Ian Cummings

iancummings2020@gmail.com | LinkedIn
+1 (267) 875-5400 | GitHub
Personal Website

Education

Northeastern University Boston, MA
BS Electrical and Computer Engineering, Minors: Mechanical Engineering and Mathematics *May 2024*

- First Place EECE Capstone Project
- Involvement: Northeastern Climbing Team, SASE

Northeastern University Boston, MA
MS Electrical and Computer Engineering *December 2024*

- Courses: Machine Learning and Pattern Recognition, Reinforcement Learning, Advances in Deep Learning, Parallel Processing, Robotics Sensing and Navigation, Mobile Robotics

Experience

Software/Electrical Engineer July 2023 – December 2023
Belmont Medical Technologies *Billerica, MA*

- Technical lead on a real-time linear regression machine learning algorithm implemented in C source code to detect a low-frequency failure mode.
- Lead of an investigation on a failure occurring on field units. Characterized test setup and conditions to develop a DOE and find root-cause of failure
- Developed an automated testing setup for a low-frequency failure with loss of memory. Used a Raspberry Pi to control solenoids, actuators, and an AC power relay to cycle charging/discharging of battery. Status was monitored over serial with the machine and the Pi would send an email if any error in testing occurred. Developed a signal routing PCB, designed the schematic, layout, BOM, and assembled the board.
- Developed an accelerated-life testing scheme to better characterize degradation of device performance.
- Developed an interfacing GUI compatible with all devices using Python. Utilized serial communication with the devices to log data, plot live data, and manually control the device.
- Developed a Python script to process raw log files from a device and various DAQs that were used in testing. Data was loaded into a Jupyter notebook with plots as well as exported into an excel sheet.
- Analyzed electrical schematics and developed a unit simulator test for the temperature measurement flow (hardware-¿software-¿software calibration) of a device.

MATLAB/Simulink Controls Software Engineer July 2022 – December 2022
Nuvera Fuel Cells *Billerica, MA*

- Developed, implemented, and documented software changes to the engine control unit (ECU) using Simulink/MATLAB
- Flashed software changes to ECU and extensively tested changes with live engines in a lab
- Processed, analyzed and presented data from engine data log using MATLAB and Python
- Worked as a lead of a small project, developing test plans, implementing necessary software changes, and creating test reports to move the project into further stages of development.
- Worked on testing and debugging of a high frequency resistance (HFR) measurement device, which communicates over the CAN bus
- Tested and debugged LabView softare to control a National Instruments module used for calibration of the ECU's analog input pins
- Extensive debugging and work with C source code of the ECU, particularly regarding I2C communication between the EEPROM and microcontrollers
- Ran MIL and HIL simulations to test major software changes

Projects

MAPLEE (Northeastern University ECE Capstone Project) May 2023-May 2024

- MAPLEE (Modular, Autonomous Platform for Landscaping and Environmental Engineering) is an autonomous robot that was developed for the use of the Northeastern Landscaping team to traverse campus and pick up weeds and trash.
- Worked on the perception aspect, creating a custom implementation of SSD300 and SSD512, as well as used Ultralytics for YOLOv8-n. Merged various online datasets to develop one large training/testing dataset, and trained the models over various number of epochs.
- Implemented the perception code into the overarching ROS codebase, and assisted with tracking, localization, and mapping code.
- Developed an interfacing GUI to use GPS information from robot (with the idea of scaling to multiple robots) to show location of robots on map, show status of each robot, and include a chat window for the natural language processing controls.

Electric Scooter Security System May 2024

- My dad owns an electric scooter that does not have any security measures, so he requested a basic, cheap, and small (fit with the rest of the packed electronics) security device for it.
- Used a Raspberry Pi Pico, Adafruit accelerometer, buzzers, Pre-assembled RF key-fob, DC-DC converter, and a power relay. The system used the DC-DC converter to power from the battery (60V), disconnected the battery from the rest of the system when locked, and set off a loud buzzer alarm if any movement was detected.
- For a future iteration of the design, plans are made to connect the alarm to the horn of the scooter for a louder alarm system, as well as exploration into break-engagement while locked.

Technical Skills

Languages:	English, German (Intermediate), Mandarin (Basic)
Programming Languages:	Python, C/C++, MATLAB/Simulink
Frameworks:	PyTorch, TensorFlow, ROS1, PySpark, PyQt, REST API
Engineering Tools:	Git/Github, OrCAD, Altium, AutoCAD, SolidWorks

Hobbies

Rock Climbing, Mountaineering, Skiing, Cooking