

IMPERIAL

Advancing Lab-Based Guidance Using an AI-Driven Virtual Assistant

Staff Partner: Dr Mohamed Maher (Department of Chemical Engineering)

Student Partners: Gaspard Bouchard (Department of Chemical Engineering)
Yihan Shen (Department of Mathematics)

23/04/2026

The Problem

Why Generic AI Tools Do Not Work in Practical Lab-Based Teaching?

01

Lab sessions are unique.

Each experiment has its own data, plot types, and objectives. A generic chatbot has no way to see any of this, so its answers aren't actually about what the student is doing.

02

Instructors can't scale.

One instructor supporting 80+ students running different experiments creates unavoidable wait times, broken concentration, and lost learning moments.

03

Off-the-shelf AI misleads.

Without knowing the student's current data or mastery level, a general chatbot gives confident answers that are often wrong for this student, at this moment, on this experiment.

Key Takeaway: Generic AI tools do not help and might actively mislead students during critical practical learning.

What We Built

Our Proposed Solution

SLM

Domain-specific Small Language Model

A compact model (smaller and cheaper to run than a general chatbot) fine-tuned on lab pedagogy.

ORC

Orchestrator layer

It rephrases student questions, chooses the right explanation style, and determines the context the model sees; students never write a prompt.

CTX

Live lab context

Reads the student's LabVIEW data, plots, and current task, so answers are grounded in what the student is actually doing.

A-RAG

Module-lead-curated content

Every answer is anchored to material that the module lead has approved and not generic knowledge.

ADP

Adaptive scaffolding

Layer to adjust tone, depth, and worked-example density automatically from the student's measured mastery.

What Makes it Different?

Three Things No Generic AI Chatbot Does

01

Zero Prompt Engineering

Students ask in plain and natural language. Orchestrator translates the question into whatever structured prompt the model needs; students never learn prompt tricks.

02

Continuous Knowledge Tracing

Per-topic mastery scores (0–1) that update every turn and decay per day of inactivity during project time. The system remembers what each student actually knows, not just what they last asked.

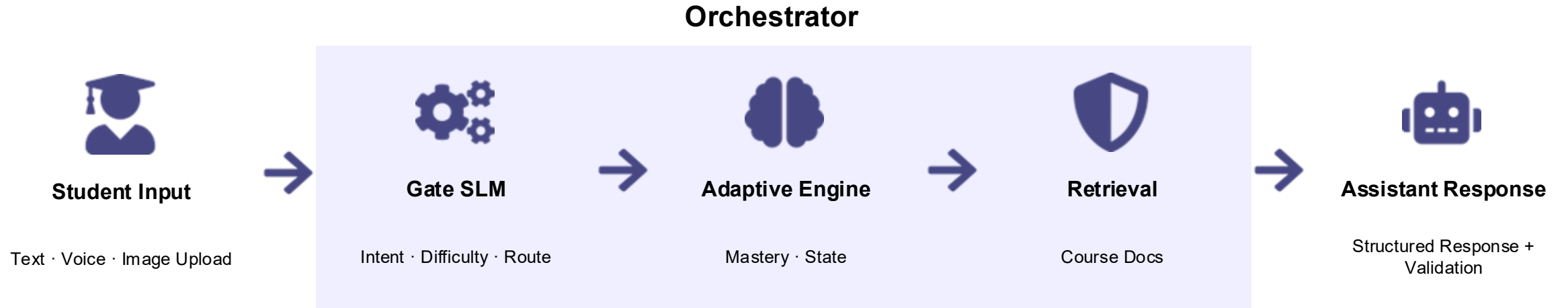
03

Pedagogically Grounded

Answers are checked against module-lead content, aligned to each student's needs, and validated for correctness.

System Architecture

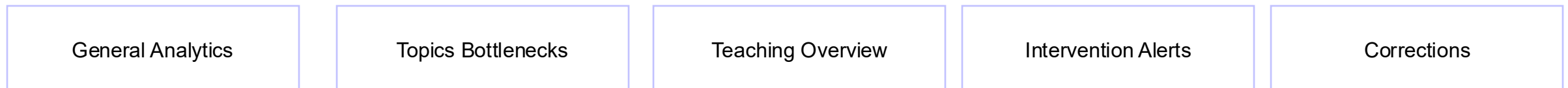
How it Works



Post Processing



Teacher Dashboard



Same Question, Different Answer

Current Student Experience

A student asks: “My PID controller is oscillating — what do I do?”

Generic AI Chatbot

Answers from training data. No idea which controller, which lab, which student.

Same explanation for a first-year and a final-year student.

Correct-sounding but ungrounded, and can hallucinate lab procedures.

Student has to prompt-engineer to get anything useful.

Our Proposed Tool

Sees the student's actual PID gains, the plot of the oscillation, and the module's approved tuning approach.

Chooses hints (high mastery) or a worked example (low mastery) based on this student's tracked knowledge state.

Answers validated against module-lead content; units and equations auto-checked.

Student asks in plain, natural language.

Student Experience

What Does A Student Actually Experience?

1

Ask Anything

Type, speak, or upload an image. No prompt crafting, just ask the question you have.

2

Auto-Classified

Orchestrator identifies and tags intent, difficulty, and whether vision/documents are needed.

3

Mastery Checked

Per-topic knowledge score is consulted to pick the explanation depth.

4

Response Tuned

Tuning parameter computed, response depth validated, scaffolding level, and tone set automatically.

5

Answer Validated

Response evaluator and validator ensure quality before you see it.

6

Learn & Progress

Mastery scores update, related questions suggested to progress understanding and next question starts better-calibrated

Student Workspace

General Lab Workspace View

Lab workspace

Centre panel: structured responses and equations. Use the left rail for session controls and the right for history.

