2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
Paper ID #50113

Hands-On Controls on a Budget: Teaching with OER, Mathematica, Python, and Arduino

Dr. Joshua Paul Steimel, Humboldt State University
JuEun Lee, Humboldt State University

Associate Professor, Mechanical Engineering, Cal Poly Humboldt Ph.D. Mechanical Engineering, Carnegie Mellon University

Hands-On Controls on a Budget: Teaching with OER, Mathematica, Python, and Arduino

Abstract

Mechanical engineering laboratory courses are often assumed to require substantial budgets and access to commercial software. This paper describes the redesign of an upper-level mechanical engineering laboratory sequence developed under resource limitations of less than $6,000 and without access to MATLAB or Simulink. The resulting 15-week sequence integrates free and low-cost tools including Python, Arduino microcontrollers, and Wolfram Mathematica, supported by educational data acquisition hardware and commercially available sensors. To extend accessibility, custom analysis functions, DAQ connectivity code, and interactive simulations were developed and shared through open-source repositories, enabling both students and the broader community to benefit from these instructional resources.

The laboratory sequence covers a diverse range of experiments, including load cell calibration and force measurement, fan RPM measurement with photogates, natural frequency determination of a beam using accelerometers and Fourier analysis, strain gauge testing with a signal amplifier, and feedback-controlled systems such as a beam-balance apparatus, fan RPM proportional control, linear actuator pendulum angle control using PD methods, and an inverted pendulum balancing experiment using Arduino-based PID control. Several of these laboratories required the design and fabrication of custom 3D-printed fixtures and circuit adaptations to integrate sensors with mechanical systems.

A significant instructional challenge was bridging gaps in student preparation, as many entered the course with limited experience in coding, electronics, or prerequisite topics such as dynamics, vibrations, or materials. These gaps were addressed through scaffolded instruction and the embedding of tool development within each lab. Assessment results indicate that students not only achieved the learning objectives but also reported increased confidence in experimental design, data acquisition, and analysis. This work demonstrates how resource-limited programs can deliver high-quality laboratory instruction while advancing equity and disseminating open educational resources, offering a replicable model for institutions facing similar constraints.

Introduction

Hands-on laboratory experiences are widely regarded as critical in undergraduate mechanical engineering education because they allow students to design experiments, collect data, validate models, and connect theoretical analysis with real-world behavior [1]. Laboratory courses provide a unique environment in which students exercise instrumentation, data acquisition, error and uncertainty analysis, model-to-experiment comparison, and engineering judgment beyond what typical lecture courses deliver [2]. Accreditation frameworks and consensus reports emphasize that well-structured laboratory experiences support ABET-style student outcomes such as experimental design, system analysis, and use of modern tools [3].

However, in many programs, laboratory instruction is constructed under strong assumptions: substantial institutional budgets, dedicated long-term capital equipment, proprietary software licenses (e.g., MATLAB/Simulink), and students with strong mathematical and instrumentation preparation [4]. These assumptions pose substantial access and equity challenges, especially at institutions with constrained budgets, high teaching loads, or geographically isolated campuses [5]. The cost of commercial closed-loop control rigs, dynamic testbeds, and related software often reaches tens of thousands of dollars— placing them out of reach for many engineering departments [6].

Instrumentation and data-acquisition technologies, however, are now more affordable and flexible. The widespread adoption of microcontrollers (e.g., Arduino) and open-source ecosystems, combined with commodity sensors and low-cost DAQ hardware, makes it possible to deliver rigorous, hands-on control and vibration experiments at a fraction of the traditional cost [7, 8]. Open software stacks including Python and Mathematica provide robust numerical computation, simulation, and visualization capabilities, enabling students to interface physical measurement with modeling and control design without needing expensive licenses [9]. Prior studies have shown that when laboratories provide modern DAQ, sensor integration, and open-software toolchains, students report higher engagement and improved confidence in experimental design, data-analysis, and troubleshooting [10].

Open Educational Resources (OER) provide a complementary access strategy: openly licensed textbooks, instructional videos, interactive simulations, and code repositories reduce financial and technical barriers while promoting reuse and adaptation [11]. Meta-analyses indicate that OER adoption can maintain or improve student outcomes in STEM contexts while lowering costs for students and institutions [12, 13]. OER also advances equity by enabling students at under-resourced institutions or remote campuses to access the same materials as their peers at resource-rich institutions [14].

