Implementation of the System of Remote Laboratories in the Area of Mechatronics for Learning without Human Supervision

Guido Soprano Machado
CETELI-PPGEE-UFAM
Federal University of Amazonas
Manaus – AM, Brazil
0000-0002-2365-9735

Marenice Melo de Carvalho
CETELI-PPGEE-UFAM
Federal University of Amazonas
Manaus – AM, Brazil
0000-0002-0309-7333

Wollace de Souza Picanço
PPGEE-UFAM and FMF Wyden
Federal University of Amazonas
Manaus – AM, Brazil
0000-0002-1455-6689

Florindo Antonio de Carvalho Ayres Jr
CETELI-PPGEE-UFAM
Federal University of Amazonas
Manaus – AM, Brazil
0000-0002-0266-1019

Renan Landau Paiva de Medeiros
CETELI-PPGEE-UFAM
Federal University of Amazonas
Manaus – AM, Brazil
0000-0002-1645-2736

Vicente Ferreira de Lucena Jr
CETELI-PPGEE-UFAM
Federal University of Amazonas
Manaus – AM, Brazil
0000-0001-9864-2850

***Abstract* — Remote Access Labs (RAL) are hardware and software tools that allow students to remotely operate actual equipment in physical labs at schools or universities. This work presents the development of a system for students to use remote access laboratories in the mechatronics area without human supervision. This system verifies the activities the registered labs can offer through algorithms that use digital twin technology to learn the actual or virtual industrial plants' operations. Registered laboratories are available for the practice of experiments where students can automatically follow the correction of activities through virtualization generated so that the student can understand their actions. With this system, students can monitor their learning in real time and receive evidence of successes and errors through gamification of their performance.**

Keywords— Mechatronic Laboratories, Remote Laboratories, Distant Learning Experiments, Online Students, Digital Twin.

I. INTRODUCTION

During the COVID-19 pandemic, conducting practical face-to-face activities in schools or universities has become increasingly complex, especially for students in the field of mechatronics engineering who need machines and equipment to accomplish tasks for their learning [1,2,3,4].

In this context, the need to create new tools to help students in their teaching and learning process is evident [5,6]. One of these tools is the Remote Access Laboratories (RAL). RALs have become essential for students to have practical experiences using current industry technologies [7,8,9]. RALs allow students to have contact with equipment shared by large companies and universities that wish to contribute to the world education process. Assume that the students have the minimum competence to use these laboratories without posing risks to the equipment that has been shared.

Therefore, the opportunity arises to offer a system where companies or universities can share equipment or physical laboratories that are fundamental for teaching students. Some universities currently offer platforms with RALs, such as Stanford University, which shares experiences within its platform called iLabs [5,10].

Due to this opportunity, a system capable of receiving didactic physical plants from the mechatronics area was implemented to be made available to students who wish to conduct experiments on these plants. This system has a three-element architecture: management platform, lab learning, and intelligent learning. The management platform is responsible for storing the necessary data to communicate with the didactic plan and monitor the student. Lab learning uses the digital twin principle to obtain the functioning data of the task passed on by the teacher through the input and output states of the didactic plant. Intelligent learning performs automatic virtualization so that students can monitor their performance.

With this system, the professor will be able to register the laboratory of his university by inserting the activities that the students can perform. As an example, in this article, programming activity in PLC will be implemented in a didactic plant of the MPS line (Module Production System) to be assessed as a student experiment.

The work is organized as follows: Section II presents the problem solved by the system. Section III shows the components developed and implemented in the system. Section IV describes the experiment to evaluate the system method, and section V presents the conclusions obtained from the work.

II. PROBLEM DESCRIPTION

In times of pandemic, we faced several problems in executing the theoretical classes and even larger ones in realizing the practical activities. The theoretical activities were conducted through web-based communication media, and the main problem was adapting the physical classroom to a virtual classroom. The execution of the practical exercises at a distance was much more challenging, and we show three examples in the following sections. The software and hardware choices were based on our group's experience developing new teaching approaches with international institutions [11] and local industries [12]. When discussing the development methodology, we adopt agile development strategies [13] based on SCRUM [14] and, whenever possible, develop reusable components [15].

