

Simulation of Research-Grade Physics, Chemistry, and Engineering Experiments in LabVIEW as a Flexible Template for Remote Laboratories

R J Lewis¹

School of Physics and Astronomy, Cardiff University
Cardiff CF10 3AT, UK

T D Drysdale

School of Engineering, The University of Edinburgh
Edinburgh EH9 3FG, UK

Conference Key Areas: Open and Online Teaching and Learning, New Notions of Interdisciplinarity in Engineering Education

Keywords: Remote Laboratories Simulation Education Authentic

ABSTRACT

We propose to repurpose sophisticated experimental simulations used for testing of control and analysis software as the foundation for flexible, realistic, and robust remote access-simulated-resource-type [1] project-based learning in higher education.

It is impractical to implement undergraduate projects or laboratories directly on “big science” experiments by virtue of their uniqueness, rarity, and running costs. By implementing simulations we can take advantage of the benefits of scalability and reduced running costs [2] in addition to the unique affordances of simulations such as zero acquisition time, enhanced opportunity for iteration of technique, and learning outcomes adaptability [3], all while maintaining a realistic learning experience.

There is a danger with virtual laboratories that students “act before thinking” as opposed to “thinking before acting”, with potentially negative effects on their learning and the realism of the experience [4]. In order to minimise this effect and to ensure as realistic an experience as possible, we propose embedding the interaction with simulations within a facsimile of a research group environment which includes time budgeting and peer accountability.

We demonstrate the practicality of this concept by implementing a LabVIEW-based simulation of the KATRIN TILO (Karlsruhe Tritium Neutrino Experiment Test of Inner Loop) tritium gas assaying system [5] which can be adapted for physics, chemistry, and engineering projects. Multiple simulations run at different physical scales and variable timescales, taking advantage of LabVIEW's inherent parallelism, including the quantum mechanics of Raman scattering, isotope exchange mechanics, transmission efficiency of the light collection system, and a realistic interface for controlling the laser, gas handling, and spectral acquisition.

1 Corresponding author: LewisR54@cardiff.ac.uk

1 SIMULATIONS IN RESEARCH AND EDUCATIONAL ENVIRONMENTS

1.1 Simulations in Research Environments

In experimental physical sciences and engineering, simulations are routinely constructed by practising professionals at scales / levels of depth from fundamental physical interactions, through individual experimental components, to comprehensive simulations of entire experimental set-ups. It is beyond the scope of this paper to provide a comprehensive review of such simulations; the following examples from physics serve only to illustrate the depth, sophistication, and universality of simulations in the physical sciences.

At the fundamental level of physical processes, one encounters highly context- and application-specific single-use simulations in fields as diverse as the particle physics of hadronic showers [6], simulation of muon backgrounds for detector commissioning [7], and gravitational waveform simulations for black hole binary merger searches [8]. Multiphysics suites do exist, however, and are also widely used. Notable examples include COMSOL [9] and ANSYS [10].

At the application level of entire detectors or elements of detectors, one encounters more frequent use of general-purpose multiphysics and simulation tool-kits such as GEANT package, applied for example to calibrate calorimeters [11] or muon flux through the KATRIN main detector [12].

Such simulations and iterative comparison with experiment are a vital part of the design and evaluation process for devices of all scales, from portable neutron flux detectors [13], to large satellites [14], to entire “big science” experimental set-ups and facilities [15], [16]. This practice is nothing new in engineering and industry, however; aerospace have long used simulation- and hardware-in-the-loop [17], [18], and it remains a central pillar of modern engineering practice [19].

In summary, simulations in research and industrial environments are vital and ubiquitous. Their implementation is usually thoroughly planned, often highly complex, cover every conceivable time, energy, and length scale and are iteratively linked to experiment / implementation by design. As a consequence of their application, such simulations are restricted to use by the host research group / institution / company, and are very rarely adapted for wider use.

1.2 Simulations in Educational Environments

When considering the use of simulations in higher education in physical sciences and engineering, the concept of the “virtual laboratory” is commonly encountered [1], [2]. This term hints at the inextricability of a simulation from its educational context. In the field of physics education, for instance, the importance of learning concepts by means of constructing models is emphasised [20]. Formal modelling frameworks have been developed which emphasise the students’ engagement as practising researchers and the iterative nature of model construction [21], construction of models in a mixed-reality setting [22], and the effect of the blurred boundaries between physical and virtual

laboratories [23]. Indeed, the dividing line between what a practising scientist might consider a simulation and a virtual environment is itself blurred.

The blurring is further complicated by the module- / programme-level framework into which the laboratories are embedded; examples include realistic “practice-centred” project-based learning frameworks with time budgets, peer-accountability, and the real possibility of failure [3], and more traditional discrete “learning units” closer in form to typical undergraduate laboratories. The latter appears to be the dominant form; educational simulations / virtual laboratories are rarely “large scale”, in both the sense of operational complexity and breadth of the underlying concepts. Rather, simulated experiments are often virtual versions of simple undergraduate experiments [24]–[29]. This is in stark contrast to the research and industrial simulations already discussed.

Within the educational context, simulations / virtual laboratories are instances of active learning, which are well-known to have an overall positive effect on student learning [30]. Virtual laboratories have a number of potential enhancements over physical laboratories, what Nolen and Koretsky refer to as the “affordances” of the virtual environment. Affordances in turn influence the instructional design of a virtual laboratory, such as the overlaying of visual representations of invisible phenomena on a user interface (UI) [3].