In the domain of control systems and structural vibration laboratories, most published work focuses either on individual experiments, short modules, or embedded laboratory components within lecture courses. A growing number of projects leverage low-cost hardware, microcontrollers, and open-source software to enable closed-loop or vibration testbeds [15, 16]. However, fewer reported implementations describe a full-semester, upper-division mechanical engineering laboratory sequence that (a) operates on a budget significantly below the norm, (b) integrates open software and microcontroller hardware comprehensively, and (c) is explicitly scaffolded to accommodate students without advanced mathematical preparation (e.g., without prior coursework in differential or partial differential equations) [17].

This paper presents the redesign of an upper-level mechanical engineering laboratory sequence developed under resource limitations (budget under $6,000 and without access to MATLAB or Simulink). The redesigned 15-week sequence integrates free and low-cost tools including Python, Arduino microcontrollers, and Mathematica, supported by educational DAQ hardware and commercially available sensors. To extend accessibility, custom analysis functions, DAQ connectivity code, and interactive simulations were developed and published in open repositories, enabling both students and the broader community to benefit from these instructional resources.

The laboratory sequence covers a diverse range of experiments including load-cell calibration and force measurement, fan RPM measurement with photogates, natural-frequency determination of beams using accelerometers and Fourier analysis, strain-gauge testing with signal amplifiers, and feedback-controlled systems such as beam-balance apparatus, fan-RPM proportional control, linear-actuator pendulum angle control using PD methods, and an Arduino-based inverted-pendulum balancing experiment. Several of these labs required custom 3-D-printed fixtures and circuit adaptations to integrate sensors with mechanical systems.

A significant instructional challenge was bridging wide variation in student preparation: many students entered the course with limited experience in coding, electronics, or prerequisite topics such as dynamics, vibrations, or materials science. These gaps were addressed through scaffolded instruction and tool development embedded within each lab. The pedagogy progressed from physical experimentation, to simulation, to mathematical abstraction, allowing students to build intuition before modeling and control theory.

Results show students not only met course learning objectives but also reported increased confidence in experimental design, data acquisition, and analysis. This work demonstrates how resource-limited programs can deliver high-quality laboratory instruction while advancing equity, disseminating open educational resources, and offering a replicable model for other institutions facing similar constraints.

Implementation, Laboratory Sequence, and Pedagogical Structure

This upper-division laboratory course was developed to integrate open theoretical resources, modern instrumentation, and scaffolded experimentation within a unified 15-week sequence. The design philosophy centered on the principle that students learn most effectively when theory, modeling, and experimentation reinforce one another [1–3].

The creation of this Mechanical Engineering Measurement, Instrumentation, and Controls course presented several challenges. The program lacked dedicated experimental infrastructure, existing laboratory equipment, and commercial software resources. In particular, students and faculty did not have access to MATLAB licenses, even though it is commonly used in control-system instruction. To address this limitation, the course was designed entirely around open and low-cost computational platforms. Python, Mathematica, and Arduino served as the primary tools for computation, modeling, and control. These choices reduced financial barriers while supporting the broader goal of creating an accessible, reproducible, and sustainable model for advanced laboratory instruction in resource-limited engineering programs.

Course Framework and Theoretical Foundation

Each weekly module began with a theory session that introduced key control and instrumentation concepts using a freely available OER textbook authored by the instructor, along with a series of instructor-created YouTube videos. In most mechanical engineering control and instrumentation courses, MATLAB serves as the primary programming language for theoretical analysis and for communication with data acquisition (DAQ) systems. Because neither students nor faculty had access to MATLAB licenses, this course used Mathematica for theoretical computation and analysis. This framework can be seen schematically in Fig.1.

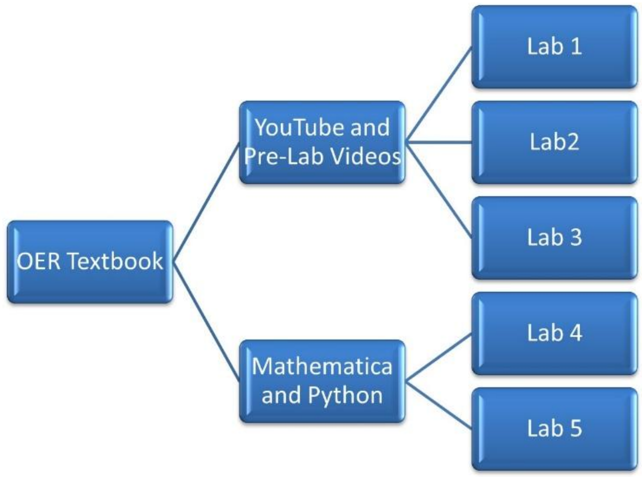

Figure 1: Integrated course framework. Students first engage with conceptual material through the OER textbook, YouTube lectures, and Mathematica notebooks. Pre-lab videos and simulation exercises reinforce these ideas and prepare students for the laboratory session, where they apply theory through hands-on data acquisition, control implementation, and analysis.

