Evaluation of student software tools for supporting an understanding of PID tuning

J. A. Rossiter *

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

Abstract: There are many software tools available for students to implement PID control laws, but these software tools are not all equally flexible, easy to use, accessible and hence effective. This paper gives a brief review of three tools: (i) author built tools in MATLAB (Mathworks, 2017); (ii) TSC software (TSC, 2017) and (iii) PISIM software (Postlethwaite, 2017). The evaluation considers aspects such as convenience and cost for the academic as well as accessibility and useability for the students. The main results in the paper are written in such a way as to assist staff in deciding in what software its worth investing time and effort developing resources and student activities.

Keywords: Virtual laboratories, staff efficiency, student engagement, independent learning.

1. INTRODUCTION

An important part of the student learning experience is the opportunity to try things out in a laboratory and indeed this is a core accreditation requirement the world over. Apart from supporting some trial and error or investigative learning, laboratories also help students engage with technical content and recognise its relevance to the real world problems them will face in employment. However, it is equally recognised that time in a hardware laboratory is subject to a number of constraints, such as:

(1) Equipment and laboratory space is often expensive so available in limited numbers.
(2) A combination of limited equipment numbers and thus a requirement for multiple cohorts alongside timetabling restrictions often implies students have limited access to hardware.

While the most significant constraints can be ameliorated by developing large laboratory spaces with large numbrs of identical equipment (DIAMOND, 2016; STEMLAB, 2016), such economies of scale effectively imply the existence of multiple engineering departments who wish to access similar equipment, but at different timetable slots. Often, this will simply not be true. In consequence, there has in recent years been a rapid growth in so-called virtual laboratores (Cameron, 2009; Fabregas et al., 2011; Dormido et al, 2012; Goodwin, 2010; Goodwin et al., 2011; Perez et al., 2011; Guzman et al., 2006), that is, laboratories which can accessed 24/7 via computer and which emulate a real system in an authentic manner. The main advantage of virtual laboratories is improved accessibility:

(1) No timetable limitations. In principle, students access the software anywhere and anytime.
(2) No number restrictions. Again, in principle, any number of students can access the laboratory simultaneously and independently.

The reader may note the emphasis on the words *in principle*: some viritual laboratories require expensive software licenses which can potentially limit the number of instantaneous users and also, may need to be installed on local hardware as opposed to hosted on a webserver.

This paper will not discuss the popular alternative of remote laboratories (Abdulwahed, 2010; Chen et al., 2013; Qiao et al., 2012; Rossiter et al., 2011, 2014; Vargas et al., 2011; de la Torre et al., 2013) as these rarely overcome the access problems for large classes, typically requiring serial access by users. Moreover, the overheads in maintenance, initial design and build can be substantial and thus they require significant in house expertise and committment.

In summary, this paper takes the arguments for virtual laboratories as now well accepted in the community and literature. The focus here is to evaluate some of the tools available for academics interested in exposing students to PI design. There is no attempt to undertake a comprehensive review as any evaluation involving students is limited to the breadth of activities students can reasonably be expected to engage in. Typically, for any fixed cohort, that would be just one piece of software although an argument could be made that one can always also use MATLAB as that is a tool most engineering students gain profficiency in regardless. In the author's institution therefore, the students are required to use two alternative pieces of software: (i) TSC (TSC, 2017) and (ii) MATLAB (Mathworks, 2017). As it was desirable to gain an evaluation of a further alternative, volunteers were also sought to make use of PISIM (Postlethwaite, 2017). These three are chosen primarily for pragmatic reasons, that is they are affordable enough and available to staff in Sheffield.

The paper is organised as follows. Section 2 describes the alternative software choices, section 3 provides a student evaluation, section 4 provides some staff reflections and the paper then finishes with some conclusions.

2. DESCRIPTION OF THREE SOFTWARE ALTERNATIVES

This section describes the alternative choices of software environment in Sheffield along with practical details of how the software might be utilised. A key assumption throughout is that the Chemical Engineering department, which is the main user for the context of this paper, is interested in two main aspects.

