

Saptak Das

Guidance, Navigation & Control · Flight Dynamics

✉ saptak.das625@gmail.com [🌐 linkedin.com/in/saptak-das1729](https://www.linkedin.com/in/saptak-das1729) [🐙 github.com/Saptak625](https://github.com/Saptak625) [🌐 saptakdas.vercel.app](https://saptakdas.vercel.app) 🇺🇸 U.S. Citizen

Aerospace engineering student at Georgia Tech (B.S. May 2027, GPA 3.96/4.00, BSMS Honors Program) building flight-proven guidance, estimation, and control: from stability theory and LQR design to embedded C on flight hardware and multi-agent reinforcement learning at Sandia National Laboratories. As President of Georgia Tech Experimental Rocketry, I led a 100-member team through the July 2026 flight of Mach 'N' Roll, a 17-ft, 283-lb, two-stage rocket, and published the roll-coupling analysis that informed its spin-stabilization strategy.

31,850 ft
club-record apogee,
two-stage Mach 'N' Roll

250,000+
designs optimized in 4 days
on GT PACE HPC

200+
flight trim points linearized;
100+ hours saved

AIAA
inertial roll-coupling paper, June 2026



Nominal flight of Mach 'N' Roll (17 ft, 283 lb, two-stage), launched July 2026 to a 31,850-ft apogee

SELECTED WORK

- 01 Inertial Roll Coupling** — stability model, AIAA paper, and flight validation on Mach 'N' Roll
- 02 Design Optimization & Dispersion at Scale** — 250,000+ designs on PACE HPC; Monte Carlo recovery-dispersion regions
- 03 Thrust-Vane Rocket GNC** — trim linearization, LQR on STM32, QUEST, EKF prototype
- 04 Multi-Agent Deep RL** — hypersonic intercept guidance (Sandia) and the CHIMERA self-play agent
- 05 Autopilot & Fault-Tolerant Sensing** — DHC-2 Beaver multi-loop autopilot, ILS, quadruplex sensor voting
- 06 Autonomy & Research** — F1Tenth racecar stack, NASA ULI IMOCAS turbulence modeling, CEREAL Lattice Boltzmann solver

+ **Impact, Publications, Toolbox & Experience**

PROBLEM

GTXR rockets are not actively controlled, so small imperfections in the vehicle and launch can induce roll. An earlier GTXR vehicle, Fire on High, held its roll rate inside a bounded band where small pitch–yaw perturbations grow rapidly through inertial coupling; the angle of attack kept increasing until aerodynamic loads tore the vehicle apart. GTXR therefore adopted spin stabilization, deliberately setting roll rate outside the unstable band. For the new Mach ‘N’ Roll vehicle, we needed to know where that band sits and how to stay out of it.

APPROACH

- Built a compact dynamical model from rigid-body rotational dynamics and linear aerodynamic restoring moments, with coupled pitch–yaw equations under a prescribed roll rate
- Analyzed stability as a function of roll rate by tracking how the system eigenvalues move; the instability region begins where a converging pair leaves the imaginary axis
- Benchmarked against the classical Mandell and LaBudde–Fetter formulations
- Computed roll-rate profiles over time for candidate sustainer fin cant angles and selected a **1.2° cant** to avoid the instability region

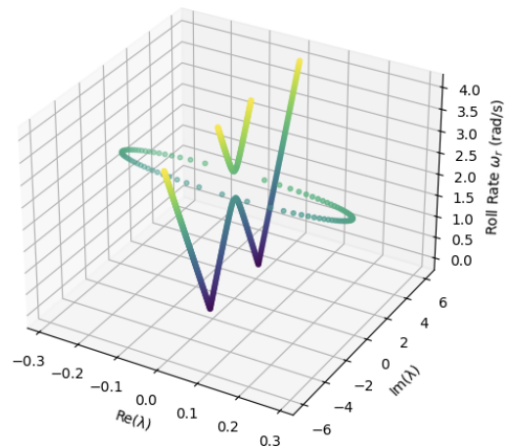
RESULT

- Identified a **bounded roll-rate instability region** where small attitude perturbations grow rapidly
- With 1.2° sustainer fin cant, roll rate stays **below** the region during full-stack flight, accelerates rapidly **through** it at sustainer staging, and stays **above** it for the rest of the flight
- **Flight data validated the strategy** on flights both below and above the dangerous coupling frequency
- Supported flight-dynamics trade studies for Mach ‘N’ Roll, which reached a club-record 31,850 ft

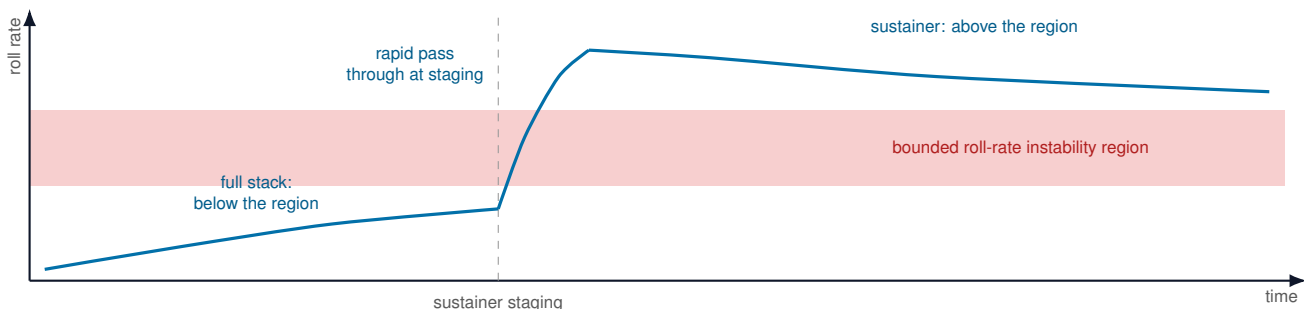
MY ROLE

Developed the model and stability analysis, wrote the paper, and led the program as GTXR President. Tools: Python, NumPy, Matplotlib, Pandas.

