Article:

Rossiter, J.A. orcid.org/0000-0002-1336-0633 (2024) A novel MATLAB toolbox for Control101 courses. European Journal of Control, 80 (A). 101041. ISSN 0947-3580

https://doi.org/10.1016/j.ejcon.2024.101041

ARTICLE IN PRESS

European Journal of Control xxx (xxxx) xxx

Contents lists available at ScienceDirect

European Journal of Control

journal homepage: www.sciencedirect.com/journal/european-journal-of-control

A novel MATLAB toolbox for Control101 courses

J.A. Rossiter

University of Sheffield, Department of Automatic Control and Systems Engineering, Sheffield, S1 3JD, S. Yorks, UK

ARTICLE INFO

Recommended by T. Parisini

Keywords:
MATLAB
Control101
Control education
Toolbox

ABSTRACT

In recent years the educationally focused parts of the global control community have given some focus to what constitutes a sensible first course in control (Rossiter et al., 2020) and how to support this with high quality learning and teaching resources. This paper contributes to that overall effort in that it provides an example of high quality, open-access resources to support students in their independent learning. One aspect of the *ideal* first course in control is the suggestion that we (the community) focus more on concepts and understanding and less on tedious paper and pen calculations; to do this the community needs suitable easy to use software for performing computations and producing illustrations. Hence the author is leading what he hopes will be a collaborative community project on creating a MATLAB toolbox to provide such software. The purpose of this paper is to highlight the toolbox, present its current contents and thus enable staff to evaluate and adopt this toolbox and moreover, to reflect on how it might be improved.

1. Introduction

Both the IFAC and IEEE technical committees on control education have a role in promoting best practice for control education. This remit is wide ranging and typically has focused on highlighting individual examples of best practice, for example specific laboratories, coding architectures to support remote laboratories, text books and so on Albertos (2017), Dormido, Sanchez-Moreno, Vargas, de la Torre, and Heradio (2011), Dormido, Vargas, and Sanchez (2012), Quanser (2022), Rossiter, Pasik-Duncan, Dormido, Vlacic, Jones et al. (2018) and Douglas (2022). More recently some effort was put into the curriculum and there was interest in whether the global community had shared views on what constituted an ideal first course in control — here summarised as Control101 (Rossiter, Serbezov, Visioli, Zakova, & Huba, 2020). Happily there was reasonable consensus on the core principles, albeit as expected, some variance on fine details which is inevitable given the variety of contexts in different institutions. In essence a simple summary (https://sites.google.com/sheffield.ac.uk/controleducation/mainindex) was that:

1. Focus on concepts and make sure students really understand the power and importance of feedback across a range of scenarios.
2. Do not let tedious paper and pen computations be an obstacle to acquiring point 1. While mathematics is important, there is time enough for rigour in later courses.
3. Following point 2, use software wisely to support reinforcement of concepts and interesting illustrations to help with motivation and context (e.g. industrial case studies).
4. Inclusion of appropriate laboratories is essential.

Following on from the above ethos, the community is now keen to provide a repository of easy and free to access high quality learning and teaching resources that international colleagues can access (Koch, Lorenzen, Pauli, & Allgower, 2020; Rossiter, Visioli, Serbezov, Hedengren, & Zakova, 2022; Serbezov, Zakova, Visioli, Rossiter, Douglas et al., 2022). Of course the internet is littered with many examples, of which only a few can be referenced here, and thus the intention is to provide a set of pathways (e.g. Douglas, 2022) for users through these resources so they can find, quickly, the most appropriate resources for their needs; this latter task is ongoing.

This particular paper gives attention to two aspects of the educational resources:

1. How do we (the community) remove the need for students to do tedious paper and pen computations so they can spend more time on learning core concepts, and indeed do not end a control course equating it to just mathematics (a common historical perspective)?
This a widely reported problem e.g. Murray and Åström (2021). Many students complete a first control course but never really appreciate what control is nor why it is important? The general feeling is that this is because they become too focused on the mathematical skills required to manage aspects such as Laplace transforms, overshoot computations, pole calculations and so forth and thus miss the bigger picture. One obvious method to overcome this issue is to remove the emphasis on tedious number crunching, that is, let the computer do the mundane computations which have little intellectual benefit; then more emphasis can be placed on core control concepts.