- The software should be as authentic as possible while recognising that software that is actually common place in industry is too expensive and thus such a requirement should be treated as idealistic and therefore is likely not to be enforced to rigidly.
- Students should be able to engage with activities that allow them to gain a realistic impression of how process control is used in industry and the sorts of choices and designs that will be required (however at an introductory level only).

2.1 TSC software

The TSC software is a relatively cheap alternative to commerical software such as Aspen and Hysys in that it almost replicates the look and feel of process control room, but with relatively limited functionality so that students can engage more quickly. The software provides a finite number of fixed installations and architectures so that users are not able to change the implied physical build. However, users can change flow rates, control law settings, disturbances, open and shut valves and so forth. Standard alarms are built in as would be expected in a process control room. The presentation of the interface and signal/component labelling is also intended to be close to authentic, see figure 1 for an an illustration of the heat exchanger.

A number of the basic scenarios support PID compensation and in fact a main focus is the behaviours of chemical engineering processes. Examples include batch reactor, columns, level control, separators, condensers, evaporators, heat exchangers, fractionation. Basic exercises and instructions are provided for each scenario. On the main window, instruments can be opened (some indicative faceplates are opened on figure 1) to view internal variables such as valve openings and PID values. Alarms are included and the ability to plot trends (see figure 2), although this display is somewhat crude. Signals and instruments have a unique numeric identifier which is on both the block diagram and the trends. Consequently, the software provides a simple environment for students to undertake observational/data based modelling of simple chemical processes by deploying step changes to selected inputs. It also allows them to practice the design and implementation of PID controllers on standard control loops, as would be appropriate for an introductory process control course. However, these investigations are somewhat clumsy (slow) and there is a lack of transparency about some of the underlying assumptions such as the definition of the PID compensator (parallel or series).

So, in summary, students have a safe and authentic environment in which to engage with process dynamics and the impact of different control room decisions on the behaviour, but TSC does not lend itself conveniently to the information students may desire for detailed analysis within a university assignment or laboratory.

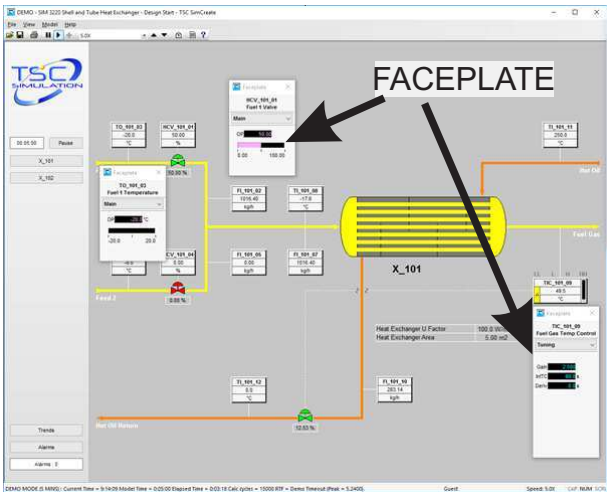

Fig. 1. Heat exchanger simulation on TSC software.

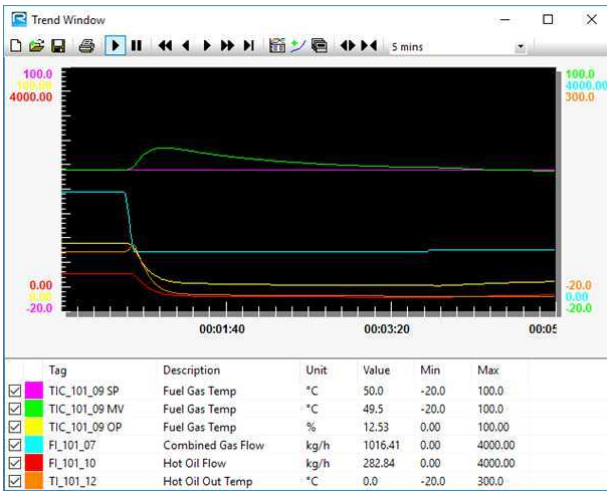

Fig. 2. Illustration of state trends on TSC software.

