Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 59-19 (2025) 508–513

Developing files for the control101 toolbox to support the learning of non-linear system behaviour and control

J.A. Rossiter *

* *School of Electrical and Electronic Engineering, University of Sheffield, Sheffield, S1 3JD, UK (e-mail:j.a.rossiter@sheffield.ac.uk*

Abstract: Many control courses focus on linear systems, behaviour and control, whereas the real world is nonlinear. This paper introduces learning resources on nonlinear systems which staff can use in introductory courses to help students appreciate this fundamental point, without being overloaded by analytical and mathematical detail. Hence the focus is on interactive resources which display simple animations and plots while allowing users to make small parameter changes and observe the impact on behaviour and control. This paper reviews some of the new resources for non-linear systems in detail and summarises all of them alongside a discussion of how they might be embedded into a first, or indeed later, course in systems, modelling and control. The resources have been embedded into the MATLAB control101 toolbox Rossiter (2024).

Keywords: Interaction within a lecture, virtual laboratories, independent learning.

1. INTRODUCTION

There has a been repeated work over the past few decades focused on what constitutes an ideal first course in control and most recently in the past few years (Silverstein et al., 2015; Rossiter et al., 2020). There was reasonable consensus on the core curriculum and emphasis, albeit a recognition that every institution has different priorities, structures and parallel courses which will impact on the actual implementation. Some, but not all, of the core points are:

(1) Focus on motivation and concepts with interesting case studies and of course laboratories.
(2) Do not over-emphasis mathematics and certainly not paper and pen based computations. Ensure students focus on understanding the importance of control and behaviours.
(3) Use software to support computation and illustration so students can do *what if* conceptual testing quickly and transparently.

It was quickly apparent that as a community, while the community has produced a wealth of high quality resources over the years, e.g. Douglas (2022); Rossiter (2021); Albertos (2017); Quansar (2022); Serbezov et al. (2022); Panza et al. (2021); Zapata-Rivera et al. (2019); de la Torre et al. (2013, 2020); Goodwin et al. (2011); Cameron (2009); Rossiter et al. (2018); Murray et al. (2004), the organisation and sharing of these has been somewhat *ad hoc* and chaotic. In order to support staff in the delivery of what we are calling **Control 101** courses, we need to provide resources that can be quickly and easily embedded into teaching and assessment, as well of course, as being easily available, easy to find, easy to use and so forth. The Technical committees for control education of both the IEEE and IFAC are currently considering how best to facilitate this sharing long term (Serbezov et al., 2022).

The focus of this paper is on resources based on non-linear systems that one might argue are motivational but superficial in analytical detail in the first instance, that is ones which allow users to do quick and easy testing of concepts, for example:

- How will behaviour change if I increase parameter A?
- What is the impact of including integral action in the feedback loop?
- How are open- or closed-loop behaviour affected by a disturbance or fault?
- How does non-linearity affect behaviours?

That is, the first priority is engagement and getting students to appreciate some real issues, and here specifically non-linear aspects, and begin to take an interest in what skills they need in order to tackle these issues. Some loose criteria that have been applied to the resources are:

(1) 24/7 access and indeed simultaneous independent access by an entire cohort.
(2) Fast run times (close to instantaneous) hence allowing rapid testing of concepts and thinking.
(3) Available on software students are already using so any toolbox installation is quick and usage is within an already familiar environment.
(4) Animation and easy user interactivity should be core on the resources.

There are some very high quality open access or licensed resources available (e.g. (Dormido et al., 2011, 2012; Matisak et al., 2023; Ferrari et al., 2023; Cameron, 2009; Goodwin et al., 2011; Quansar, 2022)). However many of these require students, and staff, to engage with an environment which is not already familiar and moreover may be neither easy nor quick to edit and tailor for local

10.1016/j.ifacol.2025.11.085

needs. Hence, while the author accepts that the proposal in this paper is not open access in the full sense [1], the decision was taken to base the resources on the MATLAB and SIMULINK environment, because:

(1) This is widely used in the control community and many institutions secure a site license for their students thus both students and staff are already familiar with the environment.
(2) The download of toolboxes is very quick and easy, from within the main window. Toolboxes are often free as indeed is the control101 toolbox.
(3) Perhaps most critically, the environment allows quick and easy development and editing of interactive resources (Rossiter, 2016, 2017; Koch et al., 2020; Nevaranta et al., 2019; Rossiter, 2022) and time is often the biggest constraint for academic staff.
(4) The toolbox also works in MATLAB online for those without a license or access to a computer.

