Next Steps in Supporting More Students in MOOL for Control Education

Christophe Salzmann*, Denis Gillet**

**EPFL, Automatic Control Lab., Station 9, CH-1015 Lausanne, Switzerland (e-mail: christophe.salzmann@epfl.ch)*
***EPFL, REACT group, Station 11, CH-1015 Lausanne, Switzerland (e-mail: denis.gillet@epfl.ch)*

Abstract: This paper explores the possible steps to support a larger number of students in a Massive Open Online Lab (MOOL) dedicated to control education. The current solutions to support a ratio greater than 1:1 (1 student accessing 1 setup) is either to multiply the setups or to limit the time the students access the setup. This paper first describes the current EPFL MOOL infrastructure and its associate MOOC for control offered to bachelor engineering students. It then provides analysis of students access and activity. Finally based on the analysis, various options to support more students are explored.

Keywords: MOOC, Massive Open Online Lab, Remote Experimentation, Remote Labs, Control Education.

1. INTRODUCTION

The Automatic Control Lab at Ecole Polytechnique Fédérale de Lausanne (EPFL) proposes a MOOC for the first course on control to mechanical, micro, and electrical engineering students. This MOOC differs from standard MOOC since it focuses on the hands-on sessions with physical lab equipment accessed at distance that are complementing classroom activities.

Most MOOCs propose videos as the main source of information and the knowledge evaluation is performed with the help of quizzes or graded assignments. Some MOOCs propose online tools with simulation in de la Croix (2014), O'Malley (2015) or for batch access in Rochat (2014), but very few propose real-time hands-on experimentation using remote laboratories, i.e. physical setups operated at distance in Hossain (2017), Garcia et al.(2014), Salzmann et al.(2016).

While videos delivery and simulations can be easily duplicated at wish, remote lab remains critical resources which cannot be concurrently shared. This implies that one student accesses one and only one lab experiment at a time, this is a major challenge when *Massive* access is expected.

This paper first describes the current MOOC and the associated Massive Open Online Lab (MOOL). Then, it presents the current scheme implemented on the MOOL to support a ratio of 10:1 (10 users for 1 lab). The current configuration has been in place for two years and successfully support the cohort of EPFL students. The next step is to open the MOOC and the associated MOOL to a wider number of users. Measurements and questionnaires have been analyzed to propose options to support an audience with an additional order of magnitude.

1.1 MOOC-TP

This section summarizes the MOOC infrastructure that has been described in Salzmann & all (2016, 2017a).

For more than twenty years, the Automatic Control Lab at EPFL offers its students the ability to remotely connect to the lab equipment for the Control Systems course, Gillet et al. (1997). Three years ago, a major effort has been put into deploying the remote lab on wider scale and enhancing it with more material. The MOOC form fitted well our goals for the new Control Systems hand-on session Salzmann et al. (2017b).

The Automatic Control course is given to 3rd year bachelor students. While the course is given face-to-face in a large auditorium, the hand-on sessions are handled via a MOOC. In addition, computer and pen & paper exercise sessions are also proposed. In the future, these exercise sessions will be also integrated into the MOOC.

For EPFL students, the MOOC is given as a flipped classroom, where students are expected to watch the video and perform initial hands-on online experimentation at home. Then, during their reserved time slots, they come to the lab facility to ask questions to the teaching assistants and complete any task of the hands-on module they prefer to conduct in presence. There is no obligation to come to the lab, as the MOOC is designed to be self-pace and that awareness is provided to the users. Students are evaluated at the end of the semester in a written exam that covers the ex-cathedra course, the exercises and the MOOC.

The MOOC is given over one semester; it is made of eight modules. Each module covers a particular aspect of the control course, for example: system identification, Ziegler/Nichols, Loop-Shaping, Rousted Margin, etc.

Each module is split into short phases (few minutes each) which follows a common structure: a short video describing the experiment, a set of hands-on exercises using the web client to control the remote lab integrated in the MOOC, data pre/post-processing using interactive simulation tools, a set of numerical questions to validate user's finding and another set of open questions based on the user observations to evaluate user's learning outcome.