Digital Twin is a technology that transforms a physical system into a digital model. It is possible to update and evolve the digital model and apply it to the physical system [16,17]. To exemplify the problem of this work, one can think of the following situation: a student wants to verify if his activity is working correctly or not. It depends on the student's didactic plan functioning, but if the student is studying at 2 A.M., who will correct this activity? Or who will help this student to solve this activity?

In this context, the problem of this work can be formalized by the following question: how can remote access laboratories be used without human intervention?

The idea of using the digital twin technology produces a system that helps the student to be able to conduct his experiment and perform the correction of the functioning of the student's activity without human intervention.

III. Proposed System

The proposed system for the student to use RALs without human intervention will consist of the following elements that form its architecture: management platform, lab learning, and intelligent learning. Figure 1 shows the structure of this system, which was designed to help students use RALs.

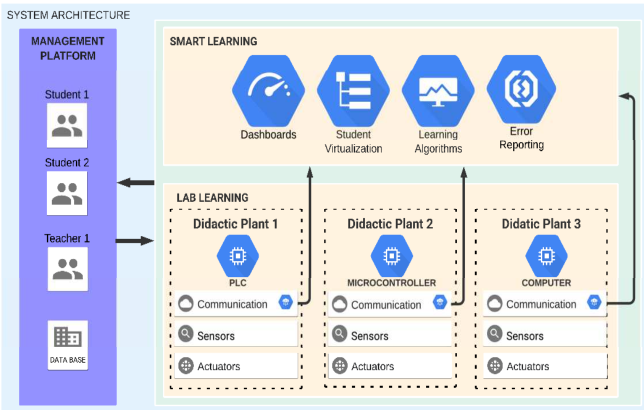

Fig. 1. Overview of the proposed system.

A. *Management Platform*

The Management Platform was developed to store student, teacher, and RAL data. This platform is a website that presents some forms for filling in these data. For the student or teacher to have access to the platform, it is necessary to register a login and password. From this register, it will be possible to store the student's performance, show their evolution, and allow teachers to register the experiments of their institutions in the mechatronics area to be shared in the system.

The most critical data to be registered are the RALs because, with these stored data, the system will be able to communicate with the controller of the didactic plant that the experiment has. Every experiment must have a communication IP/Port (link) and the inputs and outputs of the didactic plant registered so that the system can later learn how the plant works.

In addition to the registrations, this platform will allow students to browse the other system elements, manage the RALs to check which experiments are available to the student, and check which experiments have already been learned by smart learning to keep the student's progress through gamification.

The Management Platform has a database implemented in MySQL to store all the system information. Figure 2 shows the modeling of this database with its main Table and the information to link with the secondary tables. One of these secondary tables is called an experiment, as it carries all the information needed for the experiment to work correctly during the student's practice.

The information from the experiment table is initially loaded by the teacher's register of the school where the RAL is located. When the experiment is carried out, the information is updated according to the functioning of the didactic plant.

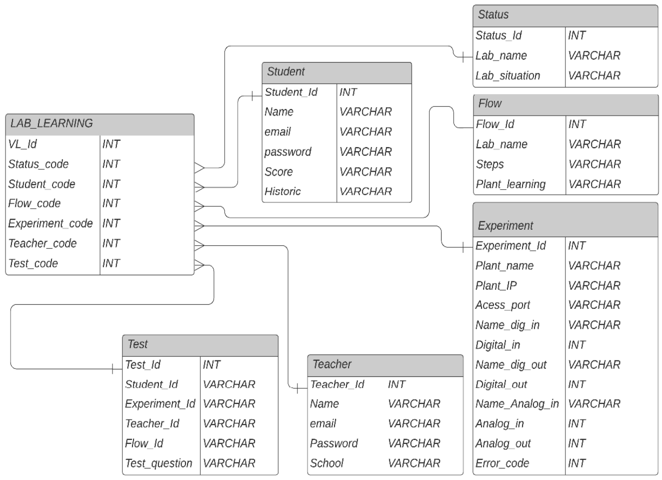

Fig. 2. DataBase of Management Platform.