Eigenvalue Trajectories in Complex Plane vs Roll Rate



System poles versus roll rate ω_r . At a critical roll rate the converging pair of eigenvalues leaves the imaginary axis and moves along the real axis; the eigenvalue with a positive real part makes the system unstable, and its magnitude is the degree of instability.

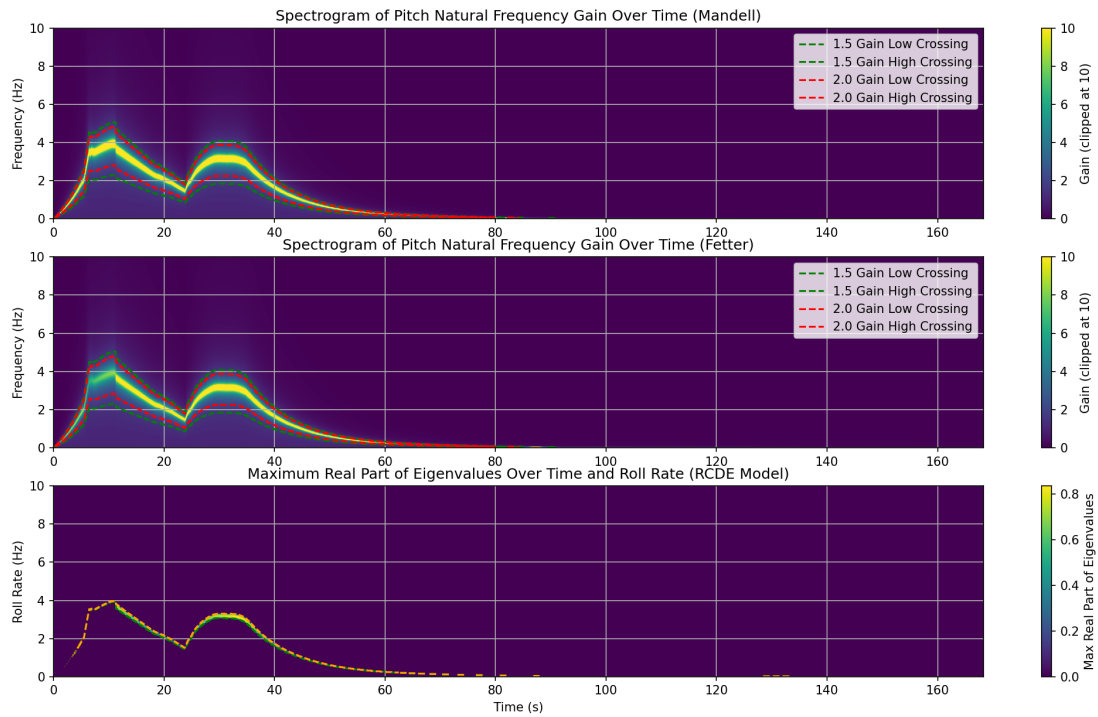


Conceptual roll-rate history with 1.2° sustainer fin cant (schematic, not to scale; see the spectrogram on the next page for the computed profile)

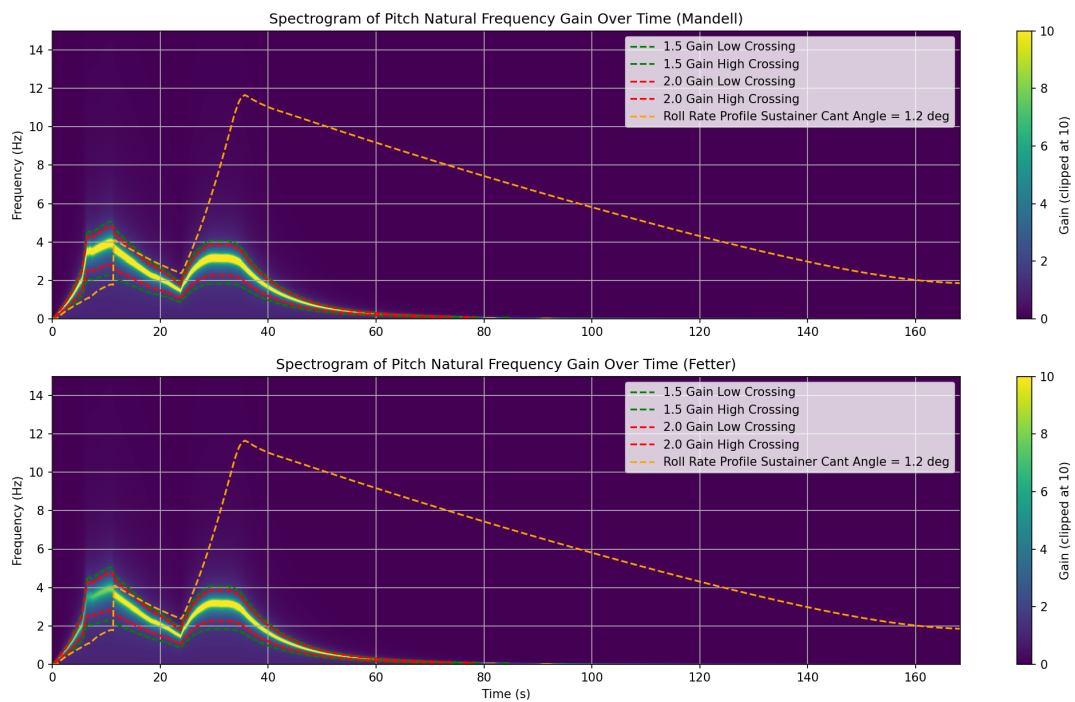
01

Inertial Roll Coupling

Model comparison and fin-cant selection for Mach 'N' Roll



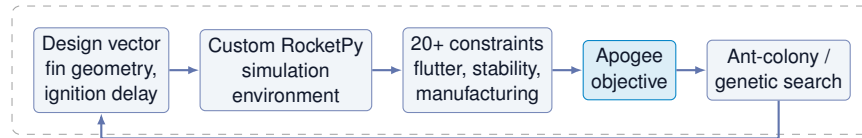
Spectrograms of pitch natural frequency gain over time for the literature models (Mandell, LaBudde–Fetter) compared with my custom RCDE model for Mach 'N' Roll.



