Nimaye Garodia

Engineer. Researcher. Innovator.

BSE Mechanical Engineering (Honors) • BS Mathematics • BA Computer Science
Certificate in Aerospace Engineering

Duke University • Class of 2026

nimaye.garodia@duke.edu nimayegarodia.com

This portfolio documents engineering and research projects spanning aerospace design, dynamical systems, computational biology, digital hardware, and field engineering. Each project was driven by a genuine technical question or real-world need, and each one produced a concrete outcome—a published paper, a flying aircraft, a built bridge, or a working chip.

Projects

Contact

Email nimaye.garodia@duke.edu

Website nimayegarodia.com

Location Durham, North Carolina, USA

Dynamics of Rotating Magnets

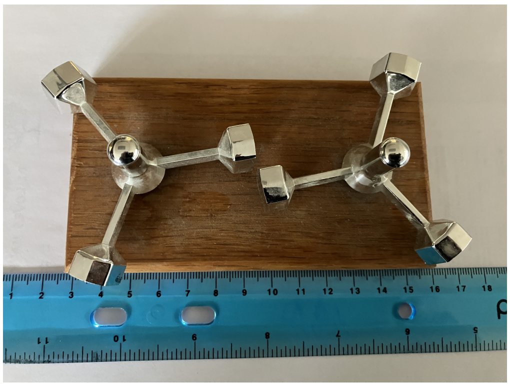
Senior Thesis | Advisor: Dr. Thomas Witelski | Mechanical Engineering & Mathematics Aug 2025 - Present

Rotating permanent magnets produce surprisingly rich nonlinear behaviour: small changes in initial conditions lead to wildly different trajectories, a hallmark of chaos. This thesis set out to characterise that behaviour rigorously—not just to simulate it, but to derive it from first principles and then build a physical system to validate the predictions.

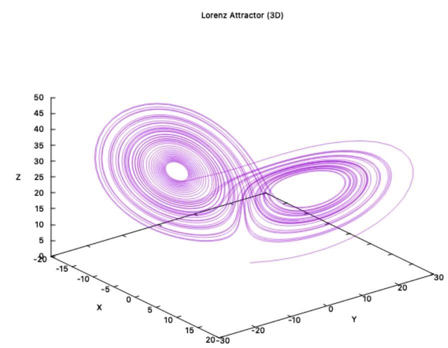
The work began analytically, formulating equations of motion for three distinct coupled-magnet configurations using Lagrangian mechanics. A mathematical model for magnetic damping was derived directly from Maxwell's equations, capturing the energy dissipation that physical systems inevitably exhibit. Solutions were first obtained in Mathematica using its built-in Runge-Kutta PDE solver, then re-implemented in a custom adaptive RK45 solver written in Python and CUDA-parallelised to handle the large parameter sweeps needed for bifurcation analysis.

To characterise the chaos quantitatively, Lyapunov exponents were computed using an iterative QR-decomposition algorithm written from scratch in C, enabling fast, low-level computation over thousands of initial conditions. Stability maps and power spectra revealed clear transitions between ordered and chaotic regimes. A to-scale physical prototype—built around axle ball bearings and permanent bar magnets—was designed to validate the numerical predictions experimentally, with video data extracted using a custom OpenCV computer-vision pipeline.

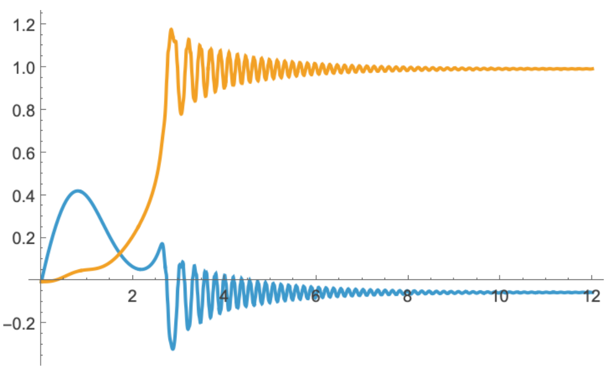
Key Technical Skills: `Lagrangian Mechanics` `C` `Python / CUDA` `Mathematica` `RK4/RK45` `Chaos Theory` `Lyapunov Analysis` `OpenCV`



