



AN IMPLEMENTATION OF A WEB LABORATORY CONVERTING OFF-LINE EXPERIMENTS INTO REMOTELY ACCESSIBLE EXPERIMENTS

Dorđe Seničić^{1*},
Milan Matijević²,
Marko Tanasković³,
Luis de la Torre⁴

¹CogniPix,
Belgrade, Serbia

²Faculty of Engineering at
University of Kragujevac,
Kragujevac, Serbia

³Singidunum University,
Belgrade, Serbia

⁴UNED,
Madrid, Spain

Abstract:

The paper presents a procedure for realizing a Web Laboratory, which enables the conversion of offline experiments into experiments available remotely. In this way, the user is provided with the same capacity for experimentation as in the case of conventional laboratory exercises. This paper aims to offer the proposed methodology to anyone who has an interest in organizing remote experiments. The proposed methodology ensures the integrity of remote experiments concerning possible cyber-attacks.

Keywords:

Web-Laboratory implementation, Remote experiments, Conversion of off-line experiments into remote experiments.

INTRODUCTION

Proper insight into experimental phenomena and real laboratory experience (according to the engineering educational principle "mind and hand") allows students to understand educational goals and theoretical concepts, understand the relationship between reality and theory, acquire professional skills, develop analytical skills, and a high level of knowledge absorption during the teaching process correctly and faster. Laboratory work requires the engagement of great resources (equipment maintenance costs, competent human resources, relevant laboratory infrastructure). To reduce the effective costs of an equipped laboratory place per student and provide relevant laboratory support to the teaching process, different universities have integrated their laboratory resources through strategic partnerships in the building of joint or open Web Laboratories (for example, see repository <https://www.golabz.eu>). A Web Laboratory is a real cyber-physical system that can be accessed through the Internet, where students can remotely operate and control a real system/plant through an experimentation interface. It is possible to access a Web Laboratory from anywhere at any time, but with only one hardware user simultaneously.

Correspondence:

Dorđe Seničić

e-mail:

djordje@cognipix.com



The existing capacities of the Web Laboratories around the globe are not enough for a huge number of students who are oriented to this approach due to the "COVID-19 - crisis".

In the context of the RELAB project, the company CogniPix has designed a technical solution for implementing the Web Laboratory, which will be presented in this paper [1, 2]. Unlike previous approaches [3, 4, 5, 6, 7, 8, 9, 10], the goal is to convert the existing laboratory cyber-physical system into a WEB Laboratory, with the purpose that existing laboratory exercises in engineering schools can be alternatively transformed into online laboratory exercises with the fullest possible capacity and cyber safety standards. The paper also contains an illustrative example that explains the pedagogical side of demonstrating or validating analytical and theoretical concepts. The proposed concept of the WEB Laboratory encourages the development of real-time programming skills to the same extent as the hands-on laboratory. This Web Laboratory performance is an advantage over common solutions for Web Laboratory design.

2. STRUCTURE OF WEB LABORATORY AND MINIMAL HW/SW REQUIREMENTS

Web Laboratory exercises in Figure 1 (three typical experiments or plants for monitoring and control were selected) are connected to a microcontroller Arduino UNO or Arduino DUE, optionally, which is further connected to a PC on the Internet. In this case, the PC contains the installed x2GO client-server (necessary software platform for WEB-Laboratory implementation), and the PC has the following performance: Giga-Byte GB-BACE-3000 BRIX Mini PC, operating system: Linux uBuntu 20.04.3, processor: Intel Celeron N3000, number of processor cores: 2, processor clock speed:

1.04 GHz (2.08 GHz)), RAM: 4 GB. The recommended minimum HW configuration of a PC (for the necessary software installation for WEB Laboratory installation) is a 2 GHz dual-core processor, 2 GB RAM, and 25 GB hard disk space.

