RHLab Scalable Software Defined Radio (SDR) Remote Laboratory

Marcos Inonan[1], Brian Chap[1], Pablo Orduña[2], Rania Hussein[1], and Payman Arabshahi[1]

[1] University of Washington, Seattle WA 98195, USA,
{minonan,brchap,rhussein,paymana}@uw.edu,
[2] LabsLand, St. Louis, MO 63101, USA
pablo@labsland.com

Abstract. Educational remote laboratories are software and hardware tools that allow students to access real equipment located somewhere else to perform tasks. One of the key features of remote laboratories is that once they are developed, the number of students who can benefit from them by accessing those labs remotely can scale fast. This contribution is focused on the development of an open-source remote laboratory for teaching SDR (Software-defined radio). The paper describes how the laboratory is developed from the technical perspective, what are the challenges of these remote laboratories compared to other types of remote laboratories, and how universities and colleges will be able to use this laboratory in the future.

Keywords: GNU Radio, Software Defined Radio, remote laboratory, embedded systems

1 Introduction

An Educational Remote Laboratory is a software and hardware solution that enables students to access real equipment located in their institution, as if they were in a hands-on-lab session, using an standard web-browser. The laboratories are typically hosted in universities or research centers.

A key factor of remote laboratories is that once they are available through the Internet their usage can be scaled up and used by students of other institutions. Thus, two or more institutions can share different equipment to reduce costs by requiring less duplicated equipment: it is typically only used in certain hours of the day and in certain days of the year.

In the literature there is a wide range of remote laboratories in many fields (e.g., robotics, electronics, physics, chemistry). However, few experiences have been developed in the field of Software-defined radio (SDR). In this field, in the traditional lab, students usually have a receiver and a transmitter and, through software, they define what radio-frequency (RF) signals are sent from one and obtain and process the signals from the receiver. Given that the communication between the components is carried through RF, having multiple setups of the same remote laboratory in a single room adds additional challenges in terms of isolation when compared with most remote laboratories in the literature.

In this paper, our research team at the Remote Hub Lab (RHLab) [3] describes an SDR Remote lab developed in the context of a National Science Foundation (NSF) funded project. The project includes both technical aspects (development of the laboratory, software, scalability) and pedagogical aspects (usage in class, evaluation): in this contribution we exclusively cover the current state of the first (technical) aspects.

The remainder of the article is structured as follows: Section 2 explains the background of the project: examples of remote laboratories, usage of SDR in class, and existing remote laboratories for SDR. Section 3 details the design of the system and the different components of the laboratory. Section 4 explains the results so far of the project. Section 5 describes the next steps of the technical aspects of this project and Section 6 will summarize the article and the conclusions.

2 Background

This section summarizes the current status of the background of the project. It includes: en explanation of how SDR labs are used in class nowadays (without a remote laboratory) in Section 2.1, later a brief explanation of the current status of remote laboratories in Section 2.2 and finally the current SDR Remote Laboratories available in the literature in Section 2.3.

2.1 Software-defined Radio labs

SDR technology has prompted many engineering educational institutions to build or upgrade their communication labs. The idea of implement multiples hardware prototypes of wireless communications by only modifying a few lines of software is optimal for engaging students in communications theory and teaching them concepts of signal processing and communications.

However, even though SDR has facilitated the configuration and experimentation with radio transmitters, students are still required to be physically present in the lab. For this reason, there are strong motivations to develop a remote Lab in this area of engineering.

In a traditional electrical engineering lab course, each student receives a kit of components to work on their lab assignments [1]. Specifically, a communication lab the kit contains one or more SDR devices, antennas, coaxial cables and connectors. Normally, universities give them the choice to either use a physical lab that is equipped with computers with the drivers and software they need installed or work at home where students are responsible of installing all the software package their computer. Since GNU Radio[4] community has motivated programmers, frequently, it is needed to update the versions of the latest version.

[3] https://rhlab.ece.uw.edu/
[4] https://www.gnuradio.org/

Figure 1 shows a diagram of a typical SDR configuration. Through SDR software environment, the user designs a radio device by connecting blocks and configuring input and output parameters. Examples of SDR environment are LabVIEW, GNU Radio Companion (GRC), or MATLAB+Simulink.