The initial focus of the control101 toolbox was fundamental resources (Rossiter, 2024). These were officially launched at the world congress in 2023. Since then efforts have focused on extending the resources to cover more breadth and indeed resources on state space systems and PID design were released in 2024. In addition, the main focus of this paper is non-linear systems which will discuss resources within this area, also released in 2024.

In summary then, this paper introduces the non-linear resources which were in the summer 2024 release of the toolbox. This is not an endpoint, and we welcome contributions from the community to fill any perceived gaps. Section 2 will give some brief discussion on SIMULINK, coding requirements and resource design as this will be helpful to any readers who want to consider developing resources of their own. Section 3 will then summarise the main resources which have been developed and the paper finishes with some conclusions.

2. AN INTRODUCTION TO THE MATLAB AND SIMULINK ENVIRONMENT

This section illustrates concisely some of the core coding requirements in support of the modelling and simulation of non-linear systems. The source code and SIMULINK files come automatically in the toolbox so users who so desire can make more fundamental edits to meet their own requirements; the toolbox has an open creative commons license.

2.1 SIMULINK models

An easy way to handle non-linear elements is within the SIMULINK environment. A simple illustration is given in Figure 1. The most obvious non-linear elements in this are the switch, square, square root, product and tangent terms. Most of the other terms manage signal input and output. What is clear is that the non-linear terms can be entered in a very transparent way which is quick and easy to code and likewise, the flow of data or signals is very clear.

[1] Mathworks is making some access to MATLAB online free

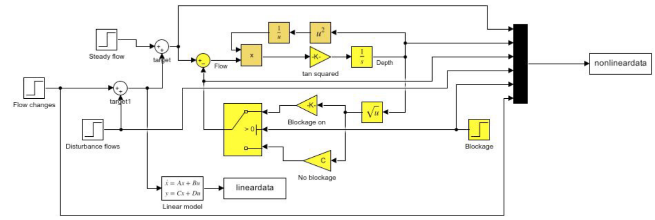

Fig. 1. Illustration of the conical tank SIMULINK model.

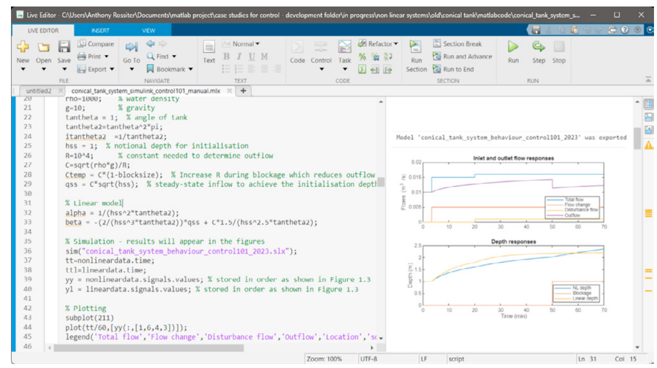

Fig. 2. Illustration of the livescript file supporting the simulation of the conical tank SIMULINK model.

Remark 1. Simulink models are not backwardly compatible. The models currently in the toolbox run on versions 2023 and later. If you run in a later version, you may give an alert.

2.2 Running a SIMULINK model

A strategic decision was taken for the control101 toolbox not to run any files using the options within the simulink editor, such as the run button. Most models have a large number of parameters and signals which need to be defined as well as output data where the plotting is ideally managed in a systematic and controlled fashion. In order for all these parameters to be flexible for the user to change, there needs to be an easy way to do that which avoids opening each box in the SIMULINK model and modifying separately.

The choice made here was to run the SIMULINK models using standard m-files or livescripts. This way, all the required data and model parameters can be defined transparently, and users can easily edit these as required and then select "run file" to update the simulation and plots. More specifically, the code was embedded into a livescript file as this allows sensible handling of background information and modelling, alongside the code snippets and plotting, e.g. see Figures 2, 3. To update a simulation, edit the code/parameters as required and then select *Run Section*.

The livescripts are organised into sections a little like a workbook so that users can step through these and gradually engage with more complex concepts and simulations. The sections all have the relevant code snippets and plots which can be run independently through the run section button. Code snippets can be copied to a simple m-file and run through that medium should the user find this more convenient.

