An open software - open hardware lab of the air levitation system

Jacobo Saenz [*] **Jesus Chacon** [*] **Luis de la Torre** [*]
Sebastian Dormido [*]

[*] *Universidad Nacional de Educacion a Distancia (UNED), Madrid, 28040 Spain (e-mail: {jsaenz, jchacon}@bec.uned.es, {ldelatorre, sdormido}@dia.uned.es).*

Abstract: Virtual, remote and hands-on experimentation can be equally important and interesting for students of control engineering. This work presents a low-cost, open-based lab implementation of the air levitation system that can be easily developed in all the previous forms.

Keywords: Hands-on labs, virtual labs, remote labs, open software, open hardware.

1. INTRODUCTION

It has been said that the automatic control discipline is mainly based on two streams of thought (Kheir et al., 1996). The first one would be practical experience, while the second one would be theory and mathematics. Nowadays, few people would discuss that control engineers need to have both a wide experience implementing solutions in real problems and plants and a deep understanding of the mathematics and theory behind these solutions. The first stream, the one based in practical experience, relies on the idea that *something* needs to be controlled and so, control systems engineering curricula should be based on hands-on and practice experiences. While this has been the traditional vision of engineering, it started changing around one hundred years ago, when the second stream, the one based in theory and mathematics, started to gain importance (Froyd et al., 2012). This one, relies on learning and understanding abstract concepts, such as the four identified by (Kheir et al., 1996) as the major ones on control systems: *stability*, *feedback*, *dynamic system*, and *dynamic compensation*. Therefore, reaching a balance between theoretical proofs and physical intuition is a major challenge in control education. Lab experimentation plays a key role as a way to connect theory and practice. Among others, lab experience helps fulfilling the following goals (Antsaklis et al., 1998):

- Introducing students to real world modeling and/or control issues, such as uncertainties, saturation, noise, sensor/actuator dynamics, etc.
- Demonstrating, validating and/or motivating analytic concepts.
- Providing facility with instrumentation and measurement tools.
- Team learning and problem solving.
- Exposing students to broader design issues, from problem specification to hardware implementation.
- Developing professional practices, including maintaining engineering notebooks and report writing.
- Comparing theoretical results with real world results so that the theory can be validated.

Traditional hands-on labs entail high costs related with space requirements, equipment, and maintenance staff (Gomes, 2009). For the last twenty years there has been a line of research that looks for reducing lab costs by taking advantage of the Internet, i.e., by replacing hands-on labs with online ones.

In order to characterize the different approaches for experimentation practices, two criteria were proposed in (Dormido, 2004):

(1) According to the way resources are accessed for experimental purposes, environments can be either *local* or *remote*.
(2) According to the physical nature of the lab, environments can be either *real* or *simulated* plants.

The combination of the two previous criteria gives us a categorization of the existing possibilities for experimentation:

(1) *Local access-real resource.* This situation represents traditional hands-on labs, where students are in front of the real plant.
(2) *Local access-simulated resource.* In this combination, the whole experimental environment is software, and the experimentation interface works with a simulated/virtual resource.
(3) *Remote access-real resource.* In this case, a real plant is accessed through the Internet. Students remotely operate and control a real plant through an experimentation interface. This approach is typically named 'remote lab'.
(4) *Remote access-simulated resource.* The last form of experimentation is similar to the previous one, but it replaces the physical system with a mathematical model. This approach is typically known as 'virtual lab'.

As the technology has progressed and some of the major concerns related to Virtual and Remote Labs (VRLs) have been solved over the recent years, their importance and use have been growing (Heradio et al., 2016; de la Torre et al.,

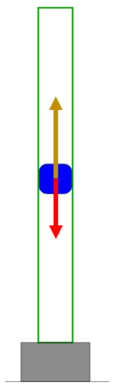


Fig. 1. Balance of forces acting over the levitating object.

2016). From the birth of VRLs, one of the above mentioned concerns has been assessing whether VRLs are capable of providing learning outcomes comparable to hands-on labs. Not only many studies have shown that VRLs and hands-on labs are equally effective (Brinson, 2015), but also VRLs provide additional advantages (Gravier et al., 2008):

(1) *Observability*: Lab sessions can be watched by many people or even recorded.
(2) *Availability*: VRLs can be used in 24/7 from anywhere, thus they support online and blended learning.
(3) *Safety*: VRLs can be a better alternative to hands-on labs for dangerous experimentation such as those related to nuclear physics, high power lasers, and so on.
(4) *Accessibility*: VRLs can be accessed by handicapped people.

