

SHARE: An Experiment in Helping Students to Collaboratively Engineer with AI

Mike Wilkins, North Carolina State University at Raleigh

Mike Wilkins is a Teaching Assistant Professor in the Lampe Joint Department of Biomedical Engineering at University of North Carolina in Chapel Hill and North Carolina State University. His research interests include bioelectronic medicine and electronics engineering education.

Cameron Roy Taylor, The University of North Carolina at Chapel Hill

Cameron R. Taylor is an Assistant Professor in the Lampe Joint Department of Biomedical Engineering at the University of North Carolina at Chapel Hill and NC State University, where he directs the High-Performance Human Interfacing Laboratory (Hi-PHI Lab). His research interests include electromagnetics, algorithm development, and medical device design.

SHARE: An Experiment in Helping Students to Collaboratively Engineer with Generative AI

Just as previous generations of engineering educators were responsible for teaching students to adapt to charts[1], calculators[2], and computers, we now face the professional responsibility of teaching students to leverage generative AI (genAI) in their respective fields[3]. The current landscape provides students with unprecedented access to powerful tools that expand the breadth of their engineering ability; however, a significant epistemological gap persists because students often lack the critical framework necessary to evaluate accuracy in AI-generated outputs[4-7]. Consistent with evidence-based active learning methodologies, this laboratory experience, as part of a 300-level bioinstrumentation electronics course, builds on STEM education research that demonstrates that cognitive engagement is a primary driver of success[8, 9]. The objective of this design is to shift students from passive consumers to erudite engineers who treat genAI as a tool rather than an oracle[10, 11]. By structuring a collaborative lab focused on active filter design for biosignals, we explore strategies to ensure students use genAI to augment rather than replace their fundamental engineering knowledge. We hope that by providing this example lab along with a more generic template, engineering educators may find this approach applicable to their own courses. This approach addresses the responsibility of the modern educator to equip future engineers with the skills needed to critically navigate genAI performance concerns.

The specific *learning objectives* center on active knowledge construction. Students are required to **define** diagnostic ECG specifications and **collaborate** with genAI tools for initial design concepts. Crucially, learners must **verify** these AI-generated topologies using theoretical transfer functions and simulations to mitigate potential hallucinations. The activity culminates in students **validating** their physical hardware against engineering constraints and **analyzing** the ethical considerations of using genAI as a design partner in biomedical workflows.

Laboratory Design: The CollABorative Workflow

(A combination of 2 x 75 minute lab sessions and homework, targeting 5-10 hours total)

- **Step 1: ASK (Defining the Overall Goal)**
 - **Action:** Students are asked to identify physiologically-based sensing requirements for biosignals (e.g., ECG, EMG) to determine passband and gain specs. We begin by giving students an example of a well-designed genAI prompt, which shrinks to no prompt support by step 3.
 - **Validation Method: Existing Discipline Knowledge.** Cross-reference frequencies against physiological standards learned in prerequisite courses.
- **Step 2: IMAGINE (Refining the Problem)**
 - **Action:** Students brainstorm solutions for the explicit problem with genAI, focusing on course content (e.g., a linear phase response filter is preferred for ECG). Students use reference tables and genAI for target filter Q and Frequency Scaling Factor values.

- **Validation Method: Literature Benchmark.** Students produce a brief table listing potential solutions and compare with lecture notes and the textbook.
- **Step 3: MODEL (Theoretical Formulation from First Principles)**
 - **Action:** Students derive the *theoretical* transfer function manually from desired corner frequencies and Q-point without genAI support before asking the genAI for the code to derive and model the desired transfer function for the specific filter in MATLAB or Python.
 - **Validation Method: Mathematical Proof.** Verification that the derived transfer functions match and mathematically yield the required frequency response.
- **Step 4: PLAN (Simulation & Idealization)**
 - **Action:** Students calculate component values to match the target transfer function by hand and using AI, then model both in circuit simulation software (e.g., LTSpice).
 - **Validation Method: Ideal Simulation vs. Theory.** Students compare the simulator's Bode plots against their Step 3 derivation. If the genAI's "ideal" circuit doesn't match the transfer function or Bode plot, the AI hallucinated the topology.
- **Step 5: CREATE (Physical Realization)**
 - **Action:** Students physically build the circuit on a breadboard.
 - **Validation Method: Benchtop Measurement.** Using function generators and oscilloscopes, students measure the *actual* cutoff frequency and gain.
- **Step 6: IMPROVE (Critical Evaluation)**
 - **Action:** Students analyze the deviation between the genAI's plan and the actual circuit. Students determine if component values chosen by the AI were reasonable.
 - **Validation Method: The Report.** Students document two specific gaps while also delivering their code, written work, and the full transcript of the conversation with the genAI:
 1. **GenAI Gap:** Errors in the genAI's initial schematic (e.g. incorrect topology, unreasonable component values, hallucinations).
 2. **Performance Gap:** Performance loss due to component standardization.