Figure 1 shows a typical scenario with *i)* the introduction video, which provides a theoretical recall (ex. closed loop) and explain the steps to be performed in the current session, *ii)* the web client which allows to perform a step response on the real equipment and save the measurements, *iii)* the numerical questions that are automatically validated by the MOOC, *iv)* the simulation tool made with SysquakeJS to process and analyze data recorded by the web client. As an example, the temporal fit tool which enables students to find the plant characteristics by superposing a 1st order model on the top of the measurements. *v)* Then, based on the identified model they design a model matching controller and test it on the real system.

There is a flow of information between various steps of a given module. This information, stored in a database within the MOOL infrastructure, is available at each step providing continuity of interaction within the MOOC tabs.

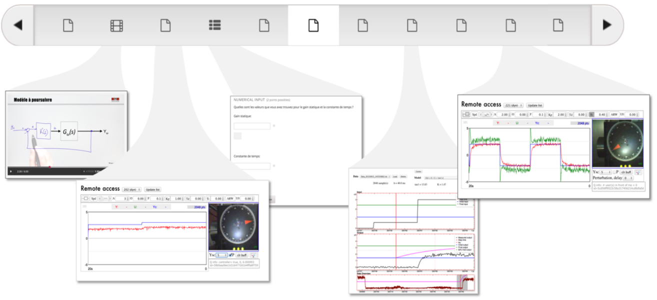

Fig. 1. The MOOC-TP interface for a given module

The modules are specially crafted to cover one theoretical aspect at a time. In order to maximize the overall interaction time with the physical lab and to minimize the user waiting time, the time to conduct each experimentation is limited to some predefined values. If a user needs more time to complete an experiment s/he will have to go back in the waiting queue.

1.2 *Massive Open Online Lab (MOOL) infrastructure*

The MOOC is powered by OPENedX running on EPFL server. Hosting the MOOC on OPENedX has been chosen for technical reasons, the main one being the possibility to easily integrate external code as LTI module. The second one is the EPFL OPENedX supporting team proximity.

The complete laboratory infrastructure encompassing the physical equipment (23 x setups: electrical drive + server), the load balancer, the database and the LTI cgi server forms the Massive Open Online Lab (MOOL) (fig.2).

The web app client that enables access to the remote setup can be “plugged” in the MOOC as external LTI module (top left of Fig.2). The LTI interface enables limited but sufficient information exchange between the MOOL and the MOOC, i.e. UserID and context. This information is encrypted to guaranty that the communication has not been compromised.

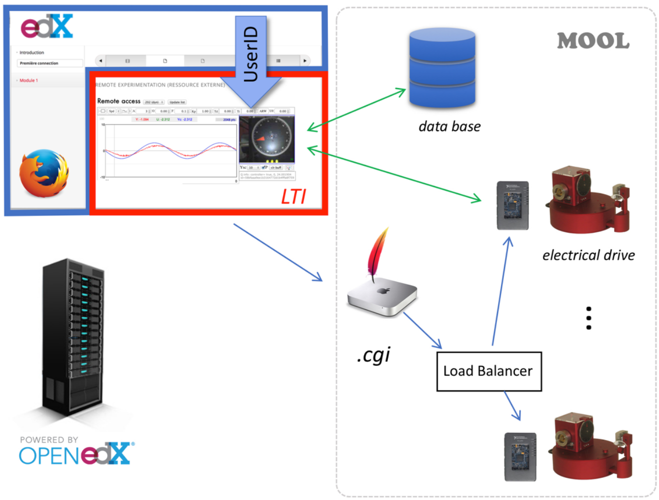

Fig. 2. MOOL infrastructure: the client web app. (red part) embedded as LTI code in the MOOC (the edx userID is passed to the MOOL), the load balancer selects the setup (electrical drive + server) with the shortest waiting queue. The web app directly communicates with the experimentation server.

2. ANALYTICS SOURCES

2.1 Analytics at the MOOL level

Analytics related to the lab usage patterns are needed at the MOOL level to manage its infrastructure. Typically, each setup reports its availability to the load balancer allowing an equal repartition of the user load. Each user connecting via the web app is redirected to the electrical drive server (setup) that has the shortest waiting queue. Other information such list of connected IP addresses can also be recorded. With this info the lab owner can see his lab global usage and load.