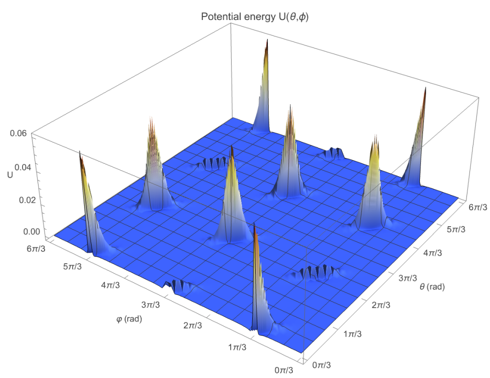
Rotor prototype



Lorenz attractor (C)



2-DOF phase plot



Potential energy surface

Glider Performance Improvement

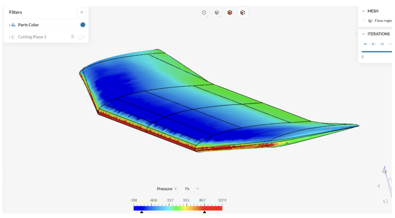
Senior Design | *Client: Dr. Michimasa Fujino, HondaJet* | *Aerospace Engineering* Oct 2025 - Present

The Skynetic Andes is a small electric glider with a commercially fixed wing geometry—but fixed does not mean optimal. Our senior design team was tasked by HondaJet's Dr. Fujino with improving the glider's efficiency by modifying its aerodynamic surfaces, specifically targeting sink rate and stall speed, two metrics that directly affect how long and how safely it can fly.

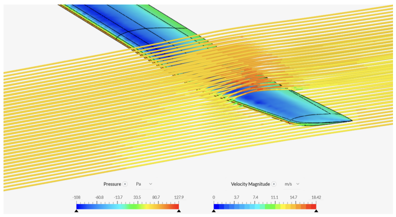
The approach was to design and integrate a flaperon—a combined flap and aileron—into the existing wing. The design process layered three levels of fidelity: thin-airfoil theory for initial sizing, XFOIL for 2D section analysis, and 3D ANSYS CFD for full-wing pressure and streamline characterisation. Custom Python scripts automated the geometric and aerodynamic parameter sweeps. The resulting flaperon reduced sink rate by 5% and stall speed by 10%, which were then confirmed experimentally in the Duke wind tunnel using carefully instrumented test runs. The final system was manufactured and integrated into a complete PID control loop with a digital RC interface, then validated through field flight tests.

Key Technical Skills: ANSYS CFD | XFOIL | Thin Airfoil Theory | Python | PID Control | Wind Tunnel Testing | Composites Manufacturing

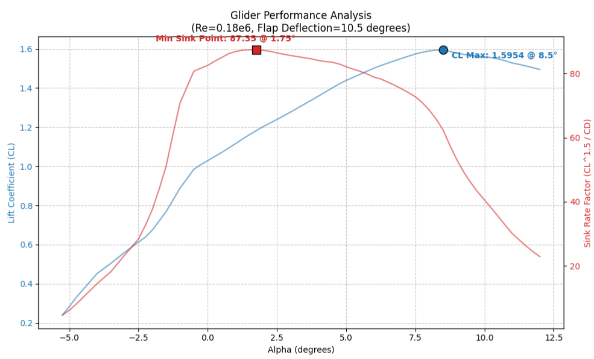
Skynetic Andes glider



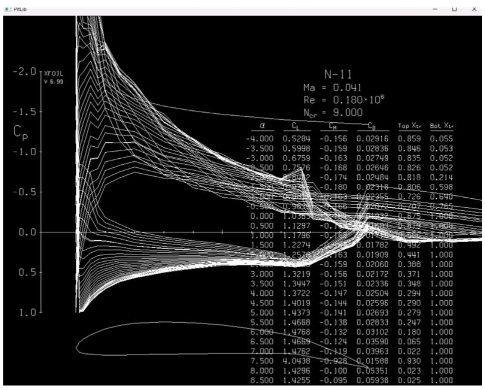
3D CFD pressure distribution



CFD streamlines



Lift curve (flaperon deflection)