Given the complementary uses of the previous experimentation approaches, it is probably best if an experiment itself is offered in several ways. This work presents a low-cost lab implementation of an air levitation system, based in open solutions. Thanks to this, it can be easily adopted to be used as both a remote lab and as a hands-on lab. The authors have also developed a simulation of the system. For those readers that might be interested in these resources, the virtual and remote laboratory can be found in UNILabs, a network of interactive online laboratories. For those readers interested in building the system to use it as a hands-on lab, the main instructions and tips to do so are given in this paper. The model of the air levitation system is presented in Section 2. Section 3 offers the guidelines for building the experimental setup and shows the software applications for the real and virtual operation of the system. Section 4 presents some experiences that can be performed with the developed lab. Finally, Section 5 contains the conclusions of this work.

2. THE SYSTEM

Newton's second law gives us the dynamic equation for the air levitation system, which has been studied by several previous works (Timmerman and van der Weelea, 1999; Jernigan et al., 2009). Since the only forces acting over the levitating object are be the upwards effect of the air flow (given by the first term in the right part of Eq. 1 and represented by the arrow going upwards in Fig. 1), and the downwards effect of the gravity (second term in the right part of Eq. 1 and represented by the arrow going downwards in Fig. 1):

$$m\ddot{z} = F = \frac{1}{2}C_d\rho A(v_w - \dot{z})^2 - mg. \qquad (1)$$

Where:

- m is the mass of the object to levitate.
- z is the vertical position of the object in the tube.
- ρ is the density of air.
- A is the objects area exposed to the upwards air flow.
- v_w is the velocity of the air inside the tube.
- g is the gravity.
- C_d is the so-called drag coefficient.

The drag coefficient is a term that depends on Reynold's number, which, in turn, depends on the relative velocity between the object that moves inside a flow and the velocity of such flow. For small velocities as the ones considered here, one can assume that C_d is constant. In that case, we can express Eq. 1 in terms of a constant named α, being: $\alpha = \frac{1}{2}C_d\rho A$:

$$\ddot{z} = \frac{\alpha}{m}(v_w - \dot{z})^2 - g. \qquad (2)$$

Now, the levitating object will be in steady state when it does not move, this is, when $\ddot{z} = \dot{z} = 0$. Let us call v_{eq} the air speed at such equilibrium point. Then:

$$g = \frac{\alpha}{m}v_{eq}^2. \qquad (3)$$

Therefore, we can finally express the dynamic equation like this:

$$\ddot{z} = g\left(\left(\frac{v_w - \dot{z}}{v_{eq}}\right)^2 - 1\right). \qquad (4)$$

2.1 Linearization

As Section 4 will show, the virtual lab features both the linear and the nonlinear models. Linearization for this system is pretty straightforward. Let $x = \frac{v_w - \dot{z}}{v_{eq}}$. Then, Eq. 4 is of the type $f = g(x^2 - 1)$, and it can be easily linearized around the equilibrium point ($v_{eq} = v_w - \dot{z}$, or $x = 1$) using Taylor's approximation ($f(x) \approx f(1) + f'(1)(x - 1)$):

$$\ddot{z} = 2g(x - 1) = \frac{2g}{v_{eq}}(v_w - \dot{z} - v_{eq}). \qquad (5)$$

3. THE LAB

Based on the low-cost and open source paradigms, several requirements were established to be satisfied by the remote lab design:

- The design should be as low-cost as possible (under 100$), so it will be affordable not only for universities, but also for students in case they would want to have their own experimentation platform.
- The design should be easily replicated, so that any educator with basic programming knowledge will be able to build one.
- It should use open-source technologies. This requirement has the twofold purpose of reducing cost to meet the first requirement, but also because this approach encourages to acquire knowledge by tinkering with

the system design, propose modifications or enhancements, and so on.