Potkonjak et al. neatly summarise the advantages of virtual laboratories such as lower operational and maintenance costs due to the lack of physical equipment, ease of reconfiguration, and multiple, simultaneous (perhaps remote) access, balanced against disadvantages such as the necessity of (possibly large) computer and software resources, the typical lack of (often instructive) “bad” outcomes in a “safe” virtual environment [2]. Interestingly, Potkonjak et al. state that “the final stage in training ... requires real equipment” [2]. The use of the word “training” suggests a predefined idea of the purpose of virtual laboratories. In the context of physics virtual laboratories this is debatable, since understanding of underlying fundamental concepts is likely the intended learning outcome, rather than mastery of a particular experimental set-up.

In summary, the operating context of simulations as part or the entirety of a virtual laboratory is very different to that of research simulations, and are typically restricted (although not always) to small-scale defined-concept implementations, although they are evidently very effective in this role.

2 PROPOSAL: BRIDGE THE GAP BETWEEN THE RESEARCH AND EDUCATIONAL REGIMES

We observe that there is a gap in the educational application of simulations in virtual laboratories that lies between the simple, small-scale virtualisation of traditional laboratories, and the simulation / computer operation of large-scale experimental set-ups in research environments. More specifically, there is little opportunity for undergraduate students to experience operation of cutting-edge “large” experimental set-ups, and hence limited access to realistic experience of grappling with the complexities research-grade experimental work.

We propose the use of simulations of practical experimental set-ups as flexible, robust, and, above all, realistic virtual laboratories.

The general principle is either to adapt existing simulation and experimental control software for educational use and / or to develop applications which simulate the operation and physical processes underlying a particular experimental set-up. It is likely that a research group will have several such applications that could be adapted, or existing simulation elements that could be combined within a framework such as a LabVIEW project or a GUI-driven Python application.

An advantage of this approach is that institutions will already have the authors / maintainers of the source software and a team of experts in the field on-site. We note, however, that development time may be significant, so it is likely that a virtual laboratory lead would actually develop the application in consultation with the research group.

Specific implementations will naturally vary between disciplines and in light of the intended learning outcomes of the planned activities. In this paper we suggest one possible specific implementation in order to illustrate the general principle; the development of a simulation of a subsystem (TILO) of a large experiment (KATRIN), based in large part on repurposing existing code. We suggest, without prescription, the use of the LabVIEW development environment due the relative ease of UI development and inherent parallelism. We further suggest how this proof-of-principle implementation can be adapted for multiple disciplines (physics, chemistry, engineering), and adapted and scaled for use in a remotely- and multiply-accessed virtual environment.

2.1 General Simulation Requirements

A simulation of an educational experimental set-up will consist of several intercommunicating simulation elements, each of which represents a context-consistent “unit” of the simulated system. The simulation as a whole should satisfy the following requirements to meet the requirements of flexibility, robustness, and realism, bearing in mind the specific local and educational implementation contexts:

Modularity and user interaction: individual elements of a simulations should be able to combine and scale with other elements in a natural way. Each simulation element should run within its own process in parallel and communicate with other elements via a standardised interface. As a consequence, a top-level application will be necessary to coordinate communication between simulation elements and to handle user input.

Timescale compatibility: physical, chemical, and mechanical processes will occur on different characteristic timescales. Depending on the time budgeting and complexity required in a specific implementation, one or more of these timescales will be dominant. A simulation should therefore account for the possible requirement to average / integrate the observables of faster processes within a slower dominant timescale, or to reliably and realistically schedule slower processes within dominant fast processes. Ideally, this scheduling should be configurable to allow for efficient code re-use.

Time budgeting configurability: not all simulations can, or need to, run in real-time. Simulations should allow for the configuration across the spectrum of “instant” results, “instant” results with a time budget cost, and “real-time” operation. This spectrum broadly aligns with the requirements of undergraduate concepts-first teaching, problem-

/ project-based teaching, and system training, respectively. Time budgeting configurability allows both for efficient code re-use and flexibility of deployment.

Adjustable complexity: simulations of physical processes typically rely on assumptions and approximations, which can be progressively relaxed and refined to achieve the desired precision of an observable. The Raman scattering of light from diatomic molecules, for instance, can be simulated with increasing precision by modelling a diatomic molecule as a rigid rotor, a non-rigid rotor, including vibration-rotation interaction, including centrifugal distortion, and so on [31].

Realism: simulations should present a coherent UI which reacts realistically to user input, taking into account the local educational requirements and the time budgeting and complexity configuration of the implementation under consideration. This requirement does not prescribe absolute verisimilitude with a research-grade interface (although this might be appropriate in context), but rather that the essential character and layout is adequately captured [32]. A parallel here can be drawn with the design choices of flight simulator interfaces as implemented on consumer PC hardware.

Appropriate use of affordances: a degree of overlap exists with research-grade simulations and control applications, which can use visual, audio, or other cues to allow efficient user interaction. Examples include: the use of highlights to show whether a fluid is present and / or flowing in a vessel, colour highlighting to allow at-a-glance estimation of temperature of components, and overlays to increase the contrast of edges in low-contrast images [3]. Context and practicability will determine which affordances could be implemented, with prioritisation determined against development resources available.

3 LABVIEW AS A MODULAR SIMULATION PLATFORM

While it is not essential that LabVIEW be the programming language in which simulations are implemented, for virtual laboratories it has a number of features which make it particularly suitable:

Ubiquity: LabVIEW is often used to implement main control applications, auxiliary supporting applications, and system prototyping in physics [33]–[39], chemistry [40]–[44], and engineering [45]–[49] research and industrial contexts. LabVIEW is also prominent in higher education in physics (usually control of simple table-top experiments) [24], [25], [50]–[52], chemistry (particularly common since a surge of interest in the 2000s) [26], [27], [53], [54], and engineering (often NI ELVIS-based, usually problem- or project-based learning) [28], [29], [55]–[57]. It is worth noting here that the engineering education LabVIEW implementations are clearly the most integrated into existing curricula and best-supported overall. In all three disciplines, a large corpus of code is already in-place.