XFOIL output

Mathematical Study of Anesthesia

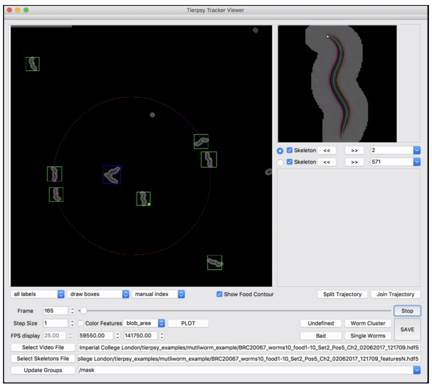
PI: Dr. Ashutosh Kowal | Mechanical Engineering & Mathematics May 2024 - Present

Anesthesia is one of medicine's most widely used and least understood phenomena. We know it works, but the precise mechanism by which it suppresses neural activity remains an open question. This research approached it mathematically, using the microscopic nematode *C. elegans*—a model organism whose 302-neuron nervous system is fully mapped—to study how anaesthetic exposure changes movement and posture at a quantifiable level.

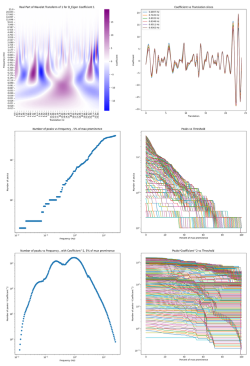
The experimental side required building the setup from scratch. A custom backlight stand and "worm motel" enclosure were fabricated using 3D printing, CAD, and laser cutting. A darkfield illumination technique was developed from first-principles optics to produce high-contrast videos suitable for the computer vision software Tierpsy, which digitised each worm's posture into a time-series. This conversion was the key insight: it reframed a biology problem as a signal processing problem, opening up a much richer analytical toolkit.

With posture represented as time-series data, custom Python algorithms were written to perform numerical integration, density-based clustering, oscillation detection, and—most importantly—wavelet transforms. Haar and Gaussian wavelet algorithms were implemented to decompose the signals across frequency and time, revealing how anaesthetic exposure suppresses specific modes of movement.

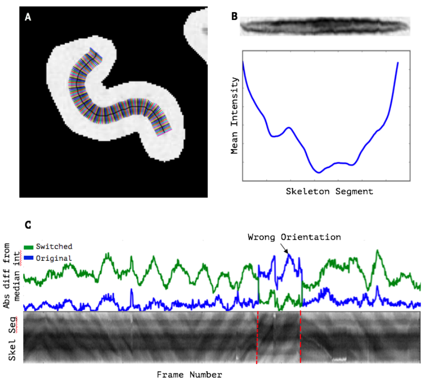
Key Technical Skills: `Python` `Wavelet Transforms` `Signal Processing` `Machine Vision` `Tierpsy` `3D Printing / CAD` `Density-Based Clustering`