Custom Mathematica notebooks were developed that included symbolic derivations, simulation examples, and controller-tuning exercises [9, 11–14]. These digital resources lowered the programming barrier for students and helped them focus on the underlying theoretical concepts. They allowed students to engage conceptually with control-system modeling and response analysis before working with physical hardware. The use of OER materials aligns with current recommendations promoting equitable access to high-quality instructional resources without financial barriers [11, 12, 14].

The video series complemented the textbook by demonstrating theoretical calculations, wiring procedures, and short parameter-variation simulations. These videos allowed students to visualize how proportional, derivative, and integral terms influence system behavior and performance. The Mathematica notebooks supported symbolic manipulation and visualization tools such as Root Locus, Bode plots, and Routh–Hurwitz stability analysis, enabling students to explore system response characteristics interactively before attending the laboratory session. Students then applied these theoretical and simulation insights to physical experiments that reinforced the same control and instrumentation concepts.

Hardware Ecosystem and Instrumentation

The core of the experimental setup is the National Instruments myDAQ, chosen for its versatility, affordability, and compatibility with both Python and Vernier instrumentation [21, 22]. Students already had prior programming experience in Python, making it an ideal platform for signal processing, analysis, and control implementation. The myDAQ serves as a dual-channel analog input and output device with additional digital I/O capability, enabling measurement of sensor voltages and generation of control signals in a compact and portable form.

The Vernier myDAQ adapter provides plug-and-play connectivity with a wide range of inexpensive educational sensors, including accelerometers, force sensors, thermocouples, and photogates. This integration minimizes wiring complexity, calibration effort, and equipment cost while maintaining high signal fidelity [22]. The adapter's standard connectors and calibrated outputs allowed students to acquire data rapidly and focus on interpretation and analysis rather than instrumentation setup.

Because the department does not maintain MATLAB or Simulink licenses, all laboratory workflows were implemented entirely in Python and Mathematica. The NI-DAQmx Python Application Programming Interface (API) enabled real-time data acquisition and control signal generation. Custom Python functions were developed for signal filtering, Fourier analysis, data visualization, and closed-loop control [9, 19, 20]. Mathematica scripts complemented these activities by providing symbolic modeling, frequency-response analysis, and numerical comparison between measured and theoretical system responses [9, 27].

To extend system flexibility and achieve faster control-loop execution, Arduino microcontrollers were incorporated in experiments that required higher sampling rates or additional analog channels [8, 19]. The open-source Arduino Integrated Development Environment (IDE) and serial communication libraries allowed students to implement proportional–derivative (PD) and proportional–integral–derivative (PID) controllers for dynamic systems such as the pendulum-angle stabilization experiment. The progression and versatility of this system can be seen schematically in Fig. 2.