B. *Lab Learning*

To develop software that could converse in all these protocols, a flow was implemented in the Node-RED platform with the functionality of a didactic plant server, where the acquisition of data from the different protocols is conducted and sent via MySQL commands to the database. Fig. 3 shows the clipping of the data reception stream via the MQQT and ModBus protocol developed.

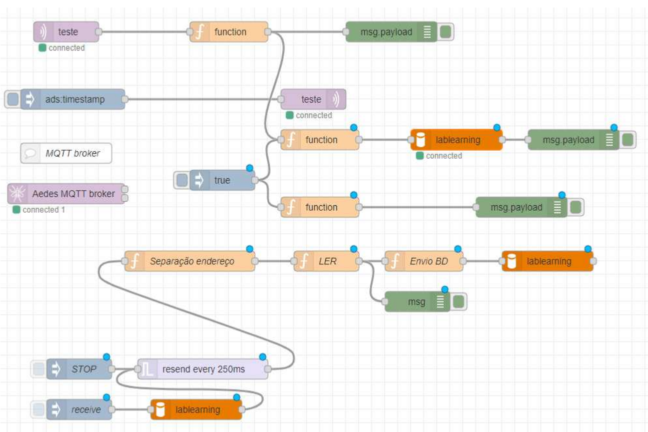

Fig. 3. Flow development in the Node-RED platform.

C. *Smart Learning*

Smart learning is a set of applications responsible for generating dashboards, virtualizing experiments, and running algorithms that can learn how the didactic plant works and correct and detect errors in experiments performed by students. Therefore, smart learning will be divided into three modules:

- *Virtual Setup:*

The virtual setup is the name given to the virtualization module. This module aims to graphically show the state of the student's experiment inputs and outputs in real time, allowing him to follow what is happening with the functioning of his task. This module monitors the Management Platform database every 250 ms and updates the student's screen through some screens. The first virtualization screen is shown in Fig. 4, where one of the automatically generated panels appears for the student to follow his activity. In this panel, squares are used to indicate the status of the inputs, and triangles show the status of the outputs.

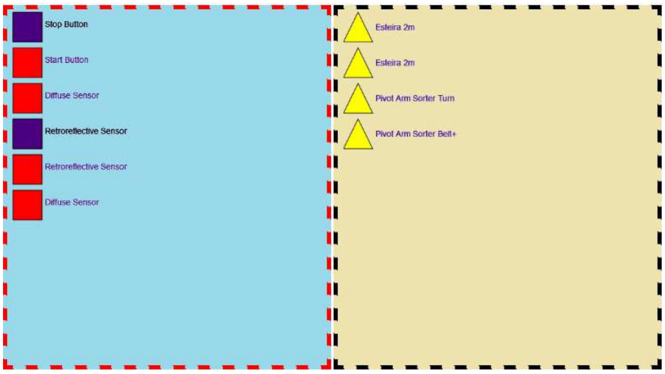

Fig. 4. Dashboard of the virtual setup module.

This monitoring is available to the student when executing his task solution. Fig. 5 shows the states of inputs and outputs differentiated by colors to identify the elements' moment of activation and deactivation.

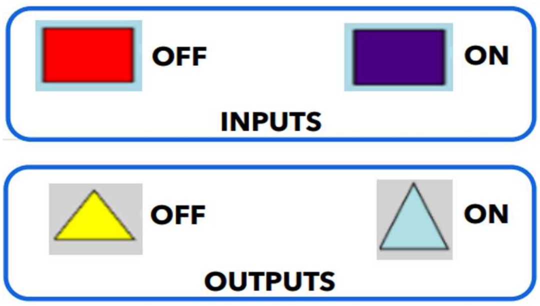

Fig. 5. Color status to identify inputs and outputs.