Roll-coupling instability region over the Mach 'N' Roll flight with the expected roll rate for 1.2° canted sustainer fins (dashed orange). This cant keeps roll rate below the region during full-stack flight, accelerates it rapidly through the region at sustainer staging, and holds it above the region for the remainder of the flight.

A. FIN GEOMETRY AND IGNITION-DELAY OPTIMIZATION

Slurm jobs on Georgia Tech PACE HPC · 250,000+ designs evaluated in four days

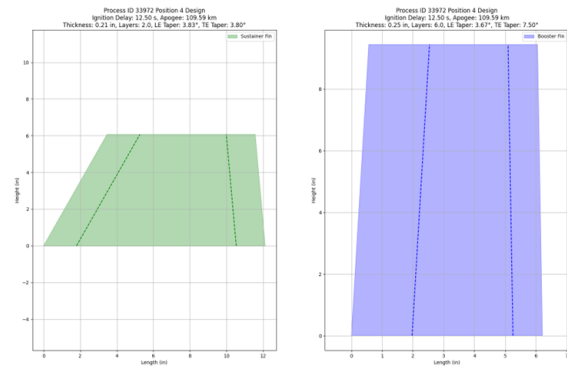


PROBLEM

Fin geometry and ignition delay for a two-stage rocket interact with flutter margins, stability, and manufacturing constraints, making manual tuning insufficient for exploring the design space.

APPROACH

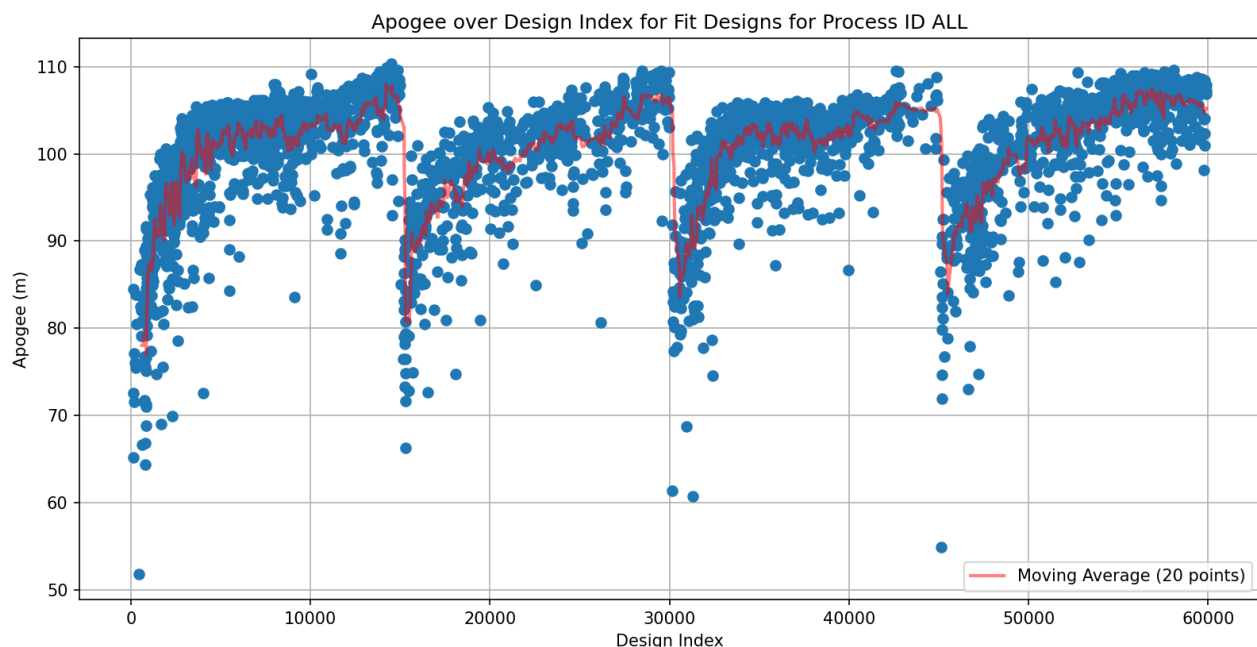
- Generalized **ant-colony/genetic optimization** in a custom **RocketPy** simulation environment under aerodynamic flutter, stability, and manufacturing constraints (20+)
- Ran **250,000+ candidate designs in four days** through Slurm on Georgia Tech's PACE cluster to identify high-performing configurations