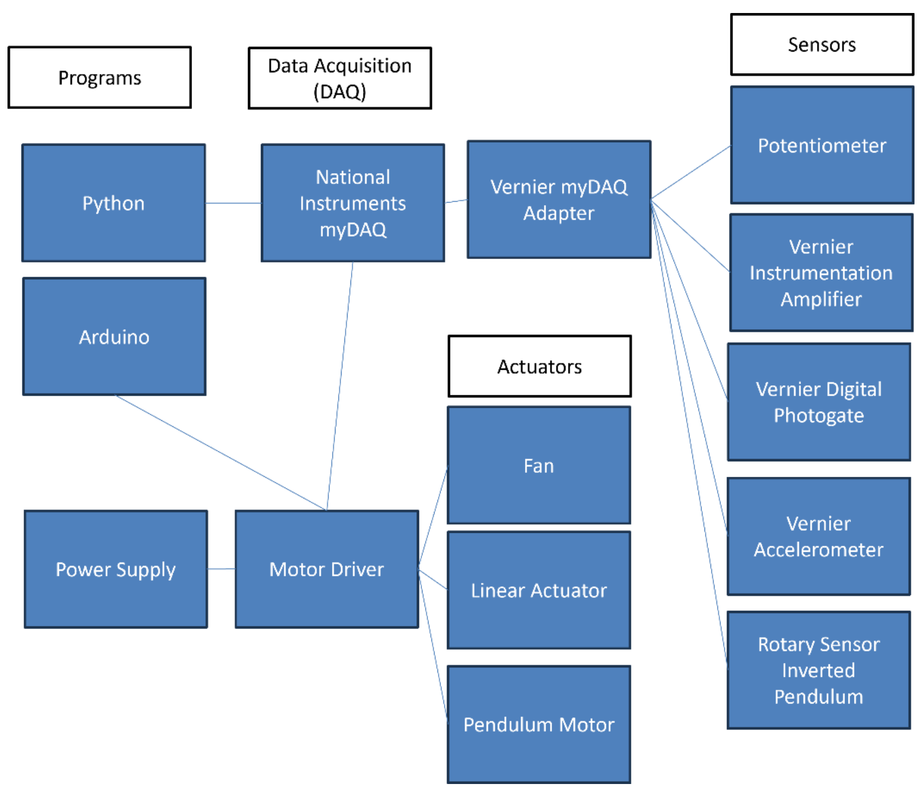

Figure 2: Experimental framework for the controls course. Python and Arduino are used for data acquisition, analysis, and control implementation. The National Instruments myDAQ interfaces

with Vernier and custom sensors, processes input signals, and sends output commands to motor drivers that control the actuators.

Custom mechanical components, including pendulum pivots, encoder brackets, and load-cell mounts, were fabricated using the department's 3D printing facilities. This followed best practices for low-cost laboratory prototyping and hardware customization [25, 26]. Additive manufacturing reduced the lead time for custom parts and allowed students to design and refine their own fixtures.

Through this process, students gained practical experience in manufacturability, design iteration, and integration of mechanical and electronic subsystems within a control environment. While several 3D-printed components required periodic replacement due to wear, this was treated as an instructional feature rather than a failure mode, reinforcing concepts of material selection, fatigue, and iterative design. Future work will quantify durability across materials and print parameters.

Laboratory Sequence and Weekly Progression

The semester consisted of eight major hands-on laboratory modules designed to progressively build students' analytical, programming, and experimental skills [10, 15, 18]. Each laboratory followed a consistent three-part structure:

1. **Pre-lab preparation** – Guided by readings from the OER textbook, tutorial videos that demonstrated the experimental systems, and Mathematica simulations that illustrated theoretical behavior.
2. **Hands-on experimentation** – Data acquisition, control implementation, and real-time parameter tuning using Python or Arduino hardware.
3. **Post-lab analysis** – Comparison of experimental data with theoretical predictions, model validation, and interpretation of system behavior.

The sequence of laboratory activities was designed to move from basic measurement and signal interpretation to advanced control implementation. The eight major laboratories included:

1. **Load-cell calibration and measurement** using a custom 3D-printed fixture and signal amplifier.
2. **Fan speed measurement** using a Vernier photogate to determine RPM and analyze time-domain and frequency-domain data.
3. **Beam vibration characterization** using accelerometer measurements and Fast Fourier Transform (FFT) analysis to extract natural frequencies.
4. **Strain-gauge instrumentation** using a Wheatstone bridge and amplifier to study stress and strain relationships.
5. **Ball-and-beam control** using a linear actuator and photogate for position feedback.

6. **Pendulum-angle control** using a linear actuator with potentiometer feedback to implement proportional–derivative (PD) control.
7. **Fan RPM control** using photogate feedback and proportional control to study steady-state error and system stability.
8. **Inverted-pendulum stabilization** using a rotary encoder for feedback and full proportional–integral–derivative (PID) control.

Each experiment required students to submit their Python scripts, acquired data files, and responses to reflection questions that linked observed physical phenomena to modeled dynamics. This consistent structure reinforced the relationship between theoretical modeling, simulation, and experimental validation throughout the course. The course is for 3rd or 4th year Mechanical Engineering students as a required course. This cohort had 20 students, a weekly 3-hour lab session, and students work in teams of 4 or more. All assessment procedures were reviewed and determined to be except educational research under institutional guidelines.

Pedagogical Structure and Scaffolding

The pedagogy followed a constructivist scaffold [16, 20] in which students progressed from guided observation and code modification to independent modeling and controller design. Early laboratory sessions emphasized sensor operation, voltage interpretation, and basic data visualization. Mid-semester activities introduced frequency-domain analysis and proportional feedback, and the final experiments required students to implement full PID control and perform stability assessments.

Pre-lab videos served as advance organizers by providing wiring walkthroughs, code demonstrations, and step-by-step parameter adjustments. These resources reduced differences in prior experience, especially for students with limited backgrounds in electronics or differential-equation modeling [4, 5, 17]. In-lab sessions encouraged collaboration, peer troubleshooting, and iterative testing to reinforce applied problem solving. Each lab concluded with a written reflection in which students analyzed discrepancies between theoretical predictions and experimental results. This process fostered critical thinking, encouraged synthesis of theoretical and experimental knowledge, and strengthened the students' ability to communicate engineering reasoning clearly.

