Proceedings Paper:

Rossiter, J.A. (2024) Creating whole story virtual laboratories for a first course in control. In: 2024 UKACC 14th International Conference on Control (CONTROL). CONTROL 2024: 14th United Kingdom Automatic Control Council (UKACC) International Conference on Control, 10-12 Apr 2024, Winchester, United Kingdom. Institute of Electrical and Electronics Engineers (IEEE). ISBN: 979-8-3503-7426-1. ISSN: 2831-5219. EISSN: 2766-6522.

https://doi.org/10.1109/CONTROL60310.2024.10532066

Creating whole story virtual laboratories for a first course in control

J. A. Rossiter[1]

***Abstract*— There has a been a growing interest in virtual laboratories as a supplement to hardware laboratories in supporting student learning and experience. This paper focuses specifically on virtual laboratories built using the MATLAB environment and highlights some recent developments. Specifically, these virtual laboratories aim to give users an overview of the core content of an entire *1st course in control* in a single virtual laboratory interface. This paper highlights two such laboratories and shows how they can be used to supplement other learning resources and activities.**

I. Introduction

The engineering educational community has always held the viewpoint that laboratories are an essential part of student learning [14] and this was re-emphasised in the recent global survey on so called Control101 courses, that is, a first course in control [15].

Nevertheless, there is also an awareness that hardware laboratories are expensive to provide and there is a practical limit to how much access most institutions can provide due to conflicting pressures of budget, space, timetabling and more. Consequently, in practice, access to hardware is relatively limited and there are even fewer opportunities for open-ended access where students can, in some sense, play or experiment to create and test their own hypothesis.

A. Virtual laboratories

As a response to these pressures, there has been an increasing recognition of the potential of both remote access and virtual laboratories, e.g. [1], [3]–[7], [9], [11], [12], [20], [21]. The theory is that access to such activities is a lot more flexible thus allowing students to do further experimentation in their own time and also potentially, to allow much more freedom over which activities they perform. Of course there may still be limits to this, and indeed remote-access laboratories may still have relatively restricted access and need careful booking and management; this paper does not discuss remote-access any further.

Here the focus is on virtual laboratories because in principle these have a number of advantages such as:

1) Unlimited numbers can use in parallel.
2) Access is truly 24/7.
3) Simulation times can be very fast so students can focus on concepts and behaviours for multiple different scenarios in a short time.
4) Inherently safe, so students can investigate a wide variety of scenarios that may be difficult with real hardware.
5) Can be tailored to support pre- and post-activities linked to real hardware use.

B. A first course in control and MATLAB

To further set the context, we return to the recent community survey on Control101 courses [15]. More recently, a follow on survey [20] has been looking at how we support staff and students with high quality open-access resources for Control101 courses. This discussion is ongoing and for now simple options such as the Resourcium website [8] are being used as convenient entry points to the wide range of resources available. A specific challenge is that we want resources to be easy to find and easy to use.

The author of this paper has taken the combined objectives, listed below, and proposed a community solution. We want resources that:

1) Focus on concepts to help students understand the power and importance of feedback across a range of interesting and varied scenarios [17].
2) Remove the need for tedious but mundane paper and pen computations so students can focus on understanding core concepts and the consequences of different decisions.
3) Incorporate visualisation and animation so that the resources bring a topic to life in a way that pages in a book cannot.
4) Ensure the resources are easy to find, download and use.
5) Academic staff can create relatively quickly and cheaply without requiring excessive coding or other expertise.

The proposed solution is to build a Control101 MATLAB toolbox focussing primarily on learning and understanding fundamental concepts linked to an introductory control course [19]. The reasons for using MATLAB are summarised as follows:

1) The broad functionality and large number of powerful built-in functions reduces and simplifies any coding requirements for staff.
2) The environment is largely intuitive and thus academics can create useful resources fairly quickly [10], [13].
3) Students can use the resources for visualisation and learning with relatively little training.
4) Tools such as MATLAB online and MATLAB web servers improve accessibility to users without computers.
5) MATLAB is very widely used for control education in the university sector.

[1] Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, S1 3JD, UK (e-mail:j.a.rossiter@sheffield.ac.uk)

6) MATLAB CENTRAL acts as a convenient repository which is more sustainable and searchable than many alternatives. Moreover, incorporating resources into a toolbox means users can access and download through the add-ons button within MATLAB.