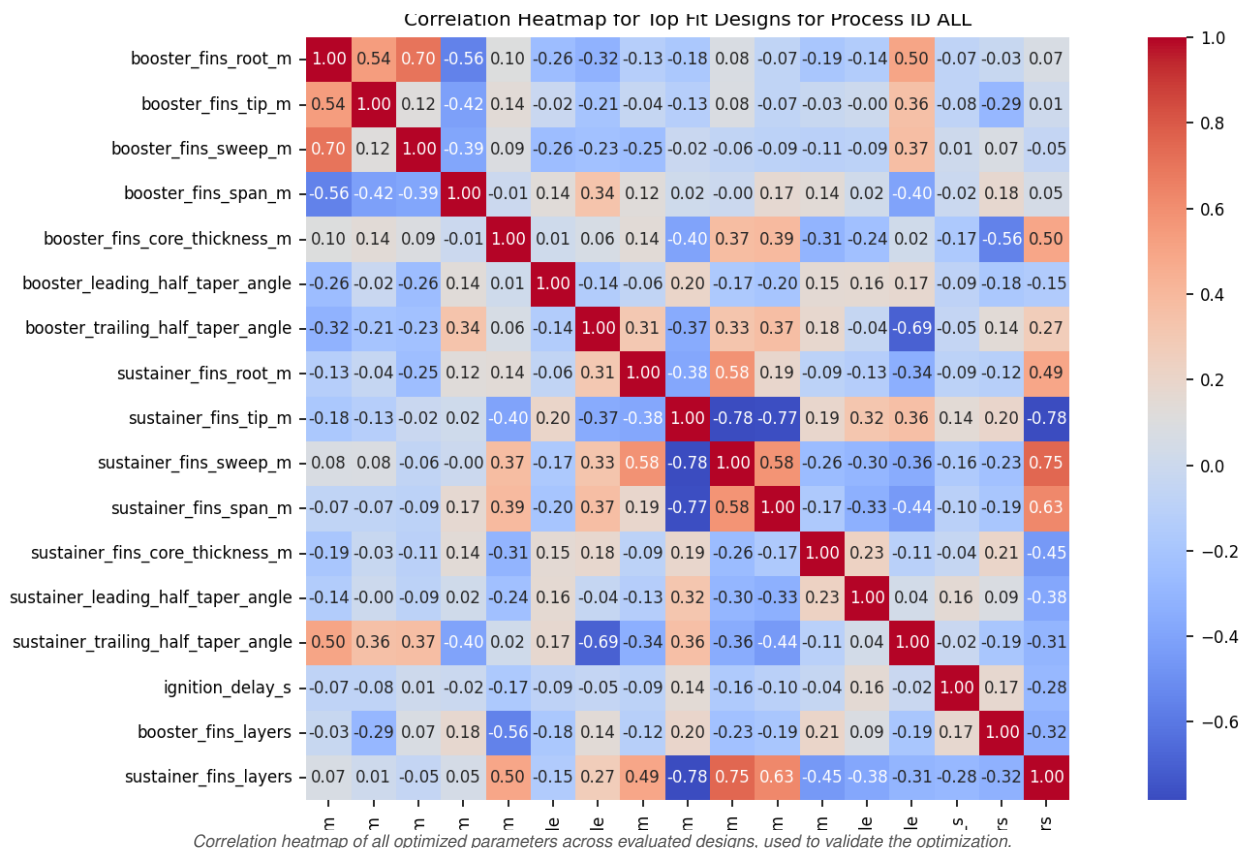
Selected hexagonal-cross-section booster (green) and sustainer (blue) fin shapes after optimization.

RESULT

- 35 km gain in simulated apogee** over the initial baseline design (about 75 km to about 110 km)
- 8 independent optimization runs, each starting from a unique initial population, converged on similar design spaces, giving confidence in the final design



Optimizer convergence for four independent optimization runs: initial-population apogee starts near 75 km and converges to final-solution apogees near 110 km.



B. RECOVERY-DISPERSION REGIONS FROM MONTE CARLO



PROBLEM

The recovery team needed knowledge of landing locations prior to launch across various nominal and off-nominal flight profiles.

APPROACH

- As president, led verification and validation across analysis, PDR, CDR, tolerance analysis, subassembly integration, and full-system integration testing for a 100-member team to reduce vehicle design uncertainty.
- Ran a **2,000-flight Monte Carlo dispersion analysis** of the trajectory and recovery and generated **three-sigma recovery-dispersion regions** so the recovery team knew where to search before launch. Inputs were dispersed separately for six cases: full stack, sustainer stack, booster, sustainer, nosecone, and full sustainer after a failed ignition, with every input limited to $\pm 3\sigma$.

RESULT

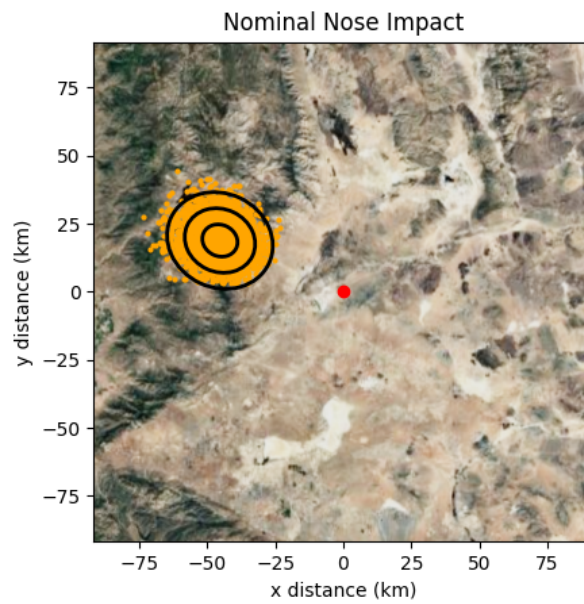
- Streamlined field logistics and cut recovery search time by 4+ hours via pre-computed dispersion mapping.
- Successfully recovered the Mach 'N' Roll vehicle intact** directly inside the modeled landing dispersion region.

DISPERSED INPUTS, FULL-STACK CASE

Input	Mean	1 σ
Mass (kg)	67.14	4
$I_{xx} = I_{yy}$ (kg m ²)	66.91	0.5
I_{zz} (kg m ²)	0.3447	0.01
Nose $\rightarrow$ CG (m)	2.399	0.0127
Drag coefficient (factor)	1	0.05
Thrust (factor)	1	0.05
Wind x, y velocity (factor)	1	0.25

02 Design Optimization & Dispersion at Scale

B. Predicted dispersion and recovered hardware



Recovered booster stage



Recovered nosecone

03

Thrust-Vane Rocket GNC

Georgia Tech Ramblin' Rocketry, Guidance, Navigation, and Controls · Aug. 2024 – May 2025

PROBLEM

A thrust-vane rocket needs controllers that hold up across its flight envelope, which means models at many operating points and a path from control design to real flight code.