Resource Leveraging and Sustainability

To remain within the budget target, all existing departmental hardware was cataloged and repurposed. Vernier sensors from introductory physics laboratories were reassigned for use in this course, and NI myDAQ units were acquired through educational discounts. All software tools used in the course (Python, Mathematica, Arduino IDE) are cross-platform, and future iterations will explicitly evaluate Linux-based workflows to further reduce system-level costs and licensing dependencies. Shared resources such as power supplies and soldering tools were borrowed from allied programs to minimize new expenditures [17, 18]. Faculty expertise in control systems, data acquisition, and additive manufacturing enabled nearly all hardware development to occur in-house, which reduced external costs and maintained flexibility for future modifications.

The resulting laboratory ecosystem is modular and adaptable. New sensors, actuators, or experimental control challenges can be incorporated with minimal additional cost or redesign. Because all instructional materials, including the textbook, instructional videos, source code, and 3D-printed components, are distributed under open licenses, other institutions can freely adopt or modify the framework to fit their own curricular needs. This open model directly supports international efforts promoting the dissemination of open educational resources and equitable access to engineering laboratory instruction [11, 14].

Assessment and Evaluation

Assessment of the redesigned laboratory course focused on two primary dimensions: technical proficiency and educational outcomes. Technical proficiency included measures of experimental accuracy, control-system performance, and modeling fidelity. Educational outcomes encompassed student learning, confidence, and engagement. Consistent with established practices in engineering education research [1, 2, 10, 15], a mixed-methods approach was adopted that combined quantitative performance benchmarks with qualitative analyses of student feedback. This dual strategy provided a comprehensive understanding of both the technical validity of the experimental systems and the pedagogical effectiveness of the instructional design.

Assessment Framework

Evaluation methods were organized around three primary domains.

1. **Technical performance:** Verification that each laboratory experiment produced credible engineering results using low-cost instrumentation.
2. **Learning outcomes:** Assessment of students' experimental design skills, data-analysis competence, and ability to connect theoretical models with experimental observations.
3. **Affective and equity measures:** Evaluation of students' confidence, engagement, and perceptions of accessibility, with special attention given to those entering the course without prior experience in ordinary differential equations, electronics, or programming.

While standardized instruments such as validated self-efficacy or motivation surveys were considered, they were not implemented during this initial course offering due to timing and institutional constraints. Instead, proxy indicators of confidence and autonomy were employed, including time-to-first-successful-measurement, reduction in instructor intervention, and convergence of performance outcomes between novice and experienced students. These measures provided convergent behavioral evidence of growing self-efficacy and independence, consistent with prior engineering education research. Future offerings of the course will incorporate validated affective instruments to enable stronger cross-study comparison and longitudinal analysis.

Technical Performance Assessment

Each laboratory experiment included measurable performance criteria derived from published standards in dynamics and control education [4, 6, 8, 19]. Representative metrics are summarized below in Table 1 and 2.

Table 1: Assessment of Technical Performance of Measurement Labs

Laboratory	Theoretical Value	Experimental Value and Total Uncertainty 95% Confidence Interval	P-value (<0.05)	Pass/Fail
Load-Cell Measurement	9.8N	9.7±0.2N	Yes	Pass
Fan RPM Measurement	1700RPM	1756±67RPM	Yes	Pass
Beam Frequency (FFT)	34.7Hz	31.3±3.7Hz	Yes	Pass
Strain-Gauge Bridge	0.0003	0.00033±0.00006Hz	Yes	Pass

Table 2: Assessment of Technical Performance of Controls Labs

Laboratory	Performance Metric	Pass/Fail
Ball-and-Beam	Ball balanced within the beam for 60s	Pass
Pendulum Angle (PD)	Steady State Error less than 0.5º	Pass
Fan RPM Control	Steady State Error less than 67RPM after 60s	Pass
Inverted Pendulum (PID)	Balance for 5s	Pass

Data were collected using Python-based DAQ scripts interfacing with NI myDAQ and Vernier sensors. Analysis was conducted with Mathematica.