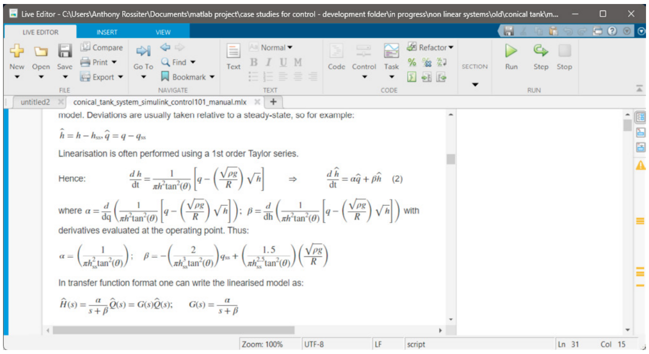

Fig. 3. Illustration of the livescript file giving supporting background information.

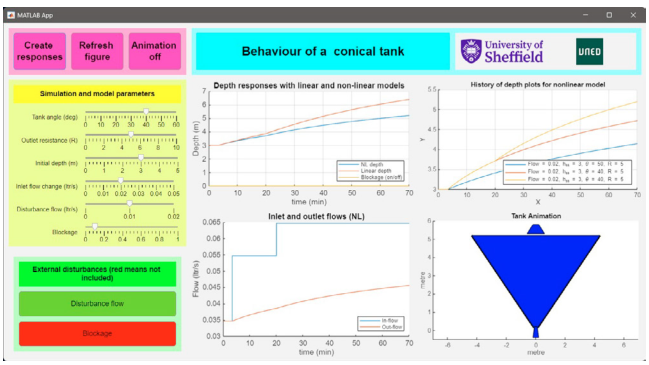

Fig. 4. Illustration of a virtual laboratory interface for the conical tank.

2.3 Virtual laboratories and MATLAB apps

The livescripts are an ideal environment for providing detailed explanation and modelling alongside simulation, but these still require the users to engage with the code snippets and make manual changes to numbers in several places which is more cumbersome than we would like for quick conceptual testing. An alternative environment which allows users to interact at a higher but less flexible level is one based solely on items such as sliders and buttons (see Figure 4). In this case, it is transparent to any educated user how to perform conceptual testing of ideas such as: what happens if I gradually change the tank angle?

- The apps can be run with or without animation. The animation attribute is to help bring the scenario to life to help students better link the scenario to real industrial practice.
- The apps also contain a history functionality, that is, at least one of the plots will retain and overlay all the previous simulations to allow compare and contrast investigations.
- The weakness of apps is that the range of investigations, parameter choices and so forth is limited to what has been coded into the interface. Students who wish to move beyond this would need to move to the livescript and source SIMULINK file.

Remark 2. The running of a SIMULINK file from within an app requires some specific coding which is not discussed here, although simple once the user knows the required syntax. The source code is available on github for those who want it. (https://github.com/jarossiter/control101)

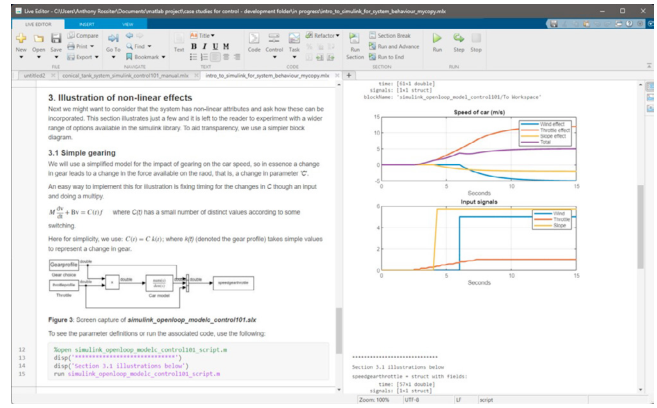

Fig. 5. Screen dump from the livescript focusing on an introduction to the use of simulink.

3. NON-LINEAR SCENARIOS WITHIN THE CONTROL101 TOOLBOX

This section will summarise the available files and scenarios within the toolbox. The hope is that the community at large will gradually add to this so that there is a good suite of scenarios to support the needs of most academics and students wishing to begin engagement with non-linear systems. Some of the toolbox basic files also which include occasional non-linear aspects such as input constraints, but these are not discussed in this paper due to space limits and focus.

3.1 Files focusing on an introduction to SIMULINK

The first set of files focuses primarily on encouraging familiarity with the SIMULINK environment and settings and thus, while there are some non-linear aspects, these are not heavily emphasised.