WEB Laboratory consists of 4 components:

1. **The experimental laboratory setup** encompasses all the necessary equipment attached for conducting a laboratory exercise (computer equipment such as the Arduino UNO or Arduino DUE in Figure 1, sensors, actuators, physical plant, etc.). This includes the installation of multiple standard Debian packages on top of Ubuntu i20.04LTS.
2. **Lab Server** is a part of the experimental laboratory setup, and it can be Ubuntu/Debian based lightweight computer (x86-based or ARM-based like RPi). Our approach to WEB Laboratory design is based on the assumption that Lab Server does not have (IPv4 or IPv6) public address. In addition, we won't discuss the VPN option as it requires more complex IT setups, which may also create other security concerns.
3. **Weblab Public Server** is a somewhat arbitrary term, but this is typically VPS (Virtual Private Server) available as a Cloud asset, provided by University Campus IT service, or rented from Public Cloud Provider (AWS, Digital Ocean, Azure, Google, or some local cloud provider). It can be used for multiple purposes, like Web server, Chat/Blog Server, File Server, Streaming server, etc. We will limit the scope of discussion in this document to additional (very few) services/configuration modifications required for the proposed solution. This is also typically a Linux-based server.



Figure 1 – An example of experimental setups belonging to the installed WEB Laboratory at the University of Kragujevac



4. **Client/Student workstation** needs very few software modifications. It can be either Windows or Linux-based. Only one or two additional software packages that can be easily installed (<5mins) are required.

3. PROCEDURE FOR WEB LABORATORY INSTALLATION

The primary intention of this approach is to allow migration of existing off-line/on-site Laboratory experiments created with Linux lab server, to online/remotely accessible setup, without additional changes to the original experiment. This is achieved via the enablement of Remote Desktop access. Though this is a very frequent approach today, multiple technologies are involved, typically requiring VPN setup and IT department support. In the follow-up approach, we tried to simplify the enablement of Remote Desktop access suitable for Web Lab experiments. Equally important is attention to a more secure approach since Lab Servers are not using public addresses and are behind the firewall /NAT. Alternative approaches can be used for this purpose, the most notable being the Apache Guacamole project (<https://guacamole.apache.org/>).

For our approach, detailed procedures with actual commands and configuration parameters are described in [2].

Lab Server configuration (Linux, Ubuntu)

Lab setup servers are affordable mini PCs with 4 GB of RAM, two cores, and 80GB of SSD storage. The same approach can be made using Raspberry PI4, whose Linux distro is also Debian-based. Linux OS used for Lab setups is Ubuntu 20.04.3 LTS. To prepare a mini PC for this role, the following modifications are required (Figure 2):

- ◆ **SSH service (open-SSH):** this allows headless operation and enables reverse-tunnelling.
- ◆ **Add Desktop environment lightdm:** This step aims to install a lightweight graphical desktop environment suitable for less demanding remote desktop operation.
- ◆ **X2GO-service:** This is an open-source Remote Desktop service based on NXv3 libs, providing quick, low-latency solutions for Linux servers. Clients are available for both Windows and Linux setups.
- ◆ **AutoSSH-service:** This service ensures a robust, always-on SSH reverse-tunnel communication link. It needs to be configured to maintain a con-

nection with a specific WebLab (aka bastion) server, using locally generated SSH keys.

- ◆ **Restrict X2GO session time:** Special cron triggered script is used to limit X2GO session time to prevent excessive logon time. This is very important if large groups of students need to share the same Lab server platform.
- ◆ **Lab server:**
 - » Generate SSH keys
 - › To be added to authorized_keys on the server, for the given weblab user account
 - » Install autoSSH service, logging into servers user account dedicated for the given weblab
 - » Install X2go server and graphical desktop environment (XFCE) to enable desktop experience
 - » As a backup, include a remotely controlled power switch.
- ◆ **WebLab Server:**
 - » Add Lab Server client SSH keys to authorized_keys
 - » Add student SSH keys to authorized_keys
 - » Reload SSHd configuration
 - » Automate this procedure via Flask/Nodejs-based dynamic content part of the Web site
 - » Add registration and reservation pages
 - » Allow linking (provide a set of pages) to other services, e.g. Moodle inclusion
- ◆ **Student/Client workstation**
 - » Install X2GO Client
 - › Configure for SSH reverse tunnel connection using client keys
 - » Stretch goal: Provide browser only access

Weblab public server configuration

This is typically VPS hosted on a Public or Private Cloud. Also, the same VPS can be used not only for this purpose as a communication node/'proxy-jump' server but also as a Web Server, Mail Server, or similar service provider. We will assume this VPS is Linux/Debian-based.