Learning-Outcome Assessment

Learning outcomes were mapped to course objectives and ABET student outcomes [3]. Lab-report rubrics assessing design quality, data acquisition, and data analysis/interpretation (each on a 4-point scale). Rubric-based assessment indicated steady improvement: average report quality increased from 2.7 to 3.6 / 4 between Weeks 3 and 15, while code documentation scores rose from 2.3 to 3.5 for a cohort of 20 students. Prior to the redesign, the course relied primarily on scripted demonstrations, limited data acquisition activities, and minimal closed-loop experimentation due to lack of dedicated equipment and software licenses. Assessment focused largely on written reports rather than iterative experimentation or controller implementation. The redesign replaced this structure with a fully hands-on, experiment-driven sequence emphasizing modeling–measurement–control integration.

Table 3: Measurements and Controls Lab Rubrics

Category	4 – Exemplary	3 – Proficient	2 – Developing	1 – Beginning
Experimental Design and Preparation	Demonstrates clear understanding of objectives. Experimental procedure is complete, logical, and justified by theory. Identifies key variables, controls, and measurement uncertainties.	Procedure is mostly complete and logically structured. Minor omissions or unclear rationale for variable selection.	Procedure is partially complete or contains conceptual gaps. Limited discussion of variables or measurement controls.	Procedure is incomplete or unclear. Little evidence of planning or understanding of experimental objectives.
Data Acquisition and Accuracy	Data are complete, accurate, and collected with appropriate calibration and sampling rate. Units and precision are correct. Uncertainty and repeatability are quantified.	Data are mostly accurate with minor inconsistencies. Calibration and sampling decisions are reasonable but not justified. Uncertainty discussed briefly.	Data show noticeable errors or missing points. Calibration and sampling inadequately documented.	Data incomplete, inconsistent, or invalid. No calibration or uncertainty discussion.
Data Analysis and Modeling	Applies correct analytical methods using Python or Mathematica. Results compared to theory with statistical justification (e.g., regression, confidence intervals, p-values). Includes plots with correct	Correct analytical methods with minor computational or presentation errors. Comparison to theory is qualitative or lacks uncertainty discussion.	Incomplete or partially incorrect analysis. Minimal quantitative comparison to theory.	Analysis incorrect or missing. No connection between data and theory.

	formatting and labels.			

Affective and Equity Measures

Student growth in confidence and autonomy was evaluated using a simple behavioral indicator collected throughout the semester. To complement self-reports, a simple time-based metric was tracked for each team: time to first successful measurement. In the early measurement labs, students typically required the full three-hour period to wire sensors and acquire valid data. By the final control experiments, average setup and validation time had decreased to approximately one hour (mean reduction = 1.9 hours, $p < 0.01$). The reduction in setup time, combined with fewer instructor assistance requests, demonstrated growing procedural fluency and independence. These results provide convergent evidence that students became more confident, efficient, and self-directed as they progressed through the laboratory sequence.

Equity and Accessibility Findings

Integration of OER materials and freely available software eliminated textbook and license costs (previously >$250 per student) and ensured all students could replicate experiments on personal laptops. Open sharing of Mathematica demonstrations and 3-D-printable hardware files aligns with documented best practices for equity and global knowledge dissemination [11, 14, 27].

The course also closed preparation gaps: students who initially self-identified as "novices" in coding achieved final lab-report scores statistically indistinguishable from those with prior programming experience ($p = 0.34$). This multi-layered assessment confirms that a resource-limited yet OER-integrated controls laboratory can produce learning gains, strong technical results, and improved student confidence comparable to traditional high-cost environments.

Pedagogical Impact

The assessment results demonstrate that low-cost, open, and scaffolded approaches to laboratory education can produce learning outcomes comparable to traditional high-cost formats. Students showed significant improvement in conceptual understanding and coding proficiency, validating the effectiveness of coupling pre-lab multimedia instruction with open-source modeling and hands-on experimentation. This aligns with prior findings that scaffolded, multimodal learning environments enhance student retention and self-efficacy in engineering courses [5, 10, 12, 16].

The three-tiered structure, theory (OER textbook and Mathematica notebooks), guided simulation (YouTube videos and modeling exercises), hands-on lab (myDAQ and Vernier sensors), proved essential for bridging knowledge gaps among students lacking strong mathematical or programming backgrounds. By emphasizing conceptual visualization and interactive learning before physical implementation, the course mitigated cognitive overload and supported inclusive participation.

Technical and Logistical Effectiveness

From an instrumentation standpoint, the integration of NI myDAQ hardware, Vernier sensors, and Arduino microcontrollers provided sufficient accuracy and bandwidth for undergraduate controls applications. The plug-and-play Vernier interface minimized wiring complexity and enabled rapid experiment setup. Using Python and Mathematica allowed for full transparency of data analysis and modeling pipelines, eliminating dependency on proprietary licenses.