Students have responded positively to this lab, and frequently commented how it changed the way they use genAI as a study aid in other courses. This is somewhat supported by fewer occurrences of genAI hallucinations in homework responses. The successful adaptation of this collaborative filter design framework by other engineering faculty is contingent upon several factors that require proactive instructional planning. Foremost, the sanctioned use of and access to genAI tools (e.g., GPT, Gemini, Claude) varies across institutions, necessitating that faculty potentially leverage university-approved licenses or locally hosted models to ensure equitable participation. Furthermore, the efficacy of the workflow depends heavily on students' technical prowess with simulation tools, and their proficiency in precise engineering prompts. To address academic integrity concerns, **educators should shift assessment weighting away from answers and toward the student's report**, which documents gaps between machine and collaborative designs, while reinforcing critical thinking skills. Ultimately, this design ensures that generative AI functions as a tool that enhances technical ability without atrophying fundamental reasoning.

- [1] G. Mottola and M. Cocconcelli, "Nomograms in the History and Education of Machine Mechanics," *Foundations of Science*, vol. 29, no. 1, pp. 125–155, 2024/03/01 2024, doi: 10.1007/s10699-022-09890-w.
- [2] D. B. McLeod, R. E. Stake, B. P. Schappelle, M. Mellissinos, and M. J. Gierl, "Setting the Standards," in *Bold Ventures: Case Studies of U.S. Innovations in Mathematics Education*, S. A. Raizen and E. D. Britton Eds. Dordrecht: Springer Netherlands, 1996, pp. 13–132.
- [3] P. M. Simelane and J. Kittur, "Use of Generative Artificial Intelligence in Teaching and Learning: Engineering Instructors' Perspectives," *Computer Applications in Engineering Education*, vol. 33, no. 1, p. e22813, 2025, doi: <https://doi.org/10.1002/cae.22813>.
- [4] G. Chaudhry, E. Gide, E. Yadegaridehkordi, and R. Tumpa, "Generative AI-Powered Teaching and Learning in Engineering and Project Management Higher Education: A Systematic Review," Cham, 2025: Springer Nature Switzerland, in The 6th Joint International Conference on AI, Big Data and Blockchain (AIBB 2025), pp. 99–113.
- [5] C. Liu, "A comprehensive review of applications of AI technologies in higher engineering education," *Discover Education*, vol. 4, no. 1, p. 528, 2025/11/28 2025, doi: 10.1007/s44217-025-00954-0.
- [6] C. Liu, G. C. Wang, and H. F. Wang, "The Application of Artificial Intelligence in Engineering Education: A Systematic Review," *IEEE Access*, vol. 13, pp. 17895–17910, 2025, doi: 10.1109/ACCESS.2025.3532595.
- [7] Y. Wan, R. Li, W. Li, and H. Du, "Impact Pathways of AI-Supported Instruction on Learning Behaviors, Competence Development, and Academic Achievement in Engineering Education," *Sustainability*, vol. 17, no. 17, p. 8059, 2025. [Online]. Available: <https://www.mdpi.com/2071-1050/17/17/8059>.
- [8] L. M. G. Costa and M. J. C. S. Reis, "Motivational Teaching Techniques in Secondary and Higher Education: A Systematic Review of Active Learning Methodologies," *Digital*, vol. 5, no. 3, p. 40, 2025. [Online]. Available: <https://www.mdpi.com/2673-6470/5/3/40>.
- [9] S. Tharayil et al., "Strategies to mitigate student resistance to active learning," (in eng), *Int J STEM Educ*, vol. 5, no. 1, p. 7, 2018, doi: 10.1186/s40594-018-0102-y.
- [10] D. Marimekala, "Impact of GenAI using Auto Content Generation Tools on Cognitive Thinking Process in Academia," in *2025 IEEE Integrated STEM Education Conference (ISEC)*, 15–15 March 2025 2025, pp. 1–8, doi: 10.1109/ISEC64801.2025.11147291.
- [11] Q. Zhu and H. Wang, "Primary Thought on Artificial Intelligence (AI) Enhanced Control Engineering Education," (in English), *ICCK Transactions on Sensing, Communication, and Control*, vol. 2, no. 3, pp. 215–225, 2025/09/23 2025, doi: 10.62762/TSCC.2025.254228.

Supplemental Lab 4: Collaborative AI Design of an Active Filter for ECG

1. Lab Objectives

Upon successful completion of this lab, you will be able to:

- Define the frequency specifications for a diagnostic ECG filter.
- Collaborate with an AI partner to research design topologies and approximations.
- Design and calculate component values for an 8th-order active bandpass filter.
- **Verify** the design using theoretical transfer functions and circuit simulation (Bode plots).
- **Validate** the design by building a physical prototype and testing its time-domain and frequency-domain performance.

