Rajiv Govindjee

http://ocf.io/rgovindjee
rajivg@berkeley.edu

EDUCATION

UNIVERSITY OF MICHIGAN
M.S. Aerospace Engineering
Autonomous Systems & Control
December 2023 | Ann Arbor, MI
GPA: 4.0 / 4.0

UC BERKELEY
B.S. Electrical Engineering & Computer Science
Minor: Mechanical Engineering
May 2022 | Berkeley, CA
Overall GPA: 3.6 / 4.0
EECS GPA: 3.8 / 4.0

LINKS

Github:// **rgovindjee**
LinkedIn:// **rgovindjee**

COURSEWORK

- Deep Neural Networks
- Artificial Intelligence
- Advanced Computer Vision
- Action and Perception
- Advanced CubeSat Design
- Flight and Trajectory Optimization
- Model Predictive Control
- Navigation and Guidance
- Stochastic State Estimation

SKILLS

PROGRAMMING
Proficient:
C/C++ • Python • MATLAB / Simulink • Java • PyTorch • Pyomo • ROS • RISC-V • Docker • Git • hg • vim • LaTeX

Familiar:
Julia • Go • CUDA • ARM mbed • SQL • GNURadio • Verilog • MuJoCo • CVX

SOFTWARE TOOLS
Proficient:
Cadence Allegro • Eagle • KiCad • SolidWorks • Fusion360 • Adobe Creative Cloud • Microsoft Office

Familiar:
CATIA • Altium • SPICE • Vivado

LANGUAGE
Spanish • German

LICENSES
High Power Rocketry: TRA Level 2
Amateur Radio: FCC General Class

EXPERIENCE

DISTRIBUTED AEROSPACE SYSTEMS AND CONTROL LAB
Research Assistant
Jan 2023 – Dec 2023 | Ann Arbor, MI

- Developed algorithms for safe visual odometry and mapping in Julia and C++
- Reduced runtime of perception pipeline by 10x, allowing for real-time operation

WING (ALPHABET) | Flight Automation, GNC Intern
Summer 2023, Summer 2021 | Palo Alto, CA

- Developed new path planning and trajectory optimization framework for precise landing, pickup, and delivery, reducing tracking error by over 30%
- Proposed and evaluated active perception algorithms, improving navigation
- Designed control mixer for concept fixed-wing eVTOL aircraft in MATLAB / Simulink and integrated embedded C++ autocode with on-vehicle code

XONA SPACE SYSTEMS | Spacecraft Software Engineer
Jan 2023 – Mar 2023 | Burlingame, CA (contract, remote)

- Wrote on-vehicle C++ for position, navigation, and timing constellation in LEO
- Created Python test client to run functional integrated tests via TCP sockets

ZOOX (AMAZON) | Safe Perception, Collision Avoidance Intern
Jun 2022 – Aug 2022 | Foster City, CA

- Developed interpretable LiDAR point cloud instance segmentation algorithms and improved performance on real-world autonomous vehicle driving data
- Debugged CUDA C++ memory layout and access issues in on-vehicle code
- Redesigned configuration parameter system to use fixed-memory size serialization for ISO 26262 / AUTOSAR compliance

APPLE | iPhone Electronics Hardware Co-Op
Jan 2020 – Jan 2021 | Cupertino, CA

- New silicon bringup, digital signal integrity, and root cause analysis for iPhone

BOEING | Flight Controls Intern
May 2019 – Aug 2019 | Everett, WA

- Developed a first-principles dynamic model and ran Monte Carlo simulations in MATLAB to assess safety issue for the 787 high lift system (AD 2019-20-07)
- Created a Simulink model to assess proposed electric motor design change

EXTRACURRICULARS

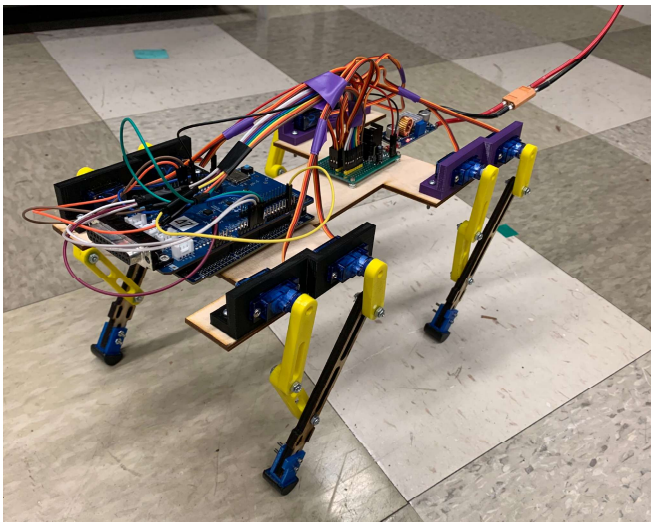
SPACE TECHNOLOGIES & ROCKETRY CLUB (STAR)
Systems Engineering Lead, Operations and Safety Officer
Aug 2017 – May 2022 | UC Berkeley