The cost reduction was significant, less than $6,000 for an entire 15-lab sequence, compared to $20,000–$40,000 for equivalent commercial systems [6]. Custom 3-D printed fixtures extended the hardware's versatility while reducing mechanical alignment issues noted in earlier lab implementations [25, 26].

Equity and Accessibility

The course achieved equity gains on two fronts: financial accessibility and pedagogical inclusivity. Removing software licensing costs and adopting OER materials reduced financial barriers for students while allowing them to access all learning materials on personal computers. The open publication of Mathematica demonstrations and Python scripts enables replication and modification at other institutions, extending the impact beyond a single classroom [11, 14, 27]. Pedagogically, the course successfully supported underprepared students through scaffolded design. The pre-lab videos and OER materials allowed them to review foundational concepts asynchronously, which aligns with recent evidence showing that flipped and scaffolded lab environments can improve persistence and confidence in engineering majors [10, 15, 16, 20].

Institutional and Practical Implications

The success of this implementation suggests a replicable model for programs seeking to modernize laboratories under resource constraints. The modular nature of the hardware ecosystem (myDAQ, Vernier, Arduino, 3-D printed fixtures) and the adaptability of the open software tools (Python, Mathematica) make the approach scalable for diverse engineering disciplines. Additionally, for instructors the preparation to create these lab activities will be minimal, less than 5 hours per lab, due to the open OER resources and pre-lab videos that are freely available via YouTube.

Institutions aiming to transition toward open, low-cost laboratory instruction may find that the greatest challenges are not technical but logistical: coordinating procurement, standardizing data acquisition interfaces, and training faculty in open-source workflows. The present work demonstrates that these barriers can be overcome through strategic reuse of departmental assets, faculty collaboration, and student co-development of open tools.

Limitations

Several limitations should be acknowledged. The assessment was based on a single course offering with a moderate cohort size, limiting generalizability. While technical performance metrics were robust, long-term retention and transfer of knowledge were not yet measured. Future studies should include multi-term tracking, cross-institutional comparisons, and longitudinal analysis of alumni performance in senior design or research settings.

Implications for Engineering Education

This study contributes to the broader conversation on open, equitable, and experiential learning in mechanical engineering. It supports three central propositions emerging in the literature:

1. Low-cost instrumentation can provide sufficient fidelity for high-level experimental learning outcomes [7, 8, 21, 22].
2. OER integration improves accessibility and democratizes advanced laboratory learning [11–14].
3. Scaffolded multimodal instruction (videos, simulations, experiments) supports heterogeneous cohorts and fosters inclusion [5, 10, 16, 20].

For educators, these findings emphasize that impactful laboratory reform does not necessarily require expensive hardware, but rather intentional design, integration of open tools, and clear pedagogical alignment. For institutions, they provide a roadmap for expanding laboratory capacity without prohibitive investment, while reinforcing commitments to equity and sustainability.

Conclusions

This paper presented the design, implementation, and evaluation of a resource-limited, OER-based control systems laboratory that integrates Python, Mathematica, Arduino, and NI myDAQ/Vernier hardware into a cohesive, scaffolded learning environment. The course demonstrates that:

1. High-quality control and measurement experiments can be conducted using low-cost, open-source tools.
2. Integration of OER materials—textbook, videos, and Mathematica demonstrations—enables equitable access and reproducibility.
3. Scaffolded, multimodal pedagogy supports diverse student preparation levels while improving confidence and competence in experimental design and analysis.
4. Technical results and learning outcomes met or exceeded traditional benchmarks.

The project contributes a replicable framework that other programs can adopt to modernize laboratory instruction affordably and equitably. Future work will include longitudinal tracking of student outcomes, expansion of remote-laboratory capabilities, and continued dissemination of OER materials through repositories and the Wolfram Demonstrations Project.

References

[1] Feisel, L. D., and Rosa, A. J., "The Role of the Laboratory in Undergraduate Engineering Education," *Journal of Engineering Education*, vol. 94, no. 1, pp. 121–130, 2005.

[2] Holmes, N. G., Olsen, J., Thomas, J. L., and Wieman, C. E., "Value Added or Misattributed? A Multi-Institution Study on the Educational Benefit of Labs for Reinforcing Physics Content," *Physical Review Physics Education Research*, vol. 13, no. 1, 2017.

[3] ABET, *Criteria for Accrediting Engineering Programs, 2025–2026*. Baltimore, MD: Accreditation Board for Engineering and Technology, 2024.