With these requirements in mind, there are some aspects that have been considered in order to keep low the costs of building the remote lab:

- Use materials that are easy to obtain and have a reasonable cost. For example, the IR sensor and single-board computer are cheap and can be bought in virtually any electronic component shop.
- Reuse components whenever it is possible. The fans can be easily gathered from an old PC or other electronic devices.
- Benefit from the boom rapid prototyping technologies. 3D printing allows to reduce greatly the cost of mechanic prototyping, and it is relatively easy to have access to a 3D printer, either at the university, specialized shop or online services which print your designs.

The following paragraphs give an insight of all the stages of the lab building, from the 3D modelling of the structural pieces and printing to the electronics and software setup, describing the Air Levitator System to have it as a reference design.

3.1 Air Levitator System

The air levitator system is composed of a cylinder in which a forced air flow is used to lift a small object levitating on a desired position. The structure is simple on purpose, there are only a few elements: a methacrylate tube with a nozzle, at one end, coupled with a blower fan. Both elements are supported by an open and movable stand, which let the air flow into the fan. The system has been constructed using only the following components:

- A methacrylate tube.
- A small and light object.
- 3D printed parts.
- A single-board computer (Beaglebone)
- An infrared distance measuring sensor (PIR).
- A PC fan.
- Some discrete electronic components and a PCB.
- A webcam.

3.2 Printed Parts

Most structural elements have been printed in a *Prusa Mendel i3* 3D printer (Figure 3), a very popular and affordable *RepRap* printer, available at the authors' department. The 3D parts has been modeled with FreeCAD (Figure 2), a mature open source CAD software for parametric design which is very popular to the 3D printing community. Among other features, *FreeCAD* is multiplatform, provides a graphical framework design, and can be easily extended with *Python* scripts.

3.3 The electronics

The air levitator system is controlled by a single-board computer, running a GNU/Linux distribution. The Beaglebone provide general purpose input/output (GPIO) pins to interconnect with external components. Since the

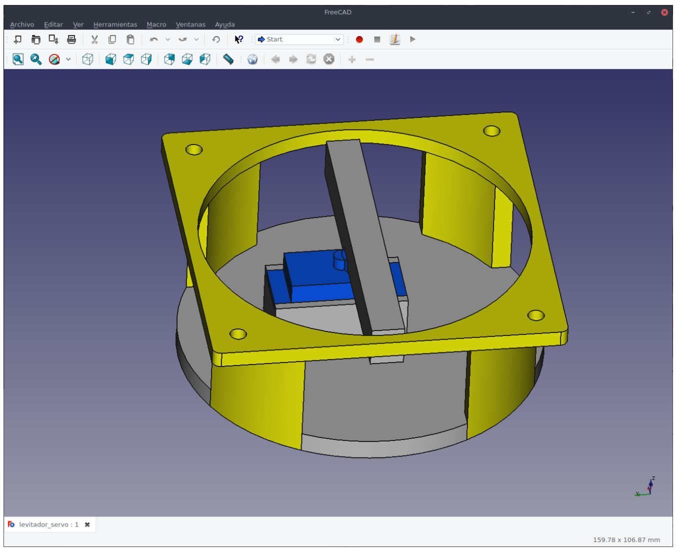


Fig. 2. 3D printed parts designed with FreeCAD software tool.

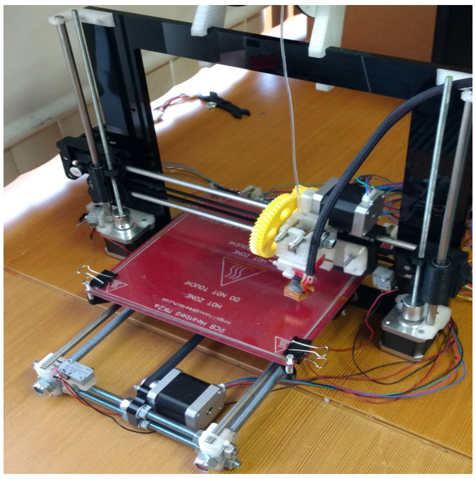


Fig. 3. 3D printer Prusa Mendel i3.

range of the voltage signal provided by the sensor (PIR) lies outside the one admitted by the analog inputs of the board $(0, 1.8V)$, it must be adapted before being connected. Similarly, the actuator (fans) requires voltages and currents that can not be directly handled by the board, so a signal conditioning circuit have to be used. The electronic circuits and the PCBs has been created with the software *KiCad*, a multiplatform and open-source tool that have the support of the *CERN*, which started the *KiCad* project and have made important contributions to it as part of the *Open Hardware Initiative* (OHI).