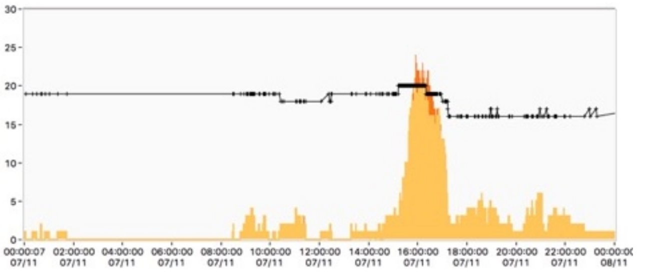

Fig 3. Typical lab usage, black line: # available setups, orange bar: # of connected users, red portion # waiting users.

Figure 3 shows the typical lab usage for an afternoon with a reserved time slots (14h15-17h00). The black line represents the number of currently available setups, the orange bars indicate the total number of connected users for a given time.

The red portion shows how many of them are waiting. We can see that the lab is slightly saturated during the reserved time. This indicates that rare users have to wait 1 round (max 90 sec) before interacting with the setup.

Note that waiting users can sees in the client web app what peers are experimenting. Outside of the reserved time slot, the

MOOL is also used, especially at night, but not from the lab premise.

At the MOOL level, clients are identified via their IP address. If needed, the IP address can be linked to the MOOC UserID. IP accesses are recorder by the MOOL infrastructure. This enable withe/black lists IP filtering. In addition, the location of the IP address can be estimated. Measurements show that only 10.7% of the total connections were made from within the lab premise. All the other connections are performed from the university wired network (8.5%) or Wifi/VPN (61.2%) or from outside the university premise (19.6%). Privacy regulation does not allow us to learn about the VPN locations, thus the users can either be at home or within the university premise.

In addition to reporting its availability to the load balancer, each setup reports the actions performed in the cleint web app. (red part of Fig. 2) For example, each time the user modifies the value of Kp (of the PID controller), an entry containing a timestamp, the MOOC userID and MOOC context, and additional parameters are added to the activity stream for that setup. This list of actions is recorded for each server on the MOOL infrastructure.

2.2 Measurements at the MOOC level

The MOOC is hosted on OPENedX which provides various tools to track user activity and provide analytics. Figure 4 shows the number of learner who accessed the MOOC. This figure is similar to what the MOOL is seeing (Fig 3.) with less granularity. Other statistics derived from user profiles can also be extracted.

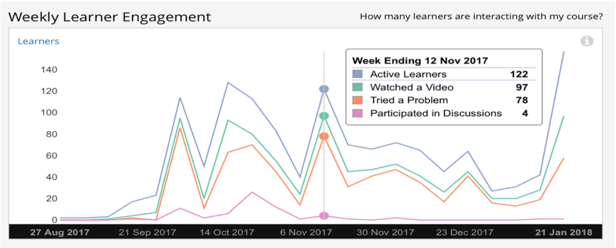

Fig 4. MOOC # active learner per week

3. ANALYTICS, WHAT FOR?

The previous sections showed some of the available analytics. While some of them are needed for example to distribute user load among the setups, other are needed to ensure a proper MOOL operation or to provide feedback to the teacher about his/her class progress.

3.1 Analytics goals

As a lab owner, the main concern is the user satisfaction at the technical level. Two indicators are the overall availability of the MOOL and the user average waiting time.

This information can be directly seen in Figure 4 which represents a typical lab usage with a reserved time slot. Over the semester, the MOOL was never saturated outside the reserved time slot and very rarely saturated during the reserved time slot. There were more than 200 students during the semester. The current MOOC contains 23 tasks requiring MOOL access. Each of this task lasts between 30 s and 3 min. On average there is 20 setups available in the MOOL. We can see that with this configuration a ratio of 10:1 is achievable without impacting user waiting time.

Supporting more students concurrently is the next challenge. While the current MOOC is accessible to everyone who register, it is not widely advertised. The aim is to run the MOOC on the main edX instance where we could expect an order of magnitude more concurrent users. Knowing that the setups in the MOOL will not be expended at will, more insights about the MOOC usage need to found to enable a larger number of concurrent users.