E-mail address: j.a.rossiter@sheffield.ac.uk.

https://doi.org/10.1016/j.ejcon.2024.101041
Received 9 May 2024; Accepted 10 June 2024
Available online 12 June 2024

Please cite this article as: J.A. Rossiter, *European Journal of Control* , https://doi.org/10.1016/j.ejcon.2024.101041

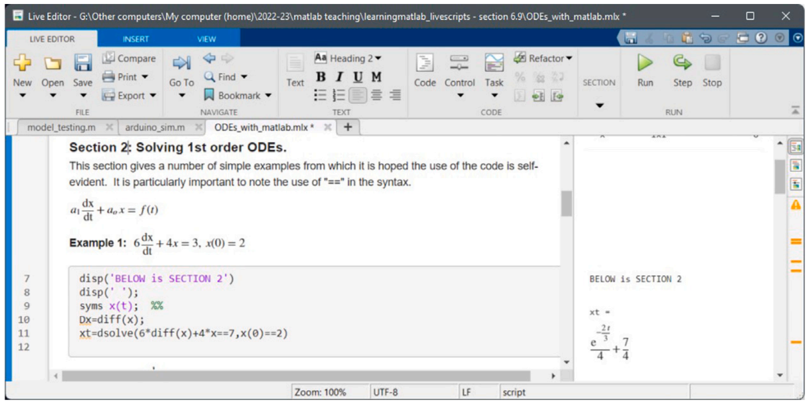

Fig. 1. Illustration of a livescript interface.

2. How does the community demonstrate the full breadth of control (Rossiter et al., 2023) with interesting and visually authentic illustrations that allow students to relate control and feedback to real and important challenges for the future?
   There is a consensus that a first course should include numerous case studies which demonstrate the breadth and usefulness of control. However, these are more convincing if they include video/animation and thus capture the interest of the students and allow them to appreciate the power of feedback in action. Consequently, it is good to choose case studies that are both visual and also in scenarios students will readily relate to.

The proposed solution summarised in this paper is unsurprisingly split into two, but these are then combined. The proposal in this paper is to produce a MATLAB toolbox which covers both components. Section 2 gives a brief historic review and rationale for using MATLAB. Then Section 3 covers what could be considered the more straightforward aspect of supporting mathematical computations and Section 4 looks at visualisation of important concepts. Section 5 completes the paper with some reflections and challenges to the community.

2. Historic overview and why use MATLAB?

Visualisation is important for learning, it helps students to see engineering in context and thus understand the role and importance better. Moreover, a concept is no longer just a mathematical equation, rather it can be related to some real system and its behaviour. Thus, a core motivating factor behind the toolbox was to allow students to visualise control and feedback.

One could argue that a good historical way of enabling visualisation and relating topics to the real world was through laboratories. However, it is well known that laboratory provision has a number of weaknesses such as: high cost, poor accessibility due to numbers, space and timetabling, limited opportunity to perform open-ended studies. As a consequence, much work in the community has gone into the provision of both remote and virtual laboratories (Abdulwahed, 2010; Matisak, Pohancenik, & Zakova, 2023; Panza, Invernizzi, Giurato, Yang, Chen et al., 2021; Rossiter, 2022; Silverstein, Vigeant, & Staehle, 2016; Zapata-Rivera, Larrondo-Petrie, & Weinthal, 2019). This paper focuses specifically on virtual laboratories (Cameron, 2009; de la Torre et al., 2013; de la Torre, Neustock, Herring, Chacon, Clemente et al., 2020; Ferrari & Visioli, 2023; Rossiter, 2016, 2017) because they can provide close to authentic interfaces and visualisations alongside important attributes such as:

1. 24/7 access.
2. Parallel or independent and simultaneous access by an entire cohort, no matter the class size.
3. Fast run times (essentially instantaneous) hence allowing rapid testing and design and thus learning.
4. Notionally cost free and available to unlimited numbers.

However, even virtual laboratories can have their limitations, for example due to the need for staff expertise and time to develop and code or the use of specific software to run them. In truth this trade off has no easy solution. The author's viewpoint is that the most severe pinch point or constraint is academic staff time hence the community needs solutions that typical academics can utilise in reasonable time. Many of the more flexible solutions and nicer looking (e.g. Matisak et al., 2023) such as those that run solely on a web browser require far more coding expertise and time and thus are not feasible for most teaching staff.

In summary then, the author has chosen to use the MATLAB software package. Obviously this entails some licensing cost, but many institutions (including the author's) provide this centrally for all their students and thus such a choice will mean the toolbox is available to large numbers of students across the world, free at the point of use! More importantly, as compared to some of the free software alternatives:

1. It has very broad functionality and a large number of powerful built-in functions for standard problems and tasks. This reduces and simplifies any coding requirements.
2. The environment is largely intuitive with simple coding, so it is quick and easy to create useful resources for the students to use (Rossiter, 2016, 2017).
3. Students can use the resources for visualisation and learning with relatively little training.
4. Tools such as MATLAB online and MATLAB web servers, in principle at least, allow students to run files on mobile phones and other simple devices.

The expectation is that well-designed interactive resources help students become more engaged and provide important feedback opportunities and thus are a valuable supplement to independent learning.