Tierpsy tracking output



Wavelet transforms & power spectrum



Worm orientation data

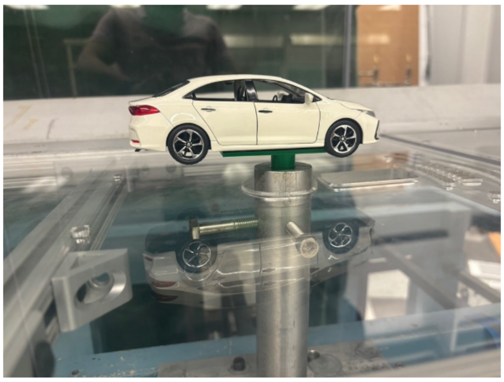
Aerodynamics of Bluff Body Automobiles

PI: Dr. Zbigniew Kabala | Aerospace Engineering Jan 2023 - Apr 2024

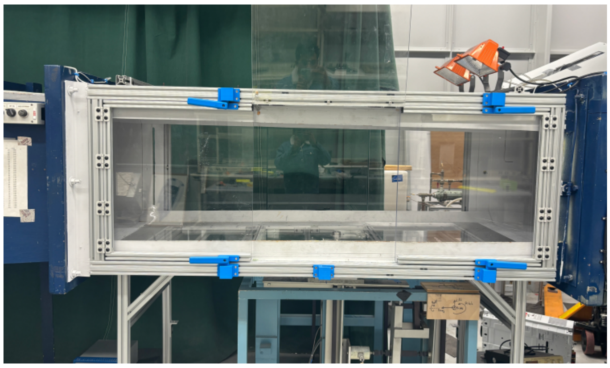
Car shape is one of the most significant determinants of fuel efficiency, yet the relationship between body geometry and aerodynamic drag is rarely studied rigorously at an accessible scale. This project set out to change that, asking: can CFD predictions for bluff-body vehicles be validated with low-cost physical experiments, and how much does shape alone account for drag differences between popular car models?

Three representative geometries were chosen—the Hyundai Elantra (sedan), Toyota Corolla (fastback), and Kia Soul (box)—to span a meaningful range of bluffness. 2D RANS simulations were run in ANSYS Fluent to generate an initial ranked hypothesis on drag. To test it physically, a custom measurement jig was designed and built for the Duke Wind Tunnel, capable of resolving the small drag forces produced by to-scale 3D-printed car models at low Reynolds numbers. The experimental results confirmed the CFD rankings and provided quantitative drag coefficients for comparison. The work was published in *Pioneer Academics* (2024): Martoma, Garodia, and Kabala, "Drag on Cars: CFD Analysis of 2-D Body Design via ANSYS Fluent vs 3-D Wind-Tunnel Study with Toy Car Replicas."

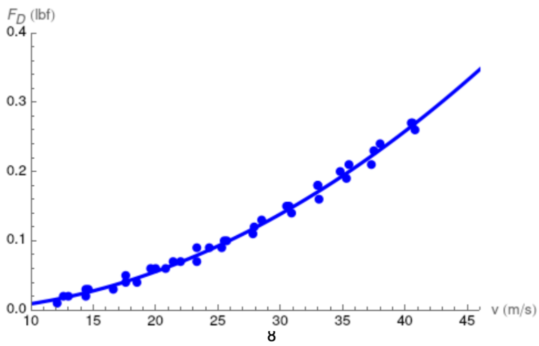
Key Technical Skills: `ANSYS Fluent` `RANS CFD` `Wind Tunnel Design` `3D Printing` `Statistical Analysis` `Technical Writing`



Car model in wind tunnel test section



Wind tunnel setup



Drag force measurements

Model Rocket Avionics System

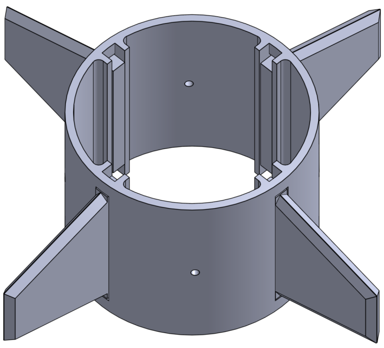
EGR 190: Special Projects in Engineering | *Duke University* Aug 2023 - Dec 2023

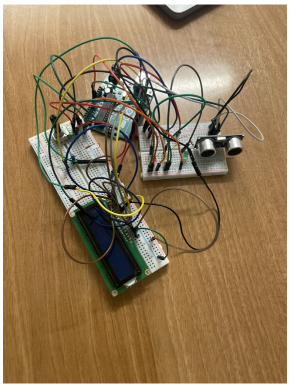
The question behind this project was deceptively simple: can you build a rocket that knows where it is, how fast it's going, and how much stress it's under—and transmit all of that in real time to the ground? The answer required building every layer of the system from sensors to software to airframe.