Bastion server is public-facing (i.e., host with public IP), hardened systems that serve as an entry point to systems behind a firewall or other restricted location. These systems may reside on the same LAN or multiple LANs, even behind different firewalls.

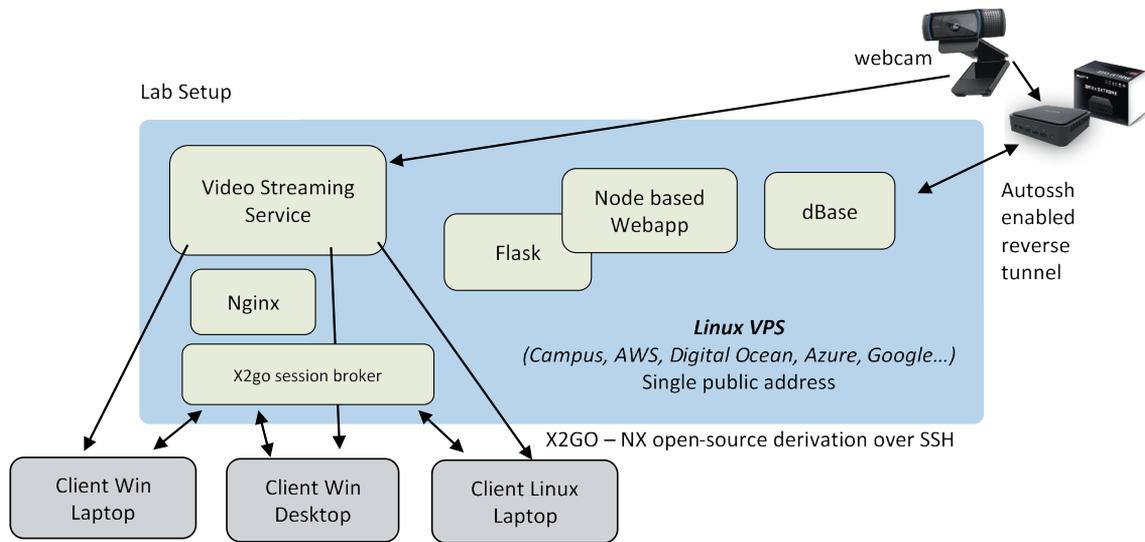


Figure 2 – The proposed concept of WEB Laboratory design and implementation

A single bastion server is sufficient for multiple groups of Lab setups and is limited by available bandwidth and compute power to less extent. Bastion server is the only Linux server accepting public SSH connections. This reduces “surface-of-attack” to a single server only. If a user wants to access a “hidden” (from public Internet) Lab setup machine, they need to connect to the bastion first and make another SSH connection from the bastion to the destination. This process is sometimes called “proxy jump” and can be automated.

In our solution, Lab setups (Figure 3) establish an SSH reverse tunnel (i.e., Lab setup initiates and maintains a connection to the Bastion server) in such a way creating an opportunity for external users (Student/Client machines) to access Lab setups not exposed to the public internet and being behind a firewall/NAT.

Typical VPS server configuration has most of the services included. Still, additional step is needed to create Lab Setup dedicated user accounts just for Lab setup connections, specially created with a limited set of permissions “weblabcpa”, “weblabcpa2”, ..., “weblabcpa5”. This is accomplished using Linux standard ‘chroot’ commands. In this way, only a subset of commands on the Bastion server can be used by these user accounts.