A final important point is that MATLAB CENTRAL can act as an easy to find and well maintained repository. Hence, the decision to create a formal MATLAB toolbox means that the resources are easy to find, download and use with the minimum of expertise from the user. These days download and installation of a MATLAB toolbox is done simply by selecting *add-ons* in the MATLAB window, searching for the desired toolbox (here control101) and then pressing install.

3. Supporting the typical computations needed in a control101 course

3.1. Context

In many historical Control101 courses, students spent a significant amount of time and effort reinforcing core concepts through laborious

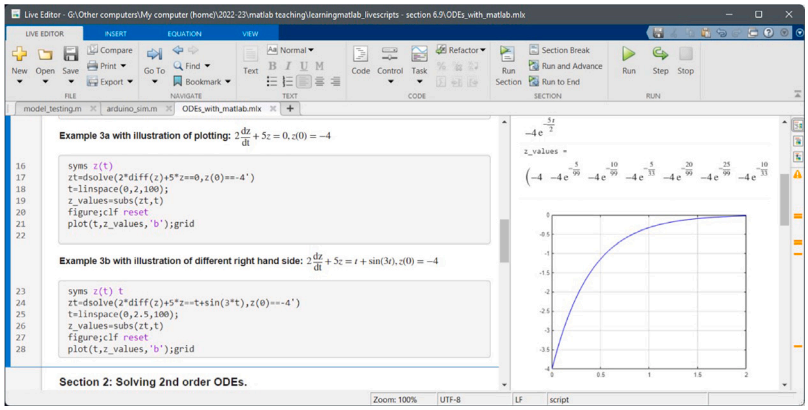

Fig. 2. Illustration of a livescript interface including a figure.

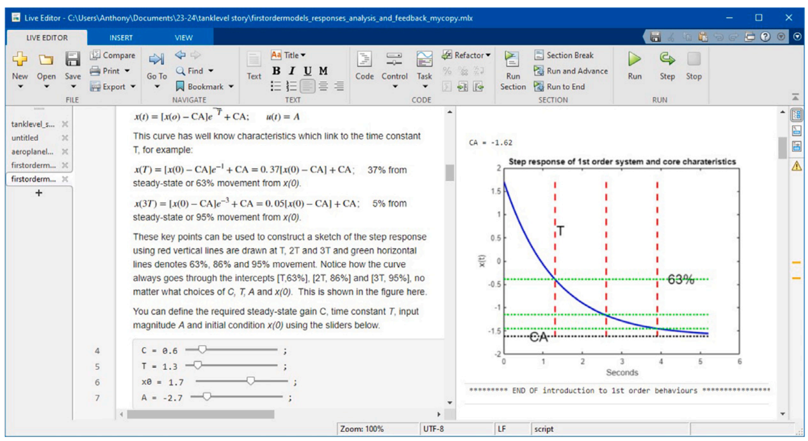

Fig. 3. Illustration of the use of sliders in a livescript.

paper and pen computations on simple examples. While this is important at some level, there comes a point where the pen and paper computations are a barrier to efficient progress and understanding because the process is too slow. Once students have mastered the core concepts, it makes more sense to let a computer do the tedious number crunching so students can focus more on interpretation, analysis, evaluation and design. Effective software enables easy visualisation of procedures, analytical solutions, graphs and more which allows students to engage with concepts and explore questions such as: *what if I change parameter X?*

3.2. The livescript environment

For many years the author has provided students with simple MATLAB code snippets (in m-files) for doing basic control related computations, but to his continued surprise, increasingly students seem to have difficulty interpreting even the simplest few lines of code, and thus exploiting effectively for their own studies.

However, in recent years MATLAB has introduced *livescript* files (Antunes, 2021; Nevaranta, Jaatinen, Gräsbeck, & Pyrhönen, 2019). These act as a smoother buffer to the code as they can be surrounded by explanation, figures and equations in the format of standard notes. Fig. 1 gives an illustration for the simple context of solving a 1st order linear ODE. The left hand window gives a easy to read explanation of the context alongside areas with a light grey background which are the associated code. The mathematical solution, numbers and/or plots appear in the right hand window (see Fig. 2). The hope is that the code snippets should be easier to understand to the user, and specifically, how they can change parameters of their choice to solve their own problems (more figures available in Rossiter et al., 2022).

Indeed one can also include simple interactive devices such as sliders (and others) into livescripts with automated update of the computations to make it even easier for students to change the parameters without the need to understand the underlying code (e.g. Fig. 3).

With livescripts, students see a well presented engineering argument and development alongside suitable code to support computation and solving. This means the association between suitable code and mathematics is more evident. This means they can now change the coding parts in real time and immediately re-run them to update computations and figures, thus gaining an appreciation of the links between coding syntax and mathematical problem solving, as well as the advantage of rapid number crunching and visualisation. In summary, livescripts have the potential to improve student learning of engineering concepts where MATLAB computations and visualisation are helpful and, as a useful aside, to learn how to code in parallel. Consequently, a suitable suite of livescript files is a logical inclusion in a Control101 toolbox.