Inherent parallelism: LabVIEW implements the “G” dataflow language [58], whose visual design allows a “natural understandability of dataflow diagrams”, as noted by Kodosky et al. [59]. In G code, data flows between nodes by means of wires. If two sections of G code are not dependent on each other under the rules of dataflow, they are in parallel, and will run as separate processes. This dramatically simplifies the top-level architecture of simulation applications, and allows a large degree of modularity.

As of April 2018, the latest version of LabVIEW is LabVIEW 2018 [60], which is the version of LabVIEW used in this work. Note that there is also a parallel, mutually incompatible version of LabVIEW (LabVIEW NXG). Although LabVIEW NXG can be deployed successfully in an education environment [61], LabVIEW 2018 is used here as it is currently more mature, stable, and has better driver support [62].

3.2 Implementing a Modular Simulation in LabVIEW

A modular simulation can be implemented very straightforwardly in LabVIEW. The inherent parallelism of the language allows each simulation element to reside within its own for loop, which can run asynchronously with respect to other loops. Communication between loops can be achieved by means of queues, although notifiers (essentially lossy, single-element queues) or files might be more appropriate in different contexts. Dedicated state machines, also residing within their own for loops, handle UI monitoring and time budgeting accountancy.

The desired level of complexity and per-loop clock can be set either by controls on the front panel (UI) or by reading configuration settings from a file. The UI is monitored by a dedicated event-driven state machine which can communicate to loops and update UI elements programmatically. A dedicated time budgeting loop synchronises and logs the duration of user interactions and processes, particularly important in the “instant results with a time budget cost” operation mode.

The requirements of “realism” and “appropriate use of affordances” are primarily addressed by careful context-specific UI design, and in the manner in which the simulation activity is embedded in the module- and programme-level activities.

4 TILO PROOF-OF-CONCEPT: A REALISTIC LABORATORY SIMULATION

Having discussed the general motivations, simulation requirements, and a general LabVIEW template, we apply our proposed approach to a concrete example, TILO.

The Test of Inner Loop (TILO) experimental set-up was a mock-up of the Windowless Gaseous Tritium Source (WGTS) inner-loop gas handling system for the Karlsruhe Tritium Neutrino (KATRIN) experiment [5].

Hydrogen isotopologue mixtures are circulated around TILO’s gas handling system and a Raman detection system measures their relative abundances. Dissociation of hydrogen isotopologues can be achieved by passing gas mixtures through a heated permeator. The TILO system was capable of observing changes in the relative abundances of hydrogen isotopologues passing through the permeator with an acquisition time of about 100s. A schematic layout of TILO is shown in Figure 1.

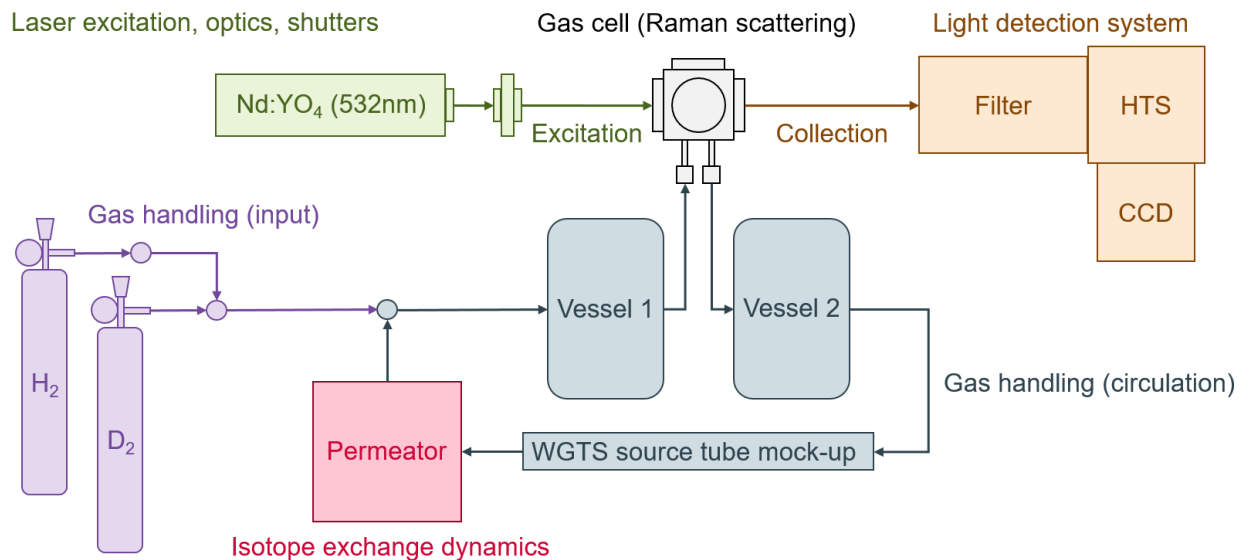


Figure 1. Schematic layout of TILO's principal experimental components. Note that principal components broadly map onto simulation elements. Adapted from [5].

TILO is a particularly useful example of an experimental set-up that contains multiple interacting elements. Simulation of TILO requires multiple mutually-interacting elements modelled at different physical and time scales, as summarised in Table 1. Simulation complexity would need to be chosen to suit local context. TILO's dominant timescale is $>1s$, primarily due to the long exposure times for the CCD detector, requiring integration / averaging of faster simulation elements.