2.2 PISIM

PISIM is a new software variant orginally developed by staff at Strathclyde University, which is currently under testing with a view to releasing soon; contact the author for more information on status (pisimdev@gmail.com). Like TSC, the intention is for this to be far cheaper and more accessible to common commerical alternatives, but in this case an order of magnitude cheaper than TSC! A focus was given on some priorities (Postlethwaite, 2017), to quote: *(i) The process and control system needed to be represented on a diagram as close to the ANSI/ISA-5.1-2009 standard as possible and be configured from this diagram (see figure 3); (b) On-screen devices that looked like real control room equipment should be generated automatically when the P&ID symbols were created. (iii) The devices should be capable of being transferred to simulated control panels or SCADA screens to allow operator HMIs to be simulated; (iv) The system (process+controllers) should be simulated directly from the P&ID interface with all the on-screen devices being updated in real or accelerated time. (v) The software should be stand-alone and capable*

of being run on normal students personal computers. (vi) The software should include a system for learner support, running course material and recording performance.

Hence, one could emphasis that a key point was the desire of authenticity, the look and feel of the interface should be as close as possible to what students would encounter in a real control room.

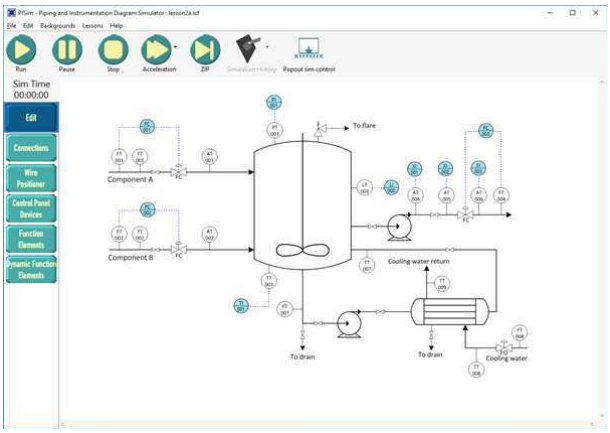

Fig. 3. Screen capture from main interface on PISim

PISIM has a very different mindset from TSC. In this case the intention is for students to be much more involved in the architecture, so that although a base architecture might be supplied (figure 3), this will have key gaps such as missing sensors, connections and/or compensators (figure 4). Students need to learn how to define and connect these in. The control panel displays are also authentic (figure 5). The default line graphs (not shown) are low quality (similar to figure 2) although an option exists to produce higher quality versions on laboratory completion. A further critical difference with PISIM which is a potentially significant advantage over competitors is that it can embed a lesson management system. Not only do users have individual logins and individual records of activity, but the laboratory steps can also be managed so that progress is prevented when a precursor step has been done incorrectly. Helpful notes such as powerpoint slides can be embedded into the system so students are given specific guidance at each step, as required. Individual student records of activity are stored and can be marked automatically although the software is still under development so the author could not fully observe this functionality.

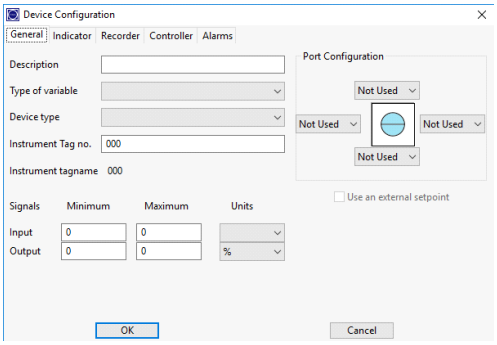

Fig. 4. Device editor on PISIM

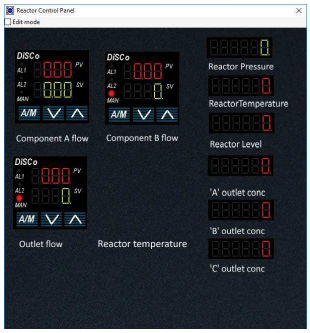

Fig. 5. Control panel display on PISIM

2.3 Background on using MATLAB to develop Virtual Laboratories

An assumption in this paper (Rossiter, 2016) is that most engineering students are proficient with MATLAB, and thus any activities using MATLAB require minimal additional learning of the environment. Moreover, many institutions supply site licenses, including for personal computers, thus access to any MATLAB activities is 24/7 and fully parallel. Consequently, author has chosen to develop MATLAB GUIs to support students learning across multiple departments Rossiter (2012, 2016, 2017).