A custom flight computer was engineered to sample an inertial measurement unit, barometric pressure sensor, and strain gauges at flight frequencies, then encode and transmit the data via radio to a companion ground station. The strain gauges were bonded to the fin structure to measure bending loads during ascent, and the results were compared against analytical beam-bending predictions to validate the structural model. The completed rocket was launched to 3,000 ft, successfully telemetering flight data throughout the burn and coast phases.

KEY TECHNICAL SKILLS: Embedded Systems RF Telemetry Strain Gauges Data Acquisition Structural Analysis Sensor Fusion


Ready for launch


Avionics bay (CAD)


Sensor prototyping


Post-flight recovery

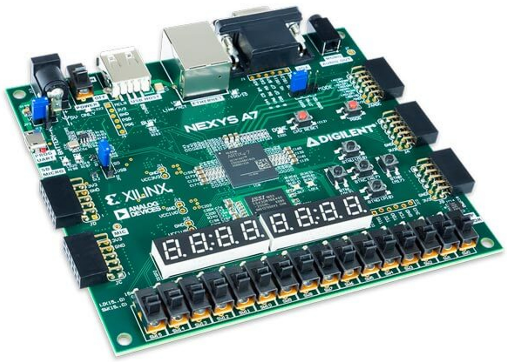
Pipelined Processor on an FPGA

ECE 350L: Digital Systems | Duke University Spring 2025

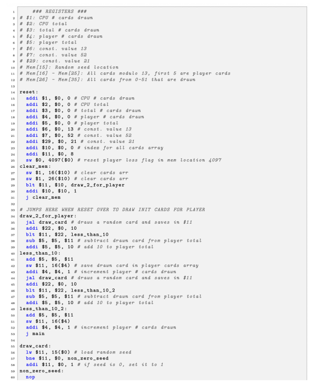
Modern processors are fast not because individual transistors are fast, but because of pipelining—the insight that you can overlap the execution of multiple instructions by breaking the processor into stages, like an assembly line. This project built that idea from the ground up: a complete 32-bit RISC processor, designed in Verilog and deployed on an FPGA, implementing every stage of a real pipeline with no shortcuts.

The five-stage pipeline (fetch, decode, execute, memory, write-back) was designed with full data-hazard forwarding so that results computed in one stage are immediately available to the next without introducing stalls. A hazard detection unit was built to handle the one case forwarding cannot solve—load-use hazards—by inserting the appropriate stall cycles automatically. Branch prediction and exception handling were added to complete the control-flow logic. The processor was validated against a comprehensive suite of hand-written assembly programs covering arithmetic, memory operations, and complex control flow, then integrated with I/O peripherals on the FPGA board to demonstrate real system interaction.

Key Technical Skills: `Verilog / HDL` `FPGA` `Computer Architecture` `Pipeline Design` `Hazard Detection` `Assembly Programming`