APPROACH

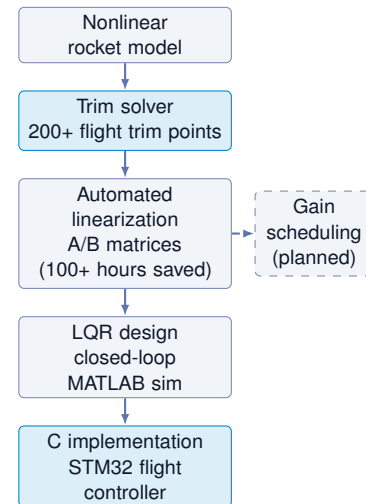
- Derived **linearized state-space models across 200+ flight trim points** for gain-scheduled control design
- Automated** trim-point refinement and A/B-matrix generation, **eliminating 100+ hours** of manual analysis
- Developed, tuned, and tested **LQR controllers** in open- and closed-loop MATLAB simulation
- Implemented the design **in C for an STM32 flight controller**, translating control-law design into flight software

ESTIMATION

- Developed a static **QUEST** quaternion-estimation algorithm for launch-pad attitude determination
- Prototyped an **Extended Kalman Filter (EKF)** in simulation

RESULT

Automating trim refinement and A/B-matrix generation eliminated 100+ hours of manual analysis, and the LQR design was carried from simulation into flight-controller C code.



METHOD (STANDARD FORMULATION) At each trim point i : $\delta \dot{x} = A_i \delta x + B_i \delta u$. LQR minimizes $J = \int_0^\infty (\delta x^\top Q \delta x + \delta u^\top R \delta u) dt$, giving $\delta u = -K_i \delta x$ with $K_i = R^{-1} B_i^\top P_i$, where P_i solves the algebraic Riccati equation.

04

Multi-Agent Deep Reinforcement Learning

A. Hypersonic intercept guidance, Sandia National Laboratories (June 2025 – May 2026)

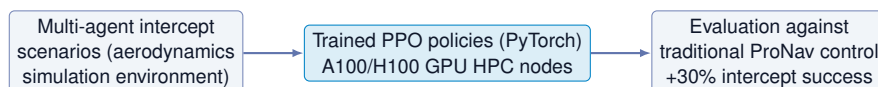
Problem. Guidance for hypersonic vehicles and extra-long-range surface-to-air missiles is a multi-agent problem: engagement outcomes depend on how several agents behave together.

APPROACH

- Investigated **multi-agent intercept dynamics** for hypersonic and extra-long-range surface-to-air missile applications
- Trained GNC **deep reinforcement learning** policies with **PyTorch Proximal Policy Optimization (PPO)** on **NVIDIA A100/H100 GPU HPC nodes**
- Evaluated multi-agent intercept scenarios in an aerodynamics simulation environment

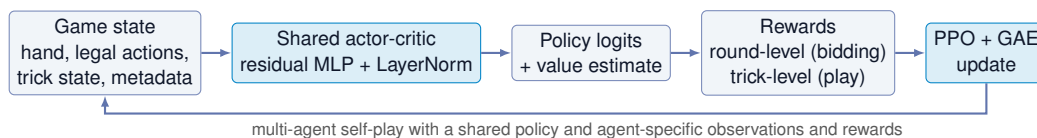
+30%

multi-agent intercept success over traditional ProNav control

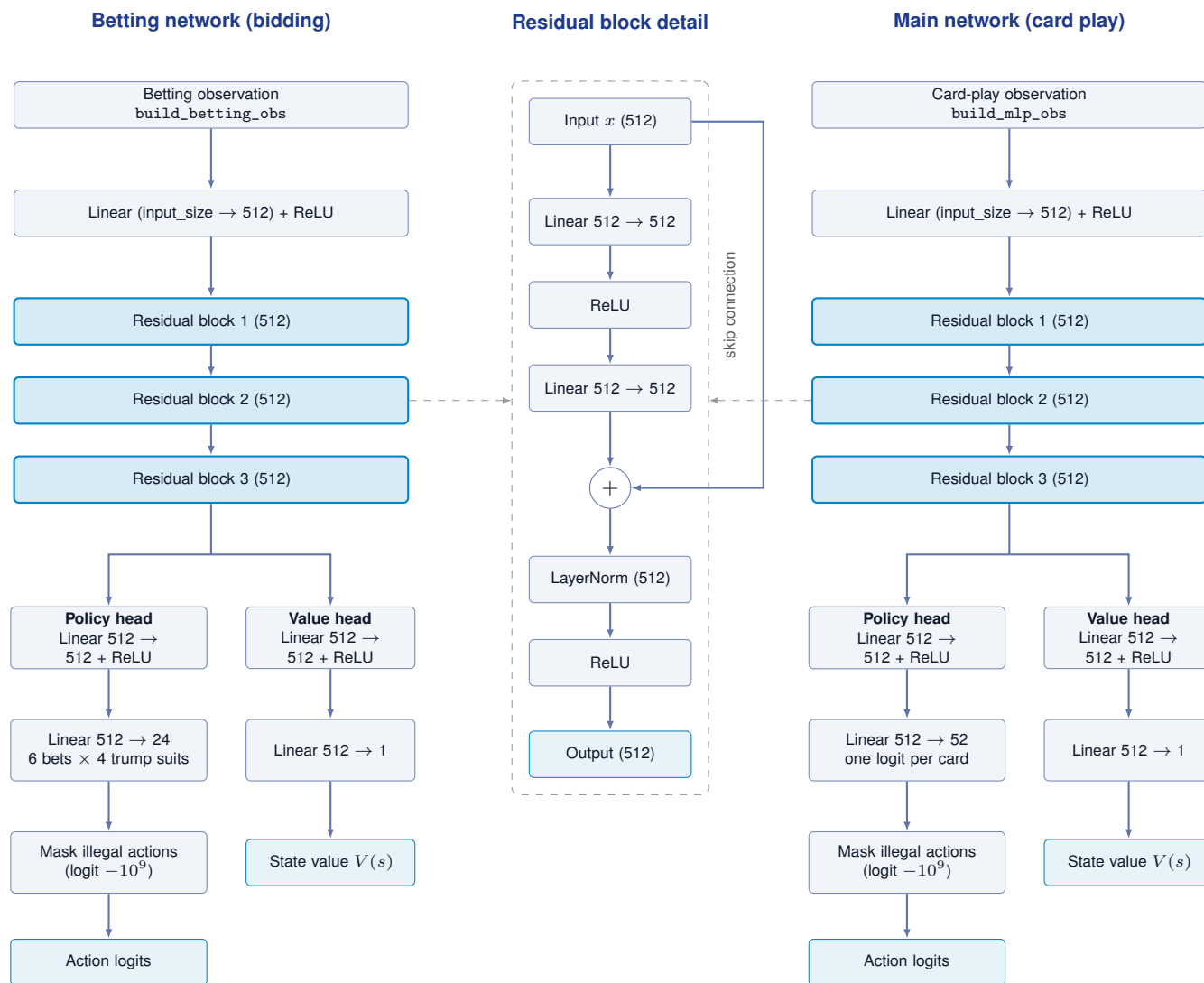
Additional technical details omitted due to project restrictions.*Generic workflow schematic; scenario and vehicle details are restricted*