3.2 Maximizing the MOOL usage

A MOOC module is split in various tasks. Some of these tasks required MOOL access (limited by the number of setups) while other tasks such as simulation or video can be duplicated without limit. To maximize the MOOL usage, the load needs to be spread over time. Various options to achieve this goal are presented hereafter.

3.3 Inform users

The first measure to maximize the MOOL usage is to inform users about the current load and ideally a prediction of the future load based on learned pattern. With this information the user can decide to come at a later time, when the MOOL load is lower.

3.4 Dynamically adapt experimentation time

The second option is to dynamically adapt the time allowed for each experimentation tasks. Currently the experiment time is hard-coded. This value could be set dynamically, based on activity measurements average. The initial measurements suggest that the experiment time was under evaluated by a factor of two, which means that the users need to run the task twice to complete it. This measurement is subject to caution since we only measure the average time users spend on a given module and compare it to the sum of hard-coded experimentation time for the given module.

Likewise, if there is some time left to complete the experiment but no actions are performed by the user, the next waiting user could get his turn earlier. The difficulty comes with the detection of user end of actions: is the mouse movements the indicator or does it exist another indicator? The user might be observing the live measurements without moving the mouse. Switching to another tab in the MOOC should be the sign that the current task is completed. We observed that while running the experiment, some users wanted to get information from another tab and open two windows, one with each tab, thus defeating our potential indicator.

Securing the OPENedX server via https forced non-SSL LTI modules to be open in a separate window. As a result, the client web app opens in a separate window. It remains open for an extended period of time, until the user decides to eventually close it. Thus, the MOOL load increased significantly (double) compared to the previous year and reached the point where users had to take turns to access it (Fig. 5). Students noticed this new behavior. We informed them that the web app window should be manually closed after completing a task. Students expected an automatic mechanism to end the task. When informally asked about the indicator for the end of the current experiment, no consensus could be reached among students.

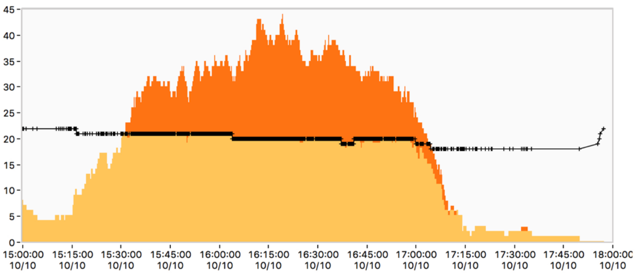

Fig. 5. Securing the MOOC access induced saturation in the lab usage.

There is a trade-off between experimenting for an extended period of time vs. waiting for the same extended period of time (or multiple of). Do users prefer to wait longer and experiment for a longer of period of time or is it the opposite? To maximize the MOOL load, the initial choice was to have small periods of time.

3.5 Tasks shuffling

A module is formed of sets of tasks. For example, a set of tasks aims at identifying the electrical drive parameters in position and another set of tasks is to identify the same plant parameters in speed. Both these sets of tasks follow the same pattern: read instructions, perform experiment, save measurement, analyze measurement to get parameters, validate parameters in the MOOC. These two set of tasks could be performed in any order and thus be exchanged. They may also carry different experimentation time. In case of high MOOL load a scheduler may exchange the two set of tasks. This would require re-ordering tasks at the MOOC level which is not currently possible.

We analyze the experimentation tasks for each module to see if a pattern (sequence of actions) exists in student activity and if it influences student performance, i.e. grade. The analysis did not show any statistical significant results for any of the module.

3.6 Priority queue

Priority queues is a mechanism to guaranty preferential access to a given set of users. The selection mechanism can be based on the UserID or location (IP address). Such static access permits to favor, for example, students from the hosting institution for a given period of time. It can be made dynamic by adding priority rules based on the student current performances and/or advancement. While it is primordial from a pedagogical point of view to test a bad/wrong solution on the real system, rules could be added to favor users with a valid set of parameters. This set of parameters could be found in a previous simulation step (Fig. 6). The users with the wrong set of parameters could be asked to redo the simulation, the one with correct set of parameters could have the priority to the one with outbound parameters.