3.4 The server

The software in the target computer must implement several capabilities, including:

- *Hardware interface*, to be able to read measures from the sensors, send values to the actuators, and implement the control system.
- *Datalogging*, to register any important information about the system state.

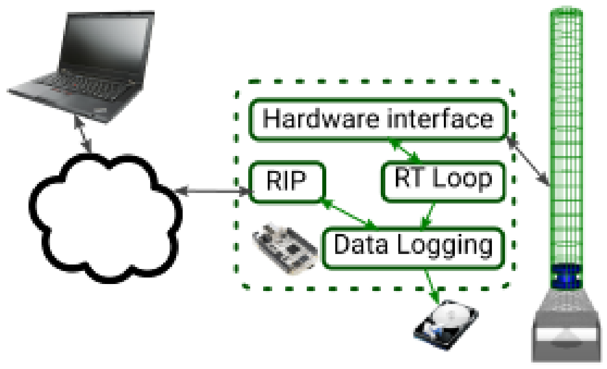


Fig. 4. The software architecture of the remote lab.

- *Communication*, to provide an API for the client to be able to interact with the system.

Though there are other alternatives that also meet the server requirements, it has been developed in *Node.js* because of several reasons: it is installed by default in the beaglebone official distro (*debian*), it provides a good performance and the *bonescript* library provides an arduino-like API to interact with the hardware.

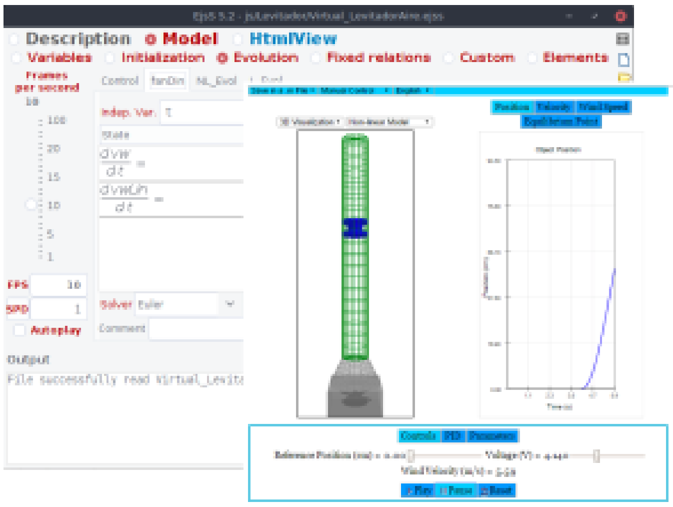
Hardware Interface This task is accomplished using the *bonescript* library, which basically mimics the Arduino API to cope with Beaglebone and the GPIO pins of the board. There is a *real-time loop* implementing the time critical actions: read sensors, update the controller and write ouputs. Technically, it is not actually real-time, because currently it is not supported by the *Node.js bonescript* library. But for the time scale of the system, which is sampled at a $100ms$ rate, it performs correctly. In case hard real-time is needed, there are other alternatives (such as C++) supported by the Beaglebone board.

Datalogging The datalogging capabilities have been separated into a low priority task that periodically dumps measures and control actions to a database, so the data is stored and can be accesed to perform off-line processing of past sessions.

Communication To make the server functionality accessible from outside of the lab computer, the middleware (in the diagram, the RPC module) implements the *Remote Interoperability Protocol* (Chacón et al. (2015)), which provides a standard API to control and monitorize the hardware. That basically means that any RIP enabled application can easily interconnect with the server to read and modify variables and plant parameters, so it is easy to decouple the GUI design from the rest of the system.

3.5 EjsS

EjsS is a tool which offers an easy way to create simulations with a GUI for users with no programming skills. These simulations can be made according to the user needs of interactivity and visualization. It is also an open source authoring tool designed for students and teachers. EjsS have been made originally using Java but, due to the security problems with Java, the last versions of EjsS



File successfully read Virtual_Levita

Fig. 5. The main view of the EjsS editor.