Table 1. Principal simulation elements for the realistic simulation of TILO. Note that these simulation elements broadly map onto principal experimental components.

Elements	Basic timescale	Primary complexities (not exhaustive)
Raman scattering	10^{-14} to $10^{-13}s$	Scattering model depth, polarisation
Gas handling systems	10^{-3} to 100s	Connectivity model, diffusion, equilibration
Permeator vessel	1 to 100s	Isotope exchange dynamics, temperature
Light detection system	1 to 1000s	Transmission efficiency, noise, aberrations
Laser excitation	1 to 5000s	CW laser line shape, warm-up, mode hops
Safety infrastructure	$>10^{-6}s$	Interdependence of simulation elements
UI monitoring	Asynchronous	Implementation of simulated failures
Time budgeting	Asynchronous	Synchronising to dominant timescale ($>1s$)

The simulation elements summarised in Table 1 reside in separate for loops running in parallel. The UI monitoring loop is an event-driven state machine monitoring the front panel and acting as the master loop. Individual elements are state machines, slaved to the master loop. Inter-loop communication is via queues. A flexible, tab-based GUI provides a realistic, colour-coded, focussed view of the simulation, shown in Figure 2.

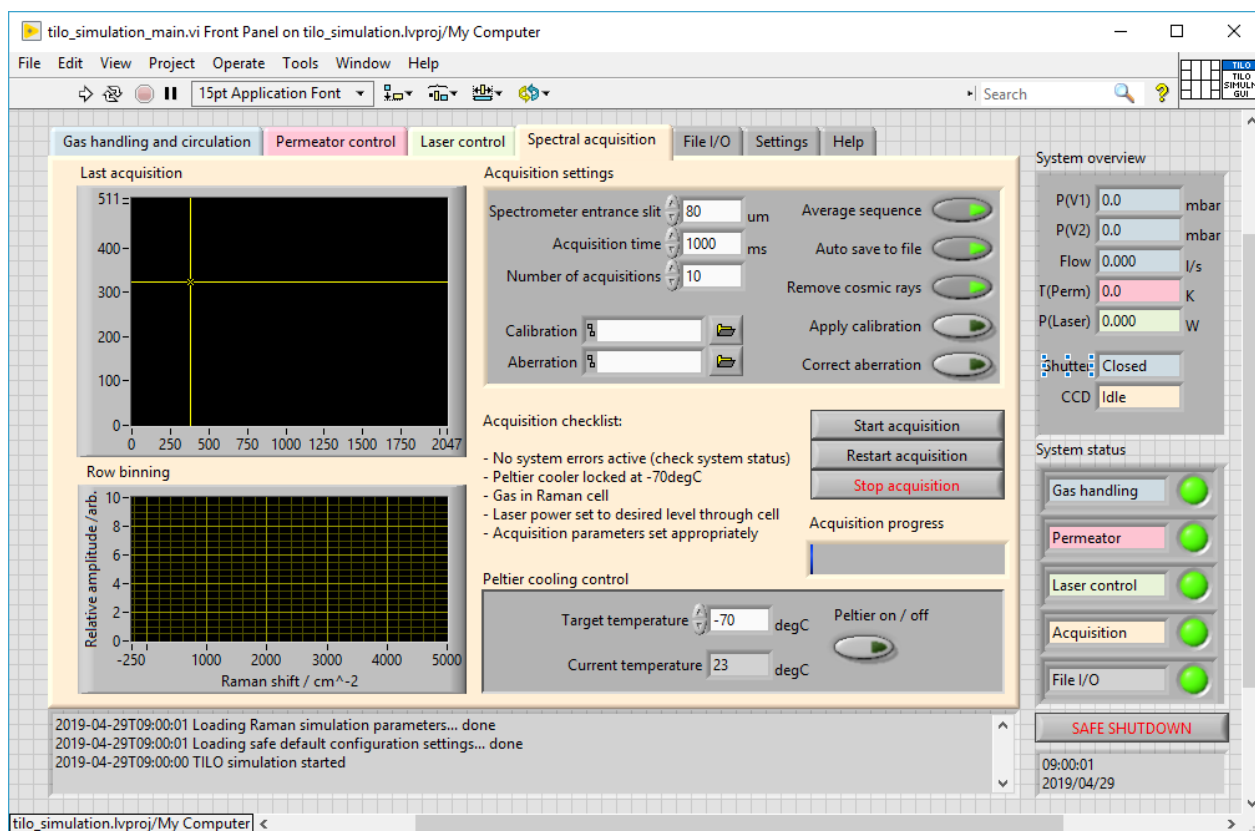


Figure 2. Tab-based GUI for the TILO simulation, with “Spectral acquisition” in focus.

4.2 Possible Operational Contexts for the TILO Simulation

The TILO simulation outlined in this work could be deployed appropriately in a number of educational contexts. Keeping the same UI and using the three time budgeting models outlined in Section 2.1, one can envisage the following three prototypical use cases:

Undergraduate problem-based learning (instant results): this model emphasises exploration of concepts and rapid iteration. Students would work in pairs or small groups. Activities would centre around correctly feeding in partial pressures of gas mixtures and finding a S/N optimal acquisition time / number of acquisitions for the Raman system. Measurements with / without gas circulation would be compared with fundamental calculations made off-line. Interactions between students would be primarily concerned with protocol, data I/O, and data analysis. Active learning and quick, iterative interactions with the simulation, together support developing an intuitive feel for the fundamental processes at work.