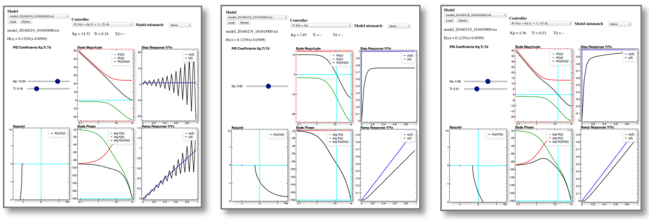

Fig. 6. Simulation of the system with of *a)* outbound parameters, *b)* wrong set of parameters, *c)* correct set of parameters

3.7 Reservation

Reservation is similar to priority queues where a given user identified with his/her UserID has access priority for a given period of the day. Implementing a reservation process can become demanding according to the granularity level. In addition, it requires information exchange between the MOOL and the MOOC.

3.8 Batch access

In batch mode, the experimentation is performed in a delayed manner: the user prepares the experiment parameters, sent it to the server and watch the result afterward. This works efficiently for some tasks, for example automatic testing and grading electronic circuits in Rochat (2014). But the main gain of interacting in real-time with physical system is lost. If one parameter was not correctly defined, the whole experience has to be performed again, while in the real-time interactive mode it can be changed immediately, and its effect directly observed. Interacting in real-time with real equipment and real data even at a distance is highly beneficial to user, in Jona et al. (2011).

3.9 Substitution with simulation

Similarly, in case of high load, the experimentation task could be replaced with a simulation Lowe D. (2014), Salzmann et al. (2016). In such case, the interaction is guaranteed, but the simulation is not the real system (Fig. 8). Furthermore, in our MOOC a significant part of the control course deals with learning how to adequately model and simulate a physical system with all its imperfections, thus simulating a simulation defeat the purpose of the course. As a matter of fact, both the modeling and the system parameter identification require observation and measurement on the real system.

3.10 Inter-MOOL sharing

There are many online labs similar to the one located at EPFL. Exchanging access time between institutions is a long-term goal that requires all the parties involved to agree on a set of practices. Such an effort has been proposed in the EU Go-Lab project which offered many tools to support large scale dissemination (http://golabz.eu): a portal where lab owners could advertise their labs; a self-describing API to access and manage smart devices in Salzmann (2015), a platform to integrate labs with educational material as open education resources (http://graasp.eu), it also proposes a tool to exchange reservations among labs, Ordunã et al (2015). Once the technical issues are managed, the remaining difficulty is linked to the access modalities that needs to be negotiated between parties.

4. QUESTIONNAIRE

To get students feedback regarding MOOL session time sharing, we sent questionnaire to all students (>200) and receive 21 responses.

This first question aims at getting a student feedback compared to the measurement we have internally for the MOOL load balancer (Fig. 3 & 6). The load-balancer gives aggregated values for all the experimentation servers, while students see a specific experimentation server at a time. 23.8% of the student never had to wait, none had to wait at each connection and 76% of the students had to wait sometime (Fig. 7).

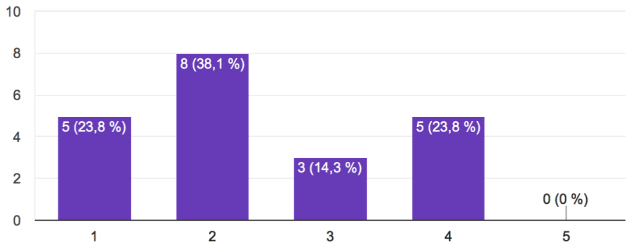

Fig. 7. How often did you have to wait before being able to control the setup? (0: never, 5: always)

When waiting, is the delay acceptable? Note that there is an indicator in the client web app to inform the user about the number of users in front of them. About 2/3 of the students found the delay very acceptable, none found the delay inacceptable (Fig.8).