When performing his task, the second screen offered to the student is the status of inputs and outputs in the form of a list, with the names on one side of the sensors and actuators. On the other hand, the status of 1 for on or 0 for turned off. The third screen offered in this module to the student is the 3D virtualization through the Unity engine using geometric shapes already established to design the flow of the didactic experiment in the virtual environment. Nevertheless, this virtualization can only occur after the Learning Module has experimented with the didactic plan.

- *Learning Module:*

This module is responsible for learning the correct functioning of the experiment. Nevertheless, for this module to work, the teacher must conduct the didactic investigation at least once according to the expected solution of the student's task. When the teacher executes the experiment, the learn didactic plan button must be clicked so that the software responsible for the learning begins its execution. When the software of this module starts, its execution must monitor the states of inputs and outputs, verifying each change that occurred in the experiment and the time for each change of state, as well as the activation time of each element involved in the process. Table 1 presents the inputs and outputs generated from the execution of a didactic task. With the operation of the experiment, the changes that must be stored occur, where each change of state is called a step. The first step is inserted in the software as a reference because it will start a loop to compare the previous step with the next step. If there is a difference, the current step is stored, and this loop is terminated when the module's algorithm finds a repeated process. For example, in Table I, Step 0 and Step 4 are repeated, and from Step 4, the algorithm checks if the process is repeating itself. The times of each step are also taken from this learning to create warning and alarm signals. Steps and Step times are stored in the Management Platform database until the student tests his task; then, the Virtual Correction module uses these data to answer the student.

TABLE I. Test of elementary conditions for the Learning Module.

List I/O	Standard Step	Step 0	Step 1	Step 2	Step 3	Step 4
I0 – BT_STOP	1 (ON)	1 (ON)	1 (ON)	1 (ON)	1 (ON)	1 (ON)
I1 – BT_START	1 (ON)	0 (OFF)	1 (ON)	0 (OFF)	0 (OFF)	0 (OFF)
I2 – SENSOR_1	1 (ON)	0 (OFF)	0 (OFF)	1 (ON)	0 (OFF)	0 (OFF)
I3 – SENSOR_2	1 (ON)	0 (OFF)	0 (OFF)	0 (OFF)	1 (ON)	0 (OFF)
Q0 – BELT_0	1 (ON)	0 (OFF)	0 (OFF)	1 (ON)	0 (OFF)	0 (OFF)
Q1 – BELT_1	1 (ON)	0 (OFF)	0 (OFF)	0 (OFF)	1 (ON)	0 (OFF)
Step Timer	---	ST0	ST1	ST2	ST3	ST4

- *Virtual Correction:*

The virtual correction module is responsible for the following functionalities:

A) Follow experiment: When starting their investigation, the student can switch between some screen options to monitor the functioning of their programming. In addition to the dashboards and virtualization, it is possible to follow the experiment through the Steps view, where the algorithm of this module takes the learning model performed by the Learning module and compares it with the operation of the current experiment, checking if there are differences in flow and time. If there are differences, a circle that follows the process turns blue and yellow; if there is a difference in flow, the circle is no longer blue and becomes red. Fig. 6 presents the screen where the student can check if his experiment is in the correct flow or not according to the color of the circle that passes through the steps in real time.

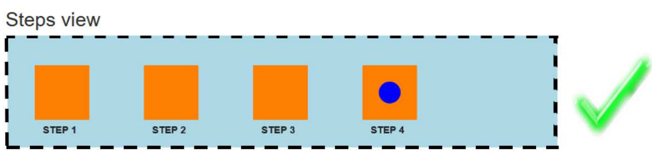

Fig. 6 Steps View screen.

B) RAL Camera: a way that the student can follow the operation of the experiment in real time is the camera located in the RAL. This camera is enabled for the student when he accesses the desired or indicated investigation.

C) Score of Experiment: A score is created for each step according to the number of Steps automatically generated by the Learning module. At the end of the experiment, the student has his score available.

IV. EXPERIMENTAL PROCEDURES

The scenario created to evaluate this system followed the following steps:

- **1st Step:** Due to COVID-19, access to physical laboratories at the university was difficult, so an experiment was developed in the factory IO software to evaluate the system. Fig. 7 presents a scenario with two belts, two buttons, three sensors, and a turning belt. This virtual scenario was developed to test the functionality of an activity performed by the teacher.