04 Multi-Agent Deep Reinforcement Learning

B. CHIMERA: self-play agent for 4-player Cinch (May 2026) · play-chimera.vercel.app

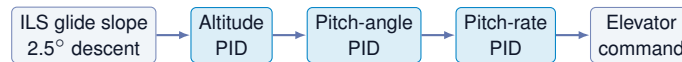


- Built a **deep RL** system that learns 4-player Cinch through **multi-agent self-play**, eliminating hand-coded gameplay heuristics
- Implemented **PPO with GAE**: clipped policy objectives, advantage estimation, rollout buffers, and actor-critic optimization
- Designed separate betting and card-play policies with round-level and trick-level rewards
- **Beat experienced human players** despite never playing humans during training
- Exported PyTorch models to ONNX and deployed a Flask web app to play the agent and inspect its architecture
- Deployed serverless on Vercel (AWS Lambda): play-chimera.vercel.app



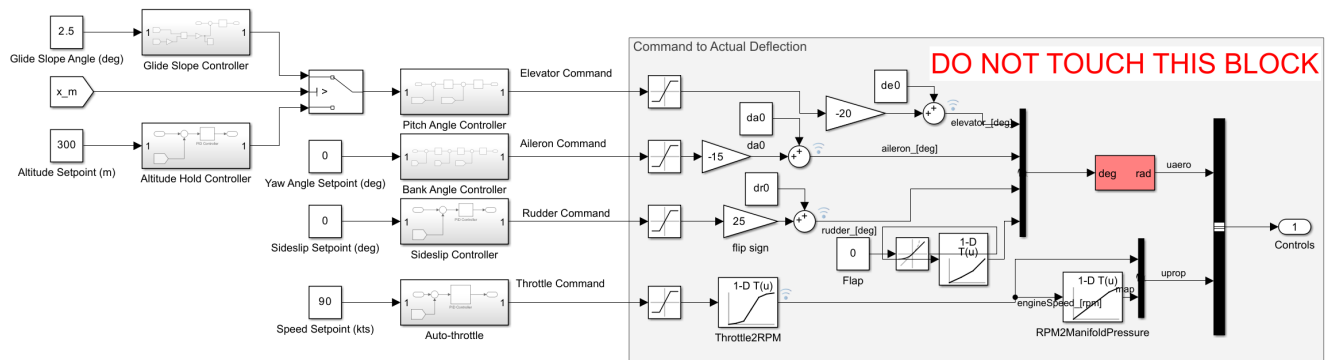
Both networks share a 512-wide trunk of three residual blocks (about 1.58M parameters) and differ in observation encoding and policy-head size; illegal actions are masked before the softmax.

inner-outer loop nesting with anti-windup clamping and output saturation

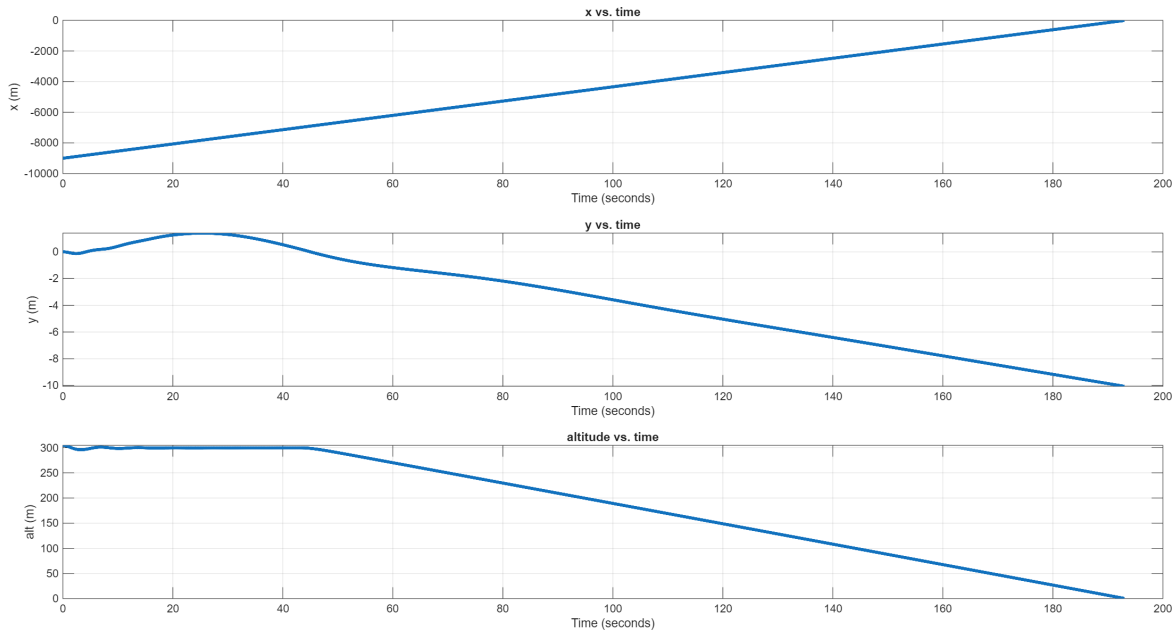


AUTOPILOT

- Developed a complete **multi-loop autopilot in Simulink** with inner-outer loop architectures for speed, bank angle, sideslip suppression, and altitude hold
- Tuned nested **PID channels** for pitch rate, pitch angle, altitude, and roll angle with **anti-windup clamping and saturation blocks** to prevent actuator chattering and give smooth command tracking
- Integrated an automated **ILS glide-slope controller** using coordinate tracking to hold a 2.5-degree descent toward the runway threshold
- Determined trim at **80 and 90 kts** and evaluated step responses, elevator deflection, angle of attack, and engine RPM