3.3. Summary of control101 toolbox livescript tutorial files

Some preliminary tests with livescript files were carried out with multiple student cohorts over the 2022–23 academic year (Rossiter et al., 2022) and the summary would be that these were far better received and used than the code snippets used previously. Based on this experience, and the template Control101 course described in the survey (Rossiter et al., 2020), a preliminary set of livescript files (Table 1) has been described for the initial toolbox in the Autumn of 2023.

In simple terms, and for brevity the content flow is summarised as: Data handling and plotting in the MATLAB environment, defining

Table 1
List of livescripts in August 2023 (not including the app manuals).

plotting_with_matlab
ODEs_with_matlab
firstordermodels_in_matlab
firstordermodels_responses_analysis_and_feedback
secondordermodels_in_matlab
secondordermodels_and_underdamping
laplace_transforms_with_matlab
partial_fractions_with_matlab
transferfunctions_and_poles
transferfunctions_and_behaviours
step_responses_with_matlab
closedloop_transferfunctions_with_feedback
closed_vs_openloop_overlay
closed_loop_compare_multiple_compensators
closedloop_offset_and_poles
freq_responses_with_matlab
bode_asymptotes
proportional_design_with_bode
lag_design_with_bode
lead_design_with_bode
lead_lag_design_with_bode
delays_and_bode
time_series_models
time_series_models_and_recursion
discrete_models_and_usage
discretisation_and_bode.m

models for low order linear systems, plotting behaviours of low order systems, Laplace transforms, poles/zeros and the complex plane, quantification of system behaviours, generating closed-loop transfer functions and analysing closed-loop behaviour, analysis of simple feedback control laws. The toolbox also contains further files which may be in a first or second course depending on the institution. A more complete summary can be seen on the web page: https://sites.google.com/sheffield.ac.uk/controleducation/matlabresources/section-6-9-matlab-live-scripts.

Remark 1. It should be emphasised that the toolbox is intended to be an ongoing community shared resource and the initial set of files is not expected to be comprehensive nor include what everyone would like. The authors welcome proposals for additions and indeed are already aware of obvious gaps such as an introduction to state space methods.

Ideally, users would implicitly gain a working knowledge of the MATLAB basic control files so that they are more empowered to create personalised files in the future, as required. In fact, the number of base functions required for Control101 can be quite small, for example the following functions cover most things a student is likely to need: *plot.m, dsolve.m, tf.m, step.m, pzmap.m, dcgain.m, feedback.m*.

The authors reiterate that the emphasis is on simple rather than advanced MATLAB usage. MATLAB is used primarily to remove the need for tedious number crunching which has little academic value, so the student can focus on core concepts and understanding. More detailed and advanced MATLAB or coding, if required in the curriculum, should be covered elsewhere. Someone looking at the example livescripts will find that the predominant or apparent complexity in the code snippets provided is actually managing the plotting, and not the underlying computations.

Remark 2. When opening a livescript from the toolbox, instead of the source (which is thus protected), MATLAB will open a copy (with file name ending in _mycopy) in the present working directory. This means the user can make and save edits to their own working area.

4. Developing of virtual laboratories and visualisations

The second core component of the toolbox is the virtual laboratories component. A virtual laboratory focuses on visualisation of core engineering, or here control, concepts, encouraging users to relate what they are learning in class to real applications. The interface should allow suitable interactions and design decisions and thus implicitly guide students to investigate the impact of core parameter changes, thus improving their understanding of the context and scenario.

An important benefit of visualisation, and having virtual laboratories in an accessible environment, is that you can also use them actively in lectures. It is straightforward for the lecturer to run some simulation from the front to illustrate core concepts and scenarios, but moreover, small interactive segments during the lecture also allow students to play with and discuss these in small groups given the run times are short (about 10 s per simulation with animation).

4.1. Summary of selected case studies

As indicated in the introduction, the prime focus on the early version of the toolbox was case studies which are both highly visual and also relate strongly to typical student life experience so that they can appreciated the power and importance of feedback. Consequently, several scenarios have been constructed, but the author is keen to receive and include more from his worldwide colleagues. A selected few are summarised next.

- Tank level systems. Most students have a good intuitive understanding of the flow of water through pipes and into and out of tanks and thus can relate to a tank level control problem quite easily, and the impact of uncertain flows, blockages and more.
- Cruise control of domestic vehicles. Many if not most students have a good experience of driving and can also relate easily to speed control problems and the impact of mass, slopes, wind and more on behaviour, thus this is a logical example to use.
- House temperature. We all know how challenging it can be to maintain a comfortable room temperature in the face of external factors (cold wind, poor insulation, open doors, etc.) and thus again this scenario is easily relatable and intuitive colours can be used to represent hot and cold.