Current objective: Explain the control error signal Review due: 1

REVIEW DUE

TRY IT NOW

Explain the control error signal

Which quantity does the control error compare?

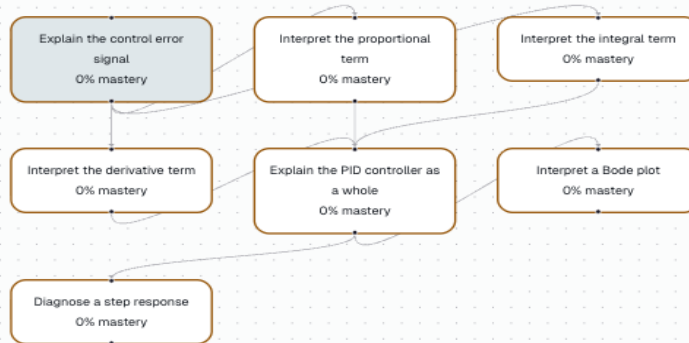
Setpoint and measurement

Control input and disturbance

Derivative and integral gains

Learning path

Current objective highlighted. Prerequisites flow from left to right.



React Flow

1

Topic mastery tiles

Per-topic knowledge scores. Update after every interaction.

2

Chat pane

Conversation with the tool, grounded in the student's current experiment.

3

Review pane

Reminds student by asking relevant questions based on their previous per-topic knowledge

Example Conversation (With Reasoning)

Process Control Question

YOU
What is PID controller?

ASSISTANT

🔊 Read aloud

Here is the rewritten response in accordance with the Reasoning-First Tutor protocol:

Internal Turn Assessment

The learner's question, "What is a PID controller?", indicates a fundamental conceptual understanding is required. The internal state S_{val} is high (0.972), indicating the learner is ready for a direct, comprehensive explanation rather than hints.

Scaffold Response

Definition

A **PID controller** (Proportional-Integral-Derivative) is a control loop feedback mechanism widely used in industrial control systems. It continuously calculates an error value as the difference between a measured process variable (the actual output) and a setpoint (the desired output). The controller attempts to minimize this error by adjusting the control output.

Explanation

The controller's output is a weighted sum of three distinct terms:

1. **Proportional (P):** Reacts to the *current* error. It provides an immediate correction but may leave a steady-state error.
2. **Integral (I):** Reacts to the *accumulated* past errors. It eliminates steady-state error by pushing the output until the error is zero.
3. **Derivative (D):** Reacts to the *rate of change* of the error. It predicts future error based on current trends, adding damping to prevent overshoot.

Equation(s)

The control signal $u(t)$ is calculated as:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

Symbols and Units

Symbol	Description	SI Unit
$u(t)$	Control output signal	Dimensionless (or Volts, Amps, etc.)
K_p	Proportional gain	Dimensionless
K_i	Integral gain	s^{-1} (or 1/s)
K_d	Derivative gain	s
$e(t)$	Error signal (<i>Setpoint - Measurement</i>)	Same as process variable (e.g., m, K)
τ	Dummy variable for integration	s

1

Student asked a question

Question is in plain and natural language (i.e., as expected in a practical lab session)

2

Assistant Response

Worked with units, the correct equation, and a depth matched to their current mastery. The student never had to ask for the 'right' level — it was picked automatically.

3

Overall Output

Equation and symbols come from module-lead-approved content, not generic training data. Every fact can be traced back to a curated source.

Note: Internal routing signals (mastery values, confidence scores) are visible in this debug view but hidden from students in production.

Outcomes and Expected Impact

What the Tool Is Designed to Do?

0

Prompt Engineering Needed

By design. Orchestrator handles all routing;
students just ask in plain language.

4×

Explanation Levels

Intuitive → Structured → Formal → Advanced
Automatically selected per turn.

24/7

Lab Support

Available during and after lab sessions

100%

Grounded Answers

Every response is validated against curated
module materials

Planned Evaluation

How We will Measure Whether It Works?

Grounding rate

% of answers traceable to module-lead-approved content.

Target $\geq 85\%$ in pilot.

Mastery gain

Per-topic score movement over a lab module.

Compared to the previous year's stats.

Instructor-wait-time

Average wait time during lab sessions.

Measured via observation and post-session survey.

Accuracy audit

Instructor-reviewed sample of 100+ conversations per module.

Scoring correctness, pedagogical fit, and unit/equation validity.

Student trust & usability

Post-lab survey.

Perceived helpfulness, willingness to re-use, specific failure modes.

Instructor workload

Time spent answering repetitive lab questions.

Measured before and after the pilot.

Challenges and Future Work

What We are Still Working On

Current Challenges and Limitations

- **Infrastructure overhead:** Running a local SLM plus orchestrator is heavier than calling a public API.
- **Content curation cost:** Module leads need to prepare and approve material before deployment.
- **Residual hallucination risk:** Despite grounding mechanisms, responses may still deviate in edge cases. Validation layer mitigates this risk but does not fully eliminate it.
- **Privacy & data governance:** Handling of student lab data (e.g., lab inputs and topic mastery scores) requires careful management, including anonymisation and obtaining ethical approval prior to the pilot.

Future Directions

- **Staff & Student Evaluation:** Evaluation of the tool across two lab-based modules
- **Progress dashboard:** Development of a student-facing interface that visualises mastery across the different modules, enabling ongoing self-monitoring and targeted revision
- **Peer-learning environments:** Introduction of a collaborative layer that supports interaction among students working on the same project within defined groups.
- **Integration Within LabVIEW:** Interface with LabVIEW via API to embed the tool within existing lab-based workflows.

IMPERIAL

Thank you

IMPERIAL

Questions?