The author of this paper has created a first draft of the toolbox to help support students and initiate discussion, but the intention is that in the long term ownership is shared by the community so other individuals can make proposals for additions and modifications. The author hopes that these interactive resources will help students by engaging them and providing important feedback opportunities.

Remark 1: Those who are interested can download the current version of the toolbox by doing a search on **control101** *under the add-ons button. Also see the author's website for some further background.* (`https://sites.google.com/sheffield.ac.uk/controleducation/matlabresources/community-control-toolbox`)

C. Story line apps

The main purpose of this paper is to highlight a novel aspect of the toolbox as compared to historical contributions [16], [18]. In general, early contributions were, individually, relatively narrow in focus and thus concerned a single learning outcome such as a system behaviour or PI tuning for a given system. This was deliberate so as not to over face users with too much complexity.

Of course, at some points students need to be ready to engage with a more holistic view of a feedback scenario, thus covering issues such as:

1) How does a system behave in the open-loop?
2) How do uncertainty and disturbances affect the behaviour?
3) What can I do to improve the behaviour and also to manage disturbance rejection?
4) What role might or should feedback take in managing the behaviour?
5) The benefits of using automated and well-tuned feedback.

Consequently, the plan is to include multiple so-called *story line* apps within the toolbox that cover the entire design process from behaviour analysis through to automated control design. The hope is that engagement with these will help students link all the elements of their control course together, for example;

1) Why do I need to do first principles modelling?
2) Why do I need to analyse and understand behaviours?
3) How do disturbances affect performance and how do I reject them?
4) The core role of feedback in ensuring effective system performance.
5) The need for systematic feedback design so that we can automate feedback.

D. Paper organisation

The paper is organised a follows. For completeness, Section II gives a brief overview of how to develop interactive resources using the MATLAB app tool and how to incorporate visualisation. Then Section III looks at how to build so called story line apps, followed by a an overview of the two story line apps within the current toolbox. Section IV completes the paper with some reflections and challenges to the community.

II. Developing interactive resources in MATLAB

Visualisation is important for learning because it helps students appreciate the context and therefore the significance of the concepts being discussed much better than line charts and equations. Consequently, for the author, a fundamental driver for developing the toolbox was to include effective visualisation and animation of multiple different engineering scenarios, obviously with a focus on feedback and control. The initial draft has relatively conventional scenarios, but in line with the CONTROL2030 [2], [17] vision, the hope is to extend these in due course.

A. Creating animations in MATLAB

MATLAB has a few very simple tools that support animation of pictures/figures. although we need to be clear at the outset that the timing is not precisely handled as it is difficult to guarantee across different computing hardware.

The first thing to note is that any object/line plotted to a figure window (or axes) has a handle which contains all its properties such as: i) coordinates; ii) colour; iii) line thickness and style; iv) font size and v) more. The object can be changed by changing the value in the handle, for example, for an object with a handle 'h': *h.linewidth = 3* changes the linewidth to 3. Similar statements can be used to change the co-ordinates, colours and so forth and thus to make objects change position, shape, appearance and more.

In order to create what appears to be a continuously moving object, one needs to update the coordinates with a relative fixed sampling period within a loop. This can be handled using *tic, toc* which give time differences; thus one can arrange for a pause statement for a known time before performing any updates. By doing this in a loop, one sees what seems like, to human vision anyway, continuous changes. However, this process can be somewhat sensitive as the time consumed updating plots on the screen, or indeed computations, may exceed what is desired. Nevertheless, the author has found that in the main, for the purposes of the toolbox, this simple method is ample and thus fit for purpose (about 10 sec per simulation with animation).

B. Graphical user interfaces and the app environment

Graphical user interfaces (GUIs) make it easier for the user to engage with a scenario. Instead of code, the user need only engage with simple input options such as sliders, buttons and drop-down menus (e.g. see Figure 1) in order to explore the impact on behaviour of different choices/designs. In recent years, MATLAB updated its GUI environment to what is now called the app environment.