Further enhancement by addition of custom X2GO session-broker protocol (automatic redirection to available Lab server) and Web app for timeslot reservation will improve usability and resilience of the proposed solution.

Client/Student workstation configuration

Student setup requires installation of two packages only: X2GO client and a specific set of files with custom SSH keys, X2GO session file, and BATCH / SHELL script files.

In this way, a single click or command is sufficient to initiate a connection to remote Lab setup – only a password needs to be entered before a remote desktop connection via the Bastion server is established.

X2GO client installation

- ♦ Windows client setup: http://code.x2go.org/releases/X2GoClient_latest_mswin32-setup.exe
- ♦ Linux client setup: it is available in standard Ubuntu repositories, so the following command is sufficient: `sudo apt-get install x2goclient`

It is important to select XFCE as the session type (since this was selected in the X2GO server type)

- ♦ Installation of custom files and login procedure is described in a separate document “Manual for accessing Lab Setups over X2GO server” (in Serbian “UPUTSTVO ZA PRISTUPANJE LABORATORIJSKIM VEŽBAMA PREKO X2GO SERVERA” on <https://relab.kg.ac.rs>).

The custom package includes the following files:

- ♦ Predefined X2GO sessions (for all N setups)
- ♦ SSH public keys, different keys for each Lab setup (generated on Lab setup using SSH-keygen)
- ♦ Window/Linux script files for direct access to Lab Setup 1,2, ..., 5.



During operation, we can monitor established connections in “/var/log/auth.log” standard files

```
grep "weblabcpa" /var/log/auth.log
```

and a number of connections successfully created by Lab Setups using the following command:

```
sudo lsof -iTCP -sTCP:LISTEN | grep SSHd | grep "weblabcpa"
```

Connection is initiated from command prompt/terminal: to_cpa.bat/.sh (or to_cpa2.bat/.sh ...)

Only a previously set Lab Setup password needs to be provided at login, and Students can access the Lab setup desktop.

4. PILOT WEB LABORATORY AT UNIVERSITY OF KRAGUJEVAC USED FOR FREQUENCY RESPONSE IDENTIFICATION EXPERIMENT

Students can apply the prepared Arduino program for experiment realization to demonstrate the following learning outcomes in figure 4: real-time programming understanding, setting of sampling time, using DA and AD converters, generation of the control signal, measurement and acquisition of output signal, visualization and analysis of experimental signals, understanding frequency responses, and characteristics, and student knows how to comment on the filter properties of a system. WEB Laboratory allows access to equipment from anywhere at any time, and a much larger number of students can use the equipment 24/7. In the specific illustrative case, the use of the WEB Lab does not negatively affect any aspect of laboratory work.

5. CONCLUSION

An important and innovative performance of the presented WEB Laboratory installation procedure is that the achieved performance of locally prepared experiments is not lost (the experiment is not limited only by changing parameters but enables safe implementation of the arbitrary control-monitoring real-time program).

The solution is created with simplicity criteria in mind for end-users, i.e., students, to allow quick access, easy installation, yet similar experience as being physically present. Another goal is to enable easy conversion of existing Linux box-based lab setups, with attached lab equipment, EVMs or similar – used for offline work, with student physical presence, to remotely accessible/online setups. In addition, we opted for open-source software packages with liberal licenses only. Among various open-source remote desktop solutions, we selected X2GO as a reliable, secure, and low-latency option. All network connections are either SSH or reverse SSH-based, using a single public-facing “proxy jump” server, thus significantly reducing security concerns.

The presented web lab solution was verified over a population of 150 students during the previous semester.