4.2. Simple design principles deployed in the toolbox

With any context, it is always possible to add more realism, more facets, more behaviours and so forth. However, for students studying a first course in control the increased authenticity may be counter productive as students may be dazzled by the complexity and thus lose sight of the simple concepts with which they should begin. Thus, largely the author has focused on simple interfaces (de la Torre et al., 2020; Rossiter, 2016, 2022) which allow relatively limited user interaction. Indeed, in some cases a good virtual laboratory may communicate only one simple concept: if the interface is easy to use and the concept is communicated clearly then the designer has succeeded.

In summary then, a core principle adopted in the toolbox is that sometimes it is better to have numerous simple interfaces, each covering just one principle concept, as opposed to having fewer interfaces each covering multiple concepts. Again, it is emphasised that the focus here is a first course in control and the objective is to enthuse and excite students about why control is important? There is plenty of time in the future, once you have enticed them, to cover more complete scenarios and to add complexity.

A second core principle is the visualisation aspect. Although the MATLAB apps largely have *cartoon* like drawings, the aim is for these to be authentic enough for users to link them to the real scenario and thus recognise the relevance to the real world. Moreover, most of the apps include animation so that users can see the real system changing dynamically. Again, the intention is that the animation helps the user relate what they are viewing to real world behaviours and thus to develop an appreciation for what a behaviour is, why it is important and moreover, how parameters changes affect behaviour?

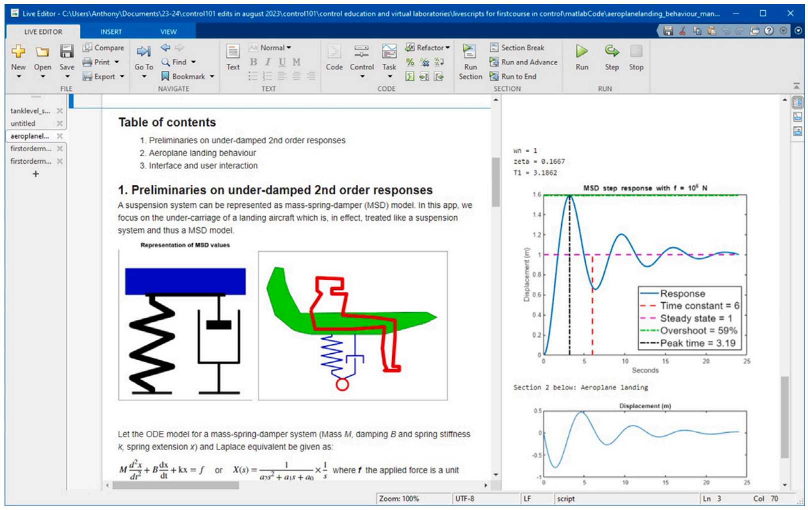

Fig. 4. Illustration of the livescript manual for the aeroplane landing app.

Table 2
List of apps in August 2023 (see toolbox for more recent additions).

1st order model behaviour characterisation
1st order modelling from responses
Mixing tank behaviour
Tank level behaviour
Tank level control with PI
Tank level story file
Car speed behaviour
Cruise control with PI
House temperature and thermal system behaviour
DC servo modelling and control
Mass–spring–damper systems and landing aircraft behaviour
Inverted pendulum and human in the loop control
PI tuning practice
Aeroplane roll control with lead compensation
Behaviours of linear models

A third and important principle is the help documentation. It is important for the background information to be embedded in a simple and consistent way, that is, within the toolbox itself. Hence, a decision was taken to generate a manual for each app that is in the form of a livescript. The manual contains background discussion on the engineering scenario within the app including core equations and analysis, and some MATLAB code for core computations. It also gives indications on how to use the app. Finally, a critical point is that, assuming the apps are being deployed from within the toolbox, the manual is opened automatically so that users do not need to search for it. An example of a manual is given in Fig. 4. Manuals can be opened separately if needs be as the syntax is always *appfilename_manual.mlx*.

4.3. Summary of virtual laboratories/apps in first release

For the time being, the initial release of the toolbox has relatively traditional engineering scenarios (Table 2), although the plan is to augment this with more diverse scenarios (Rossiter et al., 2023) in the future, as time and resource permits, and indeed hopefully with contributions from others. The discussion below introduces some, but not all due to space, of the apps available. For a complete and up to date list see the toolbox (e.g. "doc Control101 toolbox") or the author's website https://sites.google.com/sheffield.ac.uk/controleducation/matlabresources/section-6-8-matlab-apps-for-interactive-learning.