A MATLAB GUI interface is less authentic than TSC and PISIM (figures 6,7). However they are far more easily tailored to be intuitive to use, so the focus can be on learning outcomes such as understanding and learning PI, without the need for several hours familiarisation with notation and conventions. The guide interface allows the user to develop an interface in an intuitive fashion containin numerous interactive elements such as sliders, pushbuttons, editable text, real-time line graphs and animations, radio buttons and more. Basic competence with MATLAB programming is sufficient to develop moderately elaborate interfaces. Critically, it is straightforward to include significant animation and pictures to represent real systems behaviour and indeed this aspect is typically much better presented than TSC or PISIM.

The author has developed several GUIS which support the learning of PI design, but here illustrate just two of those.

Heat exchanger GUI with uncertainty The dynamics of a heat exchanger, volume V, F the flow rate through the heat exchanger and Q the flow rate of steam (condensing steam provides the heat supply) are approximated by a simple 1st order ODE with two inputs: (i) the heat supply and (ii) temperature of the inlet flow T_{in}. Simple disturbances are changes T_{in} and F. The interface is shown in figure 6 allows intuitive modification of parameters and observation of different dynamics.

GUI for control of level in a tank with disturbances Level control in a tank can be modelled with a 1st order ODE where A is the cross-sectional area, F_{in} is the flow in, R is in effect a resistance to outlet flow and h is the depth. Uncertainty is represented by allowing changes in R and F_{in}. It should be clear from figure 7 that parameter modification and PI design is transparent thus enabling qualitative investigations of impacts on behaviour.

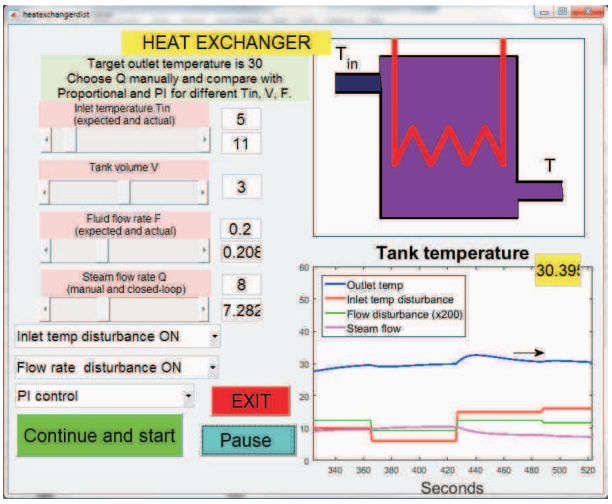

Fig. 6. MATLAB GUI for heat exchanger under PI control.

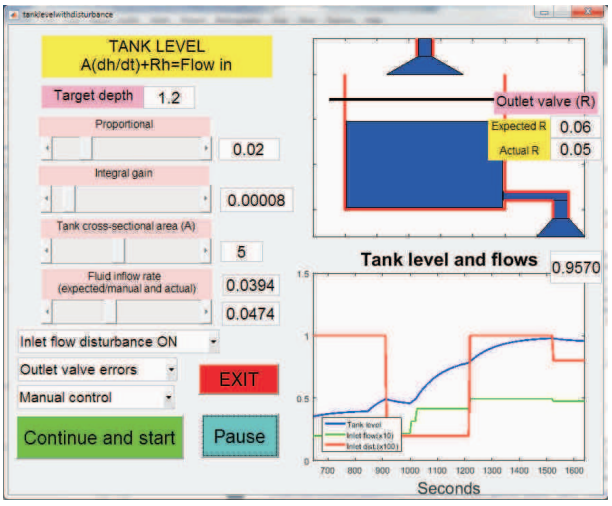

Fig. 7. MATLAB GUI to illustrate impact of uncertainty on level control of a tank.