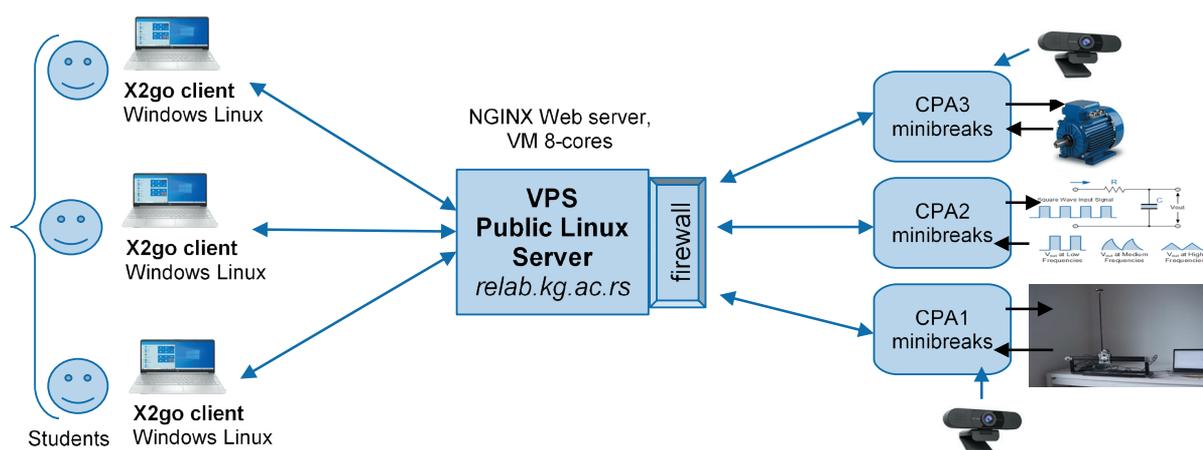


Figure 3 – Configuration pilot installation of WEB Laboratory at University of Kragujevac