Users build an interface by moving and sizing the desired objects into a panel. These could be axes, sliders, text, push

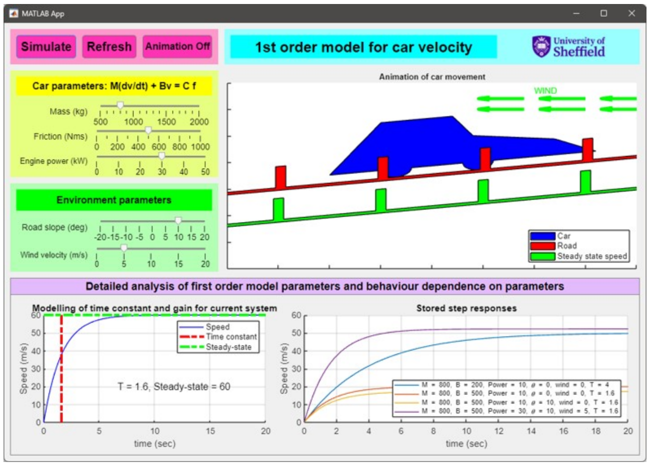

Fig. 1. Illustration of the app on car speed.

buttons and many more options; in essence whatever it might be useful for the user to change. In general terms it better to first imprint a grid as this ensures the objects are neatly arranged and can be scaled as necessary. such as when the user has a different sized screen.

Having laid out the interface design, the next step is to code what happens behind the scenes. Each interactive object (e.g. sliders and push buttons) can contain what is denoted *a callback subfunction*. Thus, whenever a user interacts with that object, the associated callback code runs automatically. Hopefully it is therefore fairly clear that it is moderately straightforward for a non-computer scientist, that is the typical engineering academic, to code up a simple app in relatively short time. It is sufficient to decide what you want to happen when a particular object is selected, and enter that code in the appropriate callback.

Remark 2: There are of course some subtleties which are too detailed for this paper.

C. Simple design principles deployed in the virtual laboratories

While it is always possible to design a very authentic interface with multiple different parameters, in truth that can confuse students at first as the complexity may distract them from a simple message or concept. Consequently, there is a lot of value in some interfaces having relatively little interactive choices, as seen in Figure 1. Here the user can only change the mass, friction and engine power for the car and, road slope and wind in the environment.

Of course such simple virtual laboratories conversely have limited use but the intention is that, initially at least, lecturers will use virtual laboratories with a single simple message and thus enable students to engage easily with the associated concepts. Later on, once some of those concepts have begun to be understand, will be time to start combining them.

In the toolbox, the introductory apps focus on basic behaviours, that is, how do engineering systems behave and how does that behaviour change as I change the parameters and external factors/environment? Later apps may, for example, focus solely on PI design, for a given engineering system.

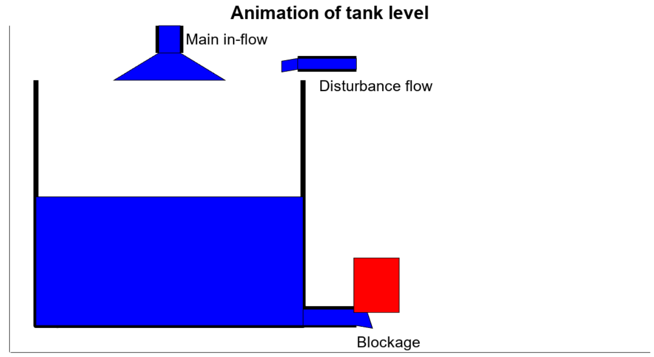

Fig. 2. Illustration of the tank level picture.

In terms of the visualisation and animation, cartoon like pictures are used because these are easy to code and manage within the code. Nevertheless, they are usually representative enough for the user to relate the scenario to an equivalent real system, e.g. see Figure 2 for a tank level system with inlet flow, disturbance flow and outlet pipe partial blockage.

III. Developing of story line virtual laboratories and visualisations

There are multiple apps with a relative limited focus [19] so students can learn core concepts. However, ultimately we want students to appreciate a holistic view of control and feedback (see section I-C). For convenience, rather than having to jump through 3 or 4 apps, it would be easier to have a single interface allowing students to pass through the entire journey; these are what the author has denoted story line apps as they cover the entire story beginning from system analysis and modelling, the need for control and then going through to control design and finally feedback loop evaluation.