3. STUDENT ENGAGEMENT AND EVALUATION

A core part of this paper contribution is to gain some insight into student views on the efficacy of different learning experiences. There are over 150 students in the 3rd year cohort taking the control module and the use of the both the MATLAB and TSC components were included in the laboratory activities. Use of PISIM was much more limited and this discussed separately.

3.1 TSC and MATLAB evaluation questionnaire

In order to gain student views, a number of simple questions were posed using a Likert scale (5 being the most positive, 1 most negative and 3 neutral) alongside opportunities to provide textual feedback; the average scorings are summarised in table 1. Broadly these indicate:

(1) Both the choices were effective in achieving the learning goals.
(2) MATLAB scored better on accessibility.

The students gave a number of helpful textual comments which there is not space to include. On the whole these are positive but they also give clear indications of potential improvements. In all cases TIME was the major factor; students wanted more dedicated familiarisation time and teaching with the software before using in earnest.

Question	TSC	MATLAB
How easy was the software to use (include familiarisation time)?	2.99	2.87
How accessible was the software?	2.88	3.24
How effective was the environment in giving a good learning experience?	2.82	3.36
Was the environment effective at capturing the relevance to chemical engineering?	3.27	3.32
Did using the environment help you improve your knowledge of process dynamics and behaviours?	3.19	3.16
Did using the environment help you improve your understanding of the importance of process control/feedback to chemical engineering?	3.23	3.31
Did using the environment help you improve your understanding of PI compensation?	2.93	2.96

Table 1. Student evaluation of two software environments on scale of 1-5 (5 is best, 3 is good student satisfaction).

[TSC]: On the whole students liked the software and supporting materials (professionally sourced) but were frustrated by poor accessibility which made preparation and reinforcement difficult. Should be taught TSC earlier in programme.

[MATLAB] Good for qualitative issues but axes too small for acquiring quantitative details, need a reset button, want to be taught MATLAB earlier in degree programme, wanted more detailed supporting materials.

3.2 PISIM student evaluation

It was not possible to get the same student cohort to perform a comprehensive evaluation of PISIM for the simple factor of avoiding student overload. However, the author of PISIM has carried out some longitudinal evaluation at the University of Strathclyde (Postlethwaite, 2017) with the following conclusions.

- The majority felt the PISIM component significantly helped with their basic control knowledge (in the context of chemical engineering).
- The majority felt the PISIM component significantly helped with their understanding of the importance of control (in chemical engineering).

More importantly, student confidence seem to grow each year as the PISIM usage was improved and increased. Of course there were a number of minor criticisms, mostly covered in the staff evaluation to follow, but on balance the students felt it was a positive component of the control module. Of course, having the software delivered and supported by the author will always help and thus one might also ask to what extent the feedback may change with a less skilled producer of the embedded activities?

It should be noted here that a key current weakness of PISIM environment is that the PI parameters (and indeed any other parameters) cannot be changed on the fly. A

simulation needs to be stopped and restarted from zero before any parameter changes are affected.

4. STAFF EVALUATION OF THE SOFTWARE ALTERNATIVES

A key observation is that no software, no matter how good or easy to access, is likely to be effective if students do not use it. It is well accepted that often a majority of students is unlikely to engage with optional activities: if you do not mark it, they will not do it! Consequently this means that the software has to be embedded into module assessment, either through laboratories, assignments or quizzes and this means it is rarely possible to take an off the shelf product or off the shelf activities prepared by the original author. Activities need to be carefully designed and planned to fit into the module assessment.

In summary, a key evaluation criteria is the extent to which it is straightforward and efficient for staff to develop the activities they want students to do. What level of staff proficiency is required and how long will it take to produce a resource/laboratory session that is effective for student learning? The following subsections gives some reflections on the three alternatives from a staff perspective.