Simple introduction - open-loop and closed-loop The first resource uses the app environment and is delivered like a virtual laboratory. Two other files are opened automatically: i) the manual which gives detailed step-by-step instructions and ii) a SIMULINK file containing the components required for the tasks. The tasks introduce students to the block diagrams within the SIMULINK window and encourage the user to become familiar with a limited toolset and how data is returned to the workspace for further processing.

Introductory examples Building on the previous resource, this time a livescript is used to expose students to how integration with MATLAB code can make running the SIMULINK file very efficient to allow systematic investigations for a range of parameter values (see Figure 5). The tasks, built around feedback loops, gradually build student competence and the breadth of scenarios.

3.2 Introductory case studies based around livescript files

In the context of a first course in modelling, simulation and control, it is unlikely that we want students to go beyond a relatively superficial engagement with SIMULINK. Here a number of relative simple case studies covering a range of scenarios are used to give students exposure to a range of possibilities. The resources are centred around a main livescript file which describes the scenarios and investigations. Within that there are short and straightforward code snippets to run the simulations and indeed the longest part of the code is for producing neat plots.

Heating system example Simple heating systems are already in the basic toolbox resources however, the purpose of this illustration is to show how SIMULINK allows the easy inclusion of non-linear, or other, components. The resource introduces two forms of non-linearity, that is delay and dead-zones. All 3 scenarios are within a single SIMULINK file as parallel but independent simulations and all the numerical parameters are predefined within the model.

Hydraulic system example Single tanks are common in the existing basic toolbox resources, but this resource considers different arrangements of tanks and management of the interconnections, as well as the chance to add non-linear components as required. The main focus is comparing parallel and series arrangements of tanks. The required parameter values definition, simulation and plotting is handled within the code sections of the livescript.

Car suspension example This resource introduces a car suspension scenario where the car drives over a ramp in the road at different speeds. The SIMULINK model illustrates common sources and sinks that can be used for importing and exporting data, both visually and into the MATLAB workspace. Specifically, here the source allows for non-simple input signals which are needed to capture the road profile. It also shows how deviation variables can be defined within the model and corrected to absolute values for plotting.

Position system example The focus here is a simple position system which uses a DC motor to move a lever (Figure 6). It shows how integration with MATLAB code can allow systematic and efficient investigations for a range of parameter values and scenarios including the use or not of non-linear components. This livescript file defines the required parameters and runs the simulations within code snippets in the different sections.

Submersible example This resource considers a simple autonomous underwater vehicle. The modelling equations can be approximated by a standard state-space model (Figure 7) but there are some non-linear elements. SIMULINK allows a simple and transparent implementation of the state dependencies with these nonlinear components, using transfer function blocks and lines.

3.3 Resources based on virtual laboratory interfaces

Position control systems and non-linear behaviours This resource is similar to the position control above, but now based on an interactive app and with core learning outcomes linked to non-linear behaviour. The user can investigate the impact of different nonlinear components in the system such as: measurement noise, input saturation,

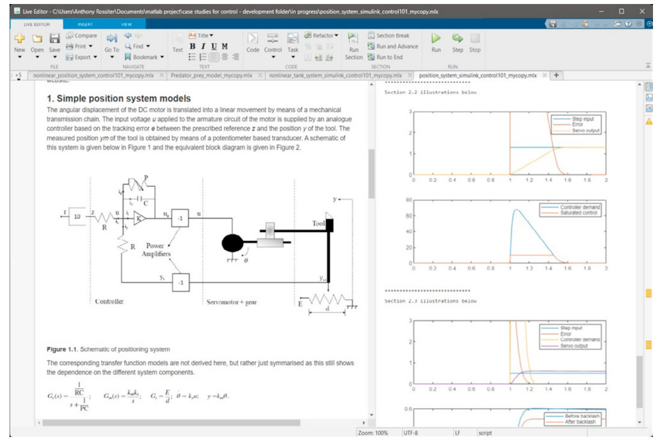

Fig. 6. Screen dump from the non-linear tank control interface.