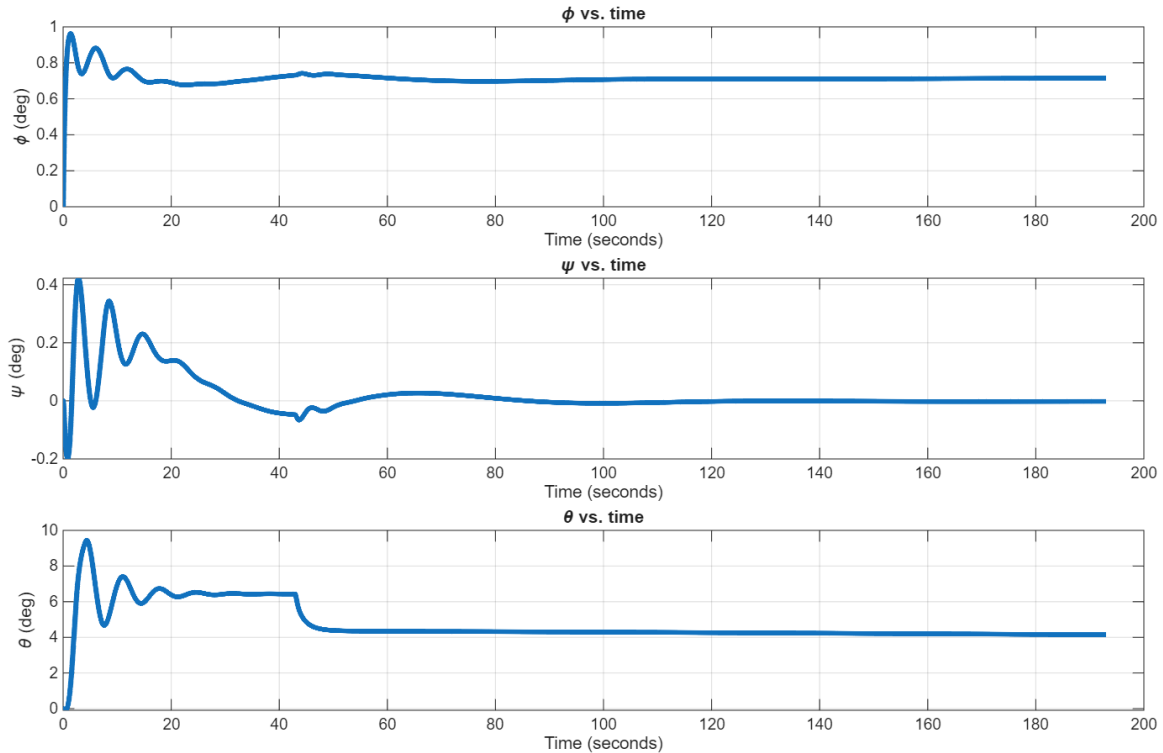
Top-level Simulink autopilot architecture.



x , y , and altitude (z) versus time for the ILS landing simulation.

05 Autopilot & Fault-Tolerant Sensing

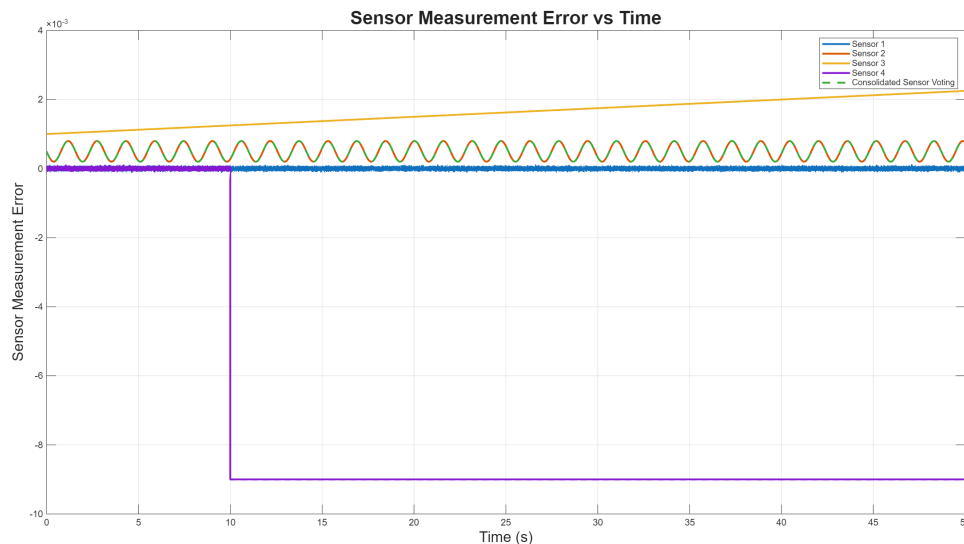
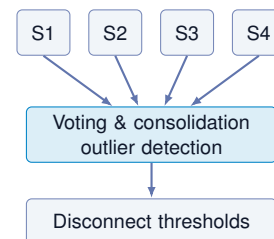
Attitude response and quadruplex sensor voting



Roll (ϕ), yaw (ψ), and pitch (θ) angles versus time for the ILS landing simulation.

QUADRUPLUX SENSOR VOTING

Build a **quadruplex sensor voting and consolidation algorithm in MATLAB** with dynamic outlier detection and automated sensor-disconnect thresholds to handle **hard-over, soft-over, and oscillatory failure modes**, so a single bad sensor cannot drive the control loops.



Voting response to injected sensor failures. Sensor 4 is disconnected at 10.00 s and Sensor 3 at 28.59 s; the voted and consolidated signal matches Sensor 2 with minimal measurement error.