A. Basic structure of story line apps

Currently there are just two story line apps in the Toolbox, although the long term intention is to create more. Ideally they can be used towards the end of a course to help students see how the entire curriculum fits together coherently, and of course, to re-motivate them on the need for feedback.

The basic structure of these apps is to create a number of tabs on the interface. Each tab contains information and user inputs related to a particular part of the design process. For example:

1) Analyse the behaviour of a system (tab 1).
2) Undertake systematic modelling (tab 2).
3) Understand why manual or open-loop control is inappropriate (tab 3).
4) Undertake systematic PI design (tab 4).
5) Assess the closed-loop behaviour with a wide range of authentic uncertainties (tab 5).

Hence, each tab of a story line app is maintained as a relatively simple interface and students move between the

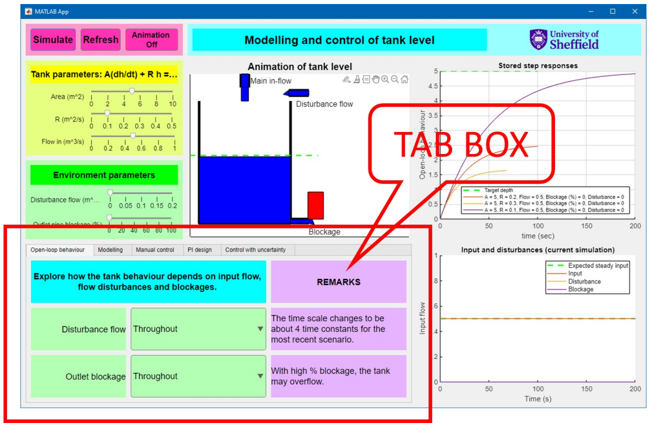

Fig. 3. Illustration of the tab box within the tank level story line app.

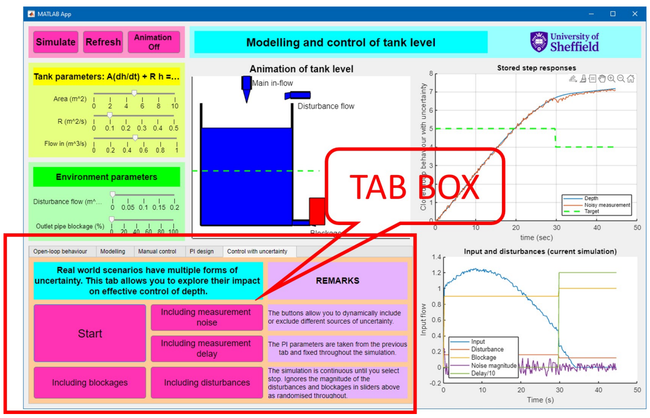

Fig. 4. Illustration of a different tab selection within the tank level story line app.

tabs to undertake the different steps in a design process. Information which remains constant throughout is displayed separately from the tab and is always visible.

The design of the overall app will have some parts which are constant and some parts which are tab dependent. For simplicitly, the tab dependent parts are within the tab part of the window which for now has been placed in the bottom left, as shown in in Figure 3. The contents of the tab box will change for each tab, for example see Figure 4 which is clearly different from Figure 3.

The remainder of the interface has the parts which are used throughout, so for the example in Figure 3, the system and environment parameters are in the top left, the animation of the system in the top middle, the line plots of behaviour in the top right and other line plots (such as the system input and disturbances) in the bottom right.

B. Memory features and animation

The default setting is that the app will add new simulations to previous simulations so that users can investigate trends, for example, what happens as I gradually increase the blockage to the outlet pipe, or what happens as I gradually increase the tank cross sectional area? The legend in the top

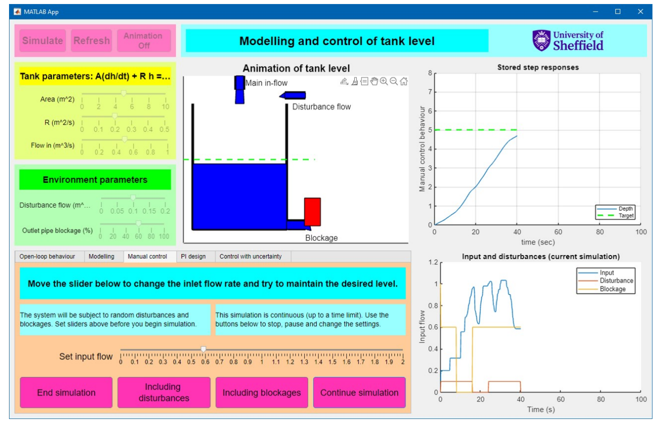

Fig. 5. Illustration of the tank level story line app under manual control.

right figure keeps track of the core parameters so the user can easily see which line plot goes with which choices, e.g. see Figure 3.