The GRC[5] is a Simulink-like graphical SDR software environment to design radio devices by adding and connecting blocks, which is defined as a flowgraph. When a GRC file is "compiled" it creates a Python Script. This Script imports the libraries to connect the blocks and plot the data trough graphical QT widgets.

Figure 2 shows a example of the design of a sine wave transmitter in GRC. The flowgraph is composed by 4 blocks and one graphical QT-widget that once it is compiled generates of plot of the streaming data.

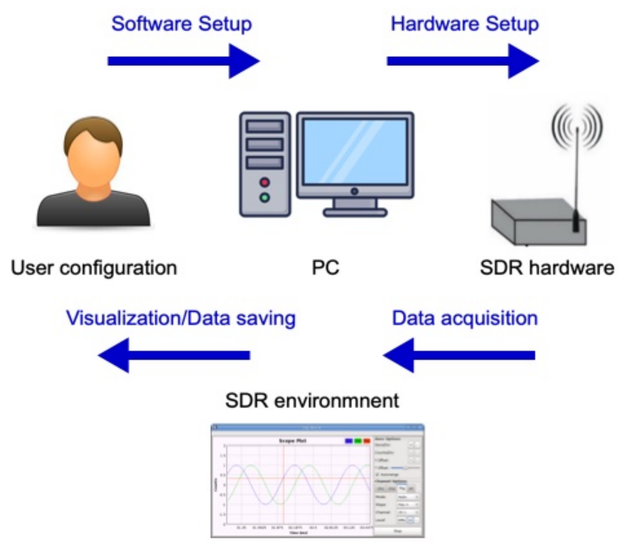


Fig. 1: SDR general configuration

2.2 Remote Laboratories

A remote laboratory is a software and hardware tool that allows students to remotely access real equipment located in the university. Users access this equipment as if they were in a traditional hands-on-lab session, but through the Internet. To show a clear example, Figure 3 shows the LabsLand Arduino Robot

[5] https://wiki.gnuradio.org/index.php/Guided_Tutorial_GRC

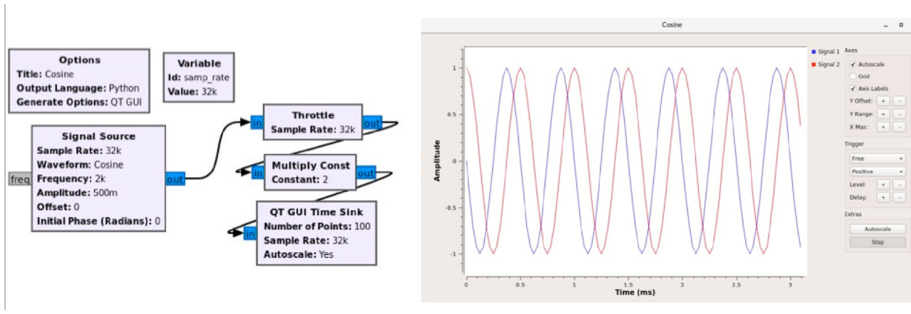


Fig. 2: GRC flowgraph of a source of sine wave and a QT-based widget that plot the signal source

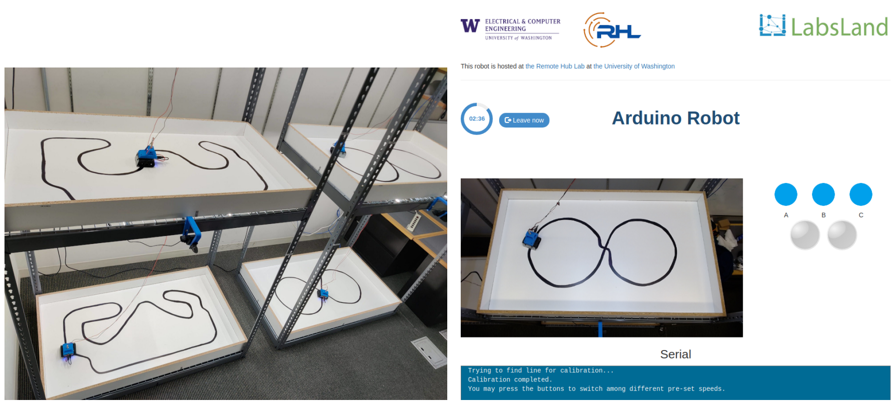


Fig. 3: Arduino Robot Laboratory at the University of Washington. On the left, the robot itself. On the right, the user interface once the program has been submitted.