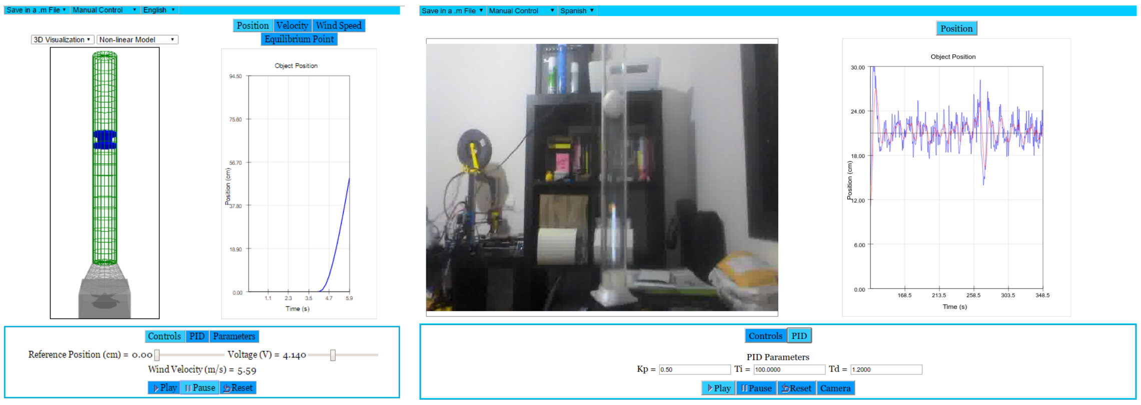
allows the user to create applications in both Java or Javascript. Other VRL and applications have been developed using EjsS and the related papers defines it as a tool that facilitates the development of applications by researcher, teachers and students who want to focus in the simulation theory and not in the technical programming aspects, (Farias et al., 2010; Chacon et al., 2015). EjsS also allows the user to run a finished application directly from the editor. If the aim is to publish the VRL as an online application, the developer can package it in order to run it later in a standalone mode (in the case of Java) or inside a web page to be run in a web browser (Java and Javascript). EjsS have been used before to develop different laboratories in their remote and virtual versions, (Bermudez-Ortega et al., 2015; Chaos et al., 2013; Galan et al., 2016; Heradio et al., 2014; de la Torre et al., 2015, 2012; Christian and Esquembre, 2007; Pastor et al., 2005; Saenz et al., 2015).

The EjsS Editor Figure 5 shows the Javascript editor. The top part in the editor contains description, model and view tabs. The editor allows to build a simulation or remote laboratory by adding the mathematical behaviour and a graphical interface. Then, the main application will be divided in two parts:

- The *model*. Using the same named tab in the editor, the developer can define differential equations, write their own code and/or make connections to other software or hardware. The complexity of the simulation and model depends only on the system implemented, the requirements and their knowledge about it.
- The *view* provides to the users a GUI that determines the interaction and visualization capabilities of the application. This view can be built using the editor by adding single view elements from the right panel of the EjsS editor (right side of Figure 5).

3.6 The Virtual/Remote Lab GUI

The application window is divided into three sections: a selection menu, a 2D and 3D graphical visual representation of the system, and the evolution graphs and indicators.



Reference Position (cm) = 0.00 Voltage (V) = 4.140
Wind Velocity (m/s) = 5.59

Kp = 0.50 Ti = 100.0000 Td = 1.2000

Fig. 6. The air levitation system virtual lab on the left, and the remote lab on the right.

Figure 6 shows the web interface of the virtual (left) and remote laboratory (right), designed with *EjsS* The interface design is clean and simple, sharing the same layout in both versions: there is a view of the system on the left, which is obtained from the laboratory webcam (or the 3D visualization in the virtual lab), some plots on the right showing the time evolution of the interesting variables (the height of the lifting object measured by the IR sensor, the setpoint and the control signal sent to the fan). Finally, at the bottom there is a control panel which allows to modify some system parameters, as the controller gains or the setpoint, and the connection buttons, in the remote lab, which are analogous to the simulation execution control ones, in the virtual lab.