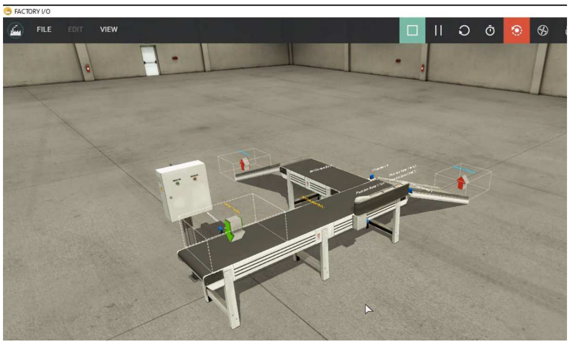

Fig. 7 Virtual Experiment of the System.

- **2nd step:** the experiment with the communication information and controller inputs and outputs must be registered to execute it. According to the activity developed by the teacher, it is necessary to learn this scenario. The teacher must run the correct program and click on the button that starts learning the plant. Remember that to access this record of experiments, it is necessary to have a history of the teacher in the system. After learning the activity, it is available for student use.
- **3rd step:** at this point, it is necessary for the student to develop a program using the teacher's guidelines to fulfill the task in question. Two programs were created in Ladder language with the SoMachine software, a program with correct and another incorrectly functioning. Both will serve to evaluate this system. After developing the programs, the student must carry out his activity. For the student to be able to perform this experiment, it is necessary to use the communication API on their computer.
- **4th step:** When the student is running their experiment, they can use the virtualization environment, dashboards, and the camera to keep track of what is happening, and they can use the Steps View screen to check the flow of their program right or wrong. Fig. 8 shows the incorrectly designed program to test when the student had an error in the flow of their activity program.

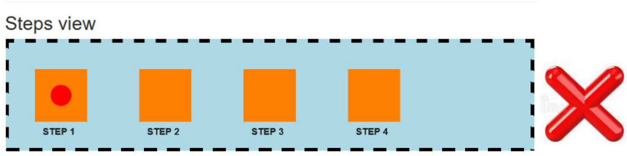

Fig. 8 Steps View screen with the wrong flow.

V. CONCLUSIONS

The system proposed in this article achieved its objective, as it developed tools that give access to RALs and correct experiments and tests inserted in the system by teachers.

These developed tools help to establish an architecture capable of assisting the students in running experiments without human intervention to correct their practical activities. This was made possible by digital twin technology and algorithms that could learn an investigation and use that learning to correct student experiments.

The next stage of development is to conduct tests on physical didactic plans, suggest practical activities for each student profile, and implement this architecture in real classrooms in new laboratories at universities or technical schools.

ACKNOWLEDGMENT

This research, carried out within the scope of the Samsung-UFAM Project for Education and Research (SUPER), according to Article 39 of Decree n° 10.521/2020 (SUFRAMA), was funded by Samsung Electronics of Amazonia Ltda., under the terms of Federal Law n° 8.387/1991, through agreement 001/2020, signed with Federal University of Amazonas and FAEPI, Brazil. We also would like to acknowledge the financial support from CAPES, CNPq, FAPEAM, and Flex Institute of Technology of the Amazon for providing the equipment for the virtualization remote of this research.

REFERENCES

[1] K. K. Naji, X. Du, F. Tarlochan, U. Ebead, M. A. Hasan, and A. K. Al-Ali, "Engineering students' readiness to transition to emergency online learning in response to COVID-19: Case of Qatar," EURASIA J Math.

[2] F. Guc, J. Viola and Y. Chen. "Digital Twins Enabled Remote Laboratory Learning Experience for Mechatronics Education". IEEE 1st International Conference on Digital Twin and Parallel intelligence, 2021.