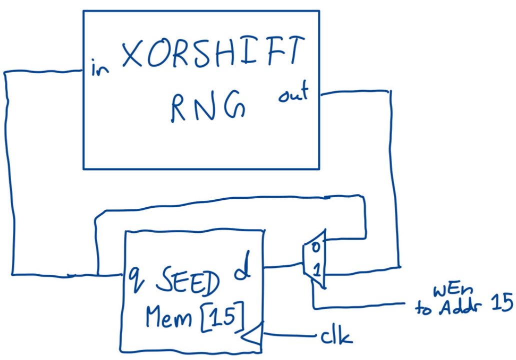


FPGA development board

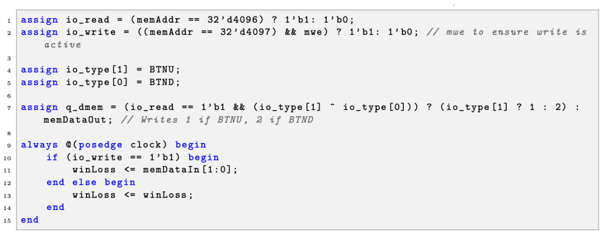


```
### REGISTERS ###
# $1: CPU # cards drawn
# $2: CPU total
# $3: total # cards drawn
# $4: player # cards drawn
# $5: player total
# $6: const. value 13
# $7: const. value 52
# $20: const. value 21
# Mem[15]: Random seed location
# Mem[16] - Mem[25]: All cards modulo 13, first 5 are player cards
# Mem[26] - Mem[51]: All cards from 0-51 that are drawn

reset:
  addi $1, $0, 0 # CPU # cards drawn
  addi $2, $0, 0 # CPU total
  addi $3, $0, 0 # total # cards drawn
  addi $4, $0, 0 # player # cards drawn
  addi $5, $0, 0 # player total
  addi $6, $0, 13 # const. value 13
  addi $7, $0, 52 # const. value 52
  addi $20, $0, 21 # const. value 21
  addi $10, $0, 0 # index for all cards array
  addi $11, $0, 8
  sw $0, 4097($0) # reset player loss flag in mem location 4097
clear_mem:
  sw $1, 16($10) # clear cards arr
  sw $1, 26($10) # clear cards arr
  blt $11, $10, draw_2_for_player
  addi $10, $10, 1
  j clear_mem

# JUMPS HERE WHEN RESET OVER TO DRAW INIT CARDS FOR PLAYER
draw_2_for_player:
  jal draw_card # draws a random card and saves in $11
  addi $22, $0, 10
  blt $11, $22, less_than_10
  sub $5, $5, $11 # subtract drawn card from player total
  addi $5, $5, 10 # add 10 to player total
less_than_10:
  add $5, $5, $11
  sw $11, 16($4) # save drawn card in player cards array
  addi $4, $4, 1 # increment player # cards drawn
  jal draw_card # draws a random card and saves in $11
  addi $22, $0, 10
  blt $11, $22, less_than_10_2
  sub $5, $5, $11 # subtract drawn card from player total
  addi $5, $5, 10 # add 10 to player total
less_than_10_2:
  add $5, $5, $11
  sw $11, 16($4)
  addi $4, $4, 1 # increment player # cards drawn
  j main

draw_card:
  lw $11, 15($0) # load random seed
  bne $11, $0, non_zero_seed
  addi $11, $0, 1 # if seed is 0, set it to 1
non_zero_seed:
  nop
```

Assembly test program



XORShift routine



```
assign io_read = (memAddr == 32'd4096) ? 1'b1: 1'b0;
assign io_write = ((memAddr == 32'd4097) && mwe) ? 1'b1: 1'b0; // mwe to ensure write is
    active

assign io_type[1] = BTNU;
assign io_type[0] = BTND;

assign q_dmem = (io_read == 1'b1 && (io_type[1] ^ io_type[0])) ? (io_type[1] ? 1 : 2) :
    memDataOut; // Writes 1 if BTNU, 2 if BTND

always @(posedge clock) begin
    if (io_write == 1'b1) begin
        winLoss <= memDataIn[1:0];
    end else begin
        winLoss <= winLoss;
    end
end
```

Verilog source

Suspended Footbridge

Engineers in Action / DukeEngage | *Eswatini, Africa* May 2023 - Jul 2023

In rural Eswatini, a river crossing that takes minutes in dry season can become impassable for months during the rains—cutting off children from school and communities from markets. Engineers in Action partnered with local communities to build suspended footbridges that solve this problem permanently. Over 12 weeks, our team of Duke students designed and constructed what became the longest suspended footbridge in the country, spanning 122 metres.

The project demanded a kind of engineering that textbooks rarely teach: adapting a standard bridge design to locally sourced materials, managing a construction site where the supply chain was unpredictable, and coordinating a team of Swazi community members who brought their own construction knowledge and labour. Concrete was mixed on site, cables were tensioned by hand, and every design decision had to account for what was actually available. The outcome was tangible: school attendance in the surrounding community increased by 200% once the bridge was completed, because children who had previously been cut off for months each year could now cross safely year-round.

KEY TECHNICAL SKILLS: Structural Engineering | Suspension Bridge Design | Site Coordination | Adaptive Engineering | Community Engagement | Team Leadership

Completed 122 m bridge

Installing the floor deck

On-site cement mixing

Community engagement

Team on completion