Practical Control Engineering Education

Requirements V2.0 (11/07/2025)

Glossary of Terms

Term	Definition
CLB	Control-Lab-in-a-Box (CLB). A modular, portable platform comprising hardware, designed to support hands-on learning and teaching of practical control engineering concepts.
Workflow	A structured engineering process that integrates theoretical instruction with practical tasks, potentially involving hardware and software exercises using the CLB platform, combined with opportunities for reflection and iterative learning.
Virtual Lab	A software-only implementation of the CLB platform, designed to deliver equivalent practical learning experiences without requiring access to physical hardware.

1. General System Requirements (GSR)

GSR-1.1 The teaching tool platform shall be named Control-Lab-in-a-Box (CLB), designed to deliver practical control engineering education across a broad learner spectrum, from secondary school students to undergraduate and postgraduate university learners.

GSR-1.2 CLB shall not be designed for a single specific application; however, broader potential application areas will be identified and highlighted as illustrative use cases.

GSR-1.3 The CLB platform shall include both physical hardware and a fully virtual alternative.

GSR-1.4 The CLB system shall be modular by design, allowing either partial or full integration into existing curricula.

GSR-1.5 The CLB platform and workflows shall be institution-agnostic, supporting adoption beyond Aston University.

GSR-1.6 The CLB platform shall be designed for accessibility and inclusivity, accommodating diverse learner backgrounds and need, e.g., low cost and for ease of duplication (set-up time should be low)

GSR-1.7 Teaching workflow materials shall be designed to support a diverse range of learners, including:

- High school students and members of the general public engaging in outreach or introductory content,
- Applied implementation-focused learners following an industry-oriented pathway, and
- Design and analysis-focused learners pursuing academic or technically advanced industrial roles.

GSR-1.8 The CLB system shall allow students to customise their learning paths based on their academic or career goals.

GSR-1.9 The CLB system shall include workflows that explicitly teach the software/system development lifecycle, such as the V-model.

GSR-1.10 Learning activities shall encourage failure and iteration as core elements of engineering.

GSR-1.11 All materials and tools shall be suitable for outreach use, including schools and professional bodies.

2. Hardware Requirements (HR)

HR-2.1 The CLB hardware shall be physically robust, suitable for repeated use in classroom and laboratory settings.

HR-2.2 The platform shall support multiple microcontrollers, including Arduino Uno and Mega.

HR-2.3 CLB shall support labs relevant to autonomous systems.

HR-2.4 No soldering or specialised tools shall be required; kits shall be plug-and-play.

HR-2.5 Each complete CLB shall use fewer than 45 components sourced from no more than 3 suppliers.

HR-2.6 No assembly time shall be required during classroom use.

HR-2.7 Pre-setup time per unit shall be less than 15 minutes.

HR-2.8 3D-printed components shall be used only for mechanical parts.

HR-2.9 Manuals and supply lists shall be provided to support self-build and online purchase options across countries.

HR-2.10 The platform shall be expandable to include other domains such as gears, hydraulics or pneumatics.

3. Software Requirements (SR)

SR-3.1 The primary development tools shall be MATLAB and Simulink, including the automatic code generation tools.

SR-3.2 The platform shall support alternative software's (although perhaps viewed as further work for now), including Python, LabVIEW, Scratch (for schools), and open-source software.

SR-3.3 An optional Control 101 App is to be developed.

SR-3.4 The virtual lab shall replicate the physical system in functionality and learning outcomes.

SR-3.5 Virtual tools shall support debugging tasks, including signal errors and model mismatch.

SR-3.6 All workflows shall be software-agnostic, capable of being delivered without dependence on a single tool.

4. Pedagogical and Learning Design Requirements (PLDR)

PLDR-4.1 Learning experiences shall focus on engineering thinking and process, not rote learning.

PLDR-4.2 Each activity shall explicitly state:

- Why it is being done.
- What engineering goal it addresses.
- How the engineering control is 'done' using the engineering thinking and process

PLDR-4.3 Learning materials shall support structured, step-by-step progression, prioritise hands-on engagement, and integrate real-world applications from the outset to inspire and motivate learners.

PLDR-4.4 The platform shall introduce control engineering concepts without requiring prior theory-heavy knowledge.

PLDR-4.5 The curriculum shall include scenario-based and challenge-based learning activities, such as 1-2 week grand challenges.

PLDR-4.6 The CLB shall support both individual and group learning.

PLDR-4.7 Learning materials shall promote iterative learning and deliberate practice.

PLDR-4.8 Skills developed from CLB shall be transferable across engineering disciplines, e.g., including vehicle electrification and autonomous systems.

PLDR-4.9 Labs shall include system identification techniques, including black-box modelling and model mismatch analysis.

PLDR-4.10 Materials shall emphasise control-in-practice, not just mathematical formulation.

PLDR-4.11 ChatGPT or similar tools shall be integrated into the workflows, e.g., for debugging code and explaining steps

5. Control Engineering Content Requirements (CECR)

CECR-5.1 Core control content shall include PID algorithms, covering tuning, and practical implementation.

CECR-5.2 Advanced topics shall be supported, including:

- Constraint controls (e.g., input, state and output)
- Multivariable control (MIMO - Multiple Input, Multiple Output and Single Input, Multiple Output)
- Optimal control
- Model Predictive Control (MPC)
- Reinforcement learning

CECR-5.3 CLB shall simulate real-world conditions (e.g., instrumentation, diagnostics, and fault detection tasks) with real engineering problems for students to solve, including:

- Noisy sensors
- Incorrect sampling configurations
- Safety-critical systems, e.g., overflow protection

CECR-5.4 Control scenarios shall include HMI design, operability, and abnormal situation management.

CECR-5.5 Labs shall support block-based control development

6. Workflow and Curriculum Requirements (WCR)

WCR-6.1 Approximately 20 workflows shall be developed, with various pathways developed for students depending on the option selected in GSR-1.6.

WCR-6.2 Each workflow shall require approximately 6 hours of effort (blended in-class and self-study).

WCR-6.3 Each workflow shall include:

- Baseline and stretch assessment questions for high-performance students
- A question bank to check students meet the learning outcomes
- Industry-focused report writing skills and tasks

WCR-6.4 All workflows shall include mapped learning outcomes, linked to measurable competencies.

WCR-6.5 The workflows shall highlight challenges in transitioning from university to industry.

WCR-6.6 Content shall be delivered in multiple formats: PDFs, slides, videos, and web-based content.

7. Virtual Lab Requirements (VLR)

VLR-7.1 The virtual lab shall serve as a complete replacement for physical hardware when needed.

VLR-7.2 The virtual lab shall support the teaching workflows through demonstrating:

- Faster development time
- Easier debugging
- Scalable teaching delivery

VLR-7.3 The virtual lab shall include interactive simulations and graphical visualisations to illustrate system dynamics.

8. Open-Source and Open-Access Requirements (OSOAR)

OSOAR-8.1
The core documentation for the CLB platform (e.g., build guides, lab manuals, component lists) shall be made available open-access.

OSOAR-8.2
The hardware designs (e.g., 3D print files, schematics) shall be published as open-access.

OSOAR-8.3
Software components (e.g., simulation scripts, virtual lab tools) developed for the CLB shall be released as open-access.

OSOAR-8.4

The CLB workflows and teaching content shall be published in an open repository (e.g., GitHub, MathWorks File Exchange or project website), allowing educators and students to freely modify, and reuse the material.

OSOAR-8.5

An open community forum or platform shall be established to share enhancements, labs, and user-generated content, supporting continuous development and peer exchange.