After a while, the figure may become congested or the user may wish to being a new series of investigations or change tab; the *refresh* button will clean the figures.

Animation is an optional feature. The initial default is that animation is *on* and thus the user will see the tank gradually filling and the line plot moving. These animations are intended to help the user relate their experiments with a real life equivalent. Of course the advantage here is that the animations are restricted to take about 10 seconds which is far quicker than, for example, a real-life tank equipment.

Once the user has absorbed and understood the context, in most tabs animation can be switched off so that the line plots appear almost instantly in order to allow a speed up of any systematic investigations that may be desired. One exception is the *manual control* tab (e.g. see Figure 5) as in this case the intention is for the user to practice manual control and thus the simulation must be close to real time in some sense.

C. Tank level story line app

Figures 3-5 already illustrate the tank level story line app. Summaries of the learning and activities for this app are described next.

Open-loop behaviours: The first tab allows the user to explore how the tank depth responds to changes in the main in-flow or system input (slider in the light yellow box) and a disturbance in-flow (slider in the light green box). The user can change tank cross-sectional area and the conductance of the outlet pipe (sliders in the light yellow box) and blockages to the outlet pipe (slider in the light green box). Options within the dedicated tab allow the use to decide the timing of the disturbances (throughout or mid-simulation) as this will help with understanding the impact of these. For no disturbances, set the appropriate sliders to zero. Figure 6 demonstrates a typical top right figure and legend investigating just changes in A.

Modelling: The second tab encourages the students to do data based modelling, but here as the model is approximately

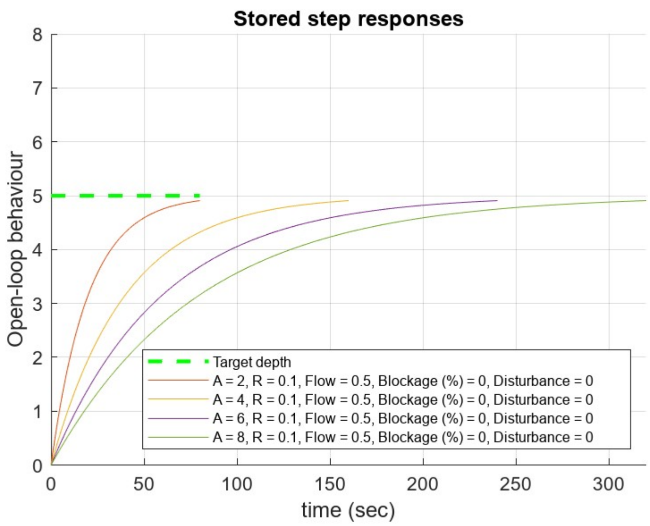

Fig. 6. Illustration of a figure from the tank level story line app under the open-loop behaviour tab.

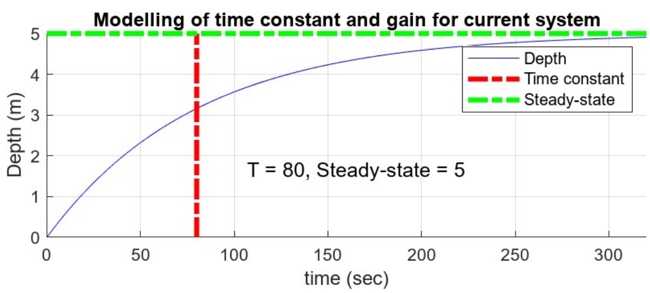

Fig. 7. Illustration of the modelling tab in the tank level app.

1st order, this can be reduced to estimating the time constant and gain from a step response. A figure in the tab dependent part is shown in Figure 7 which emphasises how the estimates of time constant and gain are made.

