# Vaman Rajagopal

856-360-0338 | vaman@umich.edu | linkedin.com/in/vamanr | vamanr06.github.io

## Education

**University of Michigan** Ann Arbor, MI

*BSE in Computer Science and Robotics* *Anticipated Graduation Date: May 2028*

- Courses: Data Structures and Algorithms, Machine Learning, Simultaneous Localization and Mapping

## Experience

**Research Assistant** May 2026 – Present

*University of Michigan* *Ann Arbor, MI*

- Developing ERAAS, a closed-loop embodied robotics system using **3 synchronized RGB views and a Qwen VLM** to select atomic robot skills and execute constrained RX200 actions through **ROS2 in Isaac Sim**
- Achieved **73.7% task success (221/300 trials)** with a **zero-shot Qwen 3.6 27B** controller and no robot-specific fine-tuning, translating text prompts into executable atomic robot skills
- Built an automated evaluation pipeline that runs repeated trials and records **tool calls, execution steps, and task outcomes** across manipulation tasks

**Software Engineer** Aug. 2025 – May 2026

*VTI Aerospace* *Ann Arbor, MI*

- Integrated a natural-language flight interface with **PX4 and MAVSDK** and exercised the system in **Gazebo** to verify request-to-flight translation, vehicle connectivity, and simulated mission execution
- Validated model-generated flight commands against **typed schemas and flight-safety constraints**, rejecting malformed or unsafe requests before they reached the drone-control stack or initiated simulated missions
- Converted live caller audio into typed, location-linked flight requests with **Whisper and OpenAI APIs** for software-mediated dispatch-to-drone handoff

**Autonomous Systems Engineer** Aug. 2024 – Aug. 2025

*MRacing* *Ann Arbor, MI*

- Implemented **C++ decoders for CAN telemetry** across vehicle ECUs, converting raw frames into typed brake, steering, and throttle signals for real-time monitoring, logging, debugging, and autonomy software
- Integrated **LiDAR into the SLAM pipeline** for localization and mapping under real-time compute constraints, keeping pose estimates available to perception and motion-planning components during vehicle operation
- Used **ROS interfaces** to connect localization outputs with the racecar's perception and motion-planning stack during vehicle operation

**Software Engineer** Sept. 2021 – June 2024

*Red Watch Robotics* *Medford, NJ*

- Implemented **AprilTag recognition and pose estimation** to measure robot-to-target distance, localize against field landmarks, and calculate approach trajectories for autonomous scoring from multiple starting positions
- Combined **PID controllers, IMU, and encoder feedback** to correct path execution and maintain closed-loop motion during autonomous runs while compensating for drivetrain drift at competition speed
- Led a **one-week** overhaul of autonomous paths, arm control, and scoring logic, helping the team rise from second-to-last to **2nd of 30 teams** in qualification rankings at the next competition

## Projects

**Firewatch** | *JavaScript, Overshoot, Arduino, WebSockets, VLMs*

- Integrated the **Overshoot streaming VLM API** with a live camera feed to translate scene observations into discrete motor commands for centering, approaching, stopping, and advancing through navigation targets
- Built a browser-to-Arduino control layer using **WebSockets and serial commands** with manual override, automatic reconnection, and **per-inference latency logging** for repeatable hardware testing

**Educational Classification Benchmark** | *PyTorch, DSPy, Scikit-learn, Embeddings*

- Built and benchmarked **logistic regression, neural network, embedding, and LLM** pipelines under a shared multiclass evaluation setup spanning classical, neural, semantic, and generative approaches
- Achieved **80% test accuracy** with structured LLM function calling across **1,000+ labeled student responses**

## Technical Skills

**Robotics**: ROS2, PX4, MAVSDK, Gazebo, Isaac Sim, LiDAR, SLAM, AprilTag, PID
**Embedded**: Arduino, Raspberry Pi, CAN bus, IMU, encoders, sensor integration
**Software & AI**: C++, Python, Java, JavaScript, TypeScript, Go, SQL, PyTorch, OpenAI APIs, Docker, Linux, Git