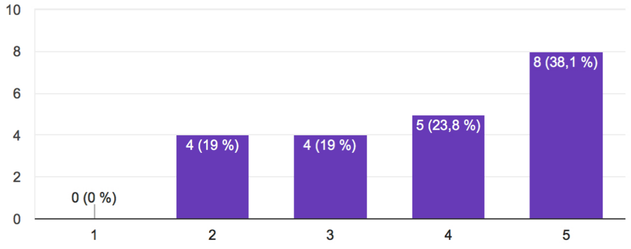

Fig. 8. Was the waiting time acceptable? (0 strongly disagree, 5 strongly agree)

We wanted to know if the hard-coded time for experiment is appropriate. The measurements at the load-balancer indicate that this time was under evaluated, but it does not inform about the user perception. About 1/3 of the users found this time sufficient while another third found it inadequate (Fig. 9). Since we did not specifically ask about the tradeoff between waiting time and experimentation time, it is difficult to derive a definitive answer regarding the hard-coded time for experiment.

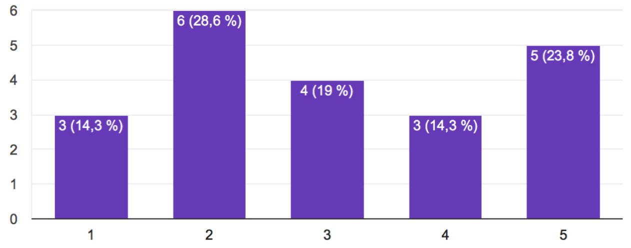

Fig. 9. Was the allocated time for you to experiment enough before you are thrown back in the queue? (0 strongly disagree, 5 strongly agree)

If a user tries to access a server that is already in use, s/he is queued. Note that the load-balancer will select the server with the shortest queue. When more than one client is in the queue, the experimentation time is shared among users. Only the user at the top of the queue can act on the system, the other users can observe what is performed by the acting user. In addition to showing the number of people waiting the client web application display the current measurement and video feeds. Unexpectedly, most of the users found this feature useless (Fig. 10).

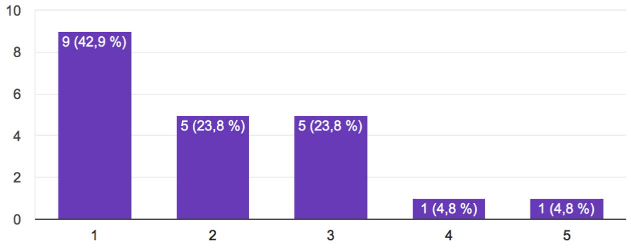

Fig. 10. While waiting (observer mode) is it useful to see what others are doing? (0 strongly disagree, 5 strongly agree)

While the load-balancer selects the server with the shortest queue, the user can still manually choose the server s/he wants to perform the experimentation. This, at the cost of not necessarily choosing the server with the shortest queue and thus inducing additional experimentation time for all the users experimenting on the given server. Most of the users had overwritten the load-balancer choice (Fig. 11). We don’t know why and if this action increased or not the total experimentation time.

A hint was given in the free comment section of the questionnaire: “The time was often too short to do the experiment properly. Often we therefore joined up (2 or 3 students) to do the same experiment on the same machine and like this enhance 'our' activity time". Another comment regarding user queuing perception: "it's a good idea but it is terribly frustrating, which is why I do the TP's outside the hours for it"

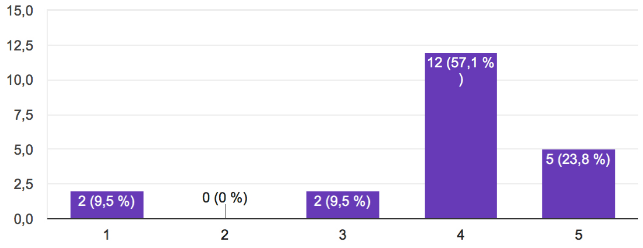

Fig. 11. Did you switch to another setup than the one that you were assigned to? (0: never, 5: always)

5. FUTURE IMPROVMENTS

Section 3 presented possible ways to improve the MOOL usage toward enabling a higher student:setups ratio. Some can be easily implemented such as displaying the MOOL current load, other are more demanding and may require not only a modification of the MOOL but also in the hosting MOOC, for example reservation.

While being well accepted according to the questionnaire, the round robin access needs to be improved. The first step is to detect the end of a task. One option is to have a decreasing time counter triggered by mouse movements in addition to the current timeout. The remaining time is displayed in the UI of the client web app. If the user does not move the mouse (or press a key) for x seconds, a counter starts down-counting, if no mouse/key event occur before the counter reaching 0, the current user is disconnected. Currently there is also a difficulty to precisely measure the user's intention: did s/he wants to start experimentation or did s/he clicks on the wrong MOOC tab. Similarly, with the end of the task, is s/he locking at received data or is s/he locking elsewhere?