4.1 Accessibility

MATLAB has the advantage in many institutions of being available via a site liences and indeed, in many cases students can download this to their own laptops and thus have unlimited access; laboratories and access do not need to be timetabled. This paper assumes such a scenario in the evaluation. Use of MATLAB may be considered an overhead for students who use the software for just one or two modules but otherwise the familiarisation load is incidental. The GUI files themselves are very small and thus easy to supply to students.

TSC is less expensive than software such as ASPEN, but still not cheap and as a consequence in Sheffield just 36 licenses are available (at an initial cost of close to 40K). However, an even more serious impediment to accessibility is that the software has to be downloaded to specified computers and thus is only available in a single computer room in the entire university. Thus, trying to guarantee access to this room, against competing requests from other departments is a challenge and becomes a serious impediment to accessibility, especially as students can be processed only 36 at a time. One mitigating factor is that a demo version is freely available on the entire network, but this is limited to very short simulation run times and thus is hard to use even for basic familiarisation. Students need about 3 hours familiarisation which is a notable overhead if not covered earlier in the degree programme.

PISIM has been written to be both cheap and accessible. The trial or demonstration version (which is quite limited) can be deployed on the entire university network through a shared memory server and thus does not need procurement and installation with central computer services. It will even run on anyone's laptop (but not MACINTOSH). The intention is that the final version will be licensed and installed more rigorously, but nevertheless the cost is expected to be of the order of just 1000 p.a. for unlimited licenses within a single institution and thus can be made available in any computer room. The software has quite specific requirements and also is designed to ensure students understand a large number of chemical engineering principles, not just control design, and thus typically students would need to do about 3 separate laboratories to get to a point where they could use the software to tackle interesting problems. However, much of this could be done in the students own time due to the network access and principle of automated marking based on login and individual student activity record!

4.2 Staff preparation

MATLAB Most academic staff have reasonable basic profficiency in MATLAB and thus would require just a 30-60min familiarisation with the operation of GUIs before they could develop activities. The author's estimate is that a reasonable GUI, with 2 page overview of functionality and underlying engineering principles, takes about one day to develop. This does not include detailed consideration of embedding into student assessment but given MATLAB is typically available 24/7, the author favours a focus more open ended activities students can do in their own time rather than formalised to a laboratory session.

TSC TSC comes with prepared scenarios and thus staff preparation requires identifcation of the particular settings and operating points that are most appropriate for different learning outcomes and in effect, this requires some time playing with the environment to discern what it can do. As the environment is likely to be unfamiliar, it will likely take about one day for a staff member to gain enough proficiency and familiarity before they can begin detailed planning of how to use the software for specific learning outcomes. A key weakness is the limited ability to edit of any of the scenarios. All you can really change is values of inputs (valve positions auto and manual) and PID parameters where control loops are included. Hence, the scenario is best for illustrating process control in situ. However, the tool is somewhat clumsy to use for demonstrating both system modelling and the tuning of PI compensators as the interface and displays are not created with these learning outcomes as central objectives.

In consequence, the lecturer needs to prepare supporting material very carefully and give quite precise sequential directions to students on what values to assign to different valves, flows, PID compensators, how to set up the trends graph, and so forth. The underlying physics/chemistry/equations (such as the structure (parallel or series) of the PID assumed) are not easy to access and have notable gaps which makes it difficult to provide precise numerical questions linked to this software. In summary, good to support general chemical engineering, authentic interfaces and scenarios, but somewhat clumsy to use for demonstrating PID tuning.

PISIM PISIM requires a higher level of chemical engineering competence but does feel authentic. The scenarios have many parameters which need to be assigned sensibly to define a template from which students can begin, including numerous specifications for each device (e.g. see figure

4). While this is valuable in the context of a chemical engineering degree, it could be a time consuming distraction where the main focus is on a understanding of basic control concepts and process dynamics. Moreover, the design is intended to be a sequenced set of activities, decisions and marking/progress points which must be coded up from the lecturer side before being released to students. This will require a substantial initial time investment for staff to gain sufficient expertise in coding the software. Moreover, there is an implied load in managing the creation and distribution of individual student login details and collation of the PISIM records they create. Adoption of PISIM is likely to need whole department buy in, so that both students and staff have time to exploit value from this tool and any laboratory preparation time is merited by improved student experience. Similar statements will apply to other tools which require fundamental first principles input by the teacher and students.

