Robotics Education in STEM

Cameron Yeoman

Robotics Education in STEM

Author/Student: Cameron Yeoman

Date: 2025-10-07

Course / Assignment: Behaviour-Based Robotics — Maze Navigation Project

Instructor / Stakeholder: Vaughn

[New Zealand's strategy for artificial intelligence: Investing with confidence]

:____

Executive Summary

This document explores the structured use of AI tools such as ChatGPT and GitHub Copilot in robotics education. Drawing from my experience in the Behaviour-Based Robotics course, it shows how AI can enhance learning, speed up development, and support reflection when used responsibly under Vaughn's evaluation-only directive.

AI is presented not as a shortcut, but as a reasoning partner that assists with coding, debugging, and design thinking. Through the vibe coding method — a natural-language-driven workflow — this report demonstrates how AI can strengthen creativity, understanding, and integrity in technical education.

Introduction

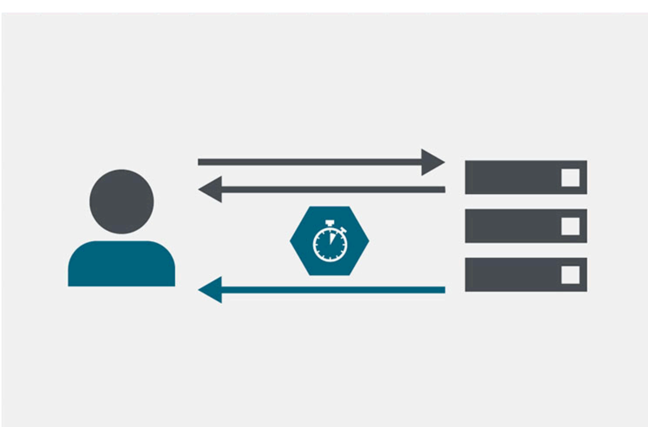
This document presents an argument for the continued and structured integration of artificial intelligence tools such as ChatGPT and GitHub Copilot within robotics and computer science education. Drawing from my final-year experience in the *Behaviour-Based Robotics* course, I argue that AI can serve as a transparent, ethical, and highly productive learning partner — one that deepens conceptual understanding, reduces unproductive trial-and-error, and enhances the quality of both code and reflection when guided by clear teaching frameworks such as Vaughn's *evaluation-only* directive.

Rather than acting as a shortcut or an unethical substitute for genuine work, AI functions as a reasoning companion — a system that helps students test logic, analyse design flaws, and iterate solutions faster while still demanding human judgement, critical thought, and accountability. In this context, the goal is not to let AI "do the work," but to use it as a *mirror for reasoning* — a tool that reveals gaps in understanding and encourages systematic problem-solving.

The purpose of this paper is twofold: first, to define and contextualise a development approach I call vibe coding — an intuitive, natural-language-driven workflow that uses AI assistants as creative and diagnostic collaborators; and second, to demonstrate how this method was applied in practice to meet the course learning outcomes and rubric expectations. Throughout the semester, our task was to design behaviour-based control systems (FSM, Behaviour Tree, or Subsumption) and demonstrate robust arbitration, sensor-driven responsiveness, and graceful degradation.

This document connects the use of AI-assisted workflows against those criteria, explaining how the chatbot was employed as both an evaluator and a design partner rather than an unreferenced code generator.

By situating AI tools within an accountable, transparent workflow — documenting prompts, citing outputs, and describing the purpose of generated suggestions — this course should aim to show that AI can augment human creativity and technical learning without compromising academic integrity. The following sections define *vibe coding*, present a reflective analysis of its integration within the project, and provide supporting artefacts that validate its educational and ethical soundness.



Understanding Vibe Coding

Definition and Rationale

Vibe coding is a natural-language-driven method of programming where the student communicates goals, conditions, and design intent to an AI model, which then produces ideas, scaffolds, or annotated suggestions. The human developer remains in control — testing, editing, and reasoning about every suggestion. This approach transforms coding into a conversation of ideas rather than a copy-and-paste task.

In robotics, this process fits seamlessly with project-based learning. Instead of spending excessive time on syntax or setup, students can focus on designing behaviours, arbitration, and degradation strategies — the deeper conceptual work the course is designed to assess.

Prompt Literacy as a Modern Engineering Skill

Prompt literacy — the ability to write, test, and interpret structured AI prompts — is becoming a vital 21st-century skill. In my own workflow, I used precise prompts to ask the AI about sensor arbitration logic, servo versus ballistic behaviours, and fallback mechanisms under sensor failure. These conversations accelerated my ability to identify design weaknesses and improve my code's robustness.

Research by Federiakin (2024) and Lee (2025) supports this approach, identifying prompt engineering as a core technical competency. Students who practice structured prompting not only improve their communication clarity but also develop the analytical discipline required to critique machine outputs effectively.