The first few apps allow students to explore first order behaviours and demonstrate that a range of simple and common systems have predominantly first order behaviour. A good example is car speed, with the dynamics represented as a mass–damper system. The users can modify the mass, the friction and thus see how these affect changes behaviour and also uncertainty represented as the road slope and wind speed (Fig. 5). In parallel, an equally simple app focuses on the time constant form for a first order model, and how behaviour is impacted by changes in the time constant and gain (Fig. 6). There also apps using tank level, house temperature and mixing tank scenarios (and more scenarios in development).

Some of the apps focus on proportional and integral (PI) design. For example, the tank level scenario has a separate app where the focus is on control in the presence of uncertainty and similarly, there is a follow on app for the car speed app where the focus is on PI design to control speed. There is also, a generic app (Fig. 7) which focuses on PI design for a range of models (low and high order).

There are also some more challenging scenarios such as control of the inverted pendulum and position and speed control of a DC servo.

4.4. Story line apps

Story line apps are more complete and currently there are not many although there is an intention to develop more. The intention of these is for students to use these at the end of a course to communicate and clarify the entire control design process in a single app, for example (Fig. 8):

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 tabs to undertake the different steps in a design process.

4.5. Future developments

The author wanted to get a minimal set of working apps available for the 2023–24 academic year, and thus recognises the limitations of the apps available at the point of writing. He is aware of several other apps in development from colleagues in his department and elsewhere and thus by the summer of 2024, the list will be more extensive, and hopefully, will keep growing. Moreover, the author re-emphasises that

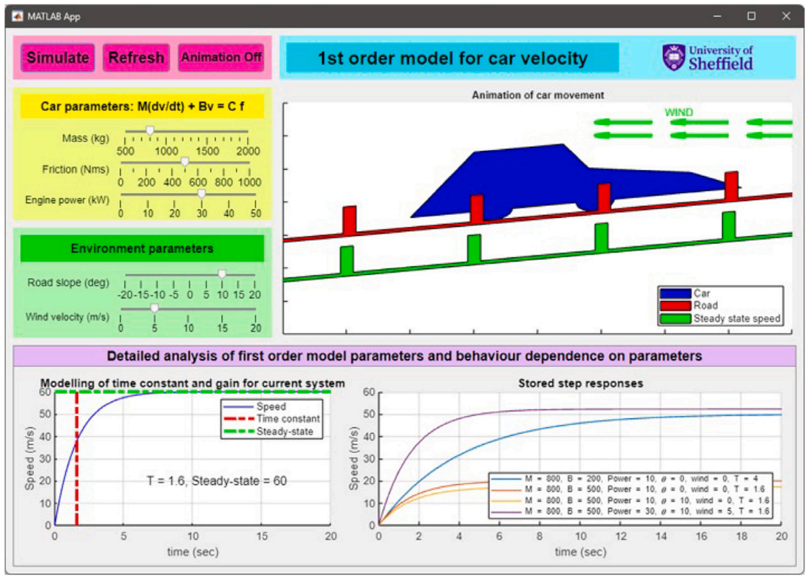

Fig. 5. Illustration of the app on car speed.

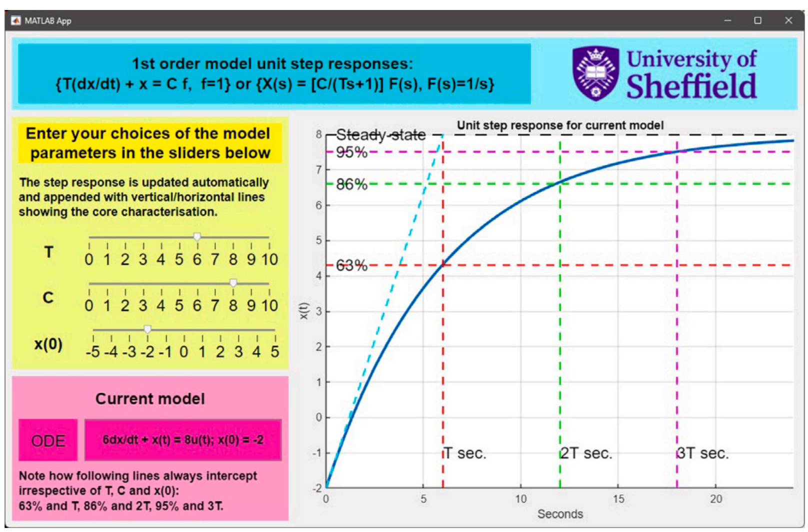

Fig. 6. Illustration of the app on first order behaviours.

he hopes this toolbox will be community owned so that anyone can approach him with a proposal for additions and/or edits to make the overall resources more useful; already some international colleagues have done this.