4. EXPERIENCES

Nowadays, there are a great collection of accessible laboratories through internet. Using some of these the student can learn new concepts or practice with real systems. UNED, as distance education university, have to enhance their labs, where the main objective is to teach the students as in hands-on classic laboratory. In this regard, any course with a VRL, have to also include a activities document and an user manual.

- The *activities document* contains guidance to the student in order to reach the objectives of a common lab. Using this guide, the knowledge about systems, control and analysis, eventually the user can obtain fundamental parameters of the system structure or dynamics, information about the best controller configuration, etc.
- The *user manual* is how-to work with the GUI of the laboratory. It contains the needed information about the available interactions with the interface and a description of the main structure of the views and menus.

4.1 Activities with the lab

The activities described in this section can be performed only in remote version while others can also be performed in a virtual way. The course structure allows the student to work with the labs in the order they choose (except in some hard-to-use or advanced labs). But it is highly recommended by the teach team to work first with the virtual version to acquire experience and to practice with the theoretical behaviour of the system. Therefore, they work with the real system in order to compare the results obtained from the real experimentation with those obtained from the simulation. The next paragraphs explain the main activities to be performed with this lab.

- *PD/PI/PID Tuning*: In the previous subsection we have seen that the user can select an automatic mode, where the system runs into a closed loop. This loop is established for controlling the position of the levitating object inside the tube, by changing the control signal to the fan by means of a PID controller. Using the PID tab in the indicators zone of the application, the user can modify the parameters of the PID controller, the proportional, integral and derivative parameters of the PID controller (k_P, T_I, T_D). The activities require a system output signal that satisfies the asked specifications, like response time, overshoot, etc.
- *Disturbances Analysis*: The virtual and remote labs allows introducing perturbations in the system. It can be done by changing the wind flux outside the fan. In the virtual version the behaviour is simulated, changing the maximum velocity that generates the fan with random signal. To obtain this kind of response in the remote version, the real equipment contains a servomechanism similar to a plane flap. This part have been also made in the 3D printer. The flap is designed to divide the base under the fan in two different parts, changing then the local airflow and the flux inside the tube.
- *System Identification*: This activity can be performed using the numeric data gathered in a lab session and using mathematical software to obtain useful information about order, poles and zeros of the system. The virtual laboratory allows the user to interact with the simulation in three ways: the linear, the non-linear models and both together. Using the acquired data in each step the student can obtain accurate information of the system. This knowledge about the virtual air levitation system and the study of the mathematical model can be useful to obtain information about the real plant in the remote laboratory, where the system identification is not easy.
- *Object Shape Analysis*: As have been shown the system description section the model depends on some

constants like density, object mass, objects area and the drag coefficient. In the activities document the student is encouraged to obtain information about the behaviour changes when they select the different object only for the virtual version of the lab. The available shapes are ball, cone and cylinder.

5. CONCLUSION

This work presents a low-cost air levitation system to be used in both a virtual and a remote version. This kind of VRL can be used by teachers to complement their classes giving to their students access to a simulated and real resources, which are very important in scientific and engineering areas. These laboratories enhance the knowledge about control experimentation, the analysis of data to obtain information about systems and the basis in laboratory practices.

ACKNOWLEDGEMENTS

This work was supported in part by the Spanish Ministry of Economy and Competitiveness under Project DPI2012-31303.

REFERENCES

Antsaklis, P., Basar, T., DeCarlo, R., McClamroch, H., Spong, M., and Yurkovich, S. (1998). NSF/CSS workshop on new directions in control engineering education. Technical report, University of Illinois at Urbana-Champaign.

Bermudez-Ortega, J., Besada-Portas, E., Lopez-Orozco, J., Bonache-Seco, J., and de la Cruz, J. (2015). Remote web-based control laboratory for mobile devices based on ejss, raspberry pi and node.js. In *3rd IFAC Workshop on Internet Based Control Education, IFAC-PapersOnLine*, volume 48, 158–163. Brescia, Italy.

Brinson, J.R. (2015). Learning outcome achievement in non-traditional (virtual and remote) versus traditional (hands-on) laboratories: A review of the empirical research. *Computers & Education*, 87, 218–237.