Enhancing Collaboration and Iterative Learning

Prompt-driven development also encourages collaborative problem-solving. When multiple students share prompts and AI-generated outputs, they can compare approaches, critique reasoning, and collectively refine solutions. This mirrors real-world engineering practices, where iterative testing, peer review, and design reflection are essential. In my experience, reviewing AI suggestions alongside classmates' interpretations often revealed alternative strategies or overlooked edge cases, reinforcing both technical understanding and critical thinking.

Vaughn’s Teaching Framework and My Experience

The “Evaluation-Only” Directive

Vaughn’s framework — where Copilot acts solely as an evaluator, not a generator — has proven to be a powerful teaching model. It enforces boundaries that protect academic integrity while allowing students to use AI responsibly. In this mode, the chatbot is trained to review, critique, or simulate analysis rather than produce final code.

Throughout the Behaviour-Based Robotics course, I worked within this structure, using AI to:

- Evaluate finite-state-machine logic and identify transition weaknesses.
- Discuss servo versus ballistic behaviour patterns.
- Test degradation handling (single-sensor failure recovery).
- Generate evaluation questions to stress-test design choices.

This reflective loop — prompt ➔ output ➔ test ➔ refine — mirrors the exact reasoning process used in autonomous systems. The more I used the model, the better I understood my robot’s decision-making and the course’s core concepts of behaviour-based control.

Student Reflection

Through this process, AI became a *discussion partner*, not a solution generator. It helped me uncover logical flaws, spot overlooked assumptions, and frame problems in new ways. I could talk through design choices in natural language — for example, asking “What would happen if the left ultrasonic fails mid-loop?” — and then test the AI’s reasoning against real results.

This ability to hold a “social conversation” with the code, exploring possible vulnerabilities or sensor issues, has made me a stronger, more reflective robotics student.

Why This Approach Benefits Students and Lecturers

Reducing Misunderstandings and Time Waste

Structured AI prompts save time on trivial misunderstandings, letting students spend their energy on conceptual understanding and physical testing. In practice, this means fewer repetitive mistakes, more efficient debugging, and deeper reflection during documentation.

Lecturers also benefit: because prompts, outputs, and reflections can be documented, the learning process becomes transparent. It's easier to see what the student asked, what the AI produced, and what changes were made — creating a traceable record of academic honesty and intellectual engagement.

Encouraging Real-World Engineering Practice

In modern engineering workplaces, AI is used daily to analyse code, generate test cases, and automate documentation. Teaching students how to use it critically and responsibly bridges the gap between university study and professional practice. Vaughn's model already aligns perfectly with industry ethics: controlled, transparent, and reflective.

This is why I believe courses like Behaviour-Based Robotics should continue to lead in AI-integrated learning. Rather than banning AI, we should guide it — using structured prompts and evaluation logs as educational tools. This not only protects integrity but also encourages innovation and higher-order thinking.

Conclusion

Vaughn's course model shows that AI use in robotics education is not a threat but a teaching evolution. Guided prompting, transparent evaluation, and documented reflection uphold academic integrity while advancing learning outcomes. AI should not be feared or hidden — it should be taught.

By continuing to refine and expand this model across robotics and computer science courses, educators can lead by example in shaping how future engineers collaborate with intelligent systems. I fully support this approach and believe it represents the next step in modern, ethical engineering education.

The following evidence and research further supports the benefits and pedagogical rationale for structured AI use in robotics education. These studies highlight both the current adoption of AI in higher education and the measurable advantages of AI-assisted learning methods such as vibe coding and prompt literacy.

Evidence and Research Sources

By continuing to refine and expand this model across robotics and computer science courses, educators can lead by example in shaping how future engineers collaborate with intelligent systems. I fully support this approach and believe it represents the next step in modern, ethical engineering education.

The following evidence and research further supports the benefits and pedagogical rationale for structured AI use in robotics education. These studies highlight both the current adoption of AI in higher education and the measurable advantages of AI-assisted learning methods such as vibe coding and prompt literacy.

- EDUCAUSE (2024–2025): Found that only 39% of universities have clear AI-use policies, recommending frameworks that emphasize transparency, reflection, and responsible engagement rather than outright prohibition. This aligns with Vaughn's evaluation-only directive, which allows safe experimentation without compromising academic integrity.
- HEPI (2025): Reported that 88% of students already use AI tools in their studies, underscoring the need for teacher guidance and structured support rather than fear-based restrictions.
- Pew Research (2024): Shows educators remain divided about AI use but increasingly recognize that guided integration improves engagement, learning outcomes, and trust.
- Microsoft Education (2025): Notes practical academic benefits of AI, including assistance with problem-solving, accessibility support, and efficiency gains.
- Federiakin, D. (2024): Argues that prompt engineering is a teachable and necessary skill for modern engineering education, helping students develop analytical and reasoning skills through structured AI interaction.
- Lee, D. (2025): Reviews strategies for integrating prompt literacy and AI reflection into university curricula, demonstrating how critical engagement with AI enhances learning depth and metacognitive skills.
- Fan et al. (2025): Demonstrates that AI-assisted pair programming improves motivation, depth of learning, and confidence, highlighting the benefits of collaborative, AI-supported STEM education.