5. Conclusions and reflections

This paper has given an overview of a new MATLAB toolbox designed specifically for a Control101 course. The idea is to immunise students as much as possible from detailed coding and tedious number crunching, so they can focus on the control engineering concepts and gaining an understanding of why feedback is important? It is emphasised that while coding and mathematics are important, these can be taught elsewhere and too often can be an unhelpful distraction from the more important control concepts in a first control course.

The reason for creating the resources in the form of a MATLAB toolbox is to optimise accessibility and usability for those with access to MATLAB, which is a large number of students and staff worldwide. Moreover, coding is relatively straightforward in the MATLAB environment which means a wide range of resources can be produced in a relatively short time compared to using other environments. Download of the toolbox typically takes 30 s to a minute and thus is very fast and easy which also means it is straightforward for both staff and students to utilise within lectures, even if they had not downloaded previously. Doing in the form of a toolbox also means that accessing and using the files does not require users to engage with the complexities of search paths, filenames or complicated run statements; easy access soft links are provided.

The toolbox is comprised of two types of files:

1. App files are in essence virtual laboratories. These are used to help students relate their control learning to numerous real world scenarios and to gain insight into core concepts, for example: why does behaviour change and why is feedback important for managing this?
2. Livescript files provide access to code snippets doing the number crunching and plotting for numerous common scenarios and showing the required analysis in a convenient format. Students can also use these to gain understanding of how to use MATLAB as a tool in supporting analysis and design for their own individual problems.

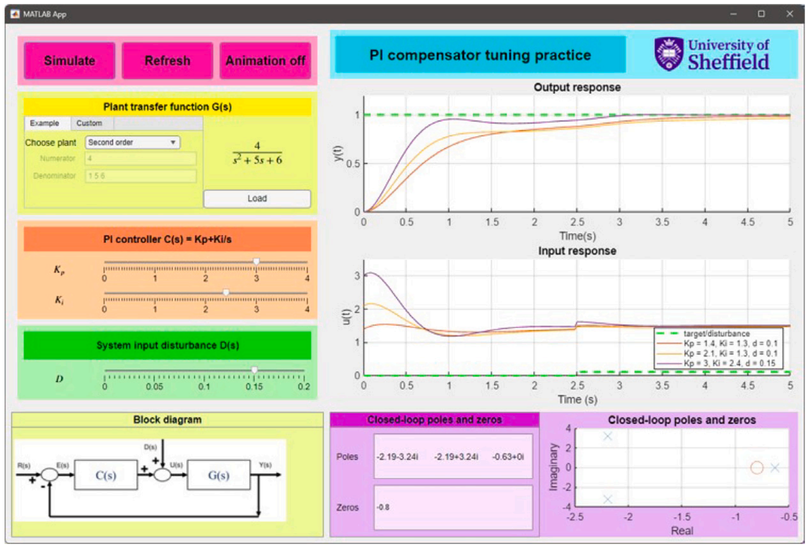

Fig. 7. Illustration of the app PI design.

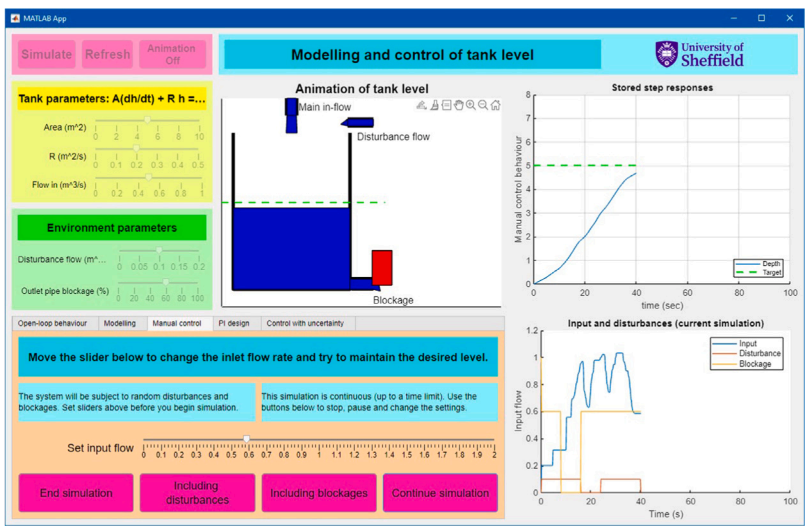

Fig. 8. Illustration of the tank level story line app.

3. The sources files are available to those who want them from the author's website and these form a good start point for those wanting to develop resources of their own.

The toolbox was formally launched in the Autumn of 2023 and so now the author would like to disseminate its existence and encourage more international colleagues to help shape its contents so that it is of best use to students and staff across the globe. Specifically, the authors want more colleagues to give feedback and indeed propose contributions of their own to be included for later releases. Assuming the toolbox help documentation is suitably presented, there is no reason why the resources cannot grow to cover a far wider range of topics such as state space, discrete control, control design methods and obviously, more varied scenarios.