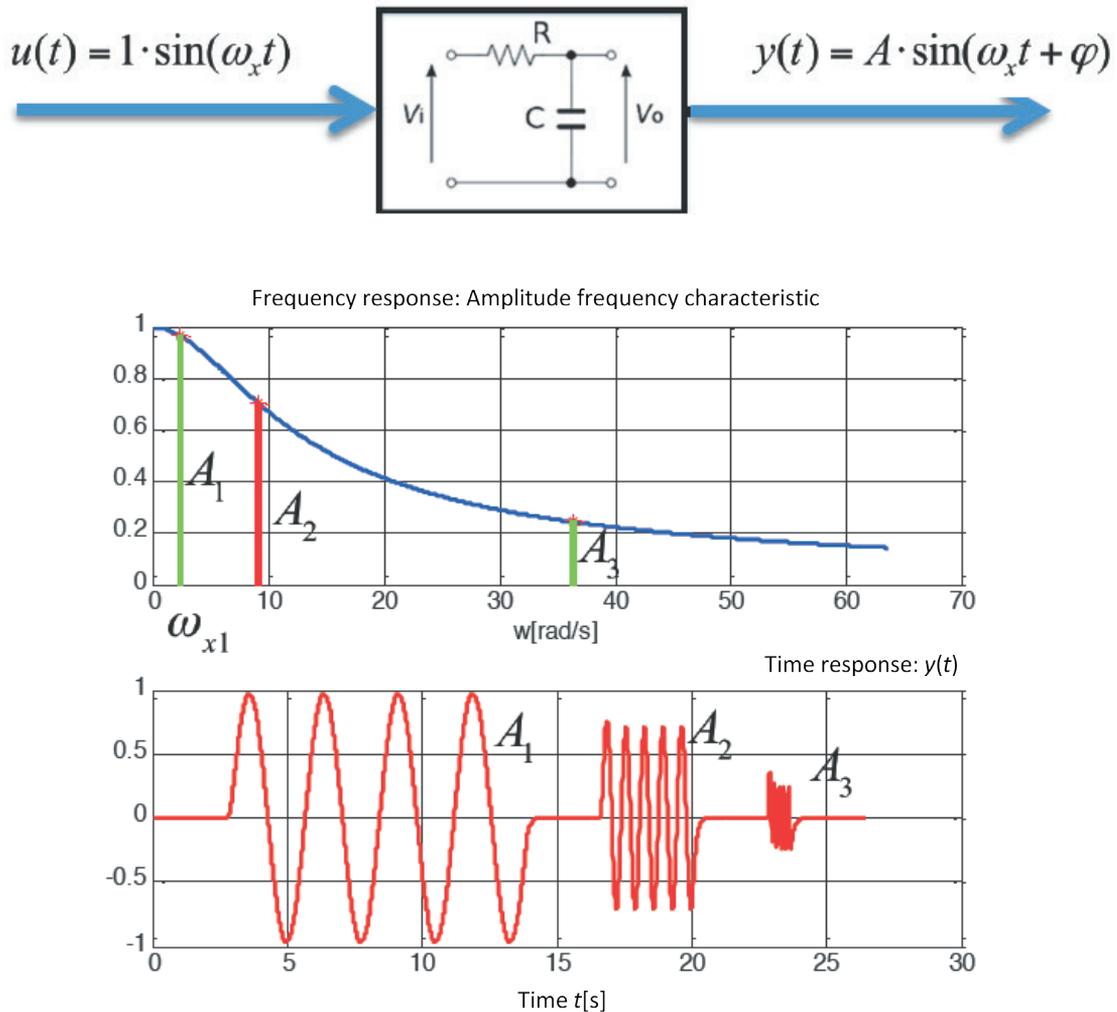


Figure 4 – Experimental laboratory setup consisting of Arduino DUE and RC circuit. The figure is illustrated experiment for the identification of frequency response

6. ACKNOWLEDGMENTS

This work has been funded by the ERASMUS+ KA226 project 2020-1-RS01-KA226-HE-094550 "Repository of Open Educational Resources for Laboratory Support in Engineering and Natural Science – RELAB", which is gratefully acknowledged.

7. REFERENCES

- [1] CogniPix, *WEB Laboratory Tutorial of Technical Design and Implementation*: "Enable conversion of off-line Lab setups to remotely accessible WebLabs - PART1/3", version 0.1, 2021, [Online] relab.kg.ac.rs.
- [2] Đ. Seničić, *WEB-Laboratory: design, implementation, maintenance* "Enable conversion of off-line Lab setups to remotely accessible WebLabs", version 1.0, 2022. [Online] relab.kg.ac.rs.
- [3] S. Dormido, J. Chac, J. Saenz, L. de la Torre, "Open virtual and remote labs for Control Education", *IEEE Report*, UNED, Madrid, Spain, 2018.
- [4] R. Heradio, L.de la Torre, S. Dormido, L. Cubillo. "Virtual and remote labs in control education A survey" *Annual Reviews in Control*. no. 42: pp. 1-10, 2016.



- [5] M. Kalúz, L. Čirka, R. Valo, M. Fikar, " ArPi Lab A Low-cost Remote Laboratory for Control Education " IFAC Proceedings Volumes, vol. 47, no. 3, pp. 9057-9062, 2014, ISSN 1474-6670.
- [6] M. Matijevic, M. Nedeljkovic, Dj. Cantrak, N. Jovic" Remote labs and problem-oriented engineering education" *IEEE Global Engineering Education Conference (EDUCON)*, Athens, Greece. pp. 1390 –1395. April 25-28, 2017.
- [7] S. B. Concari, S. T. Marcisio "The Remote Laboratory as a Teaching Resource in the Scientific and Technological Training", *Creative Education*, vol. 4, no. 10A, October 2013.
- [8] S. H. Sung, C. L. Huang, C. Xie "Enhancing distance learning of science—Impacts of remote labs 2.0 on students' behavioural and cognitive engagement", *Journal of Computer Assisted Learning*, August 2021.
- [9] G. Tsochev, "Research on Web Applications for Remote Laboratory Exercises on Computer Networks" 2021 International Conference on Information Technologies (InfoTech), September 16-17, 2021, Vrana, Bulgaria.
- [10] M. Bochicchio, "Hands-On Remote Labs: Collaborative Web Laboratories as a Case Study for IT Engineering Classes", *IEEE Transactions on Learning Technologies*, vol. 2, no. 4, October-December 2009.