- Led team of nine members to develop and manage CONOPS, requirements, FMEAs, BOMs, verification & validation plans, and checklists
- Coordinated vehicle final integration and launch operations for four launches
- Performed electrical and mechanical design and fabrication for scientific payloads, liquid bi-propellant engine controller, and recovery electronics

EE 198 (HANDS-ON PCB ENGINEERING) | Lead Instructor
Jan 2021 – May 2022 | UC Berkeley
Created and delivered weekly lectures to class of 25+ students. Created and ran PCB design labs in KiCad + 3 hands-on assembly and bringup labs with 4-person staff.

Rajiv Govindjee

http://ocf.io/rgovindjee
rajivg@umich.edu | (925) 444-5009



TART: TART Autonomously Roams Terrain

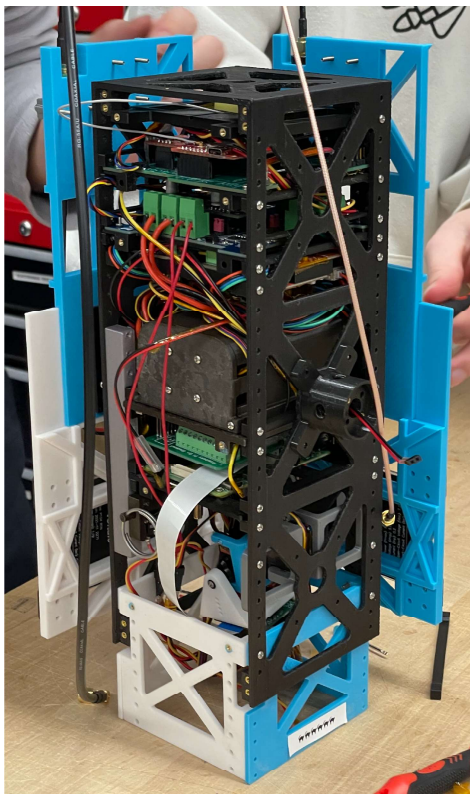
Embedded Systems (EECS 149) and
Feedback Control for Legged Robots (ME 193B)

TART is a low-cost quadrupedal robot intended to move with both dynamically and statically stable gaits. Designed for potential applications in disaster relief, TART can report environmental sensor data via Bluetooth Low Energy (BLE); sensors record air pressure, humidity, light intensity, temperature, and potentially volatile organic compounds. I designed, tested, and implemented the control system. I also designed the 5-bar leg linkages using nonlinear optimization to maximize the reachable space for each foot.
Video: https://bit.ly/TART-video

RAWB: RAWB Assists With Balancing

Robotics (EECS 106A)

RAWB uses a UR5e robotic arm to comfortably balance and move a tray with objects on it in collaboration with a human. Interaction with the robot is natural and predictable, and occurs in 4 degrees of freedom (three translational and one rotational). I worked on control system design, mechanical design, and construction of hardware.
Video: https://bit.ly/RAWB-video



MAC: MAC Advanced CubeSat

Advanced CubeSat Design (AEROSP 740)

MAC is a 3U CubeSat successfully launched and recovered with a high-altitude balloon. The flight over southern Michigan lasted one hour and reached a maximum altitude of around 26,000 m (85,000 ft) MSL, recording environmental data and imagery throughout. I led the communications subteam, successfully demonstrating half-duplex digital packet radio (70 cm) between a ground station and the satellite. I also led exploration of long-distance HF locationing techniques using WSPR (20 m), and succesfully managed recovery with commerical APRS trackers (2 m). In addition to my radio work, I helped design and assemble PCBs to organize and simplify wire harnessing. I also assisted teammates with C/C++ embedded software for the STM32L432KC on board.
Video: https://bit.ly/MAC-video