Laboratory at the University of Washington: with previous versions described in the literature in Spain [2], South Africa [3] or Colombia [4]. Students learn to program a Pololu Zumo32u4 robot with Arduino using either C code or a visual programming language (Blockly), and they can write the code anywhere, compile it with the proper tools, and then submit the binary file to a real robot through the Internet. Then, students can see how the robot performs with their program through the Internet (e.g., if it follows the black line according to the submitted program, etc.) in a real environment.

In this line, there are many examples and classifications in the literature [5,6]. Indeed, remote laboratories were born nearly two decades ago [7,8,9], and since then they have been adopted in multiple fields: Chemistry [10,11], Physics [12,13], Electronics [14,15] or Robotics, [3,4,2].

2.3 SDR Remote Laboratories

Few examples of remote laboratories using SDR have been reported. The most relevant one is [16], where a single setup of a SDR remote laboratory is detailed. Other examples include [17], relying on the FIRE federation protocol or other experiences described in [18] and more recently [19].

3 System Design

This section details the system design: what are the components and interactions in 3.2, and how is the hardware setup in Section 3.2.

3.1 Overall design

The overall design of the system is as follows:

- There is a web server with a set of services that provides the user interface that the student loads, and all the communications with the SDR kits will go using HTTP and websockets with that server. The server has multiple independent services: one is user-focused (RELIA Web), whereas another (RELIA Scheduler) manages the assignments of which particular setup will be used in each session, and another one (RELIA data exchanger) focuses in the message exchange between the SDR Kits and the server.
- Additionally, in second layer, there is a set of *setups*. Each setup consists of two Raspberry Pi, each with a SDR Kit. Whenever a student is assigned to a particular *setup*, the student will be sending two GNU Radio configurations (typically one for the receiver and another for the transmitter), that will be loaded in each of the Raspberry Pis.
- By design, there will be multiple setups, and students will be assigned automatically to one, to another, or they will be in a queue waiting for one of the systems to be available.

Figure 4 summarizes the overall design of the system.

3.2 Hardware setup

Among all the alternatives available on the market, Adalm-PLUTO[6] was chosen as it is especially designed to guide students to carry up initial experimentation in real-world Radio Frequency (RF). Another reason to choose Adalm-PLUTO is the active and large community supported by its manufacturer Analog Devices. Besides, technical specification like half and full duplex operation, a high Signal to Noise Ratio (SNR), flexible rate makes this device appealing to be the heart of RELIA.