2. Background

An electrocardiogram (ECG) signal is a small bioelectrical signal ($\sim$ low mV) that measures heart activity. The raw signal is almost always contaminated by noise:

- **DC Wander / Baseline Drift:** Low-frequency (_____ Hz) noise from patient breathing or electrode movement.
- **Powerline Interference:** (_____ Hz) noise from nearby electrical equipment.
- **Electromyographic (EMG) Noise:** High-frequency (_____ Hz) noise from muscle twitches.

Our goal is to design a filter that removes this noise, isolating the diagnostic ECG signal. A simple passive filter may not work; we need an **active filter** to provide gain (if needed), buffer stages, and avoid bulky inductors.

We will design this filter as two cascaded 4th-order filters:

1. A 4th-order **High-Pass Filter (HPF)** with a cutoff frequency of _____ Hz.
2. A 4th-order **Low-Pass Filter (LPF)** with a cutoff frequency of _____ Hz.

A 4th-order filter is created by cascading two 2nd-order stages.

3. Pre-Lab: AI-Assisted Design

In this section, you will use a large language model (LLM) or other generative AI tool as your design partner. **Your team's task is to "interview" the AI** to make critical design decisions. For that, you need to construct meaningful prompts. A good prompt includes the following categories:

- **Persona/Role:** Define “who” the AI should mimic. Can also provide a target audience.
- **Context:** Provide background information to set the stage
- **Task:** Use action verbs to state clearly what the AI should “do”
- **Constraints:** Specify tone, length, format, and things to avoid
- **Rationale:** Ask for reasoning, sources, and other relevant links

Current genAI tools will maintain this information in their context window for the length of a conversation, or will revisit the entire conversation to refresh a context window.

Step 1: Choosing a Filter Topology

We need to build four 2nd-order stages (two HPF, two LPF) and connect them. We need a topology that is

simple, stable, and easy to cascade. Non-inverting would also be more ideal.

Prompt for your AI partner:

"You are an experienced biomedical engineer with specific expertise in bioinstrumentation and electrical engineering. You are assisting a team of 3rd year undergraduate engineers with developing a device. I need to design a 2nd-order active filter. What is a common, non-inverting filter topology that uses a single op-amp and is well-suited for cascading to create higher-order filters? Please explain your reasoning."

- **Your Task:** Based on the AI's response, identify the topology we will use. (You are looking for the most common topology that fits this description). Justify your choice here using your class notes or textbook. If you change the prompt, include your new prompt here as well.

Step 2: Choosing a Filter Approximation

The "personality" or "design" of our filter (e.g., Butterworth, Chebyshev, Bessel) determines its performance. Before you design your prompt for your AI partner, consider a few questions.

When a doctor analyzes an ECG, are they looking at the *frequency* or the *shape* of the QRS complex? What does that imply for our filter design?

What happens to a complex waveform (like an ECG) if different frequency components (like the 5 Hz part and the 30 Hz part) are delayed by different amounts of time as they pass through our filter?

Prompt for your AI partner:

"For an ECG filter, _____"

- **Your Task:** Based on the AI's response, identify the filter approximation we will use. Explain *why* this approximation is superior to others for ECG analysis, and cite sources other than your AI partner. Be sure to include your prompt in the blank spot above.

Step 3: Choosing our corner frequencies

We need to choose which frequencies that we will allow through and which frequencies that we will attenuate, based on the frequency of the noise sources identified above.

Prompt for your AI partner:

"_____"

Your Task: Based on the AI's response, identify the corner frequencies of our bandpass filter. Be sure to include your prompt above. Do these values make sense based on what you know of electrophysiology from your previous courses?

Step 4: Design Calculations

Now you have your specifications:

- **Topology:**
- **Design/Approximation:**
- **Order:** 4th-order (cascaded 2nd-order stages)
- **Filter 1:** LPF, _____ Hz
- **Filter 2:** HPF, _____ Hz

A 4th-order filter may require two 2nd-order stages that are **not identical**. They have specific normalized

frequencies (f_n) and quality factors (Q) derived from the design polynomials. You will need these Q and FSF values from a look-up table. You will also need the standard transfer functions for both LPF and HPF for your chosen topology. If you use transfer functions with specific R and C variables, you should have a matching schematic for correct component placement.

Prompt for your AI partner:

"What are the standard cascade coefficients (Q and FSF) for a 4th-order _____ filter?"

Your AI should provide a table with values for two stages.