In this use case, complexity would be limited to the desired precision of the gas mixing and Raman scattering. While the light detection system would respond realistically (with noise, etc.), behaviour such as unstable system temperatures and unstable laser wavelengths would not be modelled. Session time: circa 2 to 5 hours, including analysis

Postgraduate (Master’s) project-based learning (instant results with time budget): this model emphasises group interactions, peer-accountability, and project planning. A

programme of system characterisation and optimisation, gas mixing regimes, and investigation of “unknown” gas mixtures would form the basis of a relatively long-term research-style project (several weeks) Students would be prevented from “acting before thinking” by the presence of a time budget – while results are instantaneous, the “real” generation time would be logged, and a time budget accordingly reduced. Students would meet weekly (or more frequently) to report progress and to suggest next steps.

In this use case, complexity can be increased over the undergraduate model to include the need to follow the actual system set-up and operational protocols. Behaviour that requires on- or off-line correction, such as spherical aberration in a transmission-based spectrometer would be included. Depending on the intended learning outcomes of the project, educational “booby traps” could be programmed into the simulation. For example, a failure of the Peltier cooling system for the Raman system’s detector could be coded to occur during a particular week. Students would not be warned in advance, and their response and reaction to this occurrence could be worked into an educationally highly valuable reflective discussion in a group meeting. Such occurrences should be kept to a minimum, however, and require sound pedagogical justification and careful input from the academic supervisor.

This use case is the envisaged target for our proposal in most circumstances, bridging the gap between small-scale simulations and research-grade implementations in an educationally valuable way. While this implementation of simulations can support learning of fundamental concepts, it is assumed that students operating at this level already possess a working facility with the concepts.

Research student system training (real-time): this model emphasizes realism in all aspects. Depending on the training regime, complexity could be gradually ramped up, or present in full force from the beginning of a student’s interaction. Interactions require the time they would have in a physical implementation of the set-up.

This use case is not likely to be used in an educational setting since the primary objective in training is for a student to have some simulation experience with running a particular experimental set-up. Set-ups that are inherently fast to operate, however, could directly implement a full simulation for the project-based learning case.

4.3 Extension of Proof-of-Concept to Remote Laboratories

In an era of swiftly evolving concerns around online security, adherence to standard security practices is of the utmost importance in developing distributed educational systems [63]. Unfortunately, full support in this area is lacking in the LabVIEW ecosystem [64], presumably as a result of limited demand from the traditional customer base. It is not a simple situation to rectify, because cryptography libraries are notoriously difficult to get right [65], while Autobahn-compliant WebSocket servers and clients are not trivial to build and maintain [66]. It is not practical or advisable for individual users to attempt to implement cryptographic solutions for TLS (to support https:// and wss:// connections) since they are vulnerable to attack via numerical implementation details.

Other web-native languages excel in this area, however, with open-source possibilities including Python, Golang and Rust. Python is the most widely used by scientists and

engineers, but if embarking on a major project there are significant benefits in the cleanliness with which Golang handles asynchronous programming. Golang applications can also be distributed as a single executable on all platforms, including Windows [67].

In order to hybridise with LabVIEW, the LabVIEW code needs to implement a simple unsecured WebSocket client, which is protected by the PC's firewall. That client logs into a custom-built WebSocket proxy running on the same PC (coded in Python, Golang, Rust or other web-native language), that takes care of securing and encrypting communications with the outside world and managing the communications logic of the remote lab system. A variation of this approach was used successfully in the OpenEngineering Lab at the Open University, which offered large scale remote access to ELVIS II+ boards via secure WebSockets [68].

5 DISCUSSION

Our proposal to re-purpose / adapt / re-implement research-grade simulations and control software for educational purposes would provide a much-needed bridge between the small-scale, limited-scope simulations of the typical undergraduate virtual laboratory and realistic experience of a research-grade operational environment.

We suggest that this addresses a number of criticisms levelled at virtual laboratories, particularly the “virtual laboratories are too safe” criticism of Potkonjak et al. [2], which we contend could be easily addressed in the project-based learning (instant results with time budget) use case by programmed system failures / anomalous behaviour along the lines of similar functionality already present in flight simulation software.

Additional advantages include: re-implementation of existing software bases, availability of local expertise, and the ability to relatively easily scale / adapt simulation software to a wide range of educational use cases. Although simulations are not without cost in terms of computing resources, expertise, security, and system maintenance, in the context of realistic research-grade simulations, they are much more robust, far cheaper to implement, easily modified for different use cases, and much easier to maintain.

Our proposal is not without potential disadvantages. The time and effort required to develop a realistic simulation might prove to be significant (although offset by existing software and local expertise), the necessary educational logistics (course resources, academic supervision, etc.) must be put in place, and careful thought must be put into the embedding course / module design. While the startup costs are significant, previous experience with large scale remote laboratories [68] suggests that the payback for implementing robust, autonomous systems is significant. It allows the academics to concentrate on the usual academic matters [69] rather than continually trying to coax along a less well-developed system. Having said that, it is not an easy task and so there is great need for the academic sector to work together to share a common infrastructure so as to allow academics to focus on content development, rather than continually resolving the same delivery mechanism issues [70].

In conclusion, we have identified a gap in the educational application of simulations which we propose to address by repurposing research-grade simulation and control software to provide robust, scalable, and realistic virtual laboratories.