CRediT authorship contribution statement

J.A. Rossiter: Conceptualization, Project administration, Resources, Software, Validation, Visualization, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

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

References

Abdulwahed, M. (2010). *Towards enhancing laboratory education by the development and evaluation of the trilab concept* (Ph.D. thesis), University of Loughborough.

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

Antunes, D. J. (2021). Using MATLAB live scripts to teach optimal control and dynamic programming online. https://es.mathworks.com/company/newsletters/articles/using-matlab-live-scripts-to-teach-optimal-control-and-dynamic-programming-online.html.

Cameron, I. (2009). Pedagogy and immersive environments in the curriculum. In *Blended learning conference* (pp. 290–294).

de la Torre, L., Heradio, R., Jara, C. A., Sanchez, J., Dormido, S., Torres, F., & Candelas, F. (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., & Hesselink, L. (2020). IEEE transactions on industrial informatics. *Automatic Generation and Easy Deployment of Digitized Laboratories*, *12*(16), 7328–7337. http://dx.doi.org/10.1109/TII.2020.2977113.

Dormido, S., Sanchez-Moreno, J., Vargas, H., de la Torre, L., & Heradio, R. (2011). UNED labs: A network of virtual and remote laboratories. In Javier Garca Zuba, & Gustavo R. Alves (Eds.), *Using using remote labs in education: Two little ducks in remote experimentation* (pp. 253–270). Bilbao.

Dormido, S., Vargas, H., & Sanchez, J. (2012). AutomatL@bs consortium: A spanish network of web-based labs for control engineering education. In A. Azad, M. E. Auer, & V. J. Harward (Eds.), *Internet accessible remote laboratories: Scalable E-learning tools for engineering and science discipline*: *vol. 11*, (pp. 206–225). IGI Global.

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

Ferrari, M., & Visioli, A. (2023). An educational interactive software tool to learn cascade control design. In *IFAC world congress*.

Koch, A., Lorenzen, M., Pauli, P., & Allgower, F. (2020). Facilitating learning progress in a first control course via matlab apps? In *IFAC world congress*.

Matisak, J., Pohancenik, M., & Zakova, K. (2023). Platform for virtual laboratory of mechatronic systems in augmented reality. In *EXPAT 2023 and IEEE explore*.

Murray, R. M., & Åström, K. J. (2021). *Feedback systems: An introduction for scientists and engineers* (2nd edition). Princeton University Press.

Nevaranta, N., Jaatinen, P., Gräsbeck, K., & Pyrhönen, O. (2019). Interactive learning material for control engineering education using matlab live scripts. In *IEEE 17th international conference on industrial informatics* (pp. 1150–1154). http://dx.doi.org/10.1109/INDIN41052.2019.8972282.

Panza, S., Invernizzi, D., Giurato, M., Yang, G., Chen, K., & Parisini, T. (2021). Integration of experimental activities into remote teaching using a quadrotor test-bed. In *21st IFAC workshop on aerospace control education*.

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

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

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

Rossiter, J. A. (2022). MATLAB apps to support the learning and understanding of simple system dynamics. In *IFAC symposium on advances in control education*.

Rossiter, J. A., Pasik-Duncan, B., Dormido, S., Vlacic, L., Jones, B., & Murray, R. (2018). Good practice in control education. *European Journal of Engineering Education*, http://dx.doi.org/10.1080/03043797.2018.1428530.

Rossiter, J. A., Serbezov, A., Visioli, A., Zakova, K., & 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*, *13*, Article 100092. http://dx.doi.org/10.1016/j.ifacsc.2020.100092, 15.

Rossiter, J. A., Visioli, A., Serbezov, A., Hedengren, J., & Zakova, K. (2022). Open access resources to support learning of control engineering. In *European control conference*.

(numerous), Rossiter, J. A. (2023). Control education for societal-scale challenges: A community roadmap. *Annual Reviews in Control*, *55*, 1–17. http://dx.doi.org/10.1016/j.arcontrol.2023.03.007.

Serbezov, A., Zakova, K., Visioli, A., Rossiter, J. A., Douglas, B., & Hedengren, J. (2022). Open access resources to support the first course in feedback, dynamics and control. In *IFAC symposium on advances in control education*.

Silverstein, D. L., Vigeant, M. A, & Staehle, M. (2016). How do we teach process control: 2015 survey results. In *ASEE annual conference*.

Zapata-Rivera, L. F., Larrondo-Petrie, M. M., & Weinthal, C. P. (2019). Generation of multiple interfaces for hybrid online laboratory experiments based on smart laboratory learning objects. In *IEEE frontiers in education conference* (FIE), (pp. 1–8). http://dx.doi.org/10.1109/FIE43999.2019.9028421.