A better detection of the task beginning and ending will permit a variable task duration and thus increasing the MOOL usage. It will also permit a better understanding of user access pattern.

6. CONCLUSIONS

This paper first summarizes the MOOL infrastructure and its MOOC integration. Such infrastructure is use for the Automatic Control course hands-on sessions followed by more than 200 students per semester. With a farm of 23 setups, this MOOC could sustain a ratio of 10 users for 1 setups without impacting noticeably the user perception of being the only one to access the system. Then we presented MOOL usage measurements and discussed the result of the questionnaire sent to students. The main conclusion is that in its current form the proposed time sharing via the round-robin access is well accepted but could be improved. Options to increase the number of users over time are then discussed. Finally, suggestions for future improvement are proposed.

REFERENCES

de la Croix, J. P. and Egerstedt, M. (2014) Flipping the controls classroom around a MOOC, in *American Control Conference, Portland, OR, 2014, pp. 2557-2562.*

Garcia, F. et al.,(2014) Apractice-based MOOC for learning electronics, *in IEEE Global Engineering Education Conference (EDUCON'14)*, pp. 969–974.

Gillet, D., Salzmann, C., Longchamp, R., Bonvin, D., (1997) Telepresence: An opportunity to develop real-world experimentation, *in education, In European Control Conference, (ECC97),* Brussels (1997)

Hossain, Z., Bumbacher, E., Brauneis, A. et al. (2017). Design Guidelines and Empirical Case Study for Scaling Authentic Inquiry-based Science Learning via Open Online Courses and Interactive Biology Cloud Labs. In *Int J Artif Intell Educ*, https://doi.org/10.1007/s40593-017-0150-3

Jona, K., Roque, R., Skolnik, J., Uttal, D., and Rapp, D. (2011) Are Remote Labs Worth the Cost? Insights From a Study of Student Perceptions of Remote Labs, *in iJOE Volume 7, Issue 2, May 2011*

Lowe, D., MOOLs: Massive Open Online Laboratories: An Analysis of Scale and Feasibility. (2014) *in the 11th International Conference on Remote Engineering and Virtual Instrumentation,* REV2014, Piscataway, USA.

O'Malley, P.J. Agger, J. R. and Anderson, M.W. (2015). Teaching a Chemistry MOOC with a Virtual Laboratory: Lessons Learned from an Introductory Physical Chemistry Course, in *J. Chem. Educ. 2015, 92, pp. 1661−1666*

Orduñã, P., Garbi Zutin, D., Govaerts, S., Lequerica Zorrozua, I., Bailey, P. H., Sancristobal, E., Salzmann, C., Rodriguez-Gil, L., DeLong, K., Gillet, D., (2015) "An extensible architecture for the integration of remote and virtual laboratories in public learning tools," in *Tecnologias del Aprendizaje, IEEE Revista Iberoamericana de, vol. 10, no. 4,* pp. 223–233.

Rochat, P-Y., Automatic remote correcting system for MOOCs (2014), in *Proceedings of the 6th European Embedded Design in Education and Research*, pp. 124-127

Salzmann, C., Govaerts, S., Halimi, W. and Gillet, D. (2015) The Smart Device Specification for Remote Labs, in *International Journal of Online Engineering, vol. 11, num. 4*, p. 20-29, 2015.

Salzmann, C., Gillet, D. and Piguet, Y. (2016). MOOLs for MOOCs: A first edX scalable implementation. In *Remote Engineering and Virtual Instrumentation (REV), 2016 13th International Conference on. IEEE*, pp. 246–251.

Salzmann, C., Halimi, W., and Gillet, D. (2017). Access to Massive Open Online Labs through a MOOC. In *L@S '17 Proceedings of the Fourth ACM Conference on Learning @ Scale,* pp. 331-334

Salzmann, C., Halimi, W., Gillet, D., and Govaerts, S. (2017). Deploying Large Scale Online Labs with Smart Devices. *In Cyber-Physical Laboratories in Engineering and Science Education, in press*