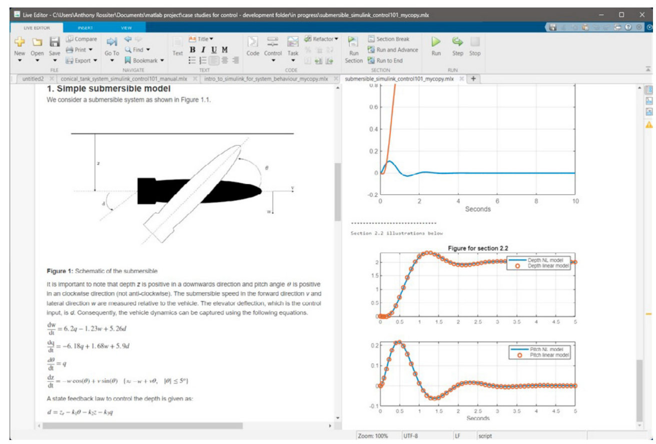

Fig. 7. Screen dump from the livescript focusing on modelling and control of a submersible.

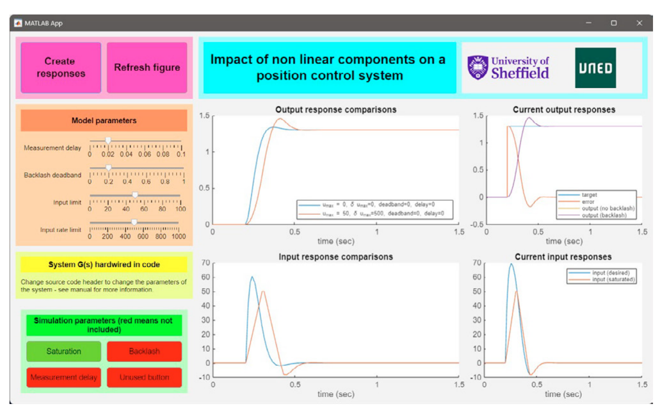

Fig. 8. Screen dump from the virtual laboratory on position control.

backlash and parameter uncertainty (Figure 8). Users can also change some core parameters.

Behaviour of a nonlinear tank level system and PI control While the basic toolbox resources used linearised tank level models, this resource explores the difference with using a more accurate non-linear model; the focus is on the non-linear dependence of the flow rate through a restriction on pressure difference. The simulation also investigates the impacts of changes in input flow and blockages in the

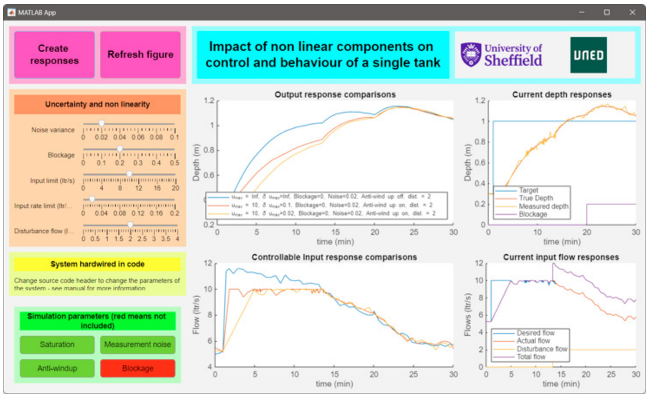

Fig. 9. Screen dump from the non-linear tank control interface.

output pipe both of which cause a movement away from the linearisation point. The manual gives guidance on the underlying scenario and how to use the SIMULINK files directly.

A separate but linked resource introduces feedback and a number of other external and non-linear impacts such as measurement noise, input saturation, anti-windup, disturbances and parameter uncertainty (Figure 9). Users can also change core parameters to perform compare and contrast simulations with different assumptions. The manual gives guidance on the underlying scenario and how to use the SIMULINK files directly.

Conical tank behaviour and control These files are quite similar to the previous tank ones but with the main difference that the tank is now conical in shape thus introducing a significantly different non-linear characteristic, the change in cross-sectional area with depth. Users are able to observe and understand the limitations of linear models by investigations on this app (Figures 1-4).

Predator - prey models Predator prey models are interesting text book examples of non-linear systems and have far more complex and sometimes complicated behaviour compared to linear systems which at times appear to behave in a somewhat chaotic fashion. This scenario also implicitly contains feedback between the different model states. The scenarios and choices in the app allow for relatively simple changes in the parameters, but these are enough for users to observe and understand the associated complexity of the behaviours (Figure 10).

3.4 State space examples

Within the new state space section, there is an introduction to using SIMULINK with state space models and state feedback. SIMULINK files are used to introduce the concepts of feedforward to improve tracking, observers to estimate states, integral action and also to demonstrate feedback efficacy in the presence of uncertainty.