- **Your Task (First Principles):**

1. Derive the ideal theoretical transfer function for your chosen topology (from Step 1) for both the LPF and HPF. Do this manually using your notes for reference.
2. Ask the AI to generate the ideal transfer functions for your LPF and HPF stages.
3. Ask the AI for MATLAB code to model the transfer function and create a Bode plot.
4. Add your ideal transfer function to this Bode plot and compare it to the AI generated result.

Now, use the AI to get a *first-draft* circuit design in stages.

Prompt for your AI partner:

"Show me a 4th-order active high-pass filter circuit using a _____ topology with a cutoff frequency of _____ Hz and a unity gain."

- **Your Task (Simulation):**

1. Use the correct design coefficients to calculate the component (R and C) values for all four stages (2 LPF, 2 HPF).
2. Ask the AI to propose a circuit and calculate component values for each of your filter stages.
3. Your complete proposed circuit will be the HPF and LPF stages connected.
4. Plot these transfer functions on a Bode plot. Do these transfer functions match the transfer functions from the previous step?

Do not trust the AI's values blindly. Your job as an engineer is to *verify* the proposal.

Use the transfer function equations, FSF, and Q values to double check the proposed component values from your AI partner. Are the proposed values practical?

Consider adding a Constraint: Use standard, available capacitor values and calculate the necessary resistors. Show all calculations in your notebook.

4. In-Lab: Verification & Validation

Part 1: Design Verification (Simulation)

Before you build, you *must* verify your design in a simulator (like LTspice or Multisim).

1. **Build:** Construct your full 4-stage filters in the simulator using your calculated component values and the values suggested by your AI partner.
2. **Your Task:** In your notebook, write the symbolic transfer function for all four stages. Multiply them to find the total theoretical transfer function. This is your theoretical model.

3. Bode Plot (Frequency-Domain):

- Run an AC analysis on your simulated circuits from 0.01 Hz to 10 kHz.
- Generate a Bode plot (Magnitude and Phase) for each.
- **Verification Checkpoints:**
 - Does the theoretical Bode plot (generated by hand or in Matlab) match the Bode plot from your model?
 - Does the low-pass -3 dB corner frequency appear at _____ Hz?
 - Does the high-pass -3 dB corner frequency appear at _____ Hz?
 - Is the roll-off for the LPF -80 dB/decade? Is the HPF roll-off +80 dB/decade?
 - Is the phase response *linear* across the passband (____ Hz to ____ Hz)?
 - **Save the calculated transfer function Bode plots and the Bode plot(s) generated from your circuit model.**

Part 2: Hardware Validation (Prototyping)

Now, build the physical circuit on your breadboard using TL072, TL074 or LM358 op-amps.

Did the AI provide component values available in the lab?

1. Frequency-Domain Validation:

- Use the function generator (set to sine wave) and the oscilloscope to sweep the input frequency.
- Measure the output amplitude at various frequencies (e.g., 0.1, 0.5, 1, 10, 100, 150, 500, 1k, 10k Hz).
- Create an *experimental* Bode plot. How does it compare to your simulation?

2. Time-Domain Validation:

- This is the most important test.
- Set the function generator to a **10 Hz square wave**. This simulates the fast rise-time of a QRS complex.
- Feed this signal into your filter and observe the input and output on the oscilloscope.
- **Capture a screenshot** of the output signal. Note the amount of overshoot and ringing.
- **Comparison:** Ask your TA for a pre-built circuit. Feed the same 10 Hz square wave into it.
- **Capture a screenshot** of the prebuilt output.

5. Post-Lab Analysis & Discussion

In your lab report, include the following:

1. **AI Collaboration:** Attach a transcript or summary of your key interactions with your AI partner. Briefly reflect on the process. Was the AI helpful? Did it ever give a wrong or misleading answer? How did you verify its responses?
2. **Design Summary:** Include your final, calculated component values and the circuit schematic.
3. **Verification (Simulation):** Include your theoretical and simulated Bode plots (Magnitude and Phase). Label the corner frequencies and roll-off slopes.
4. **Validation (Hardware):**
 - Include your *experimental* Bode plot, overlaid on your *simulated* plot.
 - Include the two oscilloscope screenshots:
 - (A) The 10 Hz square wave response of your filter.
 - (B) The 10 Hz square wave response of the supplied filter.

5. Discussion Questions:

- Compare the square wave responses. Are they the same? How does this visual result confirm the "linear phase" property you researched in the pre-lab? Why is this essential for an ECG?
- Look at your Bode plot. How much attenuation (in dB) does your filter provide for 60 Hz powerline noise? Is this enough? If not, what other type of filter could you add to this design to specifically target 60 Hz noise?
- What are the main sources of error between your simulated and experimental results?
- What performance loss, if any, did you experience from "ideal" component values to realistic component values?