REFERENCES

- [1] Heradio, R., de la Torre, L., Galan, D., Cabrerizo, F. J., Herrera-Viedma, E., and Dormido, S., (2016), "Virtual and remote labs in education: A bibliometric analysis," *Computers & Education*, vol. 98, pp. 14–38.
- [2] Potkonjak, V. *et al.*, (2016), "Virtual laboratories for education in science, technology, and engineering: A review," *Computers & Education*, vol. 95, pp. 309–327.
- [3] Nolen, S. B. and Koretsky, M. D., (2018), "Affordances of Virtual and Physical Laboratory Projects for Instructional Design: Impacts on Student Engagement," *IEEE Transactions on Education*, vol. 61, no. 3, pp. 226–233.
- [4] Chien, K.-P., Tsai, C.-Y., Chen, H.-L., Chang, W.-H., and Chen, S., (2015), "Learning differences and eye fixation patterns in virtual and physical science laboratories," *Computers & Education*, vol. 82, pp. 191–201.
- [5] Lewis, R. J., Telle, H. H., Bornschein, B., Kazachenko, O., Kernert, N., and Sturm, M., (2008), "Dynamic Raman spectroscopy of hydrogen isotopomer mixtures in-line at TILO," *Laser Phys. Lett.*, vol. 5, no. 7, p. 522.
- [6] Grindhammer, G., Rudowicz, M., and Peters, S., (1990), "The fast simulation of electromagnetic and hadronic showers," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 290, no. 2, pp. 469–488.
- [7] Tang, A., Horton-Smith, G., Kudryavtsev, V. A., and Tonazzo, A., (2006), "Muon simulations for Super-Kamiokande, KamLAND, and CHOOZ," *Phys. Rev. D*, vol. 74, no. 5, p. 053007.
- [8] Nagar, A. *et al.*, (2018), "Time-domain effective-one-body gravitational waveforms for coalescing compact binaries with nonprecessing spins, tides, and self-spin effects," *Phys. Rev. D*, vol. 98, no. 10, p. 104052.
- [9] "COMSOL Multiphysics® Software - Understand, Predict, and Optimize," *COMSOL Multiphysics*®. [Online]. Available: <https://uk.comsol.com/comsol-multiphysics>. [Accessed: 24-Apr-2019].
- [10] "Multiphysics Simulation Software | ANSYS." [Online]. Available: <https://www.ansys.com/en-gb/products/platform/multiphysics-simulation>. [Accessed: 24-Apr-2019].
- [11] Zeitnitz, C. and Gabriel, T. A., (1994), "The GEANT-CALOR interface and benchmark calculations of ZEUS test calorimeters," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 349, no. 1, pp. 106–111.
- [12] Altenmüller, K. *et al.*, (2019), "Muon-induced background in the KATRIN main spectrometer," *Astroparticle Physics*, vol. 108, pp. 40–49.
- [13] Han, M.-C., Jing, S.-W., Gao, Y.-D., and Guo, Y., (2019), "Experiment and MCNP simulation of a portable tagged neutron inspection system for detection of explosives in a concrete wall," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 929, pp. 156–161.
- [14] Villa, F. *et al.*, (2002), "The Planck Telescope," *AIP Conference Proceedings*, vol. 616, no. 1, pp. 224–228.
- [15] Alion, T. *et al.*, (2016), "Experiment Simulation Configurations Used in DUNE CDR," *arXiv:1606.09550 [hep-ex, physics:hep-ph, physics:physics]*.

- [16] Kleesiek, M. *et al.*, (2019), " β -Decay spectrum, response function and statistical model for neutrino mass measurements with the KATRIN experiment," *Eur. Phys. J. C*, vol. 79, no. 3, p. 204.
- [17] Fathy, H. K., Filipi, Z. S., Hagena, J., and Stein, J. L., "Review of hardware-in-the-loop simulation and its prospects in the automotive area," in *Modeling and Simulation for Military Applications*, 2006, vol. 6228, p. 62280E.
- [18] Maclay, D., (1997), "Simulation gets into the loop," *IEE Review*, vol. 43, no. 3, pp. 109–112.
- [19] Bacic, M., "On hardware-in-the-loop simulation," in *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005, pp. 3194–3198.
- [20] Brewe, E. and Sawtelle, V., (2018), "Modelling instruction for university physics: examining the theory in practice," *Eur. J. Phys.*, vol. 39, no. 5, p. 054001.
- [21] Dounas-Frazer, D. R. and Lewandowski, H. J., (2018), "The Modelling Framework for Experimental Physics: description, development, and applications," *Eur. J. Phys.*, vol. 39, no. 6, p. 064005.
- [22] Lindgren, R., Tscholl, M., Wang, S., and Johnson, E., (2016), "Enhancing learning and engagement through embodied interaction within a mixed reality simulation," *Computers & Education*, vol. 95, pp. 174–187.
- [23] Lima, N. *et al.*, "Do Students Really Understand the Difference Between Simulation and Remote Labs?," in *Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality*, New York, NY, USA, 2017, pp. 15:1–15:10.
- [24] Tong-on, A., Saphet, P., and Thepnurat, M., (2017), "Simple Harmonics Motion experiment based on LabVIEW interface for Arduino," *J. Phys.: Conf. Ser.*, vol. 901, no. 1, p. 012114.
- [25] Saphet, P., Tong-on, A., and Thepnurat, M., (2017), "One dimensional two-body collisions experiment based on LabVIEW interface with Arduino," *J. Phys.: Conf. Ser.*, vol. 901, p. 012115.
- [26] Beussman, D. J. and Walters, J. P., (2017), "Complete LabVIEW-Controlled HPLC Lab: An Advanced Undergraduate Experience," *J. Chem. Educ.*, vol. 94, no. 10, pp. 1527–1532.
- [27] Vesztergom, S., (2018), "A short introduction to digital simulations in electrochemistry: simulating the Cottrell experiment in NI LabVIEW," *Journal of Electrochemical Science and Engineering*, vol. 8, no. 2, pp. 171–181.
- [28] Nedeljkovic, M. S., Cantrak, D. S., Jankovic, N. Z., Ilic, D. B., and Matijevic, M. S., "Virtual instruments and experiments in engineering education lab setup with hydraulic pump," in *2018 IEEE Global Engineering Education Conference (EDUCON)*, 2018, pp. 1139–1146.
- [29] Chew, M. T., Demidenko, S., Ooi, M. P.-L., and Kuang, Y. C., "Family of low-cost NI ELVIS/LabVIEW-based semiconductor testers for engineering education," in *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, 2017, pp. 165–170.
- [30] Freeman, S. *et al.*, (2014), "Active learning increases student performance in science, engineering, and mathematics," *Proceedings of the National Academy of Sciences*, vol. 111, no. 23, pp. 8410–8415.
- [31] Long, D., "Rotational and Vibration–Rotation Raman Scattering," in *The Raman Effect*, John Wiley & Sons, Ltd, 2002, pp. 153–219.