Chacon, J., Vargas, H., Farias Castro, G., Sanchez Moreno, J., and Dormido, S. (2015). EJS, JIL Server and LabVIEW: An architecture for rapid development of remote labs. *IEEE Transactions on Learning Technologies*, (99). doi:10.1109/TLT.2015.2389245.

Chacón, J., Farias, G., Vargas, H., Visioli, A., and Dormido, S. (2015). Remote interoperability protocol: A bridge between interactive interfaces and engineering systems. *IFAC-PapersOnLine*, 48(29), 247 – 252. doi: http://dx.doi.org/10.1016/j.ifacol.2015.11.244.

Chaos, D., Chacon, J., Lopez-Orozco, J.A., and Dormido, S. (2013). Virtual and remote robotic laboratory using ejs, matlab and labview. *Sensors*, 13(2), 2595–2612.

Christian, W. and Esquembre, F. (2007). Modeling physics with easy java simulations. *The Physics Teacher*, 45, 475–480.

de la Torre, L., Guinaldo, M., Heradio, R., and Dormido, S. (2015). The ball and beam system: A case study of virtual and remote lab enhancement with moodle. *IEEE Transactions on Industrial Informatics*, 11(4), 934–945.

de la Torre, L., Sanchez, J.P., and Dormido, S. (2016). What remote labs can do for you. *Physics Today*, 69, 48–53.

de la Torre, L., Sanchez, J.P., Heradio, R., Carreras, C., Yuste, M., Sanchez, J., and Dormido, S. (2012). Unedlabs - an example of ejs labs integration into moodle. In *World Conference on Physics Education.*

Dormido, S. (2004). Control learning: present and future. *Annual Reviews in Control*, 28(1), 115–136.

Farias, G., Keyser, R.D., Dormido, S., and Esquembre, F. (2010). Developing networked control labs a matlab and easy java simulations approach. *IEEE Transactions on Industrial Electronics*, 57, 32663275.

Froyd, J.E., Wankat, P.C., and Smith, K.A. (2012). Five major shifts in 100 years of engineering education. *Proceedings of the IEEE*, 100, 1344–1360.

Galan, D., Heradio, R., de la Torre, L., Dormido, S., and Esquembre, F. (2016). Automated experiments on EjsS laboratories. In *International Conference on Remote Engineering and Virtual Instrumentation*, 78–85. Madrid, Spain.

Gomes, L. (2009). Current trends in remote laboratories. *IEEE Transactions on Industrial Electronics*, 56, 47444756.

Gravier, C., Fayolle, J., Bayard, B., Ates, M., and Lardon, J. (2008). State of the art about remote laboratories paradigms - foundations of ongoing mutations. *International Journal of Online Engineering*, 4(1), 19–25.

Heradio, R., de la Torre, L., Galan, D., Cabrerizo, F.J., Herrera-Viedma, E., and Dormido, S. (2016). Virtual and remote labs in education: a bibliometric analysis. *Computers & Education*, 98, 14–38.

Heradio, R., de la Torre, L., Sanchez, J., and Dormido, S. (2014). Making EJS applications at the OSP digital library available from Moodle. In *International Conference on Remote Engineering and Virtual Instrumentation*, 112–116. Porto, Portugal.

Jernigan, S.R., Fahmy, Y., and Buckner, G.D. (2009). Implementing a remote laboratory experience into a joint engineering degree program: Aerodynamic levitation of a beach ball. *IEEE Transaction on Education*, 52, 205–213.

Kheir, N.A., Åström, K.J., Auslander, D., Cheok, K.C., Franklin, G.F., Masten, M., and Rabins, M. (1996). Control systems engineering education. *Automatica*, 32(2), 147–166.

Pastor, R., Sanchez, J., and Dormido, S. (2005). Web-based virtual lab and remote experimentation using easy java simulations. In *Proceedings of the 16th IFAC World Congress.*

Saenz, J., Esquembre, F., Garcia, F.J., de la Torre, L., and Dormido, S. (2015). An architecture to use Easy Java-Javascript Simulations in new devices. In *IFAC Workshop on Internet Based Control Education.* Brescia, Italy.

Timmerman, P. and van der Weelea, J.P. (1999). On the rise and fall of a ball with linear or quadratic drag. *American Journal of Physics*, 67, 538–546.