4. CONCLUSIONS AND REFLECTIONS

This paper has give a rapid introduction to the inclusion of non-linear system behaviour and control within the control101 toolbox. Some brief background has been given

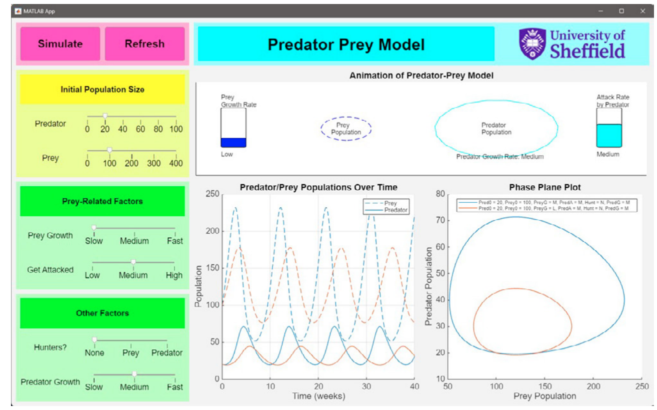

Fig. 10. Screen dump from the predator-prey behaviours interface.

on the motivation for the toolbox and the choice of the MATLAB/SIMULINK environment and then the focus has been on some specific resources on non-linear systems.

While it is accepted that dealing with non-linearity in any technical and analytical depth is beyond normal expectations for a normal first course in control, nevertheless, motivation and exposure to real issues are important parts of such a course. These resources can be used quickly and simply, both within lectures and student private study, to help students understand the challenges they will face in the future as well as gaining some realism on the extent to which the simple linear approaches they will initially study are useful.

A number of resources have been summarised concisely and it should be clear that these have been produced to allow students to engage at a conceptual level while masking the fine detail. The author recognises the limitations in coverage, both within given examples and the breadth of examples and would encourage colleagues to help him improve and add to these resources which are freely available to all.

REFERENCES

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

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

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.

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

S. Dormido, 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.

S. Dormido, 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.

Douglas, B., 2022, Resourcium, https://resourcium.org/

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

Goodwin G.C., A. M. Medioli, W. Sher, L. B. Vlacic, and J. S. Welsh, 2011, Emulation-based virtual laboratories: A low-cost alternative to physical experiments in control engineering education, IEEE Transactions on Education, 54:48-55.

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

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

Murray, R.M., Waydo, S., Cremean, L. and Mabuchi, H., 2004, A new approach to teaching feedback, IEEE Control Systems Magazine, 24, 38-42

Nevaranta, N., Jaatinen, P., Gräsbeck, K. and Pyrhönen, O., 2019, Interactive Learning Material for Control Engineering Education Using Matlab Live Scripts, IEEE 17th International Conference on Industrial Informatics, pp. 1150-1154, doi: 10.1109/INDIN41052.2019.8972282.

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.

Quanser, 2022, Quanser experience controls take home app, https://www.quanser.com/experience-controls.

Rossiter, J.A., Giaouris, D., Mitchell, R., and McKenna, P., Typical control curricula and using software for teaching/assessment: a UK perspective, IFAC world congress (2008).

Rossiter, J.A. and L. Gray, 2010, Supporting development of independent learning skills. Engineering Education.

Rossiter, J.A., 2011, Which technology can really enhance learning within engineering?, International Journal of Electrical Engineering Education, 48, 3, pp231-244.

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

Rossiter, J.A., 2017, Using interactive tools to create an enthusiasm for control in aerospace and chemical engineers, IFAC world congress (ifacpapersonline).

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 .

Rossiter, J.A., Pope, S.A., Jones, B. Ll. and Hedengren, J.D., Evaluation and demonstration of take home laboratory kit, IFAC Symposium on Advances on Control Education, Philadelphia, 2019.

Rossiter, J.A., Serbezov, A., Visioli, A., Zakova, K. and Huba, M., 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, 2020. https://doi.org/10.1016/j.ifacsc.2020.100092

Rossiter, J.A., 2021, Modelling, dynamics and control website, http://controleducation.group.shef.ac.uk/mainindex.html

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

Rossiter, J.A., 2024, A novel MATLAB toolbox for Control101 courses, European Journal of Control, 2024. https://doi.org/10.1016/j.ejcon.2024.101041

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.

D.L. Silverstein, Vigeant, M.A and Staehle, M., 2016, How do we teach process control: 2015 survey results, ASEE Annual Conference.

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