[3] Yassen Gorbounov and Hao Chen. "Low-Cost Remote Laboratory for Cyber-Physical Experiments". IEEE 29th National Conference with International Participation, October 28-29, 2021

[4] N Kapilan, P Vidhya, and Xiao-Zhi Gao. Virtual laboratory: A boon to the mechanical engineering education during covid-19 pandemic. Higher Education for the Future, page 2347631120970757, 2020.

[5] Zaman, Thorben and Hesselink, "iLabs as an online laboratory platform: A case study at Stanford University during the COVID-19 Pandemic". IEEE Global Engineering Education Conference (EDUCON), pp. 1615, 2021.

[6] P. Orduña. et al. "An Extensible Architecture for the Integration of Remote and Virtual Laboratories in Public Learning Tools". IEEE REVISTA IBEROAMERICANA DE TECNOLOGIAS DEL APRENDIZAJE, VOL. 10, NO. 4, NOVEMBER 2015.

[7] O. Zine, M. Errouha, O. Zamzoum, A. Derouich, and A. Talbi, "SEITIRMLab: A costless and effective remote measurement laboratory in electrical engineering," International journal of electrical engineering & education, vol. 56, no. 1, pp. 3–23, 2019.

[8] AGUSTÍN, Caminero C. et al. “Virtual Remote Laboratories Management System (Tutores): Using Cloud Computing to Acquire University Practical Skills”. IEEE TRANSACTIONS ON LEARNING TECHNOLOGIES, VOL. 9, NO. 2, APRIL-JUNE 2016.

[9] S. Verslype, L. Buysse, J. Peuteman, D. Pissoort, and J. Boydens, “Remote laboratory setup for software verification in an embedded systems and mechatronics course,” in 2020 XXIX International Scientific.

[10] Lars Thorben Neustock, George K Herring, and Lambertus Hesselink. Remote experimentation with massively scalable online laboratories. In Online Engineering & Internet of Things, pages 258–265. Springer, 2018.

[11] N. Jazdi, P. Göhner, A. Brito, V.F. de Lucena, “A Germany-Brazil Experience Report on Teaching Software Engineering for Electrical Engineering Undergraduate Students,” 19th Conference on Software Engineering Education & Training (CSEET’06), 2006, pp. 69-76, doi: 10.1109/CSEET.2006.6.

[12] V. Ferreira de Lucena, J. P. de Queiroz-Neto, I. B. Benchimol, A. Pereira Mendonca, V. Romao da Silva, and M. Ferreira Filho, "Teaching software engineering for embedded systems: an experience report from the Manaus research and development pole," 2007 37th Annual Frontiers In Education Conference - Global Engineering: Knowledge Without Borders, Opportunities Without Passports, 2007, pp. S4C-3-S4C-8, doi: 10.1109/FIE.2007.4417966.

[13] L. Cordeiro, R. Barreto, R. Barcelos, M. Oliveira, V. Lucena, and P. Maciel, “Agile Development Methodology for Embedded Systems: A Platform-Based Design Approach,” 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS’07), 2007, pp. 195-202, doi: 10.1109/ECBS.2007.16.

[14] L. Pinto et al., “On the use of SCRUM for the management of practical projects in graduate courses,” 2009 39th IEEE Frontiers in Education Conference, 2009, pp. 1-6, doi: 10.1109/FIE.2009.5350404.

[15] R.E.V.S. Rosa and V.F. Lucena, “Smart composition of reusable software components in mobile application product lines,” In Proceedings of the 2nd Int. Workshop on Product Line Approaches in Software Engineering (PLEASE ‘11). ACM, New York, NY, USA, 45–49. DOI:https://doi.org/10.1145/1985484.1985496

[16] J. Viola and Y. Chen. “Parallel Self Optimizing Control Framework for Digital Twin Enabled Smart Control Engineering”. IEEE 1st International Conference on Digital Twin and Parallel intelligence, 2021.

[17] Digital Twins for Students: Approaches, Advantages and Novelty J. David, A. Lobov and M. Lanz, “Learning Experiences Involving Digital Twins,” IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society, Washington, DC, 2018, pp. 3681-3686, doi: 10.1109/IECON.2018.8591460.