Manual control with uncertainty: Having understood the natural behaviours of the system, it will be clear that the disturbances and other factors affect both speed of response and steady-state. A user may like to modify the input flow directly in order to achieve the desired depth and speed of response; this tab allows the user to try. However, disturbances change randomly and thus it is a fight to keep the level at the desired depth. Hopefully users will quickly see that manual (or indeed open-loop) control is not a viable option.

PI tuning: Having decided that manual control is tiring and poor, users can automate the feedback with a PI compensator. This tab allows the user to experiment with different choices of PI tuning and decide upon their preferred choice.

Evaluation of automatic control: The final tab simply runs the PI compensator from the previous tab, but with some important differences. Disturbances are injected continuously and these vary periodically and thus the user can evaluate the efficacy of the control in a realistic scenario. Disturbances and uncertainty include the blockages and inlet disturbance

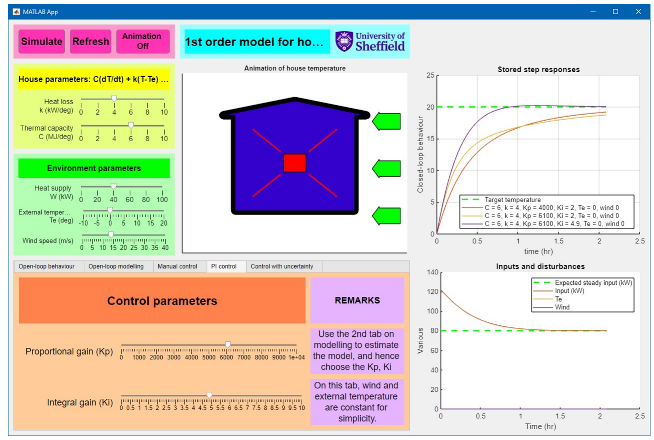

Fig. 8. Illustration of the PI tuning tab in the house temperature story line app.

flows, but also measurement delay and measurement noise. Each disturbance can be switched on and off (using the buttons in the tab, see Figure 4) as desired to observe the impact on behaviour and tracking. The target also takes the form of a square wave so that users can observe the tracking performance.

D. The house temperature story line app

Much of the description of this app would be slightly repetitive of the previous section, as by design the apps in the toolbox are intended to have a similar look and feel so users can quickly move from one to another without confusion. Hence the layout and principles are almost identical with the main differences being the context, e.g. see Figure 8.

The core parameters are now thermal inertia (or heat capacity) represented by the size of the house and insulation (heat loss), represented by wall thickness. The input is the amount of heating provided (represented by the size of the red rectangle). Temperature, which is animated, is represented by colour (red for hot and blue for cold).

Two disturbance are used, that is, the external temperature (which drives heat loss through the walls) and also a wind speed (it is assumed that heat loss increases with higher winds, the green arrows). The user can also set the initial temperature as it is interesting to observe how quickly one can move from an unheated state to the desired internal temperature.

IV. Conclusions and Reflections

This paper has given an overview of some new story line apps which have been developed as part of the new MATLAB Control101 toolbox. The toolbox is intended to support students in learning and understanding the fundamental concepts and motivation for feedback loops. While many of the apps support a single concept within that overall aim, this paper has focused on some tools within the toolbox that each attempt to capture a large part of the overall requirements, that is from initial system modelling, though control design to evaluation. The hope is that these interfaces will help students complete their first course in control with an enhanced appreciation of its importance in solving future world problems [2].

The toolbox was formally launched in the Autumn of 2023 and so far contains a relatively limited number of files, mostly from the author of this paper. The hope is that students and lecturers will find this useful and will offer to become collaborators in enhancing what is offered, so creating a more diverse range of interactive laboratories, including those with story lines. The author is particularly interested in examples that go beyond conventional engineering examples. He is aware of some resources in development by colleagues but would welcome anyone who is interested to contact him. Indeed he is already aware of 3-4 international colleagues would like to add examples to this toolbox.

Acknowledgements: Thanks to Mathworks for financial and expert support, especially from George Amarantidis Koronaios and to colleague Patricio Ortiz for help with github.

References

[1] Albertos, P., 2017, MOOC in dynamics and control, http://personales.upv.es/~palberto/

[2] Annaswamy, A.M., Johansson, K.H. and Pappas, G.J., 2023, Control for Societal-scale Challenges: Road Map 2030, https://ieeecss.org/control-societal-scale-challenges-roadmap-2030

[3] Cameron, I., 2009, Pedagogy and immersive environments in the curriculum, Blended Learning conference, 290-294.