5. CONCLUSION

The software evaluated here comes at three different levels of staff and student effort. The more effort both staff and students put in, the more the potential learning outcomes.

(1) MATLAB easy to code, relatively quick and easy to develop learning tools, good accessibility but lacks authenticity and likely to cover fairly limited scenarios. Good where staff and students have limited time.
(2) TSC is designed to be authentic and demonstrate several key issues. However, the scenarios while having the advantage of coming prepared are rather inflexible and are somewhat clunky to use so that students will find it harder work learning the core engineering principles behind basic control design. Also very expensive. If you are prepared to pay for authenticity and low staff workload, this is a good option.
(3) PISIM is an authentic interface, but with several addtional attributes as compared to TSC including automated marking and access to core system parameters. Hence, lecturers can design their own systems and their own lesson plans. However, such designs require significant expertise and time to create. PISIM has a further advantage of being cheaper and more accessible than TSC. Requires more effort than TSC, but gives far more versatility so this is a good option if you are taking a big picture view of student development and across multiple modules.

REFERENCES

Abdulwahed, M, 2010, Towards enhancing laboratory education by the development and evaluation of the trilab concept, PhD Thesis, University of Loughborough

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

Chen, X., G. Song, and Y. Zhang, 2010, Virtual and remote laboratory development: A review. In Earth and Space, Engineering, Science, Construction, and Operations in Challenging Environments, 3843-3852. doi: 10.1061/41096(366)368.

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.

DIAMOND building, www.sheffield.ac.uk/diamond (checked 14/3/18)

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.

Fabregas, E., G. Farias, S. Dormido-Canto, S. Dormido, and F. Esquembre, 2011, Developing a remote laboratory for engineering education. Computers & Education 57:1686-1697.

Goodwin, G. , 2010, Virtual laboratories for control systems design. http://www.virtual-laboratories.com/(checked 1/9/10).

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.

Guzman, J., K. Astrom, S. Dormido, T. Hagglund and Y. Piguet, 2006, Interactive learning modules for pid Control, IFAC symposium on Advances in Cont. Educ.

Mathworks, https://uk.mathworks.com/ (checked 14/3/18)

Postlethwaite, B., 2017, Demonstration of PISim software for teaching chemical process control, IFAC world congress.

Perez, J., S. Dormido and L. Vlacic, 2011, Enhancing student learning: On-line interactive laboratory for modelling of real world control system applications, IFAC world congress (IFAC papersonline), pp.7268-7273.

Y. Qiao, G. Liu, G. Zheng and C. Luo, 2012, Design and realization of networked control experiments in a web-based laboratory, Control 2012 (UKACC).

Rossiter, J.A., Y. Baradaranshokouhi, I. Lilley and C. Bacon, 2011, Developing web accessible laboratories for introductory systems and control using student projects, IFAC world congress (IFAC papersonline).

J.A. Rossiter, Br.LI. Jones, R.M. Murray, L. Vlacic, S. Dormido, 2014, Opportunities and good practice in control education: a survey, IFAC world congress.

Rossiter, J.A., 2015, Lectures and resources in modelling and control, http://controleducation.group.shef.ac.uk/indexwebbook.html (checked 14/3/18)

Rossiter, J.A., 2012, Using MATLAB to create cheap and accessible virtual laboratories, ISEE.

Rossiter, J.A., 2016, Low production cost virtual modelling and control laboratories for chemical engineering students, ACE (IFAC papersonline).

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

STEMLAB building, www.lboro.ac.uk/departments/meme/undergraduate/student-facilities/stemlab/ (checked 14/3/18)

TSC SimCreate by Technical Simulation Consultants Ltd., 2017.

Vargas, H., J. Sanchez, C. A. Jara, F. A. Candelas, F. Torres, S. Dormido, 2011, A Network of Automatic Control Web-based Laboratories, IEEE Transactions on Learning Technologies, 4, 3, 197-208.