[4] Yadav, S. R. K., "Stability Aspects of Control Systems through Python Programming," *International Journal of Engineering and Science Invention*, vol. 13, no. 4, pp. 35–42, 2024.

[5] Wahyudi, M. N. A., Budiyanto, C. W., and Widiastuti, I., "Understanding Virtual Laboratories in Engineering Education: A Systematic Literature Review," *International Journal of Pedagogical Technology and Education*, vol. 7, no. 2, 2024.

[6] Acrome, "Experiment Systems for ABET Accreditation and Control Education," Technical Brief, 2020.

[7] Álvarez Ariza, J., and Báez, H., "Understanding the Role of Single-Board Computers in Engineering and Computer Science Education: A Systematic Literature Review," *arXiv preprint*, arXiv:2203.16604, 2022.

[8] Orozco, M. Á. L., et al., "Development of Low-Cost Arduino-Based Equipment for Analytical and Educational Applications," *Proceedings of the 2023 MDPI Applied Sciences Conference*, 2023.

[9] Russell, D. A., "Acoustics and Vibration Animations (Mathematica and Other Tools)," *Wolfram Demonstrations Project*, 2022.

[10] Li, J., and Liang, W., "Effectiveness of Virtual Laboratory in Engineering Education: A Meta-Analysis," *PLOS ONE*, vol. 19, no. 12, e0316269, 2024.

[11] UNESCO, *Recommendation on Open Educational Resources (OER)*, Paris, 2019; and *Dubai Declaration on OER Implementation*, 2024.

[12] Hilton, J., "Open Educational Resources, Student Efficacy, and User Perceptions: A Synthesis of Research, 2015–2018," *Educational Technology Research and Development*, vol. 68, pp. 853–876, 2020.

[13] Frady, K., "Use of Virtual Labs to Support Demand-Oriented Engineering Pedagogy in ET and VET Programmes," *European Journal of Engineering Education*, vol. 48, no. 5, pp. 822–841, 2023.

[14] Open Education Group, "The Review Project: OER Efficacy Syntheses," OpenEdGroup.org, 2022.

[15] Tomeo-Reyes, I., Musleh, A. S., Wijenayake, C., and Yeoh, J. Z., "Online Laboratories in Modern Engineering Education: A Systematic Literature Review," *Lecture Notes in Networks and Systems*, vol. 1135, pp. 150–165, 2025.

[16] Sellberg, C., Nazari, Z., and Solberg, M., "Virtual Laboratories in STEM Higher Education: A Scoping Review," *Nordic Journal of Systematic Reviews in Education*, vol. 2, pp. 58–75, 2024.

[17] May, D., Morkos, B., Jackson, A., Hunsu, N. J., Ingalls, A., and Beyette, F., "Rapid Transition of Traditionally Hands-On Labs to Online Instruction in Engineering Courses," *ASEE Annual Conference Proceedings*, 2023.

[18] Álvarez Ariza, J., and Báez, H., "Single-Board Computers in Engineering Education: Implementation Framework," *IEEE Global Engineering Education Conference (EDUCON)*, 2023.

[19] Panda, S. K., "Teaching Control Engineering by Using Python-Based PID," *International Advanced Research Journal in Science, Engineering and Technology*, vol. 7, no. 4, pp. 30–35, 2020.

[20] Yadav, S. R. K., "Teaching Control Engineering by Using Python-Based PID," *IARJSET*, vol. 7, no. 5, 2020.

[21] National Instruments, *NI myDAQ University Kit: Portable Instruments for Hands-On Experimentation*, Austin, TX, 2022.

[22] Vernier Software & Technology, *College Engineering Sensor Systems: Vernier myDAQ Adapter Overview*, Beaverton, OR, 2023.

[23] Python Software Foundation, *SciPy and Matplotlib Documentation*, Version 1.12, 2024.

[24] Orozco, M. Á. L., et al., "Development of Low-Cost Arduino-Based Instrumentation for Analytical Laboratories," *MDPI Sensors*, vol. 23, 2023.

[25] Callens, R., and Rinaldi, R. G., "Rapid Prototyping and 3D Printing for Low-Cost Engineering Laboratories," *Advances in Engineering Education*, vol. 9, no. 3, pp. 155–170, 2021.

[26] Gonzalez, J. M., et al., "3D Printing for STEM Laboratory Fixtures and Experiments," *Journal of STEM Education: Innovations and Research*, vol. 23, no. 4, pp. 45–56, 2022.

[27] Wolfram Research, *Mathematica Demonstrations Project: Interactive Models for Mechanical Systems*, Wolfram Media, 2024.