AUTONOMY

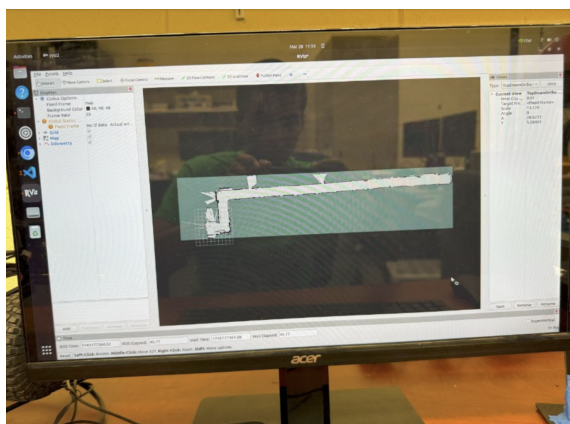
F1Tenth Autonomous Racing

Aug. 2024 – May 2025

Undergraduate Research Assistant, Georgia Tech Dynamics and Control Systems Lab



- Built an end-to-end autonomy stack for **1/10-scale racecars** on NVIDIA Jetson Orin in Python, C++, ROS, Docker, and Ubuntu Linux, tested across two cyber-physical environments; demonstrated **10+ successful time-trial laps**
- Developed a **disparity-extender reactive planner** for dynamic environments using 2D LiDAR point clouds, and tuned odometry through physical testing (servo limits, turn radius, VESC RPM-to-speed gain)



Building-corridor map used with AMCL, visualized in the Nav2 Map Server.



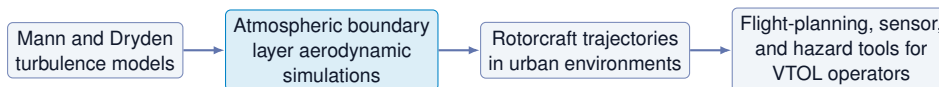
Time-trial laps on a custom racetrack with the autonomy stack described above.

RESEARCH

NASA ULI IMOCAS

Sep. 2024 – May 2026

Software Developer, Georgia Tech Nonlinear Computational Aeroelasticity Lab



- Validated and debugged atmospheric-boundary-layer aerodynamic simulations using **Mann and Dryden turbulence models** for rotorcraft trajectories in urban environments
- Built algorithms and UI for operational-analysis tools (flight planning, sensor analysis, environmental-hazard assessment) for urban-air-mobility VTOL operators, using a modular Python/PyQt5/C++ architecture (NumPy, Pandas, Matplotlib) with SOLID design

CEREAL

Aug. 2026 – present

Undergraduate Research Assistant / Incoming Graduate Research Assistant, Georgia Tech



- Developing a **real-time Lattice Boltzmann Method** solver in C++/CUDA for mid-fidelity rotorcraft aerodynamics, including disk-actuator and disk-line rotor models and in-loop control systems
- Benchmarking and optimizing simulation runtime on NVIDIA A100 GPU compute clusters

Impact, Publications, Toolbox & Experience

Summary of results, publications, technical skills, and roles

IMPACT AT A GLANCE

31,850 ft club-record apogee, Mach 'N' Roll	+35 km simulated apogee gain from 250,000+ design evaluations	+30% multi-agent intercept success over ProNav	100+ hrs manual trim and linearization analysis eliminated
1.2° sustainer fin cant selected after extensive modeling and analysis to successfully avoid roll-coupling instability	2,000 Monte Carlo flights behind 3σ recovery regions	200+ flight trim points linearized for LQR design	2 AIAA publications (2025, 2026)

PUBLICATIONS

- **Modeling and Stability Analysis of Inertial Roll Coupling in Supersonic Sounding Rockets.** AIAA Aerospace Research Central, June 18, 2026. arc.aiaa.org/doi/10.2514/6.2026-112783
- **Design and Implementation of a Data Acquisition System for Rocket Motor Testing and Performance Analysis.** AIAA Aerospace Research Central, August 12, 2025. arc.aiaa.org/doi/abs/10.2514/6.2025-99604

TOOLBOX

Controls & estimation	LQR, PID with anti-windup, EKF (simulation), QUEST, quaternion strapdown attitude estimation, trim linearization, gain/phase margin analysis, quadruplex sensor voting, ILS glide-slope tracking
Dynamics & simulation	Simulink, MATLAB, RocketPy, rigid-body dynamics, eigenvalue/stability analysis, Monte Carlo analysis, genetic/ant-colony optimization, Mann and Dryden turbulence models, Slurm HPC
Learning	PyTorch, PPO, GAE, actor-critic, multi-agent self-play, residual networks, ONNX, GPU training (A100/H100)
Software & embedded	Python, C, C++, CUDA, ROS, Nav2, Docker, Linux, Flask; STM32 embedded C (SPI, I2C, UART, CAN, DMA, interrupts)
Autonomy	2D LiDAR, SLAM, AMCL, pure pursuit, racing-line optimization, reactive planning, NVIDIA Jetson Orin

EXPERIENCE AT A GLANCE

September 2024 – May 2026	Software Developer, Nonlinear Computational Aeroelasticity Lab	NASA ULI IMOCAS
August 2024 – May 2025	Undergraduate Research Assistant, Dynamics and Control Systems Lab	F1Tenth
August 2024 – May 2025	Control Systems Engineer, Ramblin' Rocketry	GNC
August 2024 – May 2026	Avionics Engineer, Georgia Tech Experimental Rocketry	flew on Mach 'N' Roll
June 2025 – May 2026	R&D Intern, Sandia National Laboratories	hypersonic GNC
July 2025 – July 2026	President, Georgia Tech Experimental Rocketry	100-member team
August 2026 – present	Undergraduate Research Assistant, CEREAL	Lattice Boltzmann, CUDA

Looking for: new-graduate GNC and flight-dynamics roles beginning after my B.S. in May 2027. ✉ saptak.das625@gmail.com
🌐 saptakdas.vercel.app  linkedin.com/in/saptak-das1729