[4] de la Torre, L., R. Heradio, C. A. Jara, J. Sanchez, S. Dormido, F. Torres, and F. Candelas, 2013, Providing Collaborative Support to Virtual and Remote Laboratories. IEEE Transactions on Learning Technologies.

[12] Panza, S, Invernizzi, D., Giurato, M., Yang, G., Chen, K. and Parisini, T., 2021, Integration of experimental activities into remote teaching using a quadrotor test-bed, 21st IFAC Workshop on Aerospace Control Education.

[13] Rossiter, J.A., 2016, Low production cost virtual modelling and control laboratories for chemical engineering students, IFAC symposium on Advances in control education.

[14] Rossiter, J.A., B. Pasik-Duncan, S. Dormido, L. Vlacic, B. Jones, and R. Murray, 2018, Good Practice in Control Education, European Journal of Engineering Education, http://dx.doi.org/10.1080/03043797.2018.1428530.

[5] de la Torre, L., Neustock, L.T., Herring, G.K., Chacon, J., Clemente, F.J.G., and Hesselink, L., 2020, IEEE Transactions on Industrial Informatics, Automatic Generation and Easy Deployment of Digitized Laboratories, 12, 16, 7328-7337, doi = 10.1109/TII.2020.2977113

[6] Dormido, S., J. Sanchez-Moreno, H. Vargas, L. de la Torre, and R. Heradio. UNED labs: a network of virtual and remote laboratories. In Javier Garca Zuba and Gustavo R. Alves, editors, Using Using Remote Labs in Education: Two Little Ducks in Remote Experimentation, pages 253-270, Bilbao, 2011.

[7] Dormido, S., H. Vargas, J. Sanchez, 2012, AutomatL@bs Consortium: A Spanish Network of Web-Based Labs for Control Engineering Education, Internet Accessible Remote Laboratories: Scalable E-Learning Tools for Engineering and Science Discipline, 11, 206-225, A. Azad, M. E. Auer, V. J. Harward (Ed), IGI Global.

[8] Douglas, B., 2022, Resourcium, https://resourcium.org/

[9] Ferrari, M., Visioli, A., 2023, An Educational Interactive Software Tool to Learn Cascade Control Design, IFAC world congress.

[10] Koch, A., Lorenzen, M., Pauli, P. and Allgower, F., 2020, Facilitating learning progress in a first control course via Matlab apps?, IFAC world congress.

[11] Matisak, J., Pohancenik, M. and Zakova, K., 2023, Platform for Virtual Laboratory of Mechatronic Systems in Augmented Reality, EXPAT 2023 and IEEE Explore

[15] Rossiter, J.A., Serbezov, A., Visioli, A., Zakova, K. and Huba, M., 2020, A survey of international views on a first course in systems and control for engineering undergraduates, IFAC Journal of Systems and Control, Vol. 13, Article 100092, 15 pages. https://doi.org/10.1016/j.ifacsc.2020.100092

[16] Rossiter, J.A., 2022, MATLAB apps to support the learning and understanding of simple system dynamics, IFAC Symposium on Advances in Control Education.

[17] Rossiter, J.A., et al (numerous), 2023, Control Education for Societal-Scale Challenges: a Community Roadmap Annual Reviews in Control, 55, pp1-17, https://doi.org/10.1016/j.arcontrol.2023.03.007.

[18] Rossiter, J.A., 2023, A suite of MATLAB livescript files to support learning of elementary control and feedback concepts, IFAC world congress.

[19] Rossiter, J.A., A novel MATLAB toolbox for Control101 courses, 2024, ECC (submitted).

[20] Serbezov, A., Zakova, K., Visioli, A., Rossiter, J.A., Douglas, B. and Hedengren, J., 2022, Open access resources to support the first course in feedback, dynamics and control, IFAC Symposium on Advances in Control Education.

[21] Zapata-Rivera, L.F., Larrondo-Petrie, M.M. and Weinthal, C.P., 2019, Generation of multiple interfaces for hybrid online laboratory experiments based on smart laboratory learning objects, IEEE Frontiers in Education Conference (FIE), 1-8, doi=10.1109/FIE43999.2019.9028421