- [32] Couture, M., (2004), "Realism in the design process and credibility of a simulation-based virtual laboratory," *Journal of Computer Assisted Learning*, vol. 20, no. 1, pp. 40–49.
- [33] Wrobel, P. M. *et al.*, (2016), "LabVIEW interface with Tango control system for a multi-technique X-ray spectrometry IAEA beamline end-station at Elettra Sincrotrone Trieste," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 833, pp. 105–109.
- [34] Ovejero, M. C. *et al.*, (2017), "LabVIEW-based control and acquisition system for the dosimetric characterization of a silicon strip detector," *Review of Scientific Instruments*, vol. 88, no. 2, p. 025104.
- [35] Pechousek, J., Prochazka, R., Jancik, D., Frydrych, J., and Mashlan, M., (2010), "Universal LabVIEW-powered Mössbauer spectrometer based on USB, PCI or PXI devices," *J. Phys.: Conf. Ser.*, vol. 217, p. 012006.
- [36] Schlattauer, L. *et al.*, (2017), "Calibration of gamma-ray detectors using Gaussian photopeak fitting in the multichannel spectra with a LabVIEW-based digital system," *Eur. J. Phys.*, vol. 38, no. 5, p. 055806.
- [37] Hendro, Jerfi, and Viridi, S., (2019), "Automation of data acquisition Surface Plasmon Resonance (SPR) based on LabVIEW," *J. Phys.: Conf. Ser.*, vol. 1127, p. 012003.
- [38] Tamborini, A. *et al.*, (2016), "Development and characterization of a 2D scintillation detector for quality assurance in scanned carbon ion beams," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 815, pp. 23–30.
- [39] Andreassen, O., Abalo Miron, D., Augrandjean, F., Blanco Vinuela, E., Gomez De La Cruz, M. D. F., and Rijllart, A., *LabVIEW as a New Supervision Solution for Industrial Control Systems*. 2015.
- [40] Barbesi, D., Vicente Vilas, V., Millet, S., Sandow, M., Colle, J.-Y., and Aldave de las Heras, L., (2017), "A LabVIEW®-based software for the control of the AUTORAD platform: a fully automated multisequential flow injection analysis Lab-on-Valve (MSFIA-LOV) system for radiochemical analysis," *J Radioanal Nucl Chem*, vol. 313, no. 1, pp. 217–227.
- [41] Cherkasov, N., Bai, Y., Expósito, A. J., and Rebrov, E. V., (2018), "OpenFlowChem – a platform for quick, robust and flexible automation and self-optimisation of flow chemistry," *React. Chem. Eng.*, vol. 3, no. 5, pp. 769–780.
- [42] Fitzpatrick, D. E. and Ley, S. V., (2018), "Engineering chemistry for the future of chemical synthesis," *Tetrahedron*, vol. 74, no. 25, pp. 3087–3100.
- [43] Zhen, J. *et al.*, (2015), "Laboratory Photo-Chemistry of PAHs: Ionization versus Fragmentation," *ApJL*, vol. 804, no. 1, p. L7.
- [44] Cui, Y. (崔扬) and Hanley, L., (2015), "ChiMS: Open-source instrument control software platform on LabVIEW for imaging/depth profiling mass spectrometers," *Review of Scientific Instruments*, vol. 86, no. 6, p. 065106.
- [45] Hryniewicz, P., Banaś, W., Sękala, A., Gwiazda, A., Foit, K., and Kost, G., (2015), "Object positioning in storages of robotized workcells using LabVIEW Vision," *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 95, p. 012098.
- [46] Tunströmer, A., "Saab Elevates Testing of the World's Most Cost-Effective Fighter Plane - Solutions - National Instruments." [Online]. Available: <http://sine.ni.com/cs/app/doc/p/id/cs-17491#>. [Accessed: 24-Apr-2019].