[6] https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/adalm-pluto.html

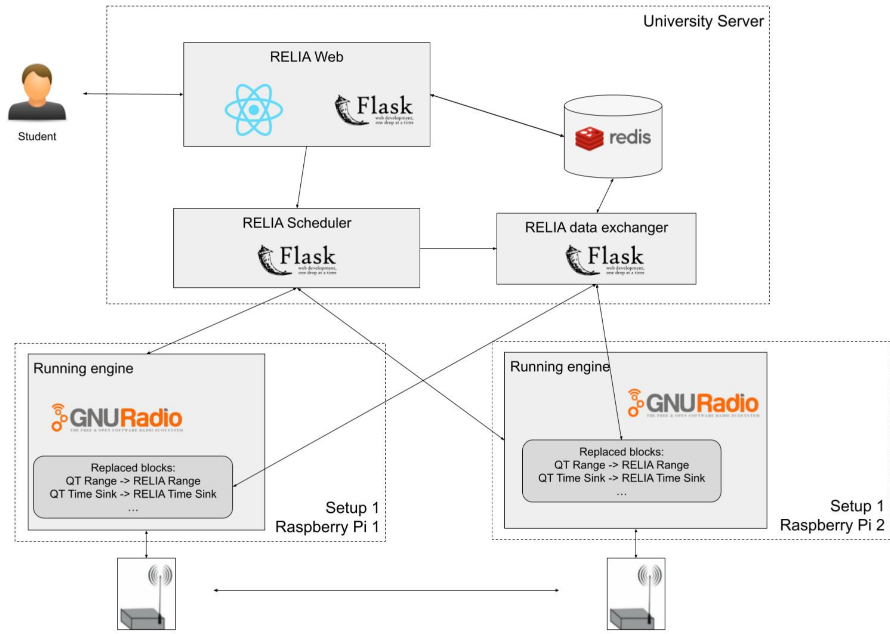


Fig. 4: RELIA architecture

RELIA lab is based on a number of modules, each module is composed by 2 Adalm-PLUTOs working together (one for transmitting and other for receiving). If various RELIA modules are working simultaneously in a room then one of them can interfere others and vice-versa. Therefore, in the case we have tens on RELIA modules, a strong interference will affect the communication between them.

One solution to solve this problem was to isolate every RELIA module within a Faraday cage (FC). Figure 5 shows one module before to be enclosed in a box covered by a nickel and copper sheet shield that acts as a case of a FC. To measure the level of isolation in every RELIA module, Packet Reception Ratio (PRR) was used as a metric. The test consisted on transmitting a sequence of 100K ASCII characters using the Binary phase-shift keying (BPSK) modulation. The FCC result is obtained by counting the number of characters received correctly then dividing it by 100K. Thereby, PRR is a kind of efficiency of transmission. Several tests were applied in order to measure the level of interference in function of the distance between modules. The results of these test are described in the section 4.2

To control the Adalm-PLUTO, the Raspberry PI 4 Model B was selected. Its four USB ports, open source platform and low cost ideal computer to control it. Besides, is important to mention that both the Adalm-PLUTO and the Raspberry PI can be fully programmed in Python, which facilitates the high speed data transfer between them.

Fig. 5: RELIA module inside a Faraday Cage

4 Results

4.1 Web version

RELIA's web platform offers a full interaction with the GNU Radio applications similar to the executed one with GRC. Thus, the user has various widgets (input) and sinks (graphical outputs) available in the RELIA package.

First of all, the user submits the GRC file designed on his computer to RELIA. RELIA searchs on it all the QT-widgets and replaces them with RELIA-widgets to creates a new GRC file. Next, this new GRC file is compiled to Python using the GNU Radio Companion Compiler (GRCC).

Once the Python script is executed, the data generated is stored in a memory database (REDIS) fast enough to send large amount of data streaming to the web. Finally, a web interface executed on the server displays the online process. The graphical widgets are based on Google Charts for the web visualization .

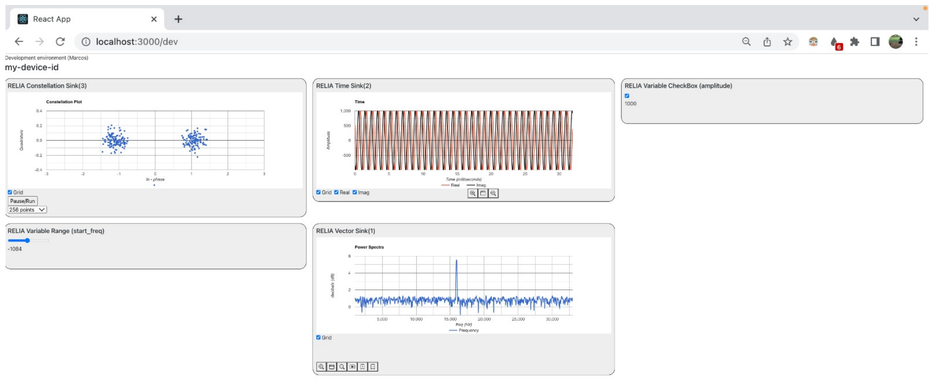


Fig. 6: Web User interface of Relia Lab based on Google Charts tools

Figure 6 shows an example of how a flowgraph designed in GRC remotely is viewed in a web interface. RELIA has widgets modules to dynamically modify input parameters such as center frequency, power, amplitude, offset, etc. Likewise, user can rearrange the upcoming data in in a customized way. For instance, RELIA has tools like zoom in/out, pause/run, averaging, noise level, etc.

