# Ralph Quartiano

516-660-3270 | ralphquartiano@gmail.com | linkedin.com/in/ralph-quartiano | ralphq.github.io | **Active TS Clearance**

## Education

**Penn State University** — State College, PA
*Bachelor of Science in Aerospace Engineering (Autonomy Focus)* — *Class of 2023*
- *Relevant Coursework: Aerospace Autonomy, Aircraft Stability & Controls, Orbital Mechanics*

**Georgia Institute of Technology** — Atlanta, GA
*Masters of Science in Computer Science (Perception & Robotics Focus)* — *Expected Grad. 2026*
- *Relevant Coursework: Bayesian State Esimation, Kalman Filters, Motion Planning*

## Experience

**Software Engineer for Advanced Computing (AI Division)** — August 2023 – Present
*Carnegie Mellon University Software Engineering Institute* — *Pittsburgh, PA*
- Profiled & benchmarked **electronic warfare** application on NVIDIA Orin **GPU** to identify algorithms with acceleration potential and develop test harness
- Designed, trained, and synthesized **neural network** for RF modulation classification on **FPGA** using **PyTorch**, **ONNX**, and open-source **HLS** tooling, speeding up inference by 25% at reduced power consumption
- Refactored PyTorch **convolution** layers into overlap-add **FFTs** for 3x speedup on digital signal processing ASICs
- Utilized neural network optimization and quantization tools in **TensorRT** to reduce size of YOLO **object-detection** models by over 4x and convert algorithms from Python scripts into C++ executables

**GNC Engineering Intern (ADCS, Starshield)** — May 2023 – July 2023
*SpaceX* — *Hawthorne, CA*
- Performed single & multi-component **fault tolerance testing** of satellite attitude determination & control systems (ADCS) using **Monte Carlo** techniques in **6-DoF C++ simulations**
- Identified attitudes in which **magnetorquer**-based angular momentum desaturation system is underactuated and orbital trajectories in which **star tracker** attitude determination system has severely reduced visibility
- Analyzed angular momentum/torque envelopes in various reaction wheel failure modes to update state transition thresholds with **MATLAB**
- Rewrote and **unit-tested flight software** to better reflect failure mode survivability in C++

**Autonomy Engineering Intern** — May 2022 – August 2022
*Palski & Associates Satellite Engineering* — *Colorado Springs, CO*
- Trained collaborative autonomous satellite agents for military rendezvous-proximity operations in geostationary orbit with several reinforcement learning techniques in **Julia**
- Developed 6-DoF simulation environment with realistic orbital mechanics and satellite performance, allowing agents to autonomously find desired orbit trajectories and calculate thruster burns required to enter them
- Combined Julia's **multiprocessing** capabilities with **Linux virtual machines** running on company servers to reduce training times by over 50% and allow for longer training runs while out of the office

**Undergraduate Dynamics & Controls Researcher** — January 2022 – January 2023
*Penn State University* — *State College, PA*
- Designed, simulated, and built momentum wheel-based stabilization system for an inverted pendulum
- Converted **Simulink** dynamics model and **PID controller** to C++ executable for use on Arduino Mega

**Lead Systems Engineer** — November 2019 – January 2021
*NASA (Oasis Project, 2020 BIG Idea Challenge Finalist)* — *State College, PA*
- Led development & testing of laser spectroscopy instrument for lunar rover with 20+ engineers and $145K budget
- Utilized **systems-engineering** methodology throughout design life cycle while adhering to NASA requirements

**Software Engineer** — April 2022
*Sauron (HackPSU 2022 AI and Overall Best Hack Winner)* — *State College, PA*
- Constructed data pipeline that used **computer vision** to classify and locate hazardous events in humanitarian crises and Google Cloud Platform to geo-tag and upload them to a live and public map

## Technical Skills

**Languages (Libraries)**: Python (PyTorch, Tensorflow, ONNX) , C/C++ (OpenCV), Matlab (Simulink), Julia
**Developer Tools**: Git, Docker, Linux, CMake/Bazel
**Hardware**: GPU Programming, HLS for FPGAs, Multiprocessing