- [47] O’Dea, D., “Designing and Testing Automated Parking Systems While Maximizing Reusability to Address Changing Requirements and Aggressive Timelines - Solutions - National Instruments.” [Online]. Available: <http://sine.ni.com/cs/app/doc/p/id/cs-17511#>. [Accessed: 24-Apr-2019].
- [48] Lazarević, S., Čongradac, V., Anđelković, A. S., Čapko, D., and Kanović, Ž., (2019), “A novel approach to real-time modelling of the district heating substation system using LabVIEW,” *Journal of Cleaner Production*, vol. 217, pp. 360–370.
- [49] Shvetsov-Shilovskiy, I. I., Boruzdina, A. B., Ulanova, A. V., Orlov, A. A., Amburkin, K. M., and Nikiforov, A. Y., “Measurement system for test memory cells based on keysight B1500A semiconductor device analyzer running LabVIEW software,” in *2017 International Siberian Conference on Control and Communications (SIBCON)*, 2017, pp. 1–4.
- [50] Kuan, W.-H., Tseng, C.-H., Chen, S., and Wong, C.-C., (2016), “Development of a Computer-Assisted Instrumentation Curriculum for Physics Students: Using LabVIEW and Arduino Platform,” *J Sci Educ Technol*, vol. 25, no. 3, pp. 427–438.
- [51] Puantha, R., Khammarew, W., Tong-on, A., and Saphet, P., (2018), “The speed of sound in air of pipe acoustic resonance via the Arduino with LabVIEW interface,” *Phys. Educ.*, vol. 54, no. 1, p. 015009.
- [52] Lyu, B., “Simulation and Visualization of Fundamental Optics Phenomenon by LabVIEW,” in *ETOP 2017 Proceedings (2017), paper 104523R*, 2017, p. 104523R.
- [53] Muyskens, M. A., Glass, S. V., Wietsma, T. W., and Gray, T. M., (1996), “Data Acquisition in the Chemistry Laboratory Using LabVIEW Software,” *J. Chem. Educ.*, vol. 73, no. 12, p. 1112.
- [54] Drew, S. M., (1996), “Integration of National Instruments’ LabVIEW Software into the Chemistry Curriculum,” *J. Chem. Educ.*, vol. 73, no. 12, p. 1107.
- [55] Johari, J., Ahmad, A., Madzhi, N. K., Buniyamin, N., and Kassim, R. A., “LabVIEW as an effective tool for problem-based learning in undergraduate engineering education,” in *2017 IEEE 9th International Conference on Engineering Education (ICEED)*, 2017, pp. 169–173.
- [56] Chacón, J., Vargas, H., Farias, G., Sánchez, J., and Dormido, S., (2015), “EJS, JIL Server, and LabVIEW: An Architecture for Rapid Development of Remote Labs,” *IEEE Transactions on Learning Technologies*, vol. 8, no. 4, pp. 393–401.
- [57] Rodríguez, F., Guzmán, J. L., Castilla, M., Sánchez-Molina, J. A., and Berenguel, M., (2016), “A proposal for teaching SCADA systems using Virtual Industrial Plants in Engineering Education,” *IFAC-PapersOnLine*, vol. 49, no. 6, pp. 138–143.
- [58] Andrade, H. A. and Kovner, S., “Software synthesis from dataflow models for G and LabVIEW/sup TM/,” in *Conference Record of Thirty-Second Asilomar Conference on Signals, Systems and Computers (Cat. No.98CH36284)*, 1998, vol. 2, pp. 1705–1709 vol.2.
- [59] Kodosky, J., MacCrisken, J., and Rymar, G., “Visual programming using structured data flow,” in *Proceedings 1991 IEEE Workshop on Visual Languages*, 1991, pp. 34–39.
- [60] Valim, A., “The Evolution of LabVIEW: Decades of Development - National Instruments.” [Online]. Available: <http://www.ni.com/newsletter/50689/en/>. [Accessed: 24-Apr-2019].
- [61] Lewis, R., “LabVIEW NXG Accelerates Problem- and Project- Based Learning for MSc Students,” 05-Mar-2019. [Online]. Available: <https://forums.ni.com/t5/Real-World-Applications/LabVIEW-NXG-Accelerates-Problem-and-Project-Based-Learning-for/ta-p/3900063>. [Accessed: 24-Apr-2019].

- [62] "Compare LabVIEW NXG and LabVIEW - National Instruments." [Online]. Available: <http://www.ni.com/en-gb/shop/labview/compare-labview-nxg-and-labview.html>. [Accessed: 24-Apr-2019].
- [63] Voas, J., Kuhn, R., Laplante, P., and Applebaum, S., "Internet of Things (IoT) Trust Concerns (Draft)," National Institute of Standards and Technology, 2018.
- [64] "Does LabVIEW support TLS Protocol for its Web Server? - National Instruments." [Online]. Available: <https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z0000019RQNSA2&l=en-GB>. [Accessed: 28-Apr-2019].
- [65] Jakobsen, J. and Orlandi, C., "On the CCA (in)security of MTPProto," 1177, 2015.
- [66] *Autobahn WebSocket protocol testsuite. Contribute to crossbario/autobahn-testsuite development by creating an account on GitHub.* Crossbar.io, 2019.
- [67] Mudrinić, M., "How To Build Go Executables for Multiple Platforms on Ubuntu 16.04," *DigitalOcean*. [Online]. Available: <https://www.digitalocean.com/community/tutorials/how-to-build-go-executables-for-multiple-platforms-on-ubuntu-16-04>. [Accessed: 28-Apr-2019].
- [68] Drysdale, T. D. and Braithwaite, N. S. J., "An internet of laboratory things," in *2017 4th Experiment@International Conference (exp.at'17)*, 2017, pp. 236–240.
- [69] Drysdale, T., "Remote Laboratories: Pedagogy, Performance, and Pooling," presented at the National Instruments Academic Forum, University of Birmingham, UK, Jan-2019.
- [70] Drysdale, T., Weightman, A., Watts, S., Dishon, V., and Lewis, R., "Remote laboratories - Requirements capture for a national pooling infrastructure," presented at the AdvanceHE STEM Teaching and Learning Conference 2019, Birmingham, UK, 30-Jan-2019.