Max Freeman

max_freeman@berkeley.edu **|** (607) 327-7804 **|** linkedin.com/in/maxfreeman **|** Mountain View, CA

Education

University of California, Berkeley **August 2024 - May 2025**
Master of Engineering in Mechanical Engineering, *Concentration in Controls & Robotics* GPA: 3.98

Cornell University **September 2020 - May 2024**
Bachelor of Science in Mechanical Engineering GPA: 3.77 **|** *Magna Cum Laude*

Skills

Embedded Systems: Sensors & Actuators **|** Communication Protocols (CAN, I2C, UART) **|** Oscilloscope Debugging
Hardware & Prototyping: System Bring-Up & Integration **|** CAD (SolidWorks) **|** 3D Printing **|** Manual Mill & Lathe
Programming & Software: Python **|** C++ **|** ROS2 **|** MATLAB **|** Simulink **|** Git **|** Linux
Robotics & Controls: Feedback Control (PID, LQR, MPC) **|** State Estimation & Sensor Fusion **|** Dynamics & Kinematics

Highlighted Experience & Projects

System Integration Intern, *Toyota Research Institute* **September 2024 - May 2025**

- Support cross-functional integration of mechanical, electrical, and software systems to enable rapid prototyping and testing of next-generation research vehicles.
- Integrate a steering wheel encoder into an existing vehicle platform, including CAN communication setup, connector and wiring fabrication, SolidWorks-based mount design with 3D printing, and integration of sensor data into a ROS2 pipeline to enhance vehicle localization accuracy.
- Enable on-vehicle deployment of an NVIDIA Jetson for real-time monitoring and diagnostics by collaborating with research scientists to define interface requirements, designing custom sheet metal mounting brackets, and developing power and communication harnessing within the vehicle architecture.

Lead Robotics Engineer, *Multimodal Autonomous Robot Design (MEng Capstone)* **September 2024 - May 2025**

- Led a team of 4 engineers in a full-stack robotics project to develop a multimodal robot capable of both driving and flying, owning system architecture, control system development, and hardware testing strategy.
- Designed and deployed a ROS2-based flight controller in Python with a cascaded PID architecture and integrated failsafes, validating flight performance through staged subsystem checks and flight tests.
- Implemented and tuned Model Predictive Control (MPC) algorithms for trajectory planning and obstacle avoidance, achieving <5 cm lateral error to final setpoints in hardware testing.
- Developed a lightweight Python simulator to validate controller behavior before hardware deployment.
- Conducted simulator and hardware testing to validate control strategies, accelerate iteration, and identify failure points early, mitigating downstream development risks.

Control of Autonomous Flight Project, *University of California, Berkeley* **September 2024 - December 2024**

- Developed a quadcopter flight controller in C++, achieving precise attitude, altitude, and position control.
- Implemented sensor fusion models in C++, utilizing data from optical flow, IMU, and Time-of-Flight sensors to improve sensor data accuracy and provide precise feedback for control.
- Diagnosed and resolved stability issues through targeted subsystem testing, sensor debugging, and PID controller tuning to validate fixes and improve system response.

Hardware Test & Integration Intern, *Lit Motors* **June 2024 - July 2024**

- Contributed to test infrastructure development and subsystem design refinement at a fast-paced startup building a two-wheeled, self-balancing vehicle actuated by Control Moment Gyroscopes (CMGs).
- Led development of a dual-plane dynamic balancing test rig for CMG evaluation, integrating structural hardware, sensors, and a belt-drive system to identify weight imbalances and reduce vibrations.

Fast Robots Project, *Cornell University* **January 2024 - May 2024**

- Managed end-to-end development of an embedded autonomous RC robot, integrating sensors and motor drivers on an Arduino Nano and implementing control logic in C++ and Python.
- Interfaced with Time-of-Flight and IMU sensors over I2C, developing sensor fusion algorithms and software-based filters in C++ to minimize sensor output noise by over 50%, enhancing sensor accuracy.
- Implemented and tuned distance-based PID controllers to improve tracking, reducing settling time by 40%.
- Diagnosed hardware and sensing issues using oscilloscope traces and telemetry data, resolving integration faults, and improving system performance.