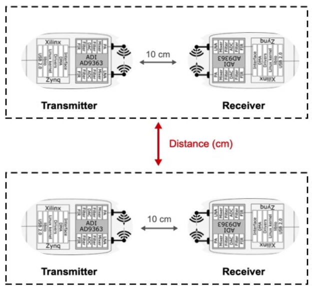
4.2 SDR isolation

In order to avoid interfering from one module to another one, a FC was used. Results of transmitting a sequence of 100k of characters using a BPSK modulation show a high efficiency when at least one module is enclosed in a FC., on the other hand, when there is no FC the transmission efficiency decrease significantly. Three test were performed to get this conclusion: first, when 2 modules work simultaneously; second when only one module is enclosed in a FC and third when both modules are isolated by a FC.

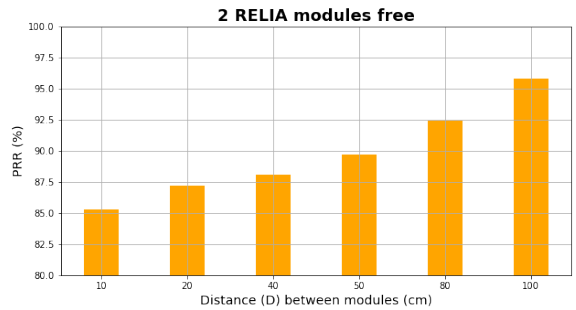
Figure 7a shows the diagram of the first test where 2 modules work simultaneously without FC isolation and the noise from one module to another caused the lost of 15% of the packages transmitted when they are at a distance of 10cm. The PRR increased when the distance between 2 modules is bigger but even when the distance is 1 meter the PRR achieves 96%, which is not ideal because it supposed that a large area would be needed to install a remote lab. This is displayed in Figure 7b.

Results improve in the next scenario. Figure 7c illustrated the test of one module enclosed in a FC and another one free. In this case, the PRR is almost ideal showing and efficiency of 99% in the module isolated by a FC even when the distance is minimal (10cm). The results keep the same performance even if the free module is further away until 1 meter. This is plotted in Figure 7d.

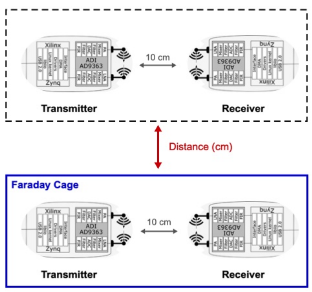
In the third test, both modules are enclosed in a FC. This is illustrated in Figure 7e. Results of PRR are very similar to the ones obtained in test 2. This can be seen in Figure 7f.



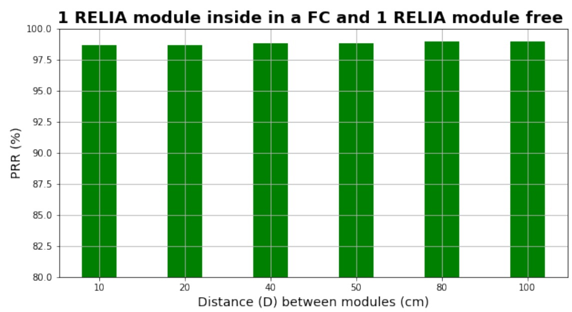
(a) Transmission of 2 modules without FC isolation



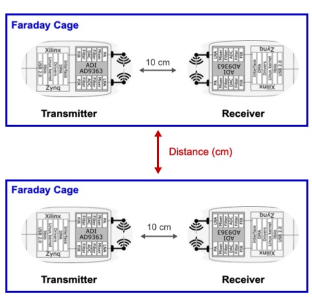
(b) PRR of transmission of 2 modules without FC isolation



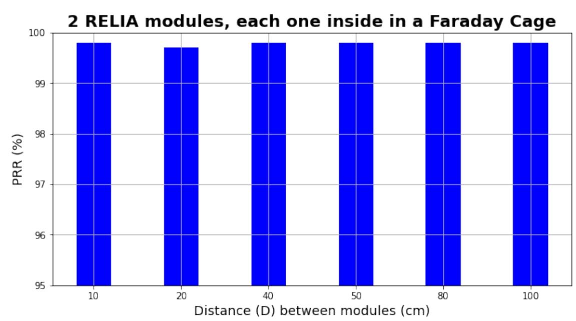
(c) Transmission of both 1 module enclosed in a FC and 1 module without FC isolation



