simulations software was the key element in the development of the Java applications used in the project due to its functionality and ease when operating, without underestimating the good result obtained. In addition, it is worth highlighting its versatility to integrate with another kind of programs focused on engineering processes such as LabVIEW and Matlab. The latter being the one chosen to perform the whole numerical part and calculations of the remote laboratory application, achieving a more robust and complex application.

A module called "Control Module" which contains a designed amplification stage was created. It is used along with a data acquisition card to work as an intermediary in the control of the speed plant. Its main purpose is the remote control of the LISA Remota executable application together with its respective block set up with Matlab Simulink.

A significant contribution was made to virtual education and the use of ICTs for the learning process by contributing with these elements designed and implemented in the project to the Surcolombiana University. These are the foundations for future implementations in the same institution or in other higher education institutions in the region.

For future projects tools different to Matlab can be used and it is also possible to work with a remote application that leads all plants arranged in a control laboratory.

REFERENCES

[1] Santana I. 2012. Herramientas para la docencia en Automática orientadas hacia la Metodología ECTS. Tesis Doctoral, Universidad Politécnica de Madrid. p. 213.

[2] López F., Bertogna L., Sánchez L., Rodríguez J., Del Castillo R. 2009. Infraestructura para laboratorios de acceso remoto. Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología (TE & ET). pp. 31-39.

[3] Fábregas E. 2013. Plataformas Interactivas de Experimentación Remota: Aplicaciones de Control y Robótica. Tesis Doctoral, Universidad Nacional de Educación a Distancia. p. 303.

[4] Aguilar E. 2012. Arquitectura de software para un Laboratorio Virtual para estanques Acuícolas vía Internet. Tesis Magister, Instituto Tecnológico de la Paz. p. 190.

[5] Khamis A. 2003. Interacción Remota con Robots Móviles Basada en Internet. Tesis Doctoral, Universidad Carlos III de Madrid, 338p.

[6] Barnaby D. 2001. Techniques for Web Telerobotics. Tesis Doctoral, University of Western Australia. p. 243.

[7] Kumar D., Radhamani R., Nizar N., Achuthan Kr., Nair B., Diwakar Sh. 2018. Virtual and remote laboratories augment self-learning and interactions: Development, deployment and assessments with direct and online feedback. PeerJ Preprints 6: e26715v1.https://doi.org/10.7287/peerj.preprints.26715v1.

[8] Esquembre F. 2013. Creación de Simulaciones Interactivas en Java, Aplicación a la enseñanza de la física. Libro. p. 346.