Rajiv Govindjee

http://ocf.io/rgovindjee
rajivg@umich.edu | (925) 444-5009

AirBears and Bear Force One

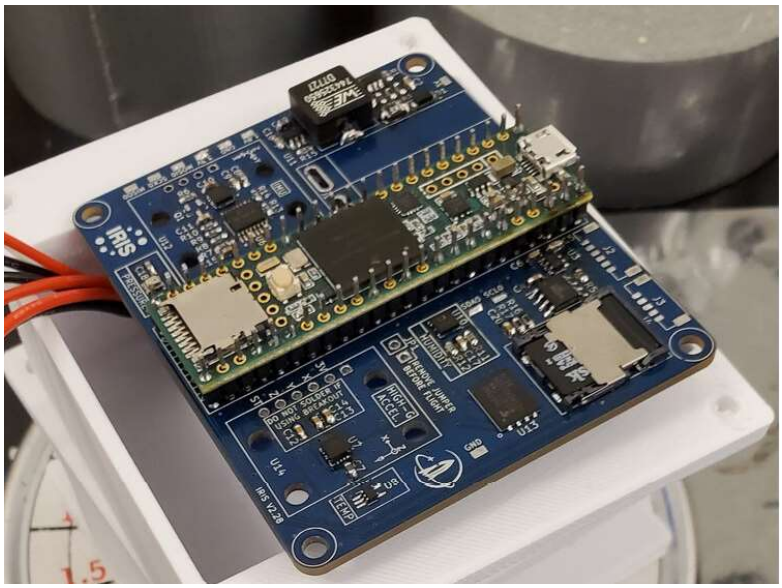
Space Technologies and Rocketry (2019-2021)

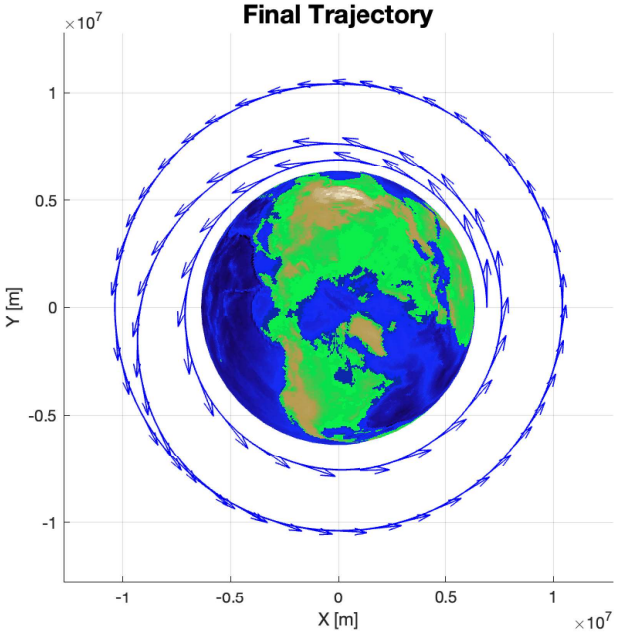
AirBears (left) and Bear Force One (right) are solid-fuel sounding rockets. AirBears is a test vehicle and has been flown twice (apogee ~7,700 ft); Bear Force One placed first in the collegiate FAR 1030 competition by carrying over 8.8 lbm of scientific payloads to over 10,000 ft. I was the team's Operations and Safety Lead for AirBears, directing vehicle final integration and coordinating launch-day procedures. I was the Systems Engineering Lead for Bear Force One and ultimately responsible for cross-functional engineering issues through design, production, and launch.
Video: https://bit.ly/BFO-video

IRIS: IRIS Records Information via Sensors

Space Technologies and Rocketry (2018-2021)

IRIS is a self-contained payload designed to measure barometric pressure, acceleration, orientation, temperature, and humidity throughout a rocket's flight. Data can be logged to both a soldered flash memory chip or removable SD card. Versions of IRIS have flown on club sounding rockets (see above), and been recovered successfully. I worked on component selection, schematic, and layout design for early versions of IRIS; I provided design review, assembled boards, debugged electrical issues, and wrote C/C++ software throughout.
ECAD (sensing board): https://bit.ly/IRIS-core
ECAD (power board): https://bit.ly/IRIS-power





MPC for Satellite Orbit Raising and Circularization

Model Predictive Control (ME C231A)

This project uses model predictive control techniques to plan trajectories for satellite orbital maneuvers. The highly nonlinear orbital dynamics are first linearized and discretized about a reference trajectory, and constraints are convexified. Control inputs are then generated using Pyomo, an open-source optimization modeling language, and an interior point solver (ipopt). I designed the overall software architecture and developed the closed-loop controller architecture, including the use of nonlinear sequential convex programming (SCPn) to increase the accuracy of generated trajectories. I also developed an extensive test suite with both unit tests and integration tests. Future work includes extending the code to work for reconfiguring constellations of small satellites.
Code: https://bit.ly/mpconstellation-code