(d) PRR of transmission of both 1 module enclosed in a FC and 1 module without FC isolation



(e) Transmission of 2 modules enclosed in a FC



(f) PRR of transmission of 2 modules enclosed in a FC

Fig. 7: PRR results

5 Future work

This project is open source[7], and documentation will be available in the RELIA website[8] as part of the Remote Hub Lab[9]. Anyone therefore will be able to replicate the software, the setup and use it in their university with the also open-source WebLab-Deusto [20] software.

However, LabsLand[10], the global network of remote laboratories, participates in the project developing part of the software and will adopt this laboratory to make it available in a federated way across institutions [21]. Once the laboratory is available in the LabsLand network, potential users of it will benefit from the features that LabsLand already provides (dashboard, different authentication mechanisms, integration of Learning Management Systems such as Moodle, Canvas, Sakai or others), and from the fact that LabsLand will have multiple copies of the laboratory in multiple universities, providing a higher reliability. It will also be possible to purchase hardware solutions to set up the remote laboratory in a new university without going through the effort of designing all the components.

6 Conclusions

This paper explains the technical details of the development of the RELIA Lab and its features. It is designed to offer the users the same quality of visualization and flexible reconfiguration than GRC. Besides, RELIA offers the advantage of building a communication system device from any location and at any time without dealing with potential hardware issues. Therefore, RELIA helps user to reduce the setup time since most experiments can be rapidly configured when they are ran remotely.

The ADALM-Pluto was chosen as the SDR device for RELIA because of its design oriented to educational purposes. Likewise, its easy configuration with Python makes this device much more attractive due to the compatibility with a low cost computer like the Raspberry Pi.

One challenge at building RELIA was the signal interference every module could generate to another when they work simultaneously. The solution proposed was to to isolate every RELIA module with Faraday Cages. Results of PRR show that transmission is 99% successfully when a module is enclosed by a Faraday Cage. On the other hand, when a RELIA module interferes other, the PRR decrease to 85% and its efficiency is not good even if the distance between modules is increased.

The modular architecture of RELIA permits a rapid growth of development since its functionality is based on widgets. Therefore, in case there are new versions of GRC, it is only necessary to work in a new widget and add this to the RELIA package.

---

[7] https://gitlab.com/relia-project/
[8] https://rhlab.ece.uw.edu/projects/relia/
[9] https://rhlab.ece.uw.edu
[10] https://labsland.com

Acknowledgements

This work was supported by the National Science Foundation's Division Of Undergraduate Education under Grant 2141798.

References

1. Hussein, R., Wilson, D.: Remote versus in-hand hardware laboratory in digital circuits courses. In: 2021 ASEE Virtual Annual Conference Content Access (2021)
2. García-Zubía, J., Angulo, I., Martínez-Pieper, G., Orduña, P., Rodríguez-Gil, L., Hernandez-Jayo, U.: Learning to program in k12 using a remote controlled robot: Roboblock. In: Proceedings of the 2017 REV Conference
3. Kwinana, P.M., Nomnga, P., Rani, M., Lekala, M.L.: Real laboratories available online: Establishment of revel as a conceptual framework for implementing remote experimentation in south african higher education institutions and rural-based schools–a case study at the university of fort hare. In: International Conference on Remote Engineering and Virtual Instrumentation. pp. 128–142. Springer (2020)
4. Buitrago, P.A., Camacho, R., Pérez, H.E., Jaramillo, O., Villar-Martinez, A., Rodríguez-Gil, L., Orduna, P.: Mobile arduino robot programming using a remote laboratory in unad: Pedagogic and technical aspects. In: International Conference on Remote Engineering and Virtual Instrumentation. pp. 171–183. Springer (2020)
5. Gomes, L., Bogosyan, S.: Current trends in remote laboratories. Industrial Electronics, IEEE Transactions on 56(12), 4744–4756 (2009)
6. Gravier, C., Fayolle, J., Bayard, B., Ates, M., Lardon, J.: State of the art about remote laboratories paradigms-foundations of ongoing mutations. iJOE 4(1) (2008)
7. Carisa, B., Burain, A., Molly H, S., Lawrence, C.: Running control engineering experiments over the internet (1995)
8. Aktan, B., Bohus, C., Crowl, L., Shor, M.: Distance learning applied to control engineering laboratories. Education, IEEE Transactions on 39(3), 320–326 (1996)
9. Henry, J.: Running laboratory experiments via the world wide web. In: ASEE Annual Conference (1996)
10. Coble, A., Smallbone, A., Bhave, A., Watson, R., Braumann, A., Kraft, M.: Delivering authentic experiences for engineering students and professionals through e-labs. In: Education Engineering (EDUCON), 2010 IEEE. pp. 1085–1090. IEEE (2010)
11. Arguedas-Matarrita, C., Montero-Miranda, E., Vargas-Badilla, L., Sánchez-Brenes, R., Rios-Badilla, E., Orduña, P., Rodríguez-Gil, L.: Design and development of an ultra-concurrent laboratory for the study of an acid–base titration (abt) at the universidad estatal a distancia (uned), costa rica. In: International Conference on Remote Engineering and Virtual Instrumentation. pp. 122–130. Springer (2021)
12. Del Alamo, J., Brooks, L., McLean, C., Hardison, J., Mishuris, G., Chang, V., Hui, L.: The mit microelectronics weblab: A web-enabled remote laboratory for microelectronic device characterization. In: World Congress on Networked Learning in a Global Environment, Berlin (Germany) (2002)
13. Gillet, D., Latchman, H., Salzmann, C., Crisalle, O.: Hands-on laboratory experiments in flexible and distance learning. Journal of Engineering Education 90(2), 187–191 (2001)
14. Gustavsson, I., Zackrisson, J., Håkansson, L., Claesson, I., Lagö, T.: The visir project–an open source software initiative for distributed online laboratories. In: Proceedings of the REV 2007 Conference, Porto, Portugal (2007)

15. Nedic, Z., Machotka, J., Nafalski, A.: Remote laboratory netlab for effective interaction with real equipment over the internet. In: Human System Interactions, 2008 Conference on. pp. 846–851. IEEE (2008)
16. Xu, Z., Chen, W., Qu, D., Hei, X., Li, W.: Developing a massive open online lab course for learning principles of communications. In: 2020 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE). pp. 586–590. IEEE (2020)
17. Mikroyannidis, A., Collins, D., Tranoris, C., Denazis, S., Pareit, D., Gerwen, V.V., Moerman, I., Jourjon, G., Fourmaux, O., Domingue, J., et al.: Forge: an elearning framework for remote laboratory experimentation on fire testbed infrastructure. In: Building the Future Internet through FIRE-2016 FIRE Book: A Research and Experiment based Approach/Serrano, M.[edit.]; et al.[edit.], pp. 521–559 (2017)
18. Somanaidu, U., Telagam, N., Nehru, K., Menakadevi, N.: Usrp 2901 based fm transceiver with large file capabilities in virtual and remote laboratory. iJOE 14(10) (2018)
19. Bhimavaram, K.R., Hiremath, P.S., Kumar, S.A.: Remote labs for communications. In: Online Engineering and Society 4.0: Proceedings of the 18th International Conference on Remote Engineering and Virtual Instrumentation. vol. 298, p. 47. Springer Nature (2021)
20. Orduña, P., Irurzun, J., Rodriguez-Gil, L., Garcia-Zubia, J., Gazzola, F., López-de Ipiña, D.: Adding new features to new and existing remote experiments through their integration in weblab-deusto. International Journal of Online Engineering (iJOE) 7(S2), pp–33 (2011)
21. Orduña, P., Rodriguez-Gil, L., Garcia-Zubia, J., Angulo, I., Hernandez, U., Azcuenaga, E.: Increasing the value of remote laboratory federations through an open sharing platform: Labsland. In: Online Engineering & Internet